Chris Piech
CS109

Lecture Handout #1
Sept 25, 2017

Counting

Based on a handout by Mehran Sahami

Although you may have thought you had a pretty good grasp on the notion of counting at the age of three, it turns out that you had to wait until now to learn how to *really* count. Aren't you glad you took this class now?! But seriously, below we present some properties related to counting which you may find helpful in the future.

Sum Rule

Sum Rule of Counting: If the outcome of an experiment can either be one of m outcomes or one of n outcomes, where none of the outcomes in the set of m outcomes is the same as the any of the outcomes in the set of n outcomes, then there are $m + n$ possible outcomes of the experiment.

Rewritten using set notation, the Sum Rule states that if the outcomes of an experiment can either be drawn from set A or set B, where $|A| = m$ and $|B| = n$, and $A \cap B = \varnothing$, then the number of outcomes of the experiment is $|A| + |B| = m + n$.

Example 1

Problem: You are running an on-line social networking application which has its distributed servers housed in two different data centers, one in San Francisco and the other in Boston. The San Francisco data center has 100 servers in it and the Boston data center has 50 servers in it. If a server request is sent to the application, how large is the set of servers it may get routed to?

Solution: Since the request can be sent to either of the two data centers and none of the machines in either data center are the same, the Sum Rule of Counting applies. Using this rule, we know that the request could potentially be routed to any of the 150 (= 100 + 50) servers.

Product Rule

Product Rule of Counting: If an experiment has two parts, where the first part can result in one of m outcomes and the second part can result in one of n outcomes regardless of the outcome of the first part, then the total number of outcomes for the experiment is mn.

Rewritten using set notation, the Product Rule states that if an experiment with two parts has an outcome from set A in the first part, where $|A| = m$, and an outcome from set B in the second part (regardless of the outcome of the first part), where $|B| = n$, then the total number of outcomes of the experiment is $|A|\,|B| = mn$.

Note that the Product Rule for Counting is very similar to "the basic principle of counting" given in the Ross textbook.

Example 2

Problem: Two 6-sided dice, with faces numbered 1 through 6, are rolled. How many possible outcomes of the roll are there?

Solution: Note that we are not concerned with the total value of the two dice, but rather the set of all explicit outcomes of the rolls. Since the first die[1] can come up with 6 possible values and the second die similarly can have 6 possible values (regardless of what appeared on the first die), the total number of potential outcomes is 36 (= 6 * 6). These possible outcomes are explicitly listed below as a series of pairs, denoting the values rolled on the pair of dice:

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

Example 3

Problem: Consider a hash table with 100 buckets. Two arbitrary strings are independently hashed and added to the table. How many possible ways are there for the strings to be stored in the table?

Solution: Each string can be hashed to one of 100 buckets. Since the results of hashing the first string do not impact the hash of the second, there are 100 * 100 = 10,000 ways that the two strings may be stored in the hash table.

The Inclusion-Exclusion Principle

Inclusion-Exclusion Principle: If the outcome of an experiment can either be drawn from set A or set B, and sets A and B may potentially overlap (i.e., it is not guaranteed that $A \cap B = \varnothing$), then the number of outcomes of the experiment is $|A \cup B| = |A| + |B| - |A \cap B|$.

Note that the Inclusion-Exclusion Principle generalizes the Sum Rule of Counting for arbitrary sets A and B. In the case where $A \cap B = \varnothing$, the Inclusion-Exclusion Principle gives the same result as the Sum Rule of Counting since $|\varnothing| = 0$.

Example 4

Problem: An 8-bit string (one byte) is sent over a network. The valid set of strings recognized by the receiver must either start with 01 or end with 10. How many such strings are there?

Solution: The potential bit strings that match the receiver's criteria can either be the 64 strings that start with 01 (since that last 6 bits are left unspecified, allowing for $2^6 = 64$ possibilities) or the 64 strings that end with 10 (since the first 6 bits are unspecified). Of course, these two sets overlap, since strings that start with 01 *and* end with 10 are in both sets. There are 2^4=16 such strings (since the middle 4 bits can be arbitrary). Casting this description into corresponding set notation, we have: $|A| = 64$, $|B| = 64$, and $|A \cap B| = 16$, so by the Inclusion-Exclusion Principle, there are 64 + 64 – 16 = 112 strings that match the specified receiver's criteria.

[1] "die" is the singular form of the word "dice" (which is the plural form).

Floors and Ceilings: They're Not Just For Buildings Anymore…

Floor and *ceiling* are two handy functions that we give below just for reference. Besides, their names sound so much neater than "rounding down" and "rounding up", and they are well-defined on negative numbers too. Bonus.

Floor Function

The **floor** function assigns to the real number x the largest integer that is less than or equal to x. The floor function applied to x is denoted $\lfloor x \rfloor$.

Ceiling Function

The **ceiling** function assigns to the real number x the smallest integer that is greater than or equal to x. The floor function applied to x is denoted $\lceil x \rceil$.

Example 5

$\lfloor 1/2 \rfloor = 0$ $\lfloor -1/2 \rfloor = -1$ $\lfloor 2.9 \rfloor = 2$ $\lfloor 8.0 \rfloor = 8$

$\lceil 1/2 \rceil = 1$ $\lceil -1/2 \rceil = 0$ $\lceil 2.9 \rceil = 3$ $\lceil 8.0 \rceil = 8$

The Pigeonhole Principle

Basic Pigeonhole Principle: For positive integers m and n, if m objects are placed in n buckets, where $m > n$, then at least one bucket must contain at least two objects.

In a more general form, this principle can be stated as:

General Pigeonhole Principle: For positive integers m and n, if m objects are placed in n buckets, then at least one bucket must contain at least $\lceil m/n \rceil$ objects.

Note that the generalized form does not require the constraint that $m > n$, since in the case where $m \leq n$, we have $\lceil m/n \rceil = 1$, and it trivially holds that at least one bucket will contain at least one object.

Example 6

Problem: Consider a hash table with 100 buckets. 950 strings are hashed and added to the table.

a) Is it possible that a bucket in the table contains no entries?

b) Is it guaranteed that at least one bucket in the table contains at least two entries?

c) Is it guaranteed that at least one bucket in the table contains at least 10 entries?

d) Is it guaranteed that at least one bucket in the table contains at least 11 entries?

Solution:

a) Yes. As one example, it is possible (albeit very improbable) that all 950 strings get hashed to the same bucket (say bucket 0). In this case bucket 1 would have no entries.

b) Yes. Since, 950 objects are placed in 100 buckets and 950 > 100, by the Basic Pigeonhole Principle, it follows that at least one bucket must contain at least two entries.

c) Yes. Since, 950 objects are placed in 100 buckets and $\lceil 950/100 \rceil = \lceil 9.5 \rceil = 10$, by the General Pigeonhole Principle, it follows that at least one bucket must contain at least 10 entries.

d) No. As one example, consider the case where the first 50 bucket each contain 10 entries and the second 50 buckets each contain 9 entries. This accounts for all 950 entries (50 * 10 + 50 * 9 = 950), but there is no bucket that contains 11 entries in the hash table.

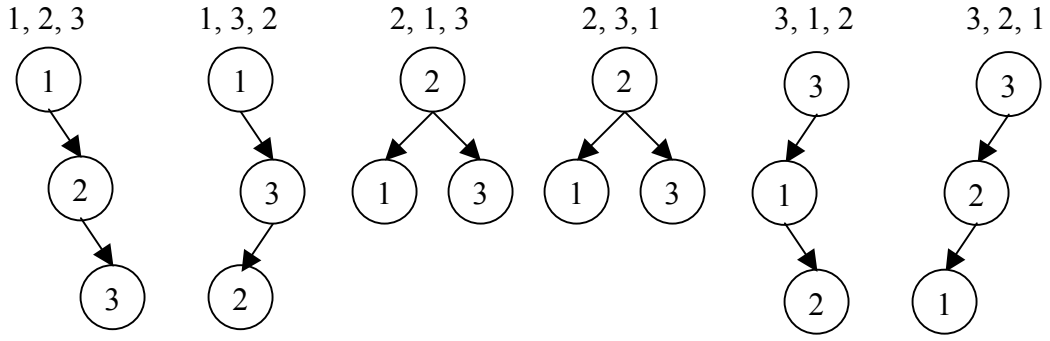
An Example with Data Structures (*Example 7*)

Recall the definition of a **binary search tree** (BST), which is a binary tree that satisfies the following three properties for *every* node n in the tree:

1. n's value is greater than all the values in its left subtree.
2. n's value is less than all the values in its right subtree.
3. both n's left and right subtrees are binary search trees.

Problem: How many possible binary search trees are there which contain the three values 1, 2, and 3, and have a degenerate structure (i.e., each node in the BST has at most one child)?

Solution: We start by considering the fact that the three values in the BST (1, 2, and 3) may have been inserted in any one of 3! (=6) orderings (permutations). For each of the 3! ways the values could have been ordered when being inserted into the BST, we can determine what the resulting structure would be and determine which of them are degenerate. Below we consider each possible ordering of the three values and the resulting BST structure.



We see that there 4 degenerate BSTs here (the first two and last two).

Bibliography

For additional information on counting, you can consult a good discrete mathematics or probability textbook. Some of the discussion above is based on the treatment in:

K. Rosen, *Discrete Mathematics and its Applications, 6th Ed.*, New York: McGraw-Hill, 2007.

Chris Piech
CS109

Lecture Handout #2
Sept 26, 2017

Combinatorics

Based on examples by Chris +Mehran Sahami

As we mentioned last class, the ideas presented in "counting" are core to probability. Counting is like the foundation of a house (where the house is all the great things we will do later in CS109, such as machine learning). Houses are awesome. Foundations on the other hand are pretty much just concrete in a hole. But don't make a house without a foundation. Trust me on that.

Permutations

Permutation Rule: A permutation is an ordered arrangement of *n* distinct object. Those *n* objects can be permuted in n x (n – 1) x (n – 2) x ... x 2 x 1 = n! ways.

This changes slightly if you are permuting a subset of distinct objects, or if some of your objects are indistinct. We will handle those cases shortly!

Example 1

Part A: iPhones have 4 digit passcode. What if there are 4 smudges over 4 digits on screen. How distinct many passcodes possible?

Solution: Since the order of codes is important we should use permutations. And since there are exactly four smudges we know that each number is distinct. Thus we can plug in the permutation formula: 4! = 24

Part B: What if there are 3 smudges over 3 digits on screen?

Solution: One of 3 digits is repeated, but don't know which one. Solve this by making three cases (each with the same number of permutations). Let A, B, C represent 3 digits:
4! permutations of: A B C_1 C_2
But need to eliminate over counting of permutations of C's
3 x [4! / (2! 1! 1!)] = 3 x 12 = 36

Part C: What if there are 2 smudges over 2 digits on screen?

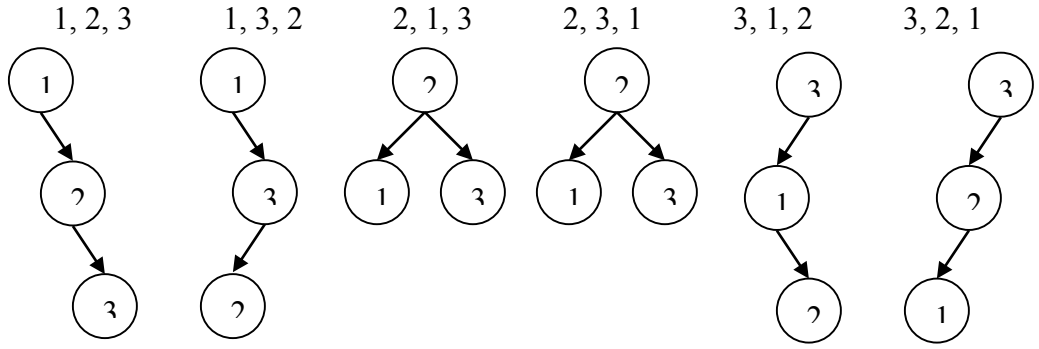
Solution: There are two possibilities, 2 digits used twice each or 1 digit of 2 digits used 3 times, and other digit used once.
[4! / (2! 2!)] + 2 x [4! / (3! 1!)] = 6 + (2 x 4) = 6 + 8 = 14

Example 2

Recall the definition of a **binary search tree** (BST), which is a binary tree that satisfies the following three properties for *every* node n in the tree:

1. n's value is greater than all the values in its left subtree.
2. n's value is less than all the values in its right subtree.
3. both n's left and right subtrees are binary search trees.

Problem: How many possible BST containing 1, 2, and 3 have a degenerate structure (i.e., each node in the BST has at most one child)?
Solution: There are 3! ways to order elements 1, 2, and 3 for insertion:



We see that there are 4 degenerate BSTs here (the first two and last two).

Permutations of Indistinct Objects

Permutation of Indistinct Objects: Generally when there are n objects and
n_1 are the same (indistinguishable) and
n_2 are the same and
...
n_r are the same, then there are $\dfrac{n!}{n_1!n_2!\ldots n_r!}$ permutations

Example 3

Problem: How many distinct bit strings can be formed from three 0's and two 1's?

Solution: 5 total digits = 5!

But that is assuming the 0's and 1's are indistinguishable (to make that explicit let's give each one a subscript). Here is a subset of the permutations.

$0_1\ 1_1\ 1_2\ 0_2\ 0_3$
$0_1\ 1_1\ 1_2\ 0_3\ 0_2$
$0_2\ 1_1\ 1_2\ 0_1\ 0_3$
$0_2\ 1_1\ 1_2\ 0_3\ 0_1$
$0_3\ 1_1\ 1_2\ 0_1\ 0_2$
$0_3\ 1_1\ 1_2\ 0_2\ 0_1$

All listed permutations are the same. For any given permutation, there are 3! ways of rearranging the 0s and 2! ways of rearranging the 1s (resulting in an indistinguishable string). We have over counted. Using the permutations of indistinct objects we correct for the over counting:

$$\text{Total} = \frac{5!}{3!2!} = \frac{120}{6\cdot 2} = \frac{120}{12} = 10$$

Combinations

Combinations: A combination is an unordered selection of r objects from a set of n objects. If all objects are distinct, then the number of ways of making the selection is:

$$\frac{n!}{r!(n-r)!} = \binom{n}{r} \text{ ways}$$

This is often stated as "n choose r"

Consider this general way to product combinations: To select r unordered objects from a set of n objects, E.g. "7 choose 3",

1. First consider permutations of all n objects. There are $n!$ ways to do that.
2. Then select the first r in the permutation. There is one way to do that.
3. Note that the order of r selected objects is irrelevant. There are $r!$ ways to permute them. The selection remains unchanged.
4. Note that the order of $(n-r)$ unselected objects is irrelevant. There are $(n-r)!$ ways to permute them. The selection remains unchanged.

$$\textbf{Total} = \frac{n!}{r!(n-r)!} = \binom{n}{r} = \binom{n}{n-r} \qquad \frac{7!}{3!4!} = 35$$

Which is the combinations formula.

Example 4

Problem: In the Hunger Games, how many ways are there of choosing 2 villages from district 12, which has a population of 8,000?

Solution: This is a straightforward combinations problem. 8,000 choose 2 = 31,996,000.

Example 5

Part A*:* How many ways to select 3 books from 6

Solution: If each of the books are distinct then this is another straightforward combination problem. There are 6 choose 3 ways:

$$\text{Total} = \binom{6}{3} = \frac{6!}{3!3!} = 20$$

Part B: How many ways to select 3 books if there are two books that should not both be chosen together (for example, don't chose both the 8th and 9th edition of the Ross textbook).

Solution: This problem is easier to solve if we split it up into cases. Consider the following three different cases:

Case 1: Select 8the Ed and 2 other non-9th Ed: There are 4 choose 2 ways of doing so.

Case 2: Select 9th Ed and 2 other non-8th Ed: There are 4 choose 2 ways of doing so.

Case 3: Select 3 from the books that are neither the eighth nor the ninth edition: There are 4 choose 3 ways of doing so.

Using our old friend the Sum Rule of Counting, we can add the cases

$$\text{Total} = 2 * \binom{4}{2} + \binom{4}{3} = 16$$

Alternatively, we could have calculated all the ways of selecting 3 books from 6, and then subtract the "forbidden" ones (eg the selections that break the constraint). Chris calls this the Beijing method because of the Forbidden City there. That's not important

Forbidden Case: Select 8th edition and 9th edition and 1 other book. There are 4 choose 1 ways of doing so (which equals 4).

Answer = All possibilities – forbidden =20 – 4 = 16

Two different ways to get the same right answer!

Group Assignment

You have probably heard about the dreaded "balls and urns" probability examples. What are those all about? They are the many different ways that we can think of stuffing elements into containers. Why people called their containers urns, ~~I have no idea~~ (I looked it up. It turns out that Jacob Bernoulli was into voting and ancient Rome. And in ancient Rome they used urns for ballot boxes). Group assignment problems are useful metaphors for many counting problems.

Example 6

Problem: Say you want to put n distinguishable balls into r urns. (no wait don't say that). Ok fine. No urns. Say we are going to put n strings into r buckets of a hashtable where all outcomes are equally likely. How many possible ways are there of doing this?
Answer: You can think of this as n independent experiments each with r outcomes. Using our friend the General Rule of Counting this comes out to $\boldsymbol{r^n}$

Divider Method: A divider problem is one where you want to place n indistinguishable items into r containers. The divider method works by imagining that you are going to solve this problem by sorting two types of objects, your n original elements and $(r-1)$ dividers. Thus you are permuting $n + r - 1$ objects, n of which are same (your elements) and $r - 1$ of which are same (the dividers). Thus:

$$\text{Total ways} = \frac{(n+r-1)!}{n!(r-1)!} = \binom{n+r-1}{r-1}$$

Example 7

Part A: Say you are an investor at a micro loan group (say the Gramin Bank) and you have $10 thousand to invest in 4 companies (in 1K increments). How many ways can you allocate it?

Solution: This is just like putting 10 balls into 4 urns. Using the Divider Method we get:

$$\text{Total ways} = \binom{10+4-1}{4-1} = \binom{13}{3} = 286$$

Part B: What if you don't have to invest all 10 K? **(Economy tight)**

Solution: Simply imagine that you have an extra company – yourself. Now you are investing 10 thousand in 5 companies. Thus the answer is the same as putting 10 balls into 5 urns.

$$\text{Total ways} = \binom{10+5-1}{5-1} = \binom{14}{4} = 1001$$

Part C: Want to invest at least 3 thousand in company 1?

Solution: There is one way to give 3 thousand to company 1. The number of ways of investing the remaining money is the same as putting 7 balls into 4 urns.

$$\text{Total ways} = \binom{7+4-1}{4-1} = \binom{10}{3} = 120$$

This handout was made fresh just for you. Did you notice any mistakes? Let Chris know and he will fix them.

Chris Piech
CS109

Handout #6
Sept 29, 2017

Probability

It is that time in the quarter (it is still week one) when we get to talk about probability. Again we are going to build up from first principles. We will heavily use the counting that we learned earlier this week.

Event Space and Sample Space

Sample space, S, is set of all possible outcomes of an experiment. For example:

1. Coin flip: S = {Head, Tails}
2. Flipping two coins: S = {(H, H), (H, T), (T, H), (T, T)}
3. Roll of 6-sided die: S = {1, 2, 3, 4, 5, 6}
4. # emails in a day: S = $\{x \mid x \in \mathbf{Z}, x \geq 0\}$ (non-neg. ints)
5. YouTube hrs. in day: S = $\{x \mid x \in \mathbf{R}, 0 \leq x \leq 24\}$

Event Space, E, is some subset of S that we ascribe meaning to. In set notation ($E \subseteq S$).

1. Coin flip is heads: E = {Head}
2. ≥ 1 head on 2 coin flips: E = {(H, H), (H, T), (T, H)}
3. Roll of die is 3 or less: E = {1, 2, 3}
4. # emails in a day ≤ 20: E = $\{x \mid x \in \mathbf{Z}, 0 \leq x \leq 20\}$
5. Wasted day (≥ 5 YT hrs.): E = $\{x \mid x \in \mathbf{R}, 5 \leq x \leq 24\}$

Probability

In the 20th century humans figured out a way to precisely define what a probability is:

$$P(E) = \lim_{n \to \infty} \frac{n(E)}{n}$$

In English this reads: lets say you perform *n* trials of an experiment. The probability of a desired event E is the ratio of trials that result in E to the number of trials performed (in the limit as your number of trials approaches infinity).

That is mathematically rigorous. You can also apply other semantics to the concept of a probability. One common meaning ascribed is that P(E) is a measure of the chance of E occurring.

I often think of a probability in another way: I don't know everything about the world. So it goes. As a result I have to come up with a way of expressing my belief that E will happen given my limited knowledge. This interpretation acknowledges that there are two sources of probabilities: natural randomness and our own uncertainty.

Axioms of Probability

Here are some basic truths about probabilities:

Axiom 1: $0 \leq P(E) \leq 1$
Axiom 2: $P(S) = 1$
Axiom 3: $P(E^c) = 1 - P(E)$

You can convince yourself of the first axiom by thinking about the definition of probability. As you perform trials of an experiment it is not possible to get more events then trials (thus probabilities are less than 1) and its not possible to get less than 0 occurrences of the event.

The second axiom makes sense too. If your event space *is* the sample space, then each trial must produce the event. This is sort of like saying; the probability of you eating cake (event space) if you eat cake (sample space) is 1.

The third axiom comes from a deep philosophical point. Everything in the world must either be a potato or not a potato. Similarly, everything in the sample space must either be in the event space, or not in the event space.

Equally Likely Events

Some sample spaces have equally likely outcomes. We like those sample spaces, because there is a way to calculate probability questions about those sample spaces simply by counting. Here are a few examples where there are equally likely outcomes:

1. Coin flip: S = {Head, Tails}
2. Flipping two coins: S = {(H, H), (H, T), (T, H), (T, T)}
3. Roll of 6-sided die: S = {1, 2, 3, 4, 5, 6}

Because every outcome is equally likely, and the probability of the sample space must be 1, we can prove that each outcome must have probability:

$$P(\text{Each outcome}) = \frac{1}{|S|}$$

If an event is a subset of a sample space with equally likely outcomes.

$$P(E) = \frac{\text{number of outcomes in E}}{\text{number of outcomes in S}} = \frac{|E|}{|S|}$$

Interestingly, this idea also applies to continuous sample spaces. Consider the sample space of all the outcomes of the computer function ``random" which produces a real valued number between 0 and 1, where all real valued numbers are equally likely. Now consider the event E that the number generated is in the range [0.3 to 0.7]. Since the sample space is equally likely, $P(E)$ is the ratio of the size of E to the size of S. In this case $P(E) = 0.4$.

When trying to solve a problem using equally likely sample spaces, you will use counting. How you set up your counting strategy for the sample space will determine if each outcome is equally likely. A nifty trick: make your objects distinct. Counting with distinct objects often makes the sample space events equally likely. Even if your objects are not distinct by default, you can make them distinct, as long as you do so in both the sample space and the event space.

Chris Piech
CS109

Lecture #4
October 2, 2017

Conditional Probability

1 Conditional Probability

In English, a conditional probability answers the question: "What is the chance of an event E happening, given that I have already observed some other event F?" Conditional probability quantifies the notion of updating one's beliefs in the face of new evidence.

When you condition on an event happening you are entering the universe where that event has taken place. Mathematically, if you condition on F, then F becomes your new sample space. In the universe where F has taken place, all rules of probability still hold!

The definition for calculating conditional probability is:

Definition of Conditional Probability
The probability of E given that (aka conditioned on) event F already happened:

$$P(E \mid F) = \frac{P(EF)}{P(F)} = \frac{P(E \cap F)}{P(F)}$$

(As a reminder, EF means the same thing as $E \cap F$—that is, E "and" F.)

A visualization might help you understand this definition. Consider events E and F which have outcomes that are subsets of a sample space with 50 equally likely outcomes, each one drawn as a hexagon:

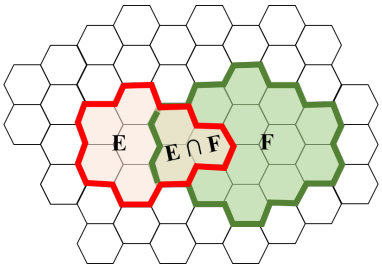


Figure 1: Conditional Probability Intuition

Conditioning on F means that we have entered the world where F has happened (and F, which has 14 equally likely outcomes, has become our new sample space). Given that event F has occured, the conditional probability that event E occurs is the subset of the outcomes of E that are consistent

with F. In this case we can visually see that those are the three outcomes in $E \cap F$. Thus we have the:

$$P(E|F) = \frac{P(EF)}{P(F)} = \frac{3/50}{14/50} = \frac{3}{14} \approx 0.21$$

Even though the visual example (with equally likely outcome spaces) is useful for gaining intuition, the above definition of conditional probability applies regardless of whether the sample space has equally likely outcomes.

The Chain Rule
The definition of conditional probability can be rewritten as:

$$P(EF) = P(E \mid F)P(F)$$

which we call the Chain Rule. Intuitively it states that the probability of observing events E and F is the probability of observing F, multiplied by the probability of observing E, given that you have observed F. Here is the general form of the Chain Rule:

$$P(E_1E_2 \ldots E_n) = P(E_1)P(E_2 \mid E_1) \ldots P(E_n \mid E_1E_2 \ldots E_{n-1})$$

2 Law of Total Probability

An astute person once observed that in a picture like the one in Figure 1, event F can be thought of as having two parts, the part that is in E (that is, $E \cap F = EF$), and the part that isn't ($E^C \cap F = E^CF$). This is true because E and E^C are mutually exclusive sets of outcomes which together cover the entire sample space. After further investigation this was proved to be a general mathematical truth, and there was much rejoicing:

$$P(F) = P(EF) + P(E^CF)$$

This observation is called the **law of total probability**; however, it is most commonly seen in combination with the chain rule:

The Law of Total Probability
For events E and F,

$$P(F) = P(F \mid E)P(E) + P(F \mid E^C)P(E^C)$$

There is a more general version of the rule. If you can divide your sample space into any number of events $E_1, E_2, \ldots E_n$ that are *mutually exclusive* and *exhaustive*—that is, *every* outcome in sample space falls into *exactly one* of those events—then:

$$P(F) = \sum_{i=1}^{n} P(F \mid E_i)P(E_i)$$

The word "total" refers to the fact that the events in E_i must combine to form the totality of the sample space.

3 Bayes' Theorem

Bayes' theorem (or **Bayes' rule**) is one of the most ubiquitous results in probability for computer scientists. Very often we know a conditional probability in one direction, say $P(E \mid F)$, but we would like to know the conditional probability in the other direction. Bayes' theorem provides a way to convert from one to the other. We can derive Bayes' theorem by starting with the definition of conditional probability:

$$P(E \mid F) = \frac{P(F \cap E)}{P(F)}$$

Now we can expand $P(F \cap E)$ using the chain rule, which results in Bayes' theorem.

Bayes' theorem
The most common form of Bayes' theorem is:

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F)}$$

Each term in the Bayes' rule formula has its own name. The $P(E \mid F)$ term is often called the **posterior**; the $P(E)$ term is often called the **prior**; the $P(F \mid E)$ term is called the **likelihood** (or the "update"); and $P(F)$ is often called the **normalization constant**.

If the normalization constant (the probability of the event you were initially conditioning on) is not known, you can expand it using the law of Total Probability:

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F \mid E)P(E) + P(F \mid E^C)P(E^C)} = \frac{P(F \mid E)P(E)}{\sum_i P(F \mid E_i)P(E_i)}$$

Again, for the last version, all the events E_i must be *mutually exclusive* and *exhaustive*.

A common scenario for applying the Bayes Rule formula is when you want to know the probability of something "unobservable" given an "observed" event. For example, you want to know the probability that a student understands a concept, given that you observed them solving a particular problem. It turns out it is much easier to first estimate the probability that a student can solve a problem given that they understand the concept and then to apply Bayes' theorem.

The "expanded" version of Bayes' rule (at the bottom of the Bayes' theorem box) allows you to work around not immediately knowing the denominator $P(F)$. It is worth exploring this in more depth, because this "trick" comes up often, and in slightly different forms. Another way to get to the exact same result is to reason that because the posterior of Bayes Theorem, $P(E \mid F)$, is a probability, we know that $P(E \mid F) + P(E^C \mid F) = 1$. If you expand out $P(E^C \mid F)$ using Bayes, you get:

$$P(E^C \mid F) = \frac{P(F \mid E^C)P(E^C)}{P(F)}$$

Now we have:

$$
\begin{aligned}
1 &= P(E \mid F) + P(E^C \mid F) && \text{since } P(E|F) \text{ is a probability} \\
1 &= \frac{P(F \mid E)P(E)}{P(F)} + \frac{P(F \mid E^C)P(E^C)}{P(F)} && \text{by Bayes' rule (twice)} \\
1 &= \frac{1}{P(F)}\left[P(F \mid E)P(E) + P(F \mid E^C)P(E^C)\right] \\
P(F) &= P(F \mid E)P(E) + P(F \mid E^C)P(E^C)
\end{aligned}
$$

We call $P(F)$ the normalization constant because it is the term whose value can be calculated by making sure that the probabilities of all outcomes sum to 1 (they are "normalized").

4 Conditional Paradigm

As we mentioned above, when you condition on an event you enter the universe where that event has taken place, all the laws of probability still hold. Thus, as long as you condition consistently on the same event, every one of the tools we have learned still apply. Let's look at a few of our old friends when we condition consistently on an event (in this case G):

Name of Rule	**Original Rule**	**Conditional Rule**
First axiom of probability	$0 \leq P(E) \leq 1$	$0 \leq P(E \mid G) \leq 1$
Corollary 1 (complement)	$P(E) = 1 - P(E^C)$	$P(E \mid G) = 1 - P(E^C \mid G)$
Chain Rule	$P(EF) = P(E \mid F)P(F)$	$P(EF \mid G) = P(E \mid FG)P(F \mid G)$
Bayes Theorem	$P(E \mid F) = \frac{P(F\|E)P(E)}{P(F)}$	$P(E \mid FG) = \frac{P(F\|EG)P(E\|G)}{P(F\|G)}$

Chris Piech
CS 109

Lecture Notes #5
October 4, 2017

Independence

Independence

Independence is a big deal in machine learning and probabilistic modeling. Knowing the "joint" probability of many events (the probability of the "and" of the events) requires exponential amounts of data. By making independence and conditional independence claims, computers can essentially decompose how to calculate the joint probability, making it faster to compute, and requiring less data to learn probabilities.

Independence
Two events, E and F, are **independent** if and only if:

$$P(EF) = P(E)P(F)$$

Otherwise, they are called **dependent** events.

This property applies regardless of whether or not E and F are from an equally likely sample space and whether or not the events are mutually exclusive.

The independence principle extends to more than two events. In general, n events $E_1, E_2, \ldots, E_n$ are independent if for every subset with r elements (where $r \leq n$) it holds that:

$$P(E_a, E_b, \ldots, E_r) = P(E_a)P(E_b)\ldots P(E_r)$$

The general definition implies that for three events E, F, G to be independent, *all* of the following must be true:

$$\begin{aligned} P(EFG) &= P(E)P(F)P(G) \\ P(EF) &= P(E)P(F) \\ P(EG) &= P(E)P(G) \\ P(FG) &= P(F)P(G) \end{aligned}$$

Problems with more than two independent events come up frequently. For example: the outcomes of n separate flips of a coin are all independent of one another. Each flip in this case is called a "trial" of the experiment.

In the same way that the mutual exclusion property makes it easier to calculate the probability of the OR of two events, independence makes it easier to calculate the AND of two events.

Example 1: Flipping a Biased Coin

A biased coin is flipped n times. Each flip (independently) comes up heads with probability p, and tails with probability $1 - p$. What is the probability of getting exactly k heads?

Solution: Consider all the possible orderings of heads and tails that result in k heads. There are $\binom{n}{k}$ such orderings, and all of them are mutually exclusive. Since all of the flips are independent, to compute the probability of any one of these orderings, we can multiply the probabilities of each of the heads and each of the tails. There are k heads and $n - k$ tails, so the probability of each ordering is $p^k(1-p)^{n-k}$. Adding up all the different orderings gives us the probability of getting exactly k heads: $\binom{n}{k}p^k(1-p)^{n-k}$.

(Spoiler alert: This is the probability density of a **binomial distribution**. Intrigued by that term? Stay tuned for next week!)

Example 2: Hash Map

Let's consider our friend the hash map. Suppose m strings are hashed (unequally) into a hash table with n buckets. Each string hashed is an independent trial, with probability p_i of getting hashed to bucket i. Calculate the probability of these three events:

A) E = *the first* bucket has ≥ 1 string hashed to it

B) E = *at least 1* of buckets 1 to k has ≥ 1 string hashed to it

C) E = *each* of buckets 1 to k has ≥ 1 string hashed to it

Part A

Let F_i be the event that string i is not hashed into the first bucket. Note that all F_i are independent of one another. By mutual exclusion, $P(F_i) = (p_2 + p_3 + \cdots + p_n)$.

$$\begin{aligned}
P(E) &= 1 - P(E^C) && \text{since } P(A) + (A^C) = 1 \\
&= 1 - P(F_1F_2 \ldots F_m) && \text{definition of } F_i \\
&= 1 - P(F_1)P(F_2) \ldots P(F_m) && \text{since the events are independent} \\
&= 1 - (p_2 + p_3 + \cdots + p_n)^m && \text{calculating } P(F_i) \text{ by mutual exclusion}
\end{aligned}$$

Part B

Let Fi be the event that at least one string is hashed into bucket i. Note that the F_i's are neither independent nor mutually exclusive.

$$\begin{aligned}
P(E) &= P(F_1 \cup F_2 \cup \cdots \cup F_k) \\
&= 1 - P([F_1 \cup F_2 \cup \cdots \cup F_k]^C) && \text{since } P(A) + (A^C) = 1 \\
&= 1 - P({F_1}^C {F_2}^C \ldots {F_k}^C) && \text{by De Morgan's law} \\
&= 1 - (1 - p_1 - p_2 - \cdots - p_k)^m && \text{mutual exclusion, independence of strings}
\end{aligned}$$

The last step is calculated by realizing that $P({F_1}^C {F_2}^C \ldots {F_k}^C)$ is only satisfied by m independent hashes into buckets other than 1 through k.

Part C

Let F_i be the same as in Part B.

$$\begin{aligned}
P(E) &= P(F_1 F_2 \ldots F_k) \\
&= 1 - P([F_1 F_2 \ldots F_k]^C) && \text{since } P(A) + P(A^C) = 1 \\
&= 1 - P({F_1}^C \cup {F_2}^C \cup \cdots \cup {F_k}^C) && \text{by De Morgan's (other) law} \\
&= 1 - P\left(\bigcup_{i=1}^{k} {F_i}^C\right) \\
&= 1 - \sum_{r=1}^{k} (-1)^{r+1} \sum_{i_1 < \cdots < i_r} P({F_{i_1}}^C {F_{i_2}}^C \ldots {F_{i_r}}^C) && \text{by General Inclusion/Exclusion}
\end{aligned}$$

where $P({F_1}^C {F_2}^C \ldots {F_k}^C) = (1 - p_1 - p_2 - \cdots - p_k)^m$ just like in the last problem.

Conditional Independence

Two events E and F are called **conditionally independent** given a third event G, if

$$P(EF \mid G) = P(E \mid G) P(F \mid G)$$

Or, equivalently:

$$P(E \mid FG) = P(E \mid G)$$

Conditioning Breaks Independence

An important caveat about conditional independence is that ordinary independence does not imply conditional independence, nor the other way around.

Knowing when exactly conditioning breaks or creates independence is a big part of building complex probabilistic models; the first few weeks of CS 228 are dedicated to some general principles for reasoning about conditional independence. We will talk about this in another lecture. I included an example in this handout for completeness:

Example 3: Fevers

Let's say a person has a fever if they either have malaria or have an infection. We are going to assume that getting malaria and having an infection are independent: knowing if a person has malaria does not tell us if they have an infection. Now, a patient walks into a hospital with a fever. Your belief that the patient has malaria is high and your belief that the patient has an infection is high. Both explain why the patient has a fever.

Now, given our knowledge that the patient has a fever, gaining the knowledge that the patient has malaria *will* change your belief the patient has an infection. The malaria explains why the patient has a fever, and so the alternate explanation becomes less likely. The two events (which were previously independent) are dependent when conditioned on the patient having a fever.

Chris Piech
CS109

Lecture #6
Oct 6th, 2017

Random Variables and Expectation

Random Variable

A Random Variable (RV) is a variable that probabilistically takes on different values. You can think of an RV as being like a variable in a programming language. They take on values, have types and have domains over which they are applicable. We can define events that occur if the random variable takes one values that satisfy a numerical test (eg does the variable equal 5, is the variable less than 8). We often think of the probabilities of such events.

As an example, let's say we flip three fair coins. We can define a random variable Y to be the total number of "heads" on the three coins. We can ask about the probability of Y taking on different values using the following notation:

- $P(Y = 0) = 1/8$ (T, T, T)
- $P(Y = 1) = 3/8$ (H, T, T), (T, H, T), (T, T, H)
- $P(Y = 2) = 3/8$ (H, H, T), (H, T, H), (T, H, H)
- $P(Y = 3) = 1/8$ (H, H, H)
- $P(Y \geq 4) = 0$

Using random variables is a convenient notation technique that assists in decomposing problems. There are many different types of random variables (indicator, binary, choice, Bernoulli, etc). The two main families of random variable types are discrete and continuous.

Probability Mass Function

For a discrete random variable, the most important thing to know is a mapping between the values that the random variable could take on and the probability of the random variable taking on said value. In mathematics, we call associations functions.

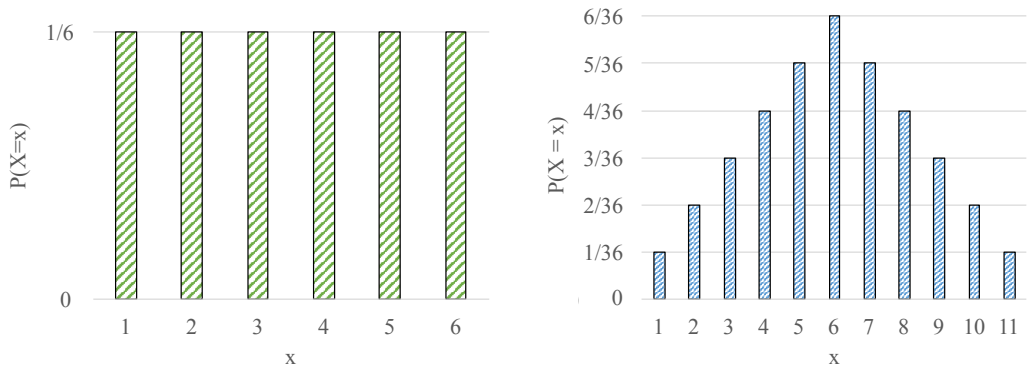


Figure 1: On the left, the PMF of a single 6 sided die roll. On the right, the PMF of the sum of two dice rolls.

The probability mass functions (PMF) maps possible outcomes of a random variable to the corresponding probabilities. Because it is a function, we can plot PMF graphs where the x-axis are the values that the random variable could take on and the y-axis is the probability of the random variable taking on said value:

There are many ways that these Probability Mass Functions can be specified. We could draw a graph. We could have a table (or for you CS folks, a Map) that lists out all the probabilities for all possible events. Or we could write out a mathematical expression.

For example lets consider the random variable X which is the sum of two dice rolls. The probability mass function can be defined by the graph on the right of figure . It could have also been defined using the equation:

$$p_X(x) = \begin{cases} \frac{x}{36} & \text{if } x \in \mathbb{R}\ , 0 \leq x \leq 6 \\ \frac{12-x}{36} & \text{if } x \in \mathbb{R}\ , x \leq 7 \\ 0 & \text{else} \end{cases}$$

The probability mass function, $p_X(x)$, defines the probability of X taking on the value x. The new notation $p_X(x)$ is simply different notation for writing $P(X = x)$. Using this new notation makes it more apparent that we are specifying a function. Try a few values of x, and compare the value of $p_X(x)$ to the graph in figure 1. They should be the same.

Expected Value

A relevant statistic for a random variable is the average value of the random variable over many repetitions of the experiment it represents. This average is called the Expected Value.

The Expected Value for a discrete random variable X is defined as:

$$E[X] = \sum_{x:P(x)>0} xP(x)$$

It goes by many other names: Mean, Expectation, Weighted Average, Center of Mass, 1st Moment.

Example 1

Lets say you roll a 6-Sided Die and that a random variable X represents the outcome of the roll. What is the $E[X]$? This is the same as asking what is the average value.

$$E[X] = 1(1/6) + 2(1/6) + 3(1/6) + 4(1/6) + 5(1/6) + 6(1/6) = 7/2$$

Example 2

Lets say a school has 3 classes with 5, 10, and 150 students. If we randomly choose a class with equal probability and let X = size of the chosen class:

$$\begin{aligned} E[Y] &= 5(1/3) + 10(1/3) + 150(1/3) \\ &= 165/3 = 55 \end{aligned}$$

If instead we randomly choose a student with equal probability and let Y = size of the class the student is in

$$\begin{aligned} E[X] &= 5(5/165) + 10(10/165) + 150(150/165) \\ &= 22635/165 = 137 \end{aligned}$$

Example 3

Consider a game played with a fair coin which comes up heads with p = 0.5. Let n = the number of coin flips before the first "tails". In this game you win $\$2^n$. How many dollars do you expect to win? Let X be a

random variable which represents your winnings.

$$E[X] = \left(\frac{1}{2}\right)^1 2^0 + \left(\frac{1}{2}\right)^2 2^1 + \left(\frac{1}{2}\right)^3 2^2 + \left(\frac{1}{2}\right)^4 2^3 + \cdots = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{i+1} 2^i$$
$$= \sum_{i=0}^{\infty} \frac{1}{2} = \infty$$

Properties of Expectation

Expectations preserve linearity which means that

$$E[aX + b] = aE[X] + b$$

It also holds in the case where you are adding random variables. Regardless of the relationship between random variables, the expectation of the sum is equal to the sum of the expectation. For random variables A and B:

$$E[A + B] = E[A] + E[B] \tag{1}$$

There is a wonderful law called the Law of the Unconcious Statistician that is used to calculate the expected value of a function $g(X)$ of a random variable X when one knows the probability distribution of X but one does not explicitly know the distribution of $g(X)$.

$$E[g(X)] = \sum_x g(x) \cdot p_X(x)$$

For example, lets apply the law of the unconcious statistician to compute the expectation of the square of a random variable (called the second moment).

$$\begin{aligned} E[X^2] &= E[g(X)] && \text{where } g(X) = X^2 \\ &= \sum_x g(x) \cdot p_X(x) && \text{by the unconcious statistician} \\ &= \sum_x x^2 \cdot p_X(x) && \text{by the unconcious statistician} \end{aligned}$$

Disclaimer: This handout was made fresh just for you. Notice any mistakes? Let Chris know.

Chris Piech
CS109

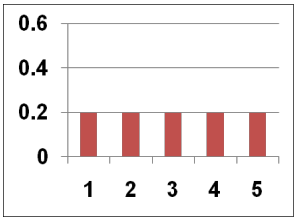
Lecture #7
Oct 9th, 2017

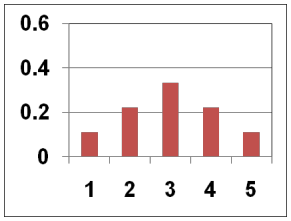
Variance, Bernoulli and Binomials

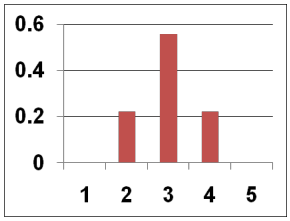
Today we are going to finish up our conversation of functions that we apply to random variables. Last time we talked about expectation, today we will cover variance. Then we will introduce two common, naturally occurring random variable types.

Variance

Consider the following 3 distributions (PMFs)







All three have the same expected value, $E[X] = 3$ but the "spread" in the distributions is quite different. Variance is a formal quantification of "spread".

If X is a random variable with mean μ then the variance of X, denoted $Var(X)$, is: $Var(X) = E[(X-\mu)^2]$. When computing the variance often we use a different form of the same equation: $Var(X) = E[X^2] - E[X]^2$. Intuitively this is the weighted average distance of a sample to the mean.

Here are some useful identities for variance:

- $Var(aX+b) = a^2 Var(X)$
- Standard deviation is the root of variance: $SD(X) = \sqrt{Var(X)}$

Example 1

Let X = value on roll of a 6 sided die. Recall that $E[X] = 7/2$. First lets calculate $E[X^2]$

$$E[X^2] = (1^2)\frac{1}{6} + (2^2)\frac{1}{6} + (3^2)\frac{1}{6} + (4^2)\frac{1}{6} + (5^2)\frac{1}{6} + (6^2)\frac{1}{6} = \frac{91}{6}$$

Which we can use to compute the variance:

$$\begin{aligned} Var(X) &= E[X^2] - (E[X])^2 \\ &= \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12} \end{aligned}$$

Bernoulli

A Bernoulli random variable is random indicator variable (1 = success, 0 = failure) that represents whether or not an experiment with probability p resulted in success. Some example uses include a coin flip, random binary digit, whether a disk drive crashed or whether someone likes a Netflix movie.

Let X be a Bernoulli Random Variable $X \sim Ber(p)$.

$$E[X] = p$$
$$Var(X) = p(1-p)$$

Binomial

A Binomial random variable is random variable that represents the number of successes in n successive independent trials of a Bernoulli experiment. Some example uses include # of heads in n coin flips, # dist drives crashed in 1000 computer cluster.

Let X be a Binomial Random Variable. $X \sim Bin(n, p)$ where p is the probability of success in a given trial.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$
$$E[X] = np$$
$$Var(X) = np(1-p)$$

Example 2

Let X = number of heads after a coin is flipped three times. $X \sim Bin(3, 0.5)$. What is the probability of different outcomes?

$$P(X = 0) = \binom{3}{0} p^0 (1-p)^3 = \frac{1}{8}$$
$$P(X = 1) = \binom{3}{1} p^1 (1-p)^2 = \frac{3}{8}$$
$$P(X = 2) = \binom{3}{2} p^2 (1-p)^1 = \frac{3}{8}$$
$$P(X = 3) = \binom{3}{3} p^3 (1-p)^0 = \frac{1}{8}$$

Example 3

When sending messages over a network there is a chance that the bits will become corrupt. A Hamming Code allows for a 4 bit code to be encoded as 7 bits, and maintains the property that if 0 or 1 bit(s) are corrupted then the message can be perfectly reconstructed. You are working on the Voyager space mission and the probability of any bit being lost in space is 0.1. How does reliability change when using a Hamming code?

Image we use error correcting codes. Let $X \sim Bin(7, 0.1)$

$$P(X = 0) = \binom{7}{0} (0.1)^0 (0.9)^7 \approx 0.468$$
$$P(X = 1) = \binom{7}{1} (0.1)^1 (0.9)^6 = 0.372$$
$$P(X = 0) + P(X = 1) = 0.850$$

What if we didn't use error correcting codes? Let $X \sim Bin(4, 0.1)$

$$P(X = 0) = \binom{4}{0} (0.1)^0 (0.9)^4 \approx 0.656$$

Using Hamming Codes improves reliability by 30%

Chris Piech
CS109

Lecture #8
Oct 11th, 2017

Poisson and More Discrete Distributions

Poisson random variables will be the third main discrete distribution that we expect you to know well. After introducing Poisson, we will quickly introduce three more. I want you to be comfortable with being told the semantics of a distribution, given the key formulas (for expectation, variance and PMF) and then using it.

Binomial in the Limit

Recall example of sending bit string over network. In our last class we used a binomial random variable to represent the number of bits corrupted out of four with a high corruption probability (each bit had independent probability of corruption $p = 0.1$). That example was relevant to sending data to space craft, but for earthly applications like HTML data, voice or video, bit streams are much longer (length $\approx 10^4$) and the probability of corruption of a particular bit is very small ($p \approx 10^{-6}$). Extreme n and p values arise in many cases: # visitors to a website, #server crashes in a giant data center.

Unfortunately X $\sim$ Bin(10^4, 10^{-6}) is unwieldy to compute. However when values get that extreme we can make approximations that are accurate and make computation feasible. Recall the Binomial distribution. First define $\lambda = np$. We can rewrite the Binomial PMF as follows:

$$\begin{aligned} P(X = i) &= \frac{n!}{i!(n-i)!}\left(\frac{\lambda}{n}\right)^i\left(1-\frac{\lambda}{n}\right)^{n-i} \\ &= \frac{n(n-1)\ldots(n-i-1)}{n^i}\frac{\lambda^i}{i!}\frac{(1-\lambda/n)^n}{(1-\lambda/n)^i} \end{aligned}$$

This equation can be made simpler by observing how some of these equations evaluate when n is sufficiently large and p is sufficiently small. The following equations hold:

$$\frac{n(n-1)\ldots(n-i-1)}{n^i} \approx 1 \qquad (1-\lambda/n)^n \approx e^{-\lambda} \qquad (1-\lambda/n)^i \approx 1$$

This reduces our original equation to:

$$P(X = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$

This simplification turns out to be so useful, that in extreme values of n and p we call the approximated Binomial its own random variable type: the Poisson Random Variable.

Poisson Random Variable

A Poisson random variable approximates Binomial where n is large, p is small, and $\lambda = np$ is "moderate". Interestingly, to calculate the things we care about (PMF, expectation, variance) we no longer need to know n and p. We only need to provide λ which we call the rate.

There are different interpretations of "moderate". The accepted ranges are $n > 20$ and $p < 0.05$ or $n > 100$ and $p < 0.1$.

Here are the key formulas you need to know for Poisson. If $Y \sim Poi(\lambda)$:

$$P(Y = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$
$$E[Y] = \lambda$$
$$Var(Y) = \lambda$$

Example

Let's say you want to send a bit string of length $n = 10^4$ where each bit is independently corrupted with $p = 10^{-6}$. What is the probability that the message will arrive uncorrupted? You can solve this using a Poisson with $\lambda = np = 10^4 10^{-6} = 0.01$. Let $X \sim Poi(0.01)$ be the number of corrupted bits. Using the PMF for Poisson:

$$\begin{aligned} P(X=0) &= \frac{\lambda^i}{i!} e^{-\lambda} \\ &= \frac{0.01^0}{0!} e^{-0.01} \\ &\sim 0.9900498 \end{aligned}$$

We could have also modelled X as a binomial such that $X \sim Bin(10^4, 10^{-6})$. That would have been computationally harder to compute but would have resulted in the same number (up to the millionth decimal).

More Discrete Random Variable

We are going to talk about these distributions on Friday. But I wanted to include so you can start thinking beyond Binomial and Poisson :-).

Geometric Random Variable

X is Geometric Random Variable: $X \sim Geo(p)$ if X is number of independent trials until first success and p is probability of success on each trial. Here are the key formulas you need to know. If $X \sim Geo(p)$:

$$P(X=n) = (1-p)^{n-1} p$$
$$E[X] = 1/p$$
$$Var(X) = (1-p)/p^2$$

Negative Binomial Random Variable

X is Negative Binomial: $X \sim NegBin(r,p)$ if X is number of independent trials until r successes and p is probability of success on each trial. Here are the key formulas you need to know. If $X \sim NegBin(p)$:

$$P(X=n) = \binom{n-1}{r-1} p^r (1-p)^{n-r} \text{ where } r \leq n$$
$$E[X] = r/p$$
$$Var(X) = r(1-p)/p^2$$

Zipf Random Variable

X is Zipf: $X \sim Zipf(s)$ if X is the rank index of a chosen word (where s is a parameter of the language).

$$P(X=k) = \frac{1}{k^s \cdot H}$$

Where H is a normalizing constant (and turns out to be equal to the Nth harmonic number where N is the size of the language.

Chris Piech
CS 109

Lecture Notes #9
Oct 13, 2017

Continuous Distributions

So far, all random variables we have seen have been *discrete*. In all the cases we have seen in CS 109, this meant that our RVs could only take on integer values. Now it's time for *continuous random variables*, which can take on values in the real number domain ($\mathbb{R}$). Continuous random variables can be used to represent measurements with arbitrary precision (e.g., height, weight, or time).

1 Probability Density Functions

In the world of discrete random variables, the most important property of a random variable was its probability mass function (PMF), which told you the probability of the random variable taking on a certain value. When we move to the world of continuous random variables, we are going to need to rethink this basic concept. If I were to ask you what the probability is of a child being born with a weight of **exactly** 3.523112342234 kilograms, you might recognize that question as ridiculous. No child will have precisely that weight. Real values are defined with infinite precision; as a result, the probability that a random variable takes on a specific value is not very meaningful when the random variable is continuous. The PMF doesn't apply. We need another idea.

In the continuous world, every random variable has a *probability density function* (PDF), which says how likely it is that a random variable takes on a particular value, relative to other values that it could take on. The PDF has the nice property that you can integrate over it to find the probability that the random variable takes on values within a range (a, b).

X is a **continuous random variable** if there is a function $f(x)$ for $-\infty \leq x \leq \infty$, called the **probability density function** (PDF), such that:

$$P(a \leq X \leq b) = \int_a^b dx\, f(x)$$

To preserve the axioms that guarantee $P(a \leq X \leq b)$ is a probability, the following properties must also hold:

$$0 \leq P(a \leq X \leq b) \leq 1$$
$$P(-\infty < X < \infty) = 1$$

A common misconception is to think of $f(x)$ as a probability. It is instead what we call a probability density. It represents probability *divided by the units of* X. Generally this is only meaningful when we either take an integral over the PDF **or** we *compare* probability densities. As we mentioned when motivating probability densities, the probability that a continuous random variable takes on a specific value (to infinite precision) is 0.

$$P(X = a) = \int_a^a dx\, f(x) = 0$$

This is very different from the discrete setting, in which we often talked about the probability of a random variable taking on a particular value exactly.

2 Cumulative Distribution Function

Having a probability density is great, but it means we are going to have to solve an integral every single time we want to calculate a probability. To save ourselves some effort, for most of these variables we will also compute a *cumulative distribution function* (CDF). The CDF is a function which takes in a number and returns the probability that a random variable takes on a value *less than (or equal to)* that number. If we have a CDF for a random variable, we don't need to integrate to answer probability questions!

For a continuous random variable X, the **cumulative distribution function** is:

$$F_X(a) = P(X \leq a) = \int_{-\infty}^a dx\, f(x)$$

This can be written $F(a)$, without the subscript, when it is obvious which random variable we are using.

Why is the CDF the probability that a random variable takes on a value *less than* (or equal to) the input value as opposed to greater than? It is a matter of convention. But it is a useful convention. Most probability questions can be solved simply by knowing the CDF (and taking advantage of the fact that the integral over the range $-\infty$ to ∞ is 1). Here are a few examples of how you can answer probability questions by just using a CDF:

Probability Query	**Solution**	**Explanation**
$P(X \leq a)$	$F(a)$	This is the definition of the CDF
$P(X < a)$	$F(a)$	Note that $P(X = a) = 0$
$P(X > a)$	$1 - F(a)$	$P(X \leq a) + P(X > a) = 1$
$P(a < X < b)$	$F(b) - F(a)$	$F(a) + P(a < X < b) = F(b)$

As we mentioned briefly earlier, the cumulative distribution function can also be defined for discrete random variables, but there is less utility to a CDF in the discrete world, because with the exception of the geometric random variable, none of our discrete random variables had "closed form" (that is, without any summations) functions for the CDF:

$$F_X(a) = \sum_{i=0}^{a} P(X = i)$$

Example 1

Let X be a continuous random variable (CRV) with PDF:

$$f(x) = \begin{cases} C(4x - 2x^2) & \text{when } 0 < x < 2 \\ 0 & \text{otherwise} \end{cases}$$

In this function, C is a constant. What value is C? Since we know that the PDF must sum to 1:

$$\int_0^2 dx\, C(4x - 2x^2) = 1$$

$$C\left(2x^2 - \frac{2x^3}{3}\right)\Big|_{x=0}^{2} = 1$$

$$C\left(\left(8 - \frac{16}{3}\right) - 0\right) = 1$$

Solving this equation for C gives $C = 3/8$.

What is $P(X > 1)$?

$$\int_1^\infty dx\, f(x) = \int_1^2 dx\, \frac{3}{8}(4x - 2x^2) = \frac{3}{8}\left(2x^2 - \frac{2x^3}{3}\right)\Big|_{x=1}^{2} = \frac{3}{8}\left[\left(8 - \frac{16}{3}\right) - \left(2 - \frac{2}{3}\right)\right] = \frac{1}{2}$$

Example 2

Let X be a RV representing the number of days of use before your disk crashes, with PDF:

$$f(x) = \begin{cases} \lambda e^{-x/100} & \text{when } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

First, determine λ. Recall that $\int Ae^{Au}du = e^{Au}$:

$$\int_0^\infty dx\, \lambda e^{-x/100} = 1$$

$$-100\lambda \int_0^\infty dx\, \frac{-1}{100} e^{-x/100} = 1$$

$$-100\lambda \cdot e^{-x/100}\Big|_{x=0}^{\infty} = 1$$

$$100\lambda \cdot 1 = 1 \quad \Rightarrow \quad \lambda = 1/100$$

What is $P(X < 10)$?

$$F(10) = \int_0^{10} dx\, \frac{1}{100} e^{-x/100} = -e^{-x/100}\Big|_{x=0}^{10} = -e^{-1/10} + 1 \approx 0.095$$

3 Expectation and Variance

For continuous RV X:

$$E[X] = \int_{-\infty}^{\infty} dx\, x \cdot f(x)$$
$$E[g(X)] = \int_{-\infty}^{\infty} dx\, g(x) \cdot f(x)$$
$$E[X^n] = \int_{-\infty}^{\infty} dx\, x^n \cdot f(x)$$

For both continuous and discrete RVs:

$$E[aX + b] = aE[X] + b$$
$$\text{Var}(X) = E[(X - \mu)^2] = E[X^2] - (E[X])^2 \qquad \text{(with } \mu = E[X])$$
$$\text{Var}(aX + b) = a^2 \,\text{Var}(X)$$

4 Uniform Random Variable

The most basic of all the continuous random variables is the uniform random variable, which is equally likely to take on any value in its range (α, β).

X is a **uniform random variable** ($X \sim \text{Uni}(\alpha, \beta)$) if it has PDF:

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{when } \alpha \le x \le \beta \\ 0 & \text{otherwise} \end{cases}$$

Notice how the density $1/(\beta - \alpha)$ is exactly the same regardless of the value for x. That makes the density uniform. So why is the PDF $1/(\beta - \alpha)$ and not 1? That is the constant that makes it such that the integral over all possible inputs evaluates to 1.

The key properties of this RV are:

$$P(a \le X \le b) = \int_a^b dx\, f(x) = \frac{b - a}{\beta - \alpha} \text{ (for } \alpha \le a \le b \le \beta)$$
$$E[X] = \int_{-\infty}^{\infty} dx\, x \cdot f(x) = \int_{\alpha}^{\beta} dx\, \frac{x}{\beta - \alpha} = \frac{x^2}{2(\beta - \alpha)}\bigg|_{x=\alpha}^{\beta} = \frac{\alpha + \beta}{2}$$
$$\text{Var}(X) = \frac{(\beta - \alpha)^2}{12}$$

5 Exponential Random Variable

An **exponential random variable** ($X \sim \text{Exp}(\lambda)$) represents the time until an event occurs. It is parametrized by $\lambda > 0$, the (constant) rate at which the event occurs. This is the same λ as in the Poisson distribution; a Poisson variable counts the number of events that occur in a fixed interval, while an exponential variable measures the amount of time until the next event occurs.

(Example 2 sneakily introduced you to the exponential distribution already; now we get to use formulas we've already computed to work with it without integrating anything.)

Properties

The probability density function (PDF) for an exponential random variable is:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

The expectation is $E[X] = \frac{1}{\lambda}$ and the variance is $\text{Var}(X) = \frac{1}{\lambda^2}$

There is a closed form for the cumulative distribution function (CDF):

$$F(x) = 1 - e^{-\lambda x} \text{ where } x \geq 0$$

Example 3

Let X be a random variable that represents the number of minutes until a visitor leaves your website. You have calculated that on average a visitor leaves your site after 5 minutes, and you decide that an exponential distribution is appropriate to model how long a person stays before leaving the site. What is the $P(X > 10)$?

We can compute $\lambda = \frac{1}{5}$ either using the definition of $E[X]$ or by thinking of how many people leave every minute (answer: "one-fifth of a person"). Thus $X \sim \text{Exp}(1/5)$.

$$\begin{aligned} P(X > 10) &= 1 - F(10) \\ &= 1 - (1 - e^{-\lambda \cdot 10}) \\ &= e^{-2} \approx 0.1353 \end{aligned}$$

Example 4

Let X be the number of hours of use until your laptop dies. On average laptops die after 5000 hours of use. If you use your laptop for 7300 hours during your undergraduate career (assuming usage = 5 hours/day and four years of university), what is the probability that your laptop lasts all four years?

As above, we can find λ either using $E[X]$ or thinking about laptop deaths per hour: $X \sim \text{Exp}(\frac{1}{5000})$.

$$\begin{aligned} P(X > 7300) &= 1 - F(7300) \\ &= 1 - (1 - e^{-7300/5000}) \\ &= e^{-1.46} \approx 0.2322 \end{aligned}$$

Chris Piech
CS109

Lecture #10
Sept 16th, 2017

Gaussian

Normal Random Variable

The single most important random variable type is the Normal (aka Gaussian) random variable, parametrized by a mean (μ) and variance (σ^2). If X is a normal variable we write $X \sim \mathcal{N}(\mu, \sigma^2)$. The normal is important for many reasons: it is generated from the summation of independent random variables and as a result it occurs often in nature. Many things in the world are not distributed normally but data scientists and computer scientists model them as Normal distributions anyways. Why? Because it is the most entropic (conservative) distribution that we can apply to data with a measured mean and variance.

Properties

The Probability Density Function (PDF) for a Normal is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

By definition a Normal has $E[X] = \mu$ and $Var(X) = \sigma^2$.

If X is a Normal such that $X \sim \mathcal{N}(\mu, \sigma^2)$ and Y is a linear transform of X such that $Y = aX + b$ then Y is also a Normal where $Y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$.

There is no closed form for the integral of the Normal PDF, however since a linear transform of a Normal produces another Normal we can always map our distribution to the "Standard Normal" (mean 0 and variance 1) which has a precomputed Cumulative Distribution Function (CDF). The CDF of an arbitrary normal is:

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

Where Φ is a precomputed function that represents that CDF of the Standard Normal.

Projection to Standard Normal

For any Normal X we can define a random variable $Z \sim \mathcal{N}(0,1)$ to be a linear transform

$$\begin{aligned} Z &= \frac{X-\mu}{\sigma} = \frac{1}{\sigma}X - \frac{\mu}{\sigma} \\ &\sim \mathcal{N}(\frac{\mu}{\sigma} - \frac{\mu}{\sigma}, \frac{\sigma^2}{\sigma^2}) \\ &\sim \mathcal{N}(0,1) \end{aligned}$$

Using this transform we can express $F_X(x)$, the CDF of X, in terms of the known CDF of Z, $F_Z(x)$. Since the CDF of Z is so common it gets its own Greek symbol: $\Phi(x)$

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ &= P\left(\frac{X-\mu}{\sigma} \leq \frac{x-\mu}{\sigma}\right) \\ &= P\left(Z \leq \frac{x-\mu}{\sigma}\right) \\ &= \Phi\left(\frac{x-\mu}{\sigma}\right) \end{aligned}$$

The values of $\Phi(x)$ can be looked up in a table. We also have an online calculator.

Example 1

Let $X \sim \mathcal{N}(3,16)$, what is $P(X > 0)$?

$$P(X > 0) = P\left(\frac{X-3}{4} > \frac{0-3}{4}\right) = P\left(Z > -\frac{3}{4}\right) = 1 - P\left(Z \leq -\frac{3}{4}\right)$$
$$= 1 - \Phi(-\frac{3}{4}) = 1 - (1 - \Phi(\frac{3}{4})) = \Phi(\frac{3}{4}) = 0.7734$$

What is $P(2 < X < 5)$?

$$P(2 < X < 5) = P\left(\frac{2-3}{4} < \frac{X-3}{4} < \frac{5-3}{4}\right) = P\left(-\frac{1}{4} < Z < \frac{2}{4}\right)$$
$$= \Phi(\frac{2}{4}) - \Phi(-\frac{1}{4}) = \Phi(\frac{1}{2}) - (1 - \Phi(\frac{1}{4})) = 0.2902$$

Example 2

You send voltage of 2 or -2 on a wire to denote 1 or 0. Let X = voltage sent and let R = voltage received. $R = X + Y$, where $Y \sim \mathcal{N}(0,1)$ is noise. When decoding, if $R \geq 0.5$ we interpret the voltage as 1, else 0. What is $P(\text{error after decoding} | \text{original bit} = 1)$?

$$P(X + Y < 0.5) == P(2 + Y < 0.5) = P(Y < -1.5) = \Phi(-1.5) = 1 - \Phi(1.5) \approx 0.0668$$

Binomial Approximation

You can use a Normal distribution to approximate a Binomial $X \sim Bin(n,p)$. To do so define a normal $Y \sim (E[X], Var(X))$. Using the Binomial formulas for expectation and variance, $Y \sim (np, np(1-p))$. This approximation holds for large n. Since a Normal is continuous and Binomial is discrete we have to use a continuity correction to discretize the Normal.

$$P(X = k) \sim P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

Example 3

100 visitors to your website are given a new design. Let X = # of people who were given the new design and spend more time on your website. Your CEO will endorse the new design if $X \geq 65$. What is $P(\text{CEO endorses change} | \text{it has no effect})$?

$E[X] = np = 50$. $Var(X) = np(1-p) = 25$. $\sigma = \sqrt{(Var(X))} = 5$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(50, 25)$.

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y - 50}{5} > \frac{64.5 - 50}{5}\right) = 1 - \Phi(2.9) = 0.0019$$

Example 4

Stanford accepts 2480 students and each student has a 68% chance of attending. Let X = # students who will attend. $X \sim Bin(2480, 0.68)$. What is $P(X > 1745)$?

$E[X] = np = 1686.4$. $Var(X) = np(1-p) = 539.7$. $\sigma = \sqrt{(Var(X))} = 23.23$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(1686.4, 539.7)$.

$$P(X > 1745) \approx P(Y > 1745.5) = P\left(\frac{Y - 1686.4}{23.23} > \frac{1745.5 - 1686.4}{23.23}\right) = 1 - \Phi(2.54) = 0.0055$$

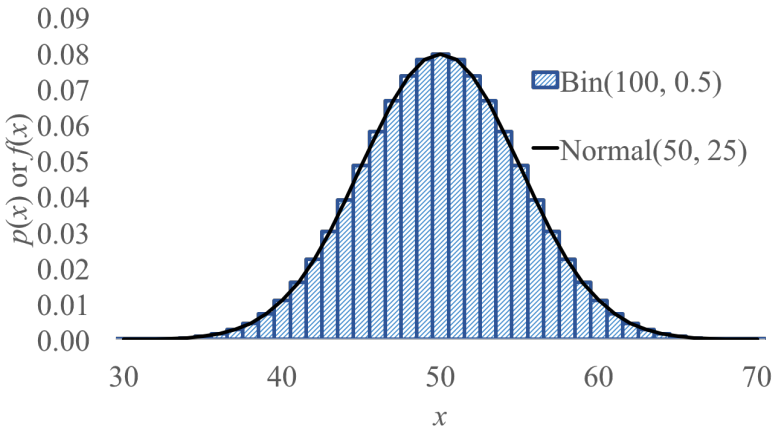
Chris Piech
CS109

Lecture #11
Oct 18th, 2017

Binomial Approximation and Joint Distributions

Binomial Approximation

For certain values, a normal can be used to approximate a Binomial. Let's take a side by side view of a normal and a binomial:



Lets say our binomial is a random variable $X \sim \text{Bin}(100, 0.5)$ and we want to calculate $P(X \geq 55)$. We could cheat by using the closest fit normal (in this case $Y \sim N(50, 25)$). How did we chose that particular Normal? Simply select one with a mean and variance that matches the Binomial expectation and variance. The binomial expectation is $np = 100 \cdot 0.5 = 50$. The Binomial variance is $np(1-p) = 100 \cdot 0.5 \cdot 0.5 = 25$.

You can use a Normal distribution to approximate a Binomial $X \sim Bin(n, p)$. To do so define a normal $Y \sim (E[X], Var(X))$. Using the Binomial formulas for expectation and variance, $Y \sim (np, np(1-p))$. This approximation holds for large n and moderate p. Since a Normal is continuous and Binomial is discrete we have to use a continuity correction to discretize the Normal.

$$P(X = k) \sim P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

You should get comfortable deciding what continuity correction to use. Here are a few examples of discrete probability questions and the continuity correction:

Discrete (Binomial) probability question	Equivalent continuous probability question
$P(X = 6)$	$P(0.5 < X < 6.5)$
$P(X \geq 6)$	$P(X > 5.5)$
$P(X > 6)$	$P(X > 6.5)$
$P(X < 6)$	$P(X < 5.5)$
$P(X \leq 6)$	$P(X < 6.5)$

Example 3

100 visitors to your website are given a new design. Let X = # of people who were given the new design and spend more time on your website. Your CEO will endorse the new design if $X \geq 65$. What is

$P(\text{CEO endorses change}|\text{it has no effect})$?

$E[X] = np = 50$. $Var(X) = np(1-p) = 25$. $\sigma = \sqrt{(Var(X))} = 5$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(50, 25)$.

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y-50}{5} > \frac{64.5-50}{5}\right) = 1 - \Phi(2.9) = 0.0019$$

Example 4

Stanford accepts 2480 students and each student has a 68% chance of attending. Let X = # students who will attend. $X \sim Bin(2480, 0.68)$. What is $P(X > 1745)$?

$E[X] = np = 1686.4$. $Var(X) = np(1-p) = 539.7$. $\sigma = \sqrt{(Var(X))} = 23.23$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(1686.4, 539.7)$.

$$P(X > 1745) \approx P(Y > 1745.5) = P\left(\frac{Y-1686.4}{23.23} > \frac{1745.5-1686.4}{23.23}\right) = 1 - \Phi(2.54) = 0.0055$$

Joint Distributions

Often you will work on problems where there are several random variables (often interacting with one another). We are going to start to formally look at how those interactions play out.

For now we will think of joint probabilities with two events X and Y.

Discrete Case

In the discrete case a joint probability mass function tells you the probability of any combination of events $X = a$ and $Y = b$:

$$p_{X,Y}(a,b) = P(X = a, Y = b)$$

This function tells you the probability of all combinations of events (the "," means "and"). If you want to back calculate the probability of an event only for one variable you can calculate a "marginal" from the joint probability mass function:

$$p_X(a) = P(X = a) = \sum_y P_{X,Y}(a,y)$$
$$p_Y(b) = P(Y = b) = \sum_x P_{X,Y}(x,b)$$

In the continuous case a joint probability density function tells you the relative probability of any combination of events $X = a$ and $Y = y$.

In the discrete case, we can define the function $p_{X,Y}$ non-parametrically. Instead of using a formula for p we simply state the probability of each possible outcome.

Multinomial Distribution

Say you perform n independent trials of an experiment where each trial results in one of m outcomes, with respective probabilities: $p_1, p_2, \ldots, p_m$ (constrained so that $\sum_i p_i = 1$). Define X_i to be the number of trials with outcome i. A multinomial distribution is a closed form function that answers the question: What is the probability that there are c_i trials with outcome i. Mathematically:

$$P(X_1 = c_1, X_2 = c_2, \ldots, X_m = c_m) = \binom{n}{c_1, c_2, \ldots, c_m} p_1^{c_1} p_2^{c_2} \ldots p_m^{c_m}$$

Example 1

A 6-sided die is rolled 7 times. What is the probability that you roll: 1 one, 1 two, 0 threes, 2 fours, 0 fives, 3 sixes (disregarding order).

$$P(X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 2, X_5 = 0, X_6 = 3) = \frac{7!}{2!3!}\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^2\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^3$$
$$= 420\left(\frac{1}{6}\right)^7$$

Fedaralist Papers

In class we wrote a program to decide whether or not James Madison or Alexander Hamilton wrote Fedaralist Paper 49. Both men have claimed to be have written it, and hence the authorship is in dispute. First we used historical essays to estimate p_i, the probability that Hamilton generates the word i (independent of all previous and future choices or words). Similarly we estimated q_i, the probability that Madison generates the word i. For each word i we observe the number of times that word occurs in Fedaralist Paper 49 (we call that count c_i). We assume that, given no evidence, the paper is equally likely to be written by Madison or Hamilton.

Define three events: H is the event that Hamilton wrote the paper, M is the event that Madison wrote the paper, and D is the event that a paper has the collection of words observed in Fedaralist Paper 49. We would like to know whether $P(H|D)$ is larger than $P(M|D)$. This is equivalent to trying to decide if $P(H|D)/P(M|D)$ is larger than 1.

The event $D|H$ is a multinomial parameterized by the values p. The event $D|M$ is also a multinomial, this time parameterized by the values q.

Using Bayes Rule we can simplify the desired probability.

$$\frac{P(H|D)}{P(M|D)} = \frac{\frac{P(D|H)P(H)}{P(D)}}{\frac{P(D|M)P(M)}{P(D)}} = \frac{P(D|H)P(H)}{P(D|M)P(M)} = \frac{P(D|H)}{P(D|M)}$$
$$= \frac{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i p_i^{c_i}}{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i q_i^{c_i}} = \frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}$$

This seems great! We have our desired probability statement expressed in terms of a product of values we have already estimated. However, when we plug this into a computer, both the numerator and denominator come out to be zero. The product of many numbers close to zero is too hard for a computer to represent. To fix this problem, we use a standard trick in computational probability: we apply a log to both sides and apply some basic rules of logs.

$$\log\left(\frac{P(H|D)}{P(M|D)}\right) = \log\left(\frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}\right)$$
$$= \log(\prod_i p_i^{c_i}) - \log(\prod_i q_i^{c_i})$$
$$= \sum_i \log(p_i^{c_i}) - \sum_i \log(q_i^{c_i})$$
$$= \sum_i c_i \log(p_i) - \sum_i c_i \log(q_i)$$

This expression is "numerically stable" and my computer returned that the answer was a negative number. We can use exponentiation to solve for $P(H|D)/P(M|D)$. Since the exponent of a negative number is a number smaller than 1, this implies that $P(H|D)/P(M|D)$ is smaller than 1. As a result, we conclude that Madison was more likely to have written Fedaralist Paper 49.

Chris Piech
CS109

Lecture #12
Oct 20th, 2017

Continuous Joints

Continuous Joint Distributions

Random variables X and Y are Jointly Continuous if there exists a Probability Density Function (PDF) $f_{X,Y}$ such that:

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = \int_{a_1}^{a_2} \int_{b_1}^{b_2} f_{X,Y}(x,y) dy\, dx$$

Using the PDF we can compute marginal probability densities:

$$f_X(a) = \int_{-\infty}^{\infty} f_{X,Y}(a,y) dy$$
$$f_Y(b) = \int_{-\infty}^{\infty} f_{X,Y}(x,b) dx$$

Lemmas

Here are two useful lemmas. Let $F(a,b)$ be the Cumulative Density Function (CDF):

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = F(a_2,b_2) - F(a_1,b_2) + F(a_1,b_1) - F(a_2,b_1)$$

And did you know that if Y is a non-negative random variable the following hold (for discrete and continuous random variables respectively):

$$E[Y] = \sum_{i=1}^{n} P(Y \geq i)$$
$$E[Y] = \int_{0}^{\infty} P(Y \geq i) di$$

Example 3

A disk surface is a circle of radius R. A single point imperfection is uniformly distributed on the disk with joint PDF:

$$f_{X,Y}(x,y) = \begin{cases} \frac{1}{\pi R^2} & \text{if } x^2 + y^2 \leq R^2 \\ 0 & \text{else} \end{cases}$$

Let D be the distance from the origin: $D = \sqrt{X^2 + Y^2}$. What is $E[D]$? Hint: use the lemmas

Example 4

Lets make a weight matrix used for Gaussian blur. In the weight matrix, each location in the weight matrix will be given a weight based on the probability density of the area covered by that grid square in a 2D Gaussian with variance σ^2. For this example lets blur using $\sigma = 3$.

In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise.

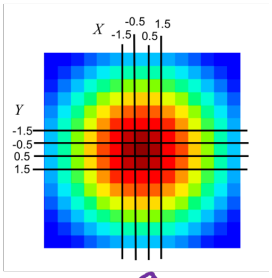
Gaussian blurring with StDev = 3, is based on a joint probability distribution:

Joint PDF

$$f_{X,Y}(x,y) = \frac{1}{2\pi \cdot 3^2} e^{-\frac{x^2+y^2}{2\cdot 3^2}}$$

Joint CDF

$$F_{X,Y}(x,y) = \Phi\left(\frac{x}{3}\right) \cdot \Phi\left(\frac{y}{3}\right)$$



Used to generate this weight matrix

Each pixel is given a weight equal to the probability that X and Y are both within the pixel bounds. The center pixel covers the area where $-0.5 \le x \le 0.5$ and $-0.5 \le y \le 0.5$ What is the weight of the center pixel?

$$\begin{aligned}
&P(-0.5 < X < 0.5, -0.5 < Y < 0.5) \\
=&P(X < 0.5, Y < 0.5) - P(X < 0.5, Y < -0.5) \\
&- P(X < -0.5, Y < 0.5) + P(X < -0.5, Y < -0.5) \\
=&\phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{0.5}{3}\right) - 2\phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) \\
&+ \phi\left(\frac{-0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) \\
=&0.5662^2 - 2 \cdot 0.5662 \cdot 0.4338 + 0.4338^2 = 0.206
\end{aligned}$$

Chris Piech
CS109

Lecture #13
Oct 23rd, 2017

Properties of Joint Distributions

Expectation with Multiple RVs

Expectation over a joint isn't nicely defined because it is not clear how to compose the multiple variables. However, expectations over functions of random variables (for example sums or multiplications) are nicely defined: $E[g(X,Y)] = \sum_{x,y} g(x,y)p(x,y)$ for any function $g(X,Y)$. When you expand that result for the function $g(X,Y) = X+Y$ you get a beautiful result:

$$\begin{aligned} E[X+Y] = E[g(X,Y)] &= \sum_{x,y} g(x,y)p(x,y) = \sum_{x,y} [x+y]p(x,y) \\ &= \sum_{x,y} xp(x,y) + \sum_{x,y} yp(x,y) \\ &= \sum_{x} x\sum_{y} p(x,y) + \sum_{y} y\sum_{x} p(x,y) \\ &= \sum_{x} xp(x) + \sum_{y} yp(y) \\ &= E[X] + E[Y] \end{aligned}$$

This can be generalized to multiple variables:

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

Independence with Multiple RVs

Discrete

Two discrete random variables X and Y are called independent if:

$$P(X = x, Y = y) = P(X = x)P(Y = y) \text{ for all } x, y$$

Intuitively: knowing the value of X tells us nothing about the distribution of Y. If two variables are not independent, they are called dependent. This is a similar conceptually to independent events, but we are dealing with multiple *variables*. Make sure to keep your events and variables distinct.

Continuous

Two continuous random variables X and Y are called independent if:

$$P(X \leq a, Y \leq b) = P(X \leq a)P(Y \leq b) \text{ for all } a, b$$

This can be stated equivalently as:

$$F_{X,Y}(a,b) = F_X(a)F_Y(b) \text{ for all } a, b$$
$$f_{X,Y}(a,b) = f_X(a)f_Y(b) \text{ for all } a, b$$

More generally, if you can factor the joint density function then your continuous random variable are independent:

$$f_{X,Y}(x,y) = h(x)g(y) \text{ where } -\infty < x, y < \infty$$

Example 2

Let N be the # of requests to a web server/day and that $N \sim Poi(\lambda)$. Each request comes from a human (probability = p) or from a "bot" (probability = $(1-p)$), independently. Define X to be the # of requests from humans/day and Y to be the # of requests from bots/day.

Since requests come in independently, the probability of X conditioned on knowing the number of requests is a Binomial. Specifically:

$$(X|N) \sim Bin(N, p)$$
$$(Y|N) \sim Bin(N, 1-p)$$

Calculate the probability of getting exactly i human requests and j bot requests. Start by expanding using the chain rule:

$$P(X = i, Y = j) = P(X = i, Y = j | X + Y = i + j) P(X + Y = i + j)$$

We can calculate each term in this expression:

$$P(X = i, Y = j | X + Y = i + j) = \binom{i+j}{i} p^i (1-p)^j$$
$$P(X + Y = i + j) = e^{-\lambda} \frac{\lambda^{i+j}}{(i+j)!}$$

Now we can put those together and simplify:

$$P(X = i, Y = j) = \binom{i+j}{i} p^i (1-p)^j e^{-\lambda} \frac{\lambda^{i+j}}{(i+j)!}$$

As an exercise you can simplify this expression into two independent Poisson distributions.

Symmetry of Independence

Independence is symmetric. That means that if random variables X and Y are independent, X is independent of Y and Y is independent of X. This claim may seem meaningless but it can be very useful. Imagine a sequence of events $X_1, X_2, \ldots$. Let A_i be the event that X_i is a "record value" (eg it is larger than all previous values). Is A_{n+1} independent of A_n? It is easier to answer that A_n is independent of A_{n+1}. By symmetry of independence both claims must be true.

Conditional Distributions

Before we looked at conditional probabilities for events. Here we formally go over conditional probabilities for random variables. The equations for both the discrete and continuous case are intuitive extensions of our understanding of conditional probability:

Discrete

The conditional probability mass function (PMF) for the discrete case:

$$p_{X|Y}(x|y) = P(X = x | Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{P_{X,Y}(x, y)}{p_Y(y)}$$

The conditional cumulative density function (CDF) for the discrete case:

$$F_{X|Y}(a|y) = P(X \leq a | Y = y) = \frac{\sum_{x \leq a} p_{X,Y}(x, y)}{p_Y(y)} = \sum_{x \leq a} p_{X|Y}(x|y)$$

Continuous

The conditional probability density function (PDF) for the continuous case:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

The conditional cumulative density function (CDF) for the continuous case:

$$F_{X|Y}(a|y) = P(X \leq a|Y = y) = \int_{-\infty}^{a} f_{X|Y}(x|y)dx$$

Example 2

Let's say we have two independent random Poisson variables for requests received at a web server in a day: X = # requests from humans/day, $X \sim Poi(\lambda_1)$ and Y = # requests from bots/day, $Y \sim Poi(\lambda_2)$. Since the convolution of Poisson random variables is also a Poisson we know that the total number of requests $(X+Y)$ is also a Poisson $(X+Y) \sim Poi(\lambda_1+\lambda_2)$. What is the probability of having k human requests on a particular day given that there were n total requests?

$$\begin{aligned}
P(X = k|X+Y = n) &= \frac{P(X = k, Y = n-k)}{P(X+Y = n)} = \frac{P(X = k)P(Y = n-k)}{P(X+Y = n)} \\
&= \frac{e^{-\lambda_1}\lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2}\lambda_2^{n-k}}{(n-k)!} \cdot \frac{n!}{e^{1(\lambda_1+\lambda_2)}(\lambda_1+\lambda_2)^n} \\
&= \binom{n}{k} \left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right)^k \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right)^{n-k} \\
&\sim Bin\left(n, \frac{\lambda_2}{\lambda_1+\lambda_2}\right)
\end{aligned}$$

Chris Piech
CS109

Lecture #14
Oct 25th, 2017

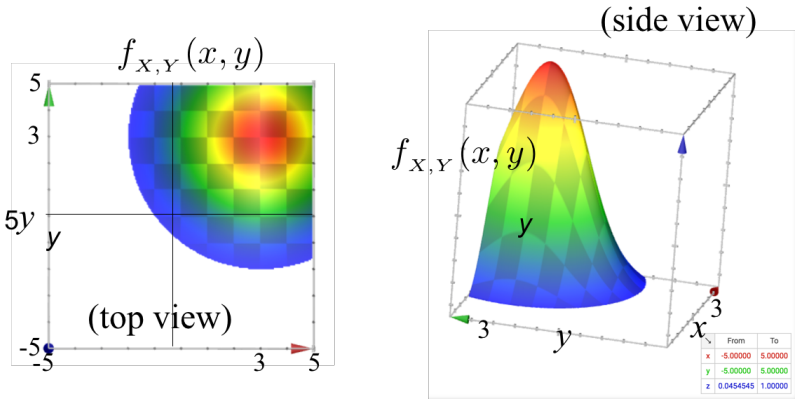
Tracking in 2D Space

In this example we are going to explore the problem of tracking an object in 2D space. The object exists at some (x,y) location, however we are not sure exactly where! Thus we are going to use random variables X and Y to represent location.

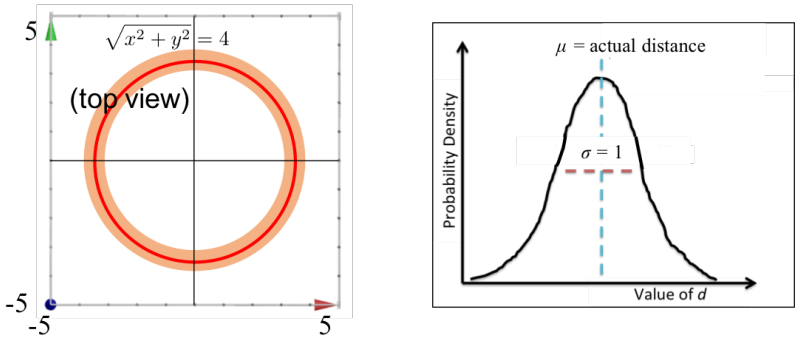
We have a prior belief about where the object is. In this example our prior both X and Y as normals which are independently distributed with mean 3 and variance 2. First let's write the prior belief as a joint probability density function

$$\begin{aligned} f(X=x,Y=y) &= f(X=x)\cdot f(Y=y) && \text{In the prior X and Y are independent} \\ &= \frac{1}{\sqrt{2\cdot 4\cdot \pi}}\cdot e^{-\frac{(x-3)^2}{2\cdot 4}}\cdot \frac{1}{\sqrt{2\cdot 4\cdot \pi}}\cdot e^{-\frac{(y-3)^2}{2\cdot 4}} && \text{Using the PDF equation for normals} \\ &= K_1\cdot e^{-\frac{(x-3)^2+(y-3)^2}{8}} && \text{All constants are put into } K_1 \end{aligned}$$

This combinations of normals is called a bivariate distribution. Here is a visualization of the PDF of our prior.



The interesting part about tracking an object is the process of updating your belief about it's location based on an observation. Let's say that we get an instrument reading from a sonar that is sitting on the origin. The instrument reports that the object is 4 units away. Our instrument is not perfect: if the true distance was t units away, than the instrument will give a reading which is normally distributed with mean t and variance 1. Let's visualize the observation:



Based on this information about the noisiness of our prior, we can compute the conditional probability of seeing a particular distance reading D, given the true location of the object X, Y. If we knew the object was at location (x,y), we could calculate the true distance to the origin $\sqrt{x^2+y^2}$ which would give us the mean for the instrument Gaussian:

$$f(D=d|X=x,Y=y) = \frac{1}{\sqrt{2\cdot 1\cdot \pi}}\cdot e^{-\frac{\left(d-\sqrt{x^2+y^2}\right)^2}{2\cdot 1}} \qquad \text{Normal PDF where } \mu=\sqrt{x^2+y^2}$$

$$= K_2\cdot e^{-\frac{\left(d-\sqrt{x^2+y^2}\right)^2}{2\cdot 1}} \qquad \text{All constants are put into } K_2$$

How about we try this out on actual numbers. How much more likely is an instrument reading of 1 compared to 2, given that the location of the object is at (1, 1)?

$$\frac{f(D=1|X=1,Y=1)}{f(D=2|X=1,Y=1)} = \frac{K_2\cdot e^{-\frac{\left(1-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}}{K_2\cdot e^{-\frac{\left(2-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}} \qquad \text{Substituting into the conditional PDF of D}$$

$$= \frac{e^0}{e^{-1/2}} \approx 1.65 \qquad \text{Notice how the } K_2 \text{ cancel out}$$

At this point we have a prior belief and we have an observation. We would like to compute an updated belief, given that observation. This is a classic Bayes' formula scenario. We are using joint continuous variables, but that doesn't change the math much, it just means we will be dealing with densities instead of probabilities:

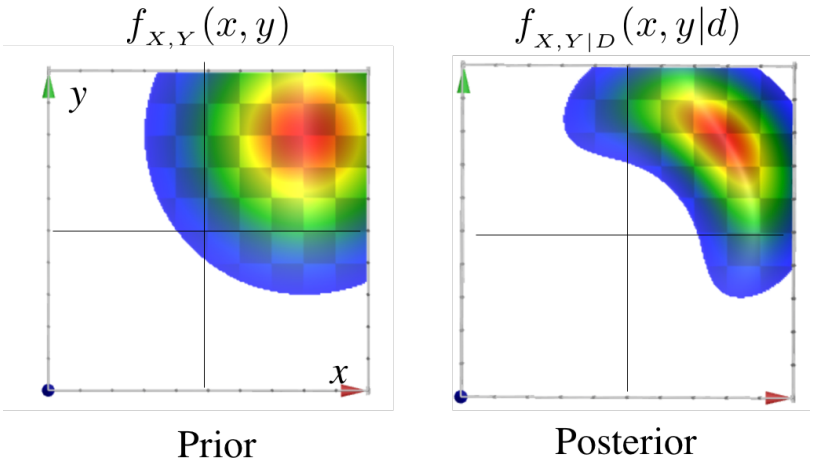
$$f(X=x,Y=y|D=4) = \frac{f(D=4|X=x,Y=y)\cdot f(X=x,Y=y)}{f(D=4)} \qquad \text{Bayes using densities}$$

$$= \frac{K_1\cdot e^{-\frac{[4-\sqrt{x^2+y^2})^2]}{2}}\cdot K_2\cdot e^{-\frac{[(x-3)^2+(y-3)^2]}{8}}}{f(D=4)} \qquad \text{Substituting for prior and update}$$

$$= \frac{K_1\cdot K_2}{f(D=4)}\cdot e^{-\left[\frac{[4-\sqrt{x^2+y^2})^2]}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} \qquad f(D=4) \text{ is a constant w.r.t. } (x,y)$$

$$= K_3\cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} \qquad K_3 \text{ is a new constant}$$

Wow! That looks like a pretty interesting function! You have successfully computed the updated belief. Let's see what it looks like. Here is a figure with our prior on the left and the posterior on the right: How beautiful



is that! Its like a 2D normal distribution merged with a circle. But wait, what about that constant! We do

not know the value of K_3 and that is not a problem for two reasons: the first reason is that if we ever want to calculate a relative probability of two locations, K_3 will cancel out. The second reason is that if we really wanted to know what K_3 was, we could solve for it.

This math is used every day in millions of applications. If there are multiple observations the equations can get truly complex (even worse than this one). To represent these complex functions often use an algorithm called particle filtering.

Chris Piech
CS109

Lecture #15
Oct 27th, 2017

Convolution

Convolution is the result of adding two different random variables together. For some particular random variables computing convolution has intuitive closed form equations. Importantly convolution is the sum of the random variables themselves, not the addition of the probability density functions (PDF)s that correspond to the random variables.

Independent Binomials with equal p

For any two Binomial random variables with the same "success" probability: $X \sim Bin(n_1, p)$ and $Y \sim Bin(n_2, p)$ the sum of those two random variables is another binomial: $X + Y \sim Bin(n_1 + n_2, p)$. This does not hold when the two distribution have different parameters p.

Independent Poissons

For any two Poisson random variables: $X \sim Poi(\lambda_1)$ and $Y \sim Poi(\lambda_2)$ the sum of those two random variables is another Poisson: $X + Y \sim Poi(\lambda_1 + \lambda_2)$. This holds when λ_1 is not the same as λ_2.

Independent Normals

For any two normal random variables $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ the sum of those two random variables is another normal: $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.

General Independent Case

For two general independent random variables (aka cases of independent random variables that don't fit the above special situations) you can calculate the CDF or the PDF of the sum of two random variables using the following formulas:

$$F_{X+Y}(a) = P(X + Y \leq a) = \int_{y=-\infty}^{\infty} F_X(a-y) f_Y(y) dy$$

$$f_{X+Y}(a) = \int_{y=-\infty}^{\infty} f_X(a-y) f_Y(y) dy$$

There are direct analogies in the discrete case where you replace the integrals with sums and change notation for CDF and PDF.

Example 1

Calculate the PDF of $X + Y$ for independent uniform random variables $X \sim Uni(0,1)$ and $Y \sim Uni(0,1)$? First plug in the equation for general convolution of independent random variables:

$$f_{X+Y}(a) = \int_{y=0}^{1} f_X(a-y) f_Y(y) dy$$

$$f_{X+Y}(a) = \int_{y=0}^{1} f_X(a-y) dy \qquad \text{Because } f_Y(y) = 1$$

It turns out that is not the easiest thing to integrate. By trying a few different values of a in the range $[0,2]$ we can observe that the PDF we are trying to calculate is discontinuous at the point $a = 1$ and thus will be easier to think about as two cases: $a < 1$ and $a > 1$. If we calculate f_{X+Y} for both cases and correctly constrain the bounds of the integral we get simple closed forms for each case:

$$f_{X+Y}(a) = \begin{cases} a & \text{if } 0 < a \leq 1 \\ 2 - a & \text{if } 1 < a \leq 2 \\ 0 & \text{else} \end{cases}$$

Chris Piech
CS109

Lecture #16
Oct 30th, 2017

Beta Distribution

In this chapter we are going to have a very meta discussion about how we represent probabilities. Until now probabilities have just been numbers in the range 0 to 1. However, if we have uncertainty about our probability, it would make sense to represent our probabilities as random variables (and thus articulate the relative likelihood of our belief).

1 Estimating Probabilities

Imagine we have a coin and we would like to know its probability of coming up heads (p). We flip the coin $(n+m)$ times and it comes up head n times. One way to calculate the probability is to assume that it is exactly $p = \frac{n}{n+m}$. That number, however, is a coarse estimate, especially if $n+m$ is small. Intuitively it doesn't capture our uncertainty about the value of p. Just like with other random variables, it often makes sense to hold a distributed belief about the value of p.

To formalize the idea that we want a distribution for p we are going to use a random variable X to represent the probability of the coin coming up heads. Before flipping the coin, we could say that our belief about the coin's success probability is uniform: $X \sim Uni(0,1)$.

If we let N be the number of heads that came up, given that the coin flips are independent, $(N|X) \sim Bin(n+m,x)$. We want to calculate the probability density function for $X|N$. We can start by applying Bayes Theorem:

$$
\begin{aligned}
f(X=x|N=n) &= \frac{P(N=n|X=x)f(X=x)}{P(N=n)} && \text{Bayes Theorem} \\
&= \frac{\binom{n+m}{n}x^n(1-x)^m}{P(N=n)} && \text{Binomial PMF, Uniform PDF} \\
&= \frac{\binom{n+m}{n}}{P(N=n)}x^n(1-x)^m && \text{Moving terms around} \\
&= \frac{1}{c}\cdot x^n(1-x)^m && \text{where } c = \int_0^1 x^n(1-x)^m dx
\end{aligned}
$$

2 Beta Distribution

The equation that we arrived at when using a Bayesian approach to estimating our probability defines a probability density function and thus a random variable. The random variable is called a Beta distribution, and it is defined as follows:

The Probability Density Function (PDF) for a Beta $X \sim Beta(a,b)$ is:

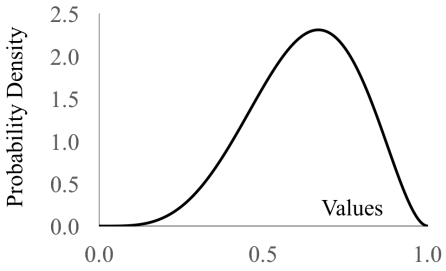
$$
f(X=x) = \begin{cases} \frac{1}{B(a,b)}x^{a-1}(1-x)^{b-1} & \text{if } 0<x<1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{where } B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx
$$

A Beta distribution has $E[X] = \frac{a}{a+b}$ and $Var(X) = \frac{ab}{(a+b)^2(a+b+1)}$. All modern programming languages have a package for calculating Beta CDFs. You will not be expected to compute the CDF by hand in CS109.

To model our estimate of the probability of a coin coming up heads as a beta set $a = n+1$ and $b = m+1$. Beta is used as a random variable to represent a belief distribution of probabilities in contexts beyond estimating coin flips. It has many desirable properties: it has a support range that is exactly $(0,1)$, matching the values

that probabilities can take on and it has the expressive capacity to capture many different forms of belief distributions.

Let's imagine that we had observed $n = 4$ heads and $m = 2$ tails. The probability density function for $X \sim \text{Beta}(5,3)$ is:



Notice how the most likely belief for the probability of our coin is when the random variable, which represents the probability of getting a heads, is $4/6$, the fraction of heads observed. This distribution shows that we hold a non-zero belief that the probability could be something other than $4/6$. It is unlikely that the probability is 0.01 or 0.09, but reasonably likely that it could be 0.5.

It works out that $Beta(1,1) = Uni(0,1)$. As a result the distribution of our belief about p before ("prior") and after ("posterior") can both be represented using a Beta distribution. When that happens we call Beta a "conjugate" distribution. Practically conjugate means easy update.

Beta as a Prior

You can set $X \sim Beta(a,b)$ as a prior to reflect how biased you think the coin is apriori to flipping it. This is a subjective judgment that represent $a+b-2$ "imaginary" trials with $a-1$ heads and $b-1$ tails. If you then observe $n+m$ real trials with n heads you can update your belief. Your new belief would be, $X|(n \text{ heads in } n+m \text{ trials}) \sim Beta(a+n, b+m)$. Using the prior $Beta(1,1) = Uni(0,1)$ is the same as saying we haven't seen any "imaginary" trials, so apriori we know nothing about the coin. This form of thinking about probabilities is representative of the "Bayesian" field of thought where computer scientists explicitly represent probabilities as distributions (with prior beliefs). That school of thought is separate from the "Frequentest" school which tries to calculate probabilities as single numbers evaluated by the ratio of successes to experiments.

Assignment Example

In class we talked about reasons why grade distributions might be well suited to be described as a Beta distribution. Let's say that we are given a set of student grades for a single exam and we find that it is best fit by a Beta distribution: $X \sim Beta(a = 8.28, b = 3.16)$. What is the probability that a student is below the mean (i.e. expectation)?

The answer to this question requires two steps. First calculate the mean of the distribution, then calculate the probability that the random variable takes on a value less than the expectation.

$$\begin{aligned} E[X] &= \frac{a}{a+b} = \frac{8.28}{8.28+3.16} \\ &\approx 0.7238 \end{aligned}$$

Now we need to calculate $P(X < E[X])$. That is exactly the CDF of X evaluated at $E[X]$. We don't have a formula for the CDF of a Beta distribution but all modern programming languages will have a Beta CDF function. In Python using the scipy stats library we can execute stats.beta.cdf which takes the x parameter first followed by the alpha and beta parameters of your Beta distribution.

$$\begin{aligned} P(X < E[X]) &= F_X(0.7238) = \text{stats.beta.cdf(0.7238, 8.28, 3.16)} \\ &\approx 0.46 \end{aligned}$$

Chris Piech
CS109

Lecture #17
Nov 1st, 2017

Great Expectations

Earlier in the course we came to the important result that $E[\sum_i X_i] = \sum_i E[X_i]$. First, as a warm up lets go back to our old friends and show how we could have derived expressions for their expectation.

Expectation of Binomial

First let's start with some practice with the sum of expectations of indicator variables. Let $Y \sim Bin(n, p)$, in other words if Y is a Binomial random variable. We can express Y as the sum of n Bernoulli random indicator variables $X_i \sim Ber(p)$. Since X_i is a Bernoulli, $E[X_i] = p$

$$Y = X_1 + X_2 + \cdots + X_n = \sum_{i=1}^{n} X_i$$

Let's formally calculate the expectation of Y:

$$\begin{aligned} E[Y] &= E[\sum_{i}^{n} X_i] \\ &= \sum_{i}^{n} E[X_i] \\ &= E[X_0] + E[X_1] + \ldots E[X_n] \\ &= np \end{aligned}$$

Expectation of Negative Binomial

Recall that a Negative Binomial is a random variable that semantically represents the number of trials until r successes. Let $Y \sim NegBin(r, p)$.

Let X_i = # trials to get success after $(i–1)$st success. We can then think of each X_i as a Geometric RV: $X_i \sim Geo(p)$. Thus, $E[X_i] = \frac{1}{p}$. We can express Y as:

$$Y = X_1 + X_2 + \cdots + X_r = \sum_{i=1}^{r} X_i$$

Let's formally calculate the expectation of Y:

$$\begin{aligned} E[Y] &= E[\sum_{i=1}^{r} X_i] \\ &= \sum_{i=1}^{r} E[X_i] \\ &= E[X_1] + E[X_2] + \ldots E[X_r] \\ &= \frac{r}{p} \end{aligned}$$

Conditional Expectation

We have gotten to know a kind and gentle soul, conditional probability. And we now know another funky fool, expectation. Let's get those two crazy kids to play together.

Let X and Y be jointly random variables. Recall that the conditional probability mass function (if they are discrete), and the probability density function (if they are continuous) are respectively:

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$
$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

We define the conditional expectation of X given $Y = y$ to be:

$$E[X|Y=y] = \sum_x x p_{X|Y}(x|y)$$
$$E[X|Y=y] = \int_{-\infty}^{\infty} x f_{X|Y}(x|y)dx$$

Where the first equation applies if X and Y are discrete and the second applies if they are continuous.

Properties of Conditional Expectation

Here are some helpful, intuitive properties of conditional expectation:

$$E[g(X)|Y=y] = \sum_x g(x) p_{X|Y}(x|y) \qquad \text{if X and Y are discrete}$$
$$E[g(X)|Y=y] = \int_{-\infty}^{\infty} g(x) f_{X|Y}(x|y)dx \qquad \text{if X and Y are continuous}$$
$$E[\sum_{i=1}^{n} X_i|Y=y] = \sum_{i=1}^{n} E[X_i|Y=y]$$

Law of Total Expectation

The law of total expectation states that: E[E[X—Y]] = E[X]

What?! How is that a thing? Check out this proof:

$$\begin{aligned}
E[E[X|Y]] &= \sum_y E[X|Y=y]P(Y=y) \\
&= \sum_y \sum_x xP(X=x|Y=y)P(Y=y) \\
&= \sum_y \sum_x xP(X=x,Y=y) \\
&= \sum_x \sum_y xP(X=x,Y=y) \\
&= \sum_x x \sum_y P(X=x,Y=y) \\
&= \sum_x xP(X=x) \\
&= E[X]
\end{aligned}$$

Example 1

You roll two 6-sided dice D_1 and D_2. Let $X = D_1 + D_2$ and let Y = the value of D_2. What is $E[X|Y=6]$

$$\begin{aligned}
E[X|Y=6] &= \sum_x xP(X=x|Y=6) \\
&= \left(\frac{1}{6}\right)(7+8+9+10+11+12) = \frac{57}{6} = 9.5
\end{aligned}$$

Which makes intuitive sense since 6 + $E[$value of $D_1] = 6 + 3.5$

Example 2

Consider the following code with random numbers:

```
int Recurse() {
   int x = randomInt(1, 3);  // Equally likely values
   if (x == 1) return 3;
   else if (x == 2) return (5 + Recurse());
   else return (7 + Recurse());
}
```

Let Y = value returned by "Recurse". What is $E[Y]$. In other words, what is the expected return value. Note that this is the exact same approach as calculating the expected run time.

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

First lets calculate each of the conditional expectations:

$$E[Y|X=1] = 3$$
$$E[Y|X=2] = E[5+Y] = 5+E[Y]$$
$$E[Y|X=3] = E[7+Y] = 7+E[Y]$$

Now we can plug those values into the equation. Note that the probability of X taking on 1, 2, or 3 is 1/3:

$$\begin{aligned} E[Y] &= E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3) \\ &= 3(1/3) + (5+E[Y])(1/3) + (7+E[Y])(1/3) \\ &= 15 \end{aligned}$$

Hiring Software Engineers

You are interviewing n software engineer candidates and will hire only 1 candidate. All orderings of candidates are equally likely. Right after each interview you must decide to hire or not hire. You can not go back on a decision. At any point in time you can know the relative ranking of the candidates you have already interviewed.

The strategy that we propose is that we interview the first k candidates and reject them all. Then you hire the next candidate that is better than all of the first k candidates. What is the probability that the best of all the n candidates is hired for a particular choice of k? Let's denote that result $P_k(Best)$. Let X be the position in the ordering of the best candidate:

$$\begin{aligned} P_k(Best) &= \sum_{i=1}^{n} P_k(Best|X=i)P(X=i) \\ &= \frac{1}{n}\sum_{i=1}^{n} P_k(Best|X=i) && \text{since each position is equally likely} \end{aligned}$$

What is $P_k(Best|X=i)$? if $i \leq k$ then the probability is 0 because the best candidate will be rejected without consideration. Sad times. Otherwise we will chose the best candidate, who is in position i, only if the best of the first $i-1$ candidates is among the first k interviewed. If the best among the first $i-1$ is not among the first k, that candidate will be chosen over the true best. Since all orderings are equally likely the probability that the best among the $i-1$ candidates is in the first k is:

$$\frac{k}{i-1} \qquad \text{if } i > k$$

Now we can plug this back into our original equation:

$$
\begin{aligned}
P_k(Best) &= \frac{1}{n}\sum_{i=1}^{n} P_k(Best|X=i) \\
&= \frac{1}{n}\sum_{i=k+1}^{n} \frac{k}{i-1} && \text{since we know } P_k(Best|X=i) \\
&\approx \frac{1}{n}\int_{i=k+1}^{n} \frac{k}{i-1}\,di && \text{By Riemann Sum approximation} \\
&= \frac{k}{n}\ln(i=1)\Big|_{k+1}^{n} = \frac{k}{n}\ln\frac{n-1}{k} \approx \frac{k}{n}\ln\frac{n}{k}
\end{aligned}
$$

If we think of $P_k(Best) = \frac{k}{n}\ln\frac{n}{k}$ as a function of k we can take find the value of k that optimizes it by taking its derivative and setting it equal to 0. The optimal value of k is n/e. Where e is Euler's number.

Chris Piech
CS109

Lecture #18
Nov 3rd, 2017

Covariance and Correlation

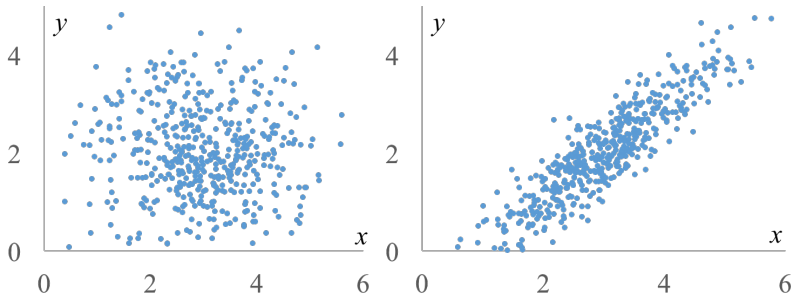
Product of Expectations Lemma

Here is a lovely little lemma to get us started:

$$E[g(X)h(Y)] = E[g(X)]E[h(Y)] \qquad \text{if and only if } X \text{ and } Y \text{ are independent}$$

1 Covariance and Correlation

Consider the two multivariate distributions shown bellow. In both images I have plotted one thousand samples drawn from the underlying joint distribution. Clearly the two distributions are different. However, the mean and variance are the same in both the x and the y dimension. What is different?



Covariance is a quantitative measure of the extent to which the deviation of one variable from its mean matches the deviation of the other from its mean. It is a mathematical relationship that is defined as:

$$\text{Cov}(X,Y) = E[(X - E[X])(Y - E[Y])]$$

That is a little hard to wrap your mind around (but worth pushing on a bit). The outer expectation will be a weighted sum of the inner function evaluated at a particular (x,y) weighted by the probability of (x,y). If x and y are both above their respective means, or if x and y are both below their respective means, that term will be positive. If one is above its mean and the other is below, the term is negative. If the weighted sum of terms is positive, the two random variables will have a positive correlation. We can rewrite the above equation to get an equivalent equation:

$$\text{Cov}(X,Y) = E[XY] - E[Y]E[X]$$

Using this equation (and the product lemma) is it easy to see that if two random variables are independent their covariance is 0. The reverse is *not* true in general.

Properties of Covariance

Say that X and Y are arbitrary random variables:

$$\text{Cov}(X,Y) = \text{Cov}(Y,X)$$
$$\text{Cov}(X,X) = E[X^2] - E[X]E[X] = \text{Var}(X)$$
$$\text{Cov}(aX+b,Y) = a\text{Cov}(X,Y)$$

Let $X = X_1 + X_2 + \cdots + X_n$ and let $Y = Y_1 + Y_2 + \cdots + Y_m$. The covariance of X and Y is:

$$\text{Cov}(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m}\text{Cov}(X_i,Y_j)$$

$$\text{Cov}(X,X) = \text{Var}(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\text{Cov}(X_i,X_j)$$

That last property gives us a third way to calculate variance. You could use this definition to calculate the variance of the binomial.

Correlation

Covariance is interesting because it is a quantitative measurement of the relationship between two variables. Correlation between two random variables, $\rho(X,Y)$ is the covariance of the two variables normalized by the variance of each variable. This normalization cancels the units out and normalizes the measure so that it is always in the range [0, 1]:

$$\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)Var(Y)}}$$

Correlation measure linearity between X and Y.

$\rho(X,Y) = 1$	$Y = aX + b$ where $a = \sigma_y/\sigma_x$
$\rho(X,Y) = -1$	$Y = aX + b$ where $a = -\sigma_y/\sigma_x$
$\rho(X,Y) = 0$	absence of linear relationship

If $\rho(X,Y) = 0$ we say that X and Y are "uncorrelated." If two varaibles are independent, then their correlation will be 0. However, it doesn't go the other way. A correlation of 0 does not imply independence.

When people use the term correlation, they are actually referring to a specific type of correlation called "Pearson" correlation. It measures the degree to which there is a linear relationship between the two variables. An alternative measure is "Spearman" correlation which has a formula almost identical to your regular correlation score, with the exception that the underlying random variables are first transformed into their rank. "Spearman" correlation is outside the scope of CS109.

Chris Piech
CS109

Lecture #19
Nov 6th, 2017

Samples and The Bootstrap

Let's say you are the king of Bhutan and you want to know the average happiness of the people in your country. You can't ask every single person, but you could ask a random subsample. In this next section we will consider principled claims that you can make based on a subsample. Assume we randomly sample 200 Bhutanese and ask them about their happiness. Our data looks like this: $72, 85, \ldots, 71$. You can also think of it as a collection of $n = 200$ I.I.D. (independent, identically distributed) random variables $X_1, X_2, \ldots, X_n$.

Estimating Mean and Variance from Samples

We assume that the data we look at are IID from the same underlying distribution (F) with a true mean (μ) and a true variance (σ^2). Since we can't talk to everyone in Bhutan we have to rely on our sample to estimate the mean and variance. From our sample we can calculate a sample mean ($\bar{X}$) and a sample variance (S^2). These are the best guesses that we can make about the true mean and true variance.

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n} \qquad\qquad S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$$

The first question to ask is, are those unbiased estimates? Yes. Unbiased, means that if we were to repeat this sampling process many times, the expected value of our estimates should be equal to the true values we are trying to estimate. We will prove that that is the case for $\bar{X}$. The proof for S^2 is in lecture slides.

$$\begin{aligned} E[\bar{X}] &= E[\sum_{i=1}^{n} \frac{X_i}{n}] = \frac{1}{n} E\left[\sum_{i=1}^{n} X_i\right] \\ &= \frac{1}{n} \sum_{i=1}^{n} E[X_i] = \frac{1}{n} \sum_{i=1}^{n} \mu = \frac{1}{n} n\mu = \mu \end{aligned}$$

The equation for sample mean seems like a reasonable way to calculate the expectation of the underlying distribution. The same could be said about sample variance except for the surprising $(n-1)$ in the denominator of the equation. Why $(n-1)$? That denominator is necessary to make sure that the $E[S^2] = \sigma^2$.

The intuition behind the proof is that sample variance calculates the distance of each sample to the sample mean, *not* the true mean. The sample mean itself varies, and we can show that its variance is also related to the true variance.

Standard Error

Ok, you convinced me that our estimates for mean and variance are not biased. But now I want to know how much my sample mean might vary relative to the true mean.

$$\begin{aligned} \text{Var}(\bar{X}) &= \text{Var}(\sum_{i=1}^{n} \frac{X_i}{n}) = \left(\frac{1}{n}\right)^2 \text{Var}\left(\sum_{i=1}^{n} X_i\right) \\ &= \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \text{Var}(X_i) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \sigma^2 = \left(\frac{1}{n}\right)^2 n\sigma^2 = \frac{\sigma^2}{n} \\ &\approx \frac{S^2}{n} && \text{Since S is an unbiased estimate} \\ \text{Std}(\bar{X}) &\approx \sqrt{\frac{S^2}{n}} && \text{Since Std is the square root of Var} \end{aligned}$$

That $\text{Std}(\bar{X})$ term has a special name. It is called the standard error and its how you report uncertainty of estimates of means in scientific papers (and how you get error bars). Great! Now we can compute all these wonderful statistics for the Bhutanese people. But wait! You never told me how to calculate the $\text{Std}(S^2)$. True, that is outside the scope of CS109. You can find it on wikipedia if you want.

Let's say we calculate the our sample of happiness has n = 200 people. The sample mean is $\bar{X} = 83$ (what is the unit here? happiness score?) and the sample variance is $S^2 = 450$. We can now calculate the standard error of our estimate of the mean to be 1.5. When we report our results we will say that the average happiness score in Bhutan is 83 ± 1.5 with variance 450.

Bootstrap

Bootstrap is a newly invented statistical technique for both understanding distributions of statistics and for calculating p-values (a p-value is a the probability that a scientific claim is incorrect). It was invented here at Stanford in 1979 when mathematicians were just starting to understand how computers, and computer simulations, could be used to better understand probabilities.

The first key insight is that: if we had access to the underlying distribution (F) then answering almost any question we might have as to how accurate our statistics are becomes straightforward. For example, in the previous section we gave a formula for how you could calculate the sample variance from a sample of size n. We know that in expectation our sample variance is equal to the true variance. But what if we want to know the probability that the true variance is within a certain range of the number we calculated? That question might sound dry, but it is critical to evaluating scientific claims! If you knew the underlying distribution, F, you could simply repeat the experiment of drawing a sample of size n from F, calculate the sample variance from our new sample and test what portion fell within a certain range.

The next insight behind bootstrapping is that the best estimate that we can get for F is from our sample itself! The simplest way to estimate F (and the one we will use in this class) is to assume that the $P(X = k)$ is simply the fraction of times that k showed up in the sample. Note that this defines the probability mass function of our estimate $\hat{F}$ of F.

```
def bootstrap(sample):
    N = number of elements in sample
    pmf = estimate the underlying pmf from the sample
    stats = []
    repeat 10,000 times:
        resample = draw N new samples from the pmf
        stat = calculate your stat on the resample
        stats.append(stat)
    stats can now be used to estimate the distribution of the stat
```

To calculate $\text{Var}(S^2)$ we could calculate S_i^2 for each resample i and after 10,000 iterations, we could calculate the sample variance of all the S_i^2s.

The bootstrap has strong theoretic grantees, and is accepted by the scientific community, when calculating any statistic. It breaks down when the underlying distribution has a "long tail" or if the samples are not I.I.D.

Chris Piech
CS109

Lecture #20
May 15th, 2017

Central Limit Theorem

The Theory

The central limit theorem proves that the averages of samples from *any* distribution themselves must be normally distributed. Consider IID random variables $X_1, X_2 \ldots$ such that $E[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$. Let

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

The central limit theorem states:

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \qquad \text{as } n \to \infty$$

It is sometimes expressed in terms of the standard normal, Z:

$$Z = \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \qquad \text{as } n \to \infty$$

At this point you probably think that the central limit theorem is awesome. But it gets even better. With some algebraic manipulation we can show that if the sample mean of IID random variables is normal, it follows that the sum of equally weighted IID random variables must also be normal. Let's call the sum of IID random variables $\bar{Y}$:

$$\begin{aligned}
\bar{Y} &= \sum_{i=1}^{n} X_i = n \cdot \bar{X} && \text{If we define } \bar{Y} \text{ to be the sum of our variables} \\
&\sim N(n\mu, n^2\frac{\sigma^2}{n}) && \text{Since } \bar{X} \text{ is a normal and } n \text{ is a constant.} \\
&\sim N(n\mu, n\sigma^2) && \text{By simplifying.}
\end{aligned}$$

In summary, the central limit theorem explains that both the sample mean of IID variables is normal (regardless of what distribution the IID variables came from) and that the sum of equally weighted IID random variables is normal (again, regardless of the underlying distribution).

Example 1

Say you have a new algorithm and you want to test its running time. You have an idea of the variance of the algorithm's run time: $\sigma^2 = 4\text{sec}^2$ but you want to estimate the mean: $\mu = t\text{sec}$. You can run the algorithm repeatedly (IID trials). How many trials do you have to run so that your estimated runtime = $t \pm 0.5$ with 95% certainty? LetX_i be the run time of the i-th run (for $1 \leq i \leq n$).

$$0.95 = P(-0.5 \leq \frac{\sum_{i=1}^{n} X_i}{n} - t \leq 0.5)$$

By the central limit theorem, the standard normal Z must be equal to:

$$\begin{aligned}
Z &= \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \\
&= \frac{(\sum_{i=1}^{n} X_i) - nt}{2\sqrt{n}}
\end{aligned}$$

Now we rewrite our probability inequality so that the central term is Z:

$$\begin{aligned}
0.95 &= P(-0.5 \leq \frac{\sum_{i=1}^{n} X_i}{n} - t \leq 0.5) = P(\frac{-0.5\sqrt{n}}{2} \leq \frac{\sum_{i=1}^{n} X_i}{n} - t \leq \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \leq \frac{\sqrt{n}}{2}\frac{\sum_{i=1}^{n} X_i}{n} - \frac{\sqrt{n}}{2}t \leq \frac{0.5\sqrt{n}}{2}) = P(\frac{-0.5\sqrt{n}}{2} \leq \frac{\sum_{i=1}^{n} X_i}{2\sqrt{n}} - \frac{\sqrt{n}}{\sqrt{n}}\frac{\sqrt{n}t}{2} \leq \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \leq \frac{\sum_{i=1}^{n} X_i - nt}{2\sqrt{n}} \leq \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \leq Z \leq \frac{0.5\sqrt{n}}{2})
\end{aligned}$$

And now we can find the value of n that makes this equation hold.

$$\begin{aligned}
0.95 &= \phi(\frac{\sqrt{n}}{4}) - \phi(-\frac{\sqrt{n}}{4}) = \phi(\frac{\sqrt{n}}{4}) - (1 - \phi(\frac{\sqrt{n}}{4})) \\
&= 2\phi(\frac{\sqrt{n}}{4}) - 1 \\
0.975 &= \phi(\frac{\sqrt{n}}{4}) \\
\phi^{-1}(0.975) &= \frac{\sqrt{n}}{4} \\
1.96 &= \frac{\sqrt{n}}{4} \\
n &= 61.4
\end{aligned}$$

Thus it takes 62 runs. If you are interested in how this extends to cases where the variance is unknown, look into variations of the students' t-test.

Example 2

You will roll a 6 sided dice 10 times. Let X be the total value of all 10 dice = $X_1 + X_2 + \cdots + X_10$. You win the game if $X \leq 25$ or $X \geq 45$. Use the central limit theorem to calculate the probability that you win.

Recall that $E[X_i] = 3.5$ and $\text{Var}(X_i) = \frac{35}{12}$.

$$\begin{aligned}
P(X \leq 25 \text{ or } X \geq 45) &= 1 - P(25.5 \leq X \leq 44.5) \\
&= 1 - P(\frac{25.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \leq \frac{X - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \leq \frac{44.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \\
&\approx 1 - (2\phi(1.76) - 1) \approx 2(1 - 0.9608) = 0.0784
\end{aligned}$$

Chris Piech
CS109

Lecture #21
Nov 10th, 2017

Maximum Likelihood

Parameters

Before we dive into parameter estimation, first let's revisit the concept of parameters. Given a model, the parameters are the numbers that yield the actual distribution. In the case of a Bernoulli random variable, the single parameter was the value p. In the case of a Uniform random variable, the parameters are the a and b values that define the min and max value. Here is a list of random variables and the corresponding parameters. From now on, we are going to use the notation θ to be a vector of all the parameters: In the real

Distribution	Parameters
Bernoulli(p)	$\theta = p$
Poisson(λ)	$\theta = \lambda$
Uniform(a,b)	$\theta = (a, b)$
Normal(μ, σ^2)	$\theta = (\mu, \sigma^2)$
Y = mX + b	$\theta = (m, b)$

world often you don't know the "true" parameters, but you get to observe data. Next up, we will explore how we can use data to estimate the model parameters.

It turns out there isn't just one way to estimate the value of parameters. There are two main schools of thought: Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP). Both of these schools of thought assume that your data are independent and identically distributed (IID) samples: $X_1, X_2, \ldots X_n$ where X_i.

Maximum Likelihood

Our first algorithm for estimating parameters is called Maximum Likelihood Estimation (MLE). The central idea behind MLE is to select that parameters (θ) that make the observed data the most likely.

The data that we are going to use to estimate the parameters are going to be n independent and identically distributed (IID) samples: $X_1, X_2, \ldots X_n$.

Likelihood

First we define the likelihood of our data give parameters θ:

$$L(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

This is the probability of all of our data. It evaluates to a product because all X_i are independent. Now we chose the value of θ that maximizes the likelihood function. Formally $\hat{\theta} = \underset{\theta}{\text{argmax}}\, L(\theta)$.

A cool property of argmax is that since log is a monotone function, the argmax of a function is the same as the argmax of the log of the function! That's nice because logs make the math simpler. Instead of using likelihood, you should instead use log likelihood: $LL(\theta)$.

$$LL(\theta) = \log \prod_{i=1}^{n} f(X_i|\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

To use a maximum likelihood estimator, first write the log likelihood of the data given your parameters. Then chose the value of parameters that maximize the log likelihood function. Argmax can be computed in many ways. Most require computing the first derivative of the function.

Bernoulli MLE Estimation

Consider IID random variables $X_1, X_2, \ldots X_n$ where $X_i \sim \text{Ber}(p)$. First we are going to write the PMF of a Bernoulli in a crazy way: The probability mass function $f(X_i|p) = p^{X_i}(1-p)^{1-X_i}$. Wow! Whats up with that? First convince yourself that when $X_i = 0$ and $X_i = 1$ this returns the right probabilities. We write the PMF this way because its derivable.

Now let's do some MLE estimation:

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i} \\
LL(\theta) &= \sum_{i=1}^{n} \log p^{X_i}(1-p)^{1-X_i} \\
&= \sum_{i=1}^{n} X_i(\log p) + (1-X_i) log(1-p) \\
&= Y \log p + (n-Y) log(1-p) \qquad \text{where } Y = \sum_{i=1}^{n} X_i
\end{aligned}$$

Great Scott! Now we simply need to chose the value of p that maximizes our log-likelihood. One way to do that is to find the first derivative and set it equal to 0.

$$\begin{aligned}
\frac{\delta LL(p)}{\delta p} &= Y\frac{1}{p} + (n-Y)\frac{-1}{1-p} = 0 \\
\hat{p} &= \frac{Y}{n} = \frac{\sum_{i=1}^{n} X_i}{n}
\end{aligned}$$

All that work and we get the same thing as method of moments and sample mean...

Normal MLE Estimation

Consider IID random variables $X_1, X_2, \ldots X_n$ where $X_i \sim N(\mu, \sigma^2)$.

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(X_i|\mu, \sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}} \\
LL(\theta) &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}} \\
&= \sum_{i=1}^{n} \left[-\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}(X_i-\mu)^2 \right]
\end{aligned}$$

If we chose the values of $\hat{\mu}$ and $\hat{\sigma^2}$ that maximize likelihood, we get:$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2$.

Chris Piech
CS109

Lecture #22
Nov 13th, 2017

Gradient Ascent

Maximum Likelihood Refresher

Our first algorithm for estimating parameters is called Maximum Likelihood Estimation (MLE). The central idea behind MLE is to select that parameters (θ) that make the observed data the most likely.

The data that we are going to use to estimate the parameters are going to be n independent and identically distributed (IID) samples: $X_1, X_2, \ldots X_n$.

Likelihood

First we define the likelihood of our data give parameters θ:

$$L(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

This is the probability of all of our data. It evaluates to a product because all X_i are independent. Now we chose the value of θ that maximizes the likelihood function. Formally $\hat{\theta} = \underset{\theta}{\text{argmax}}\, L(\theta)$.

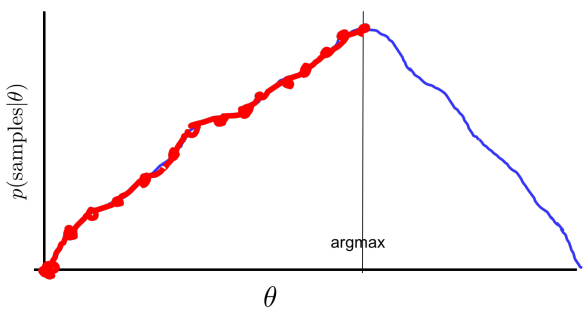
A cool property of argmax is that since log is a monotone function, the argmax of a function is the same as the argmax of the log of the function! That's nice because logs make the math simpler. Instead of using likelihood, you should instead use log likelihood: $LL(\theta)$.

$$LL(\theta) = \log \prod_{i=1}^{n} f(X_i|\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

To use a maximum likelihood estimator, first write the log likelihood of the data given your parameters. Then chose the value of parameters that maximize the log likelihood function. Argmax can be computed in many ways. Most require computing the first derivative of the function.

Gradient Ascent Optimization

In many cases we can't solve for argmax mathematically. Instead we use a computer. To do so we employ an algorithm called gradient ascent (a classic in optimization theory). The idea behind gradient ascent is that if you continuously take small steps in the direction of your gradient, you will eventually make it to a local maxima.



Start with theta as any initial value (often 0). Then take many small steps towards a local maxima. The new theta after each small step can be calculated as:

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} + \eta \cdot \frac{\partial LL(\theta^{\text{old}})}{\partial \theta_j^{\text{old}}}$$

Where "eta" (η) is the magnitude of the step size that we take. If you keep updating θ using the equation above you will (often) converge on good values of θ. As a general rule of thumb, use a small value of η to start. If ever you find that the function value (for the function you are trying to argmax) is decreasing, your choice of η was too large. Here is the gradient ascent algorithm in pseudo-code:

```
Initialize: θ_j = 0 for all 0 ≤ j ≤ m

Repeat many times:
    gradient[j] = 0 for all 0 ≤ j ≤ m

    Calculate all gradient[j]'s based on data
    and current setting of theta

    θ_j += η * gradient[j] for all 0 ≤ j ≤ m
```

Linear Regression Lite

MLE is an algorithm that can be used for any probability model with a derivable likelihood function. As an example lets estimate the parameter θ in a model where there is a random variable Y such that $Y = \theta X + Z$, $Z \sim N(0, \sigma^2)$ and X is an unknown distribution.

In the case where you are told the value of X, θX is a number and $\theta X + Z$ is the sum of a gaussian and a number. This implies that $Y|X \sim N(\theta X, \sigma^2)$. Our goal is to chose a value of θ that maximizes the probability IID: $(X_1, Y_1), (X_2, Y_2), \ldots (X_n, Y_n)$.

We approach this problem by first finding a function for the log likelihood of the data given θ. Then we find the value of θ that maximizes the log likelihood function. To start, use the PDF of a Normal to express the probability of $Y|X, \theta$:

$$f(Y_i|X_i, \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}}$$

Now we are ready to write the likelihood function, then take its log to get the log likelihood function:

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(Y_i, X_i|\theta) && \text{Let's break up this joint} \\
&= \prod_{i=1}^{n} f(Y_i|X_i, \theta) f(X_i) && f(X_i) \text{ is independent of } \theta \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{Substitute in the definition of } f(Y_i|X_i)
\end{aligned}$$

$$
\begin{aligned}
LL(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{Substitute in } L(\theta) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} + \sum_{i=1}^{n} \log f(X_i) && \text{Log of a product is the sum of logs} \\
&= n \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \theta X_i)^2 + \sum_{i=1}^{n} \log f(X_i)
\end{aligned}
$$

Remove positive constant multipliers and terms that don't include θ. We are left with trying to find a value of θ that maximizes:

$$
\hat{\theta} = \underset{\theta}{\operatorname{argmax}} - \sum_{i=1}^{n} (Y_i - \theta X_i)^2
$$

To solve this argmax we are going to use Gradient Ascent. In order to do so we first need to find the derivative of the function we want to argmax with respect to θ.

$$
\begin{aligned}
\frac{\partial}{\partial \theta} - \sum_{i=1}^{n} (Y_i - \theta X_i)^2 &= - \sum_{i=1}^{n} \frac{\partial}{\partial \theta} (Y_i - \theta X_i)^2 \\
&= - \sum_{i=1}^{n} 2(Y_i - \theta X_i)(-X_i) \\
&= \sum_{i=1}^{n} 2(Y_i - \theta X_i)(X_i)
\end{aligned}
$$

This first derivative can be plugged into gradient ascent to give our final algorithm:

```
Initialize: θ = 0

Repeat many times:
    gradient = 0
    For each training example (x, y):
        gradient += 2(y - θ x)(x)
    θ += η * gradient
```

Chris Piech
CS109

Lecture #23
Nov 15th, 2017

Maximum A Posteriori

Today we are going to cover our third Parameter Estimator, Maximum A Posteriori (MAP). The other two were Unbiased estimation and Maximum Likelihood (MLE). The paradigm of MAP is that we should chose the value for our parameters that is the most likely given the data. At first blush this might seem the same as MLE, however notice that MLE choses the value of parameters that makes the *data* most likely. Formally, for IID random variables $X_1, \ldots, X_n$:

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n)$$

In the equation above we trying to calculate the conditional probability of unobserved random variables given observed random variables. When that is the case, think Bayes Theorem! Expand the function f using the continuous version of Bayes Theorem.

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n) && \text{Now apply Bayes Theorem} \\
&= \underset{\theta}{\text{argmax}}\ \frac{f(X_1, X_2, \ldots, X_n | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Ahh much better}
\end{aligned}$$

Note that f, g and h are all probability densities. I used different symbols to make it explicit that they may have different functions. Now we are going to leverage two observations. First, the data is assumed to be IID so we can decompose the density of the data given θ. Second, the denominator is a constant with respect to θ. As such its value does not affect the argmax and we can drop that term. Mathematically:

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ \frac{\prod_{i=1}^{n} f(X_i | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Since the samples are IID} \\
&= \underset{\theta}{\text{argmax}}\ \prod_{i=1}^{n} f(X_i | \theta) g(\theta) && \text{Since } h \text{ is constant with respect to } \theta
\end{aligned}$$

As before, it will be more convenient to find the argmax of the log of the MAP function, which gives us the final form for MAP estimation of parameters.

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i | \theta)) \right)$$

Using Bayesian terminology, the MAP estimate is the mode of the "posterior" distribution for θ. If you look at this equation side by side with the MLE equation you will notice that MAP is the argmax of the exact same function *plus* a term for the log of the prior.

Parameter Priors

In order to get ready for the world of MAP estimation, we are going to need to brush up on our distributions. We will need reasonable distributions for each of our different parameters. For example, if you are predicting a Poisson distribution, what is the right random variable type for the prior of λ?

A desiderata for prior distributions is that the resulting posterior distribution has the same functional form. We call these "conjugate" priors. In the case where you are updating your belief many times, conjugate priors makes programming in the math equations much easier.

Here is a list of different parameters and the distributions most often used for their priors:

Parameter	Distribution
Bernoulli p	Beta
Binomial p	Beta
Poisson λ	Gamma
Exponential λ	Gamma
Multinomial p_i	Dirichlet
Normal μ	Normal
Normal σ^2	Inverse Gamma

You are only expected to know the new distributions on a high level. You do not need to know Inverse Gamma. I included it for completeness.

The distributions used to represent your "prior" belief about a random variable will often have their own parameters. For example, a Beta distribution is defined using two parameters (a,b). Do we have to use parameter estimation to evaluate a and b too? No. Those parameters are called "hyperparameters". That is a term we reserve for parameters in our model that we fix before running parameter estimate. Before you run MAP you decide on the values of (a,b).

Dirichlet

The Dirichlet distribution generalizes Beta in same way Multinomial generalizes Bernoulli. A random variable X that is Dirichlet is parametrized as $X \sim \text{Dirichlet}(a_1, a_2, \ldots, a_m)$. The PDF of the distribution is:

$$f(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = K \prod_{i=1}^{m} x_i^{a_i - 1}$$

Where K is a normalizing constant.

You can intuitively understand the hyperparameters of a Dirichlet distribution: imagine you have seen $\sum_{i=1}^{m} a_i - m$ imaginary trials. In those trials you had $(a_i - 1)$ outcomes of value i. As an example consider estimating the probability of getting different numbers on a six-sided Skewed Dice (where each side is a different shape). We will estimate the probabilities of rolling each side of this dice by repeatedly rolling the dice n times. This will produce n IID samples. For the MAP paradigm, we are going to need a prior on our belief of each of the parameters $p_1 \ldots p_6$. We want to express that we lightly believe that each roll is equally likely.

Before you roll, let's imagine you had rolled the dice six times and had gotten one of each possible values. Thus the "prior" distribution would be Dirichlet(2,2,2,2,2,2). After observing $n_1 + n_2 + \cdots + n_6$ new trials with n_i results of outcome i, the "posterior" distribution is Dirichlet($2 + n_1, \ldots 2 + n_6$). Using a prior which represents one imagined observation of each outcome is called "Laplace smoothing" and it guarantees that none of your probabilities are 0 or 1.

Gamma

The Gamma(k, θ) distribution is the conjugate prior for the λ parameter of the Poisson distribution (It is also the conjugate for Exponential, but we won't delve into that).

The hyperparameters can be interpreted as: you saw k total imaginary events during θ imaginary time periods. After observing n events during the next t time periods the posterior distribution is Gamma($k + n, \theta + t$).

For example Gamma(10, 5) would represent having seen 10 imaginary events in 5 time periods. It is like imagining a rate of 2 with some degree of confidence. If we start with that Gamma as a prior and then see 11 events in the next 2 time periods our posterior is Gamma(21,7) which is equivalent to an updated rate of 3.

Chris Piech
CS109

Lecture #24
Nov 17th, 2017

Naïve Bayes

Naïve Bayes is a type of machine learning algorithm called a classifier. It uses training data with feature/label pairs ($\mathbf{x}$, y), where y is one of two class labels, in order to estimate a function $\hat{y} = g(\mathbf{x})$. This function can then be used to make predictions.

In the classification task you are given N training pairs: $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$ Where $\mathbf{x}^{(i)}$ is a vector of m discrete features for the ith training example and $y^{(i)}$ is the discrete label for the ith training example. For now we are going to assume that all values in our training data-set are binary. While this is not a necessary assumption (both naive bayes and logistic regression can work for non-binary data), it makes it much easier to learn the core concepts. Specifically we assume that all labels are binary $y^{(i)} \in \{0,1\}\ \forall i$ and all features are binary $x_j^{(i)} \in \{0,1\}\ \forall i, j$.

1 Naïve Bayes algorithm

Here is the Naïve Bayes algorithm. After presenting the algorithm I am going to show the theory behind it.

Training

The objective in training is to estimate the probabilities $P(Y)$ and $P(X_i|Y)$ for all $0 < i \leq m$ features.
Using an MLE estimate:

$$\hat{p}(X_i = x_i | Y = y) = \frac{(\text{\# training examples where } X_i = x_i \text{ and } Y = y)}{(\text{training examples where } Y = y)}$$

Using a Laplace MAP estimate:

$$\hat{p}(X_i = x_i | Y = y) = \frac{(\text{\# training examples where } X_i = x_i \text{ and } Y = y) + 1}{(\text{training examples where } Y = y) + 2}$$

Prediction

For an example with $\mathbf{x} = [x_1, x_2, \ldots, x_m]$, estimate the value of y as:

$$\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \operatorname*{argmax}_y \hat{P}(\mathbf{X}|Y)\hat{P}(Y) && \text{This is equal to argmax } \hat{P}(Y = y|\mathbf{X}) \\
&= \operatorname*{argmax}_y \prod_{i=1}^{m} \hat{p}(X_i = x_i | Y = y)\hat{p}(Y = y) && \text{Naïve Bayes assumption} \\
&= \operatorname*{argmax}_y \sum_{i=1}^{m} \log \hat{p}(X_i = x_i | Y = y) + \log \hat{p}(Y = y) && \text{Log version for numerical stability}
\end{aligned}$$

Note that for small enough datasets you may not need to use the log version of the argmax.

Theory

We can solve the classification task using a brute force solution. To do so we will learn the full joint distribution $\hat{P}(Y, \mathbf{X})$. In the world of classification, when we make a prediction, we want to chose the value of y that maximizes: $g(\mathbf{x}) = \operatorname*{argmax}_y \hat{P}(Y = y|\mathbf{X})$.

$$\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \operatorname*{argmax}_y \hat{P}(Y|\mathbf{X}) = \operatorname*{argmax}_y \frac{\hat{P}(\mathbf{X}, Y)}{\hat{P}(\mathbf{X})} && \text{By definition of conditional probability} \\
&= \operatorname*{argmax}_y \hat{P}(\mathbf{X}, Y) && \text{Since } \hat{P}(\mathbf{X}) \text{ is constant with respect to Y}
\end{aligned}$$

Using our training data we could interpret the joint distribution of $\mathbf{X}$ and Y as one giant multinomial with a different parameter for every combination of $\mathbf{X} = \mathbf{x}$ and $Y = y$. If for example, the input vectors are only length one. In other words $|\mathbf{x}| = 1$ and the number of values that x and y can take on are small, say binary, this is a totally reasonable approach. We could estimate the multinomial using MLE or MAP estimators and then calculate argmax over a few lookups in our table.

The bad times hit when the number of features becomes large. Recall that our multinomial needs to estimate a parameter for every unique combination of assignments to the vector $\mathbf{x}$ and the value y. If there are $|\mathbf{x}| = n$ binary features then this strategy is going to take order $\mathcal{O}(2^n)$ space and there will likely be many parameters that are estimated without any training data that matches the corresponding assignment.

Naïve Bayes Assumption

The Naïve Bayes Assumption is that each feature of $\mathbf{x}$ is independent of one another given y. That assumption is wrong, but useful. This assumption allows us to make predictions using space and data which is linear with respect to the size of the features: $\mathcal{O}(n)$ if $|\mathbf{x}| = n$. That allows us to train and make predictions for huge feature spaces such as one which has an indicator for every word on the internet. Using this assumption the prediction algorithm can be simplified.

$$
\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \underset{y}{\text{argmax}}\ \hat{P}(\mathbf{X}, Y) && \text{As we last left off} \\
&= \underset{y}{\text{argmax}}\ \hat{P}(\mathbf{X}|Y)\hat{P}(Y) && \text{By chain rule} \\
&= \underset{y}{\text{argmax}}\ \prod_{i=1}^{n} \hat{p}(X_i|Y)\hat{P}(Y) && \text{Using the naïve bayes assumption} \\
&= \underset{y}{\text{argmax}}\ \sum_{i=1}^{m} \log \hat{p}(X_i = x_i|Y = y) + \log \hat{p}(Y = y) && \text{Log version for numerical stability}
\end{aligned}
$$

This algorithm is both fast and stable both when training and making predictions. If we think of each X_i, y pair as a multinomial we can find MLE and MAP estimations for the values. See the "algorithm" section for the optimal values of each p in the multinomial.

Naïve Bayes is a simple form of a field of machine learning called Probabilistic Graphical Models. In that field you make a graph of how your variables are related to one another and you come up with conditional independence assumptions that make it computationally tractable to estimate the joint distribution.

Example

Say we have thirty examples of people's preferences (like or not) for Star Wars, Harry Potter and Lord of the Rings. Each training example has x_1, x_2 and y where x_1 is whether or not the user liked Star Wars, x_2 is whether or not the user liked Harry Potter and y is whether or not the user liked Lord of the Rings. For the 30 training examples the MAP and MLE estimates are as follows:

Y \ X_1	0	1	MLE estimates	
0	3	10	0.10	0.33
1	4	13	0.13	0.43

Y \ X_2	0	1	MLE estimates	
0	5	8	0.17	0.27
1	7	10	0.23	0.33

Y	#	MLE est.
0	13	0.43
1	17	0.57

For a new user who likes Star Wars ($x_1 = 1$) but not Harry Potter ($x_2 = 0$) do you predict that they will like Lord of the Rings? See the lecture slides for the answer.

Chris Piech
CS109

Lecture #25
Nov 27th, 2017

Logistic Regression

Before we get started I wanted to familiarize you with some notation:

$$\theta^T \mathbf{x} = \sum_{i=1}^{n} \theta_i x_i = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \qquad \text{weighted sum}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \text{sigmoid function}$$

Logistic Regression Overview

Classification is the task of choosing a value of y that maximizes $P(Y|X)$. Naïve Bayes worked by approximating that probability using the naïve assumption that each feature was independent given the class label.

For all classification algorithms you are given n I.I.D. training datapoints $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(n)}, y^{(n)})$ where each "feature" vector $\mathbf{x}^{(i)}$ has $m = |\mathbf{x}^{(i)}|$ features.

Logistic Regression Assumption

Logistic Regression is a classification algorithm (I know, terrible name) that works by trying to learn a function that approximates $P(Y|X)$. It makes the central assumption that $P(Y|X)$ can be approximated as a sigmoid function applied to a linear combination of input features. Mathematically, for a single training datapoint $(\mathbf{x}, y)$ Logistic Regression assumes:

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(z) \text{ where } z = \theta_0 + \sum_{i=1}^{m} \theta_i x_i$$

This assumption is often written in the equivalent forms:

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(\theta^T \mathbf{x}) \qquad \text{where we always set } x_0 \text{ to be 1}$$

$$P(Y = 0|\mathbf{X} = \mathbf{x}) = 1 - \sigma(\theta^T \mathbf{x}) \qquad \text{by total law of probability}$$

Using these equations for probability of $Y|X$ we can create an algorithm that select values of theta that maximize that probability for all data. I am first going to state the log probability function and partial derivatives with respect to theta. Then later we will (a) show an algorithm that can chose optimal values of theta and (b) show how the equations were derived.

Log Likelihood

We can write an equation for the likelihood of all the data (under the Logistic Regression assumption). If you take the log of the likelihood equation the result is:

$$LL(\theta) = \sum_{i=0}^{n} y^{(i)} \log \sigma(\theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \log[1 - \sigma(\theta^T \mathbf{x}^{(i)})]$$

We will show the derivation later.

Gradient of Log Likelihood

Now that we have a function for log-likelihood, we simply need to chose the values of theta that maximize it. Unlike it other questions, there is no closed form way to calculate theta. Instead we chose it using optimization. Here is the partial derivative of log-likelihood with respect to each parameter θ_j:

$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}$$

Gradient Ascent Optimization

Once we have an equation for Log Likelihood, we chose the values for our parameters (θ) that maximize said function. In the case of logistic regression we can't solve for θ mathematically. Instead we use a computer to chose θ. To do so we employ an algorithm called gradient ascent. That algorithms claims that if you continuously take small steps in the direction of your gradient, you will eventually make it to a local maxima. In the case of Logistic Regression you can prove that the result will always be a global maxima.

The small step that we continually take given the training dataset can be calculated as:

$$\begin{aligned}\theta_j^{\text{new}} &= \theta_j^{\text{old}} + \eta \cdot \frac{\partial LL(\theta^{\text{old}})}{\partial \theta_j^{\text{old}}} \\ &= \theta_j^{\text{old}} + \eta \cdot \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}\end{aligned}$$

Where η is the magnitude of the step size that we take. If you keep updating θ using the equation above you will converge on the best values of θ!

Derivations

In this section we provide the mathematical derivations for the log-likelihood function and the gradient. The derivations are worth knowing because these ideas are heavily used in Neural Networks. To start, here is a super slick way of writing the probability of one datapoint:

$$P(Y = y | X = \mathbf{x}) = \sigma(\theta^T \mathbf{x})^y \cdot \left[1 - \sigma(\theta^T \mathbf{x})\right]^{(1-y)}$$

Since each datapoint is independent, the probability of all the data is:

$$\begin{aligned}L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)} | X = \mathbf{x}^{(i)}) \\ &= \prod_{i=1}^{n} \sigma(\theta^T \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^T \mathbf{x}^{(i)})\right]^{(1-y^{(i)})}\end{aligned}$$

And if you take the log of this function, you get the reported Log Likelihood for Logistic Regression.

The next step is to calculate the derivative of the log likelihood with respect to each theta. To start, here is the definition for the derivative of sigma with respect to its inputs:

$$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)[1 - \sigma(z)] \qquad \text{to get the derivative with respect to } \theta\text{, use the chain rule}$$

Derivative of gradient for one datapoint $(\mathbf{x}, y)$:

$$\begin{aligned}\frac{\partial LL(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} y \log \sigma(\theta^T \mathbf{x}) + \frac{\partial}{\partial \theta_j}(1-y)\log[1 - \sigma(\theta^T \mathbf{x}] && \text{derivative of sum of terms} \\ &= \left[\frac{y}{\sigma(\theta^T x)} - \frac{1-y}{1-\sigma(\theta^T x)}\right] \frac{\partial}{\partial \theta_j}\sigma(\theta^T x) && \text{derivative of } \log f(x) \\ &= \left[\frac{y}{\sigma(\theta^T x)} - \frac{1-y}{1-\sigma(\theta^T x)}\right] \sigma(\theta^T x)[1 - \sigma(\theta^T x)] x_j && \text{chain rule + derivative of sigma} \\ &= \left[\frac{y - \sigma(\theta^T x)}{\sigma(\theta^T x)[1 - \sigma(\theta^T x)]}\right] \sigma(\theta^T x)[1 - \sigma(\theta^T x)] x_j && \text{algebraic manipulation} \\ &= \left[y - \sigma(\theta^T x)\right] x_j && \text{cancelling terms}\end{aligned}$$

Because the derivative of sums is the sum of derivatives, the gradient of theta is simply the sum of this term for each training datapoint.

Chris Piech
CS109

Lecture #24
Nov 29th, 2017

Deep Learning

Deep Learning (the new term to refer to Neural Networks) is one of the greatest ideas in computer science that I have been exposed to. On a practical level they are a rather simple extension of Logistic Regression. But the simple idea has had powerful results. Deep Learning is the core idea behind dramatic improvements in Artificial Intelligence. It is the learning algorithm behind Alpha Go, Voice Recognition, Computer Vision (think facebook's ability to recognize a photo of you), Google's Deep Dream, Educational Knowledge Tracing and modern Natural Language Processing. You are about to learn math that has had a big impact on every day life and will likely continue to revolutionize many disciplines and sub-disciplines.

Let's start with intuition gained from a simple analogy. You can think of a Logistic Regression function: $\sigma(\theta^T \mathbf{x})$, as a cartoon model of a single neuron inside your brain. Neural Networks (aka Deep Learning) is the result of putting many layers of Logistic Regression functions together.

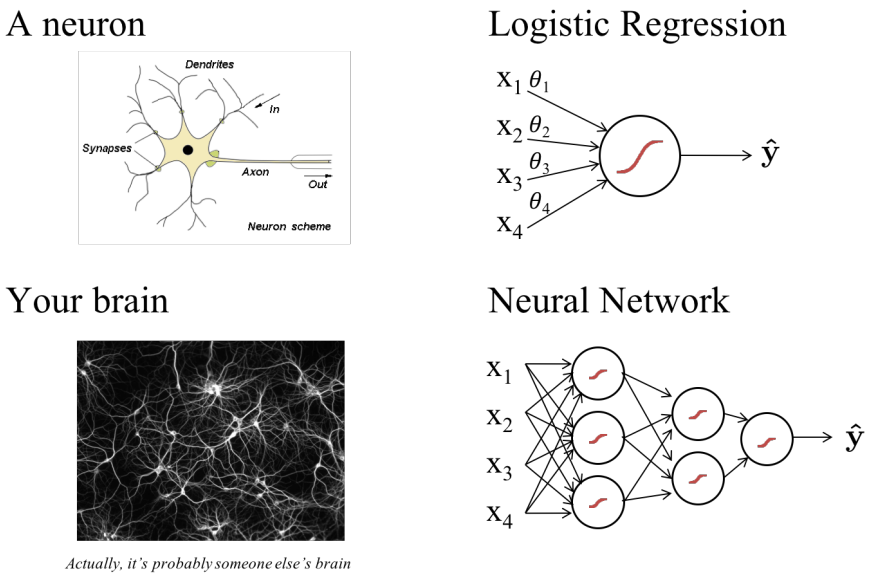


Figure 1: Logistic Regression is a cartoon model of a single neuron. Neural Networks model a brain.

This simple idea allows for models which can represent complex functions from input features ($\mathbf{x}$) to outputs ($\hat{y}$). In CS109 we are going to interpret the output of a neural network in the same was as we interpreted the output of logistic regression: as a prediction of the probability of a class label.

Simple Deep Network

As a motivating example we are going to build a simple Deep Network that can learn to classify hand written digits as either the number "1" or the number "2". Here is a diagram of a neural network that we will use. It has three "layers" of neurons. The input layer ($\mathbf{x}$) is a vector of pixel darkness in the hand drawn number. The hidden layer ($\mathbf{h}$) is a vector of logistic regression cells which are each take all the elements of $\mathbf{x}$ as input. The output layer is a single logistic regression cell that takes all of the elements of the *hidden layer* $\mathbf{h}$ as input. We are going to interpret the output value $\hat{y}$ in the same way that we interpreted the output of vanilla logistic

regression: as an estimation of $P(Y=1|\mathbf{x})$. Formally:

$$\hat{y} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right) = P(Y=1|\mathbf{x}) \tag{1}$$

$$\mathbf{h}_j = \sigma\left(\sum_{i=0}^{m_x} \mathbf{x}_i \theta_{i,j}^{(h)}\right) \tag{2}$$

These equations introduce a few new pieces of notation. Let's spell out what each term means. The parameters of the equations are all of the symbols θ. There are two groups of parameters: the weights for the logistic cells in the hidden unit ($\theta^{(h)}$) and weights for the output logistic cell ($\theta^{(\hat{y})}$). These are collections of parameters. There is a value $\theta_{i,j}^{(h)}$ for every pair of input i and hidden unit j and there is a $\theta_j^{(\hat{y})}$ for every hidden unit j. There are $m_x = |\mathbf{x}|$ number of inputs and there are $m_h = |\mathbf{h}|$ number of hidden units. Familiarize yourself with the notation. The math of neural networks isn't particularly difficult. The notation is!

For a given image (and its corresponding $\mathbf{x}$) the neural network will produce a single value $\hat{y}$. Because it is the result of a sigmoid function it will have a value in the range $[0,1]$. We are going to interpret this value as the probability that the hand written digit is the number "1". This is the same classification assumption made by logistic regression. Here are two diagrams of the same network with one layer of hidden neurons. In the

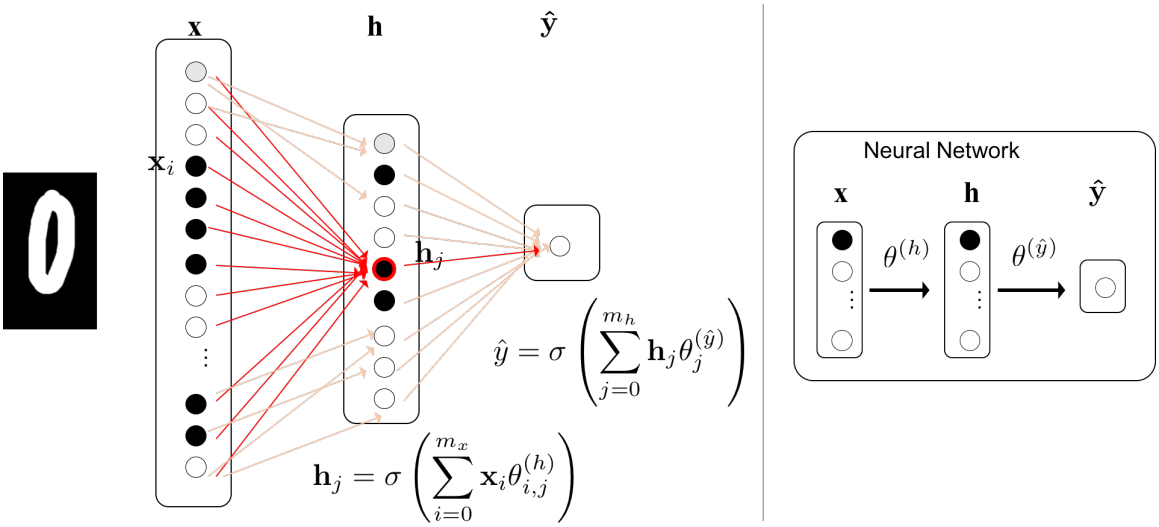


Figure 2: Two diagrams of the same neural network.

figure on the left a single hidden neuron is highlighted. Keep in mind that all hidden neurons take all values $\mathbf{x}$ as input. I could only draw so many arrows between $\mathbf{x}$ and $\mathbf{h}$ without it becoming too messy.

Once you understand the notation, and think through how you would compute a value $\hat{y}$ given θ and $\mathbf{x}$ (called the "Forward Pass") you are most of the way there. The only step left is to think through how to chose the values of θ that maximize the likelihood of our training data. Recall that the process for MLE is to (1) write the log-likelihood function and then (2) find the values of theta that maximize the log-likelihood. Just like in logistic regression we are going to use gradient ascent to chose our thetas. Thus we simply need the partial derivative of the log likelihood with respect to each parameter.

Log Likelihood

We start with the same assumption as logistic regression. For one datapoint with true output y and predicted output $\hat{y}$, the likelihood of that data is:

$$P(Y=y|X=\mathbf{x}) = (\hat{y})^y (1-\hat{y})^{1-y}$$

If you plug in 0 or 1 in for y you get the logistic regression assumption (try it)! If we extend this idea to write the likelihood of n independent datapoints $(\mathbf{x}^{(i)}, \hat{y}^{(i)})$we get:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)} | X = \mathbf{x}^{(i)}) \\ &= \prod_{i=1}^{n} \sigma(\theta^T \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^T \mathbf{x}^{(i)})\right]^{(1-y^{(i)})} \end{aligned}$$

If you take the log of the likelihood you get the following log likelihood function for the neural network:

$$LL(\theta) = \sum_{i=0}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log[1 - \hat{y}^{(i)}] \tag{3}$$

Though this doesn't look like it is an equation in terms of theta, it is. You could plug in the definition for $\hat{y}$.

Backpropagation

We are going to chose values of θ using our old friend MLE (maximum likelihood estimation). MLE applied to deep networks gets a special name "Backpropagation". To chose the optimal values of theta we are going to use gradient ascent where we continually update our thetas in a way that leads to a step up with respect to likelihood. In order to apply gradient ascent we will need to know the partial derivatives of log-likelihood with respect to each of the parameters.

Since the log likelihood of all the data is a sum of the log likelihood of each data point, we can calculate the derivative of the log likelihood with respect to a single data instance $(\mathbf{x}, y)$. The derivative with respect to all the data will simply be the sum of the derivatives with respect to each instance (by derivative of summation).

The one great idea that makes MLE simple for deep networks is that by using the chain rule from calculus we can decompose the calculation of gradients in a deep network. Let's work it out. The values that we need to calculate are the partial derivatives of the log likelihood with respect to each parameter. The big idea that is really worth wrapping your head around, is that chain rule can let us calculate gradients one layer at a time. By the chain rule we can decompose the calculation of the gradient with respect to the output parameters as such:

$$\frac{\partial LL(\theta)}{\partial \theta_j^{(\hat{y})}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} \tag{4}$$

Similarly we can decompose the calculation of the gradient with respect to the hidden layer parameters as:

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mathbf{h}_j} \cdot \frac{\partial \mathbf{h}_j}{\partial \theta_{i,j}^{(h)}} \tag{5}$$

Each of those terms is reasonable to calculate. Here are their closed form equations:

$$\frac{\partial LL(\theta)}{\partial \hat{y}} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})} \qquad \frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} = \hat{y}[1-\hat{y}] \cdot h_j$$

$$\frac{\partial \hat{y}}{\partial \mathbf{h}_j} = \hat{y}[1-\hat{y}]\theta_j^{(\hat{y})} \qquad \frac{\partial \mathbf{h}_j}{\partial \theta_{i,j}^{(h)}} = \mathbf{h}_j[1-\mathbf{h}_j]x_j$$

In this simple model, we only have one layer of hidden neurons. If we added more we could keep using the chain rule to calculate derivatives with respect to parameters deeper in the network.

Example 1

As an example, consider evaluating the partial derivative $\frac{\partial LL(\theta)}{\partial \hat{y}}$. To do so first write out the function for LL in terms of $\hat{y}$ then differentiate:

$$LL = y \log \hat{y} + (1-y) \log[1-\hat{y}]$$
$$\frac{\partial LL(\theta)}{\partial \hat{y}} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

It is that simple! Let's try another example.

Example 2

Let's calculate the partial derivative of $\hat{y}$ with respect to an output parameter $\theta_j^{(\hat{y})}$:

$$\hat{y} = \sigma(z) \qquad \text{Where } z = \sum_{i=0}^{m_h} \mathbf{h}_i \theta_i^{(\hat{y})}$$
$$\frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} = \sigma(z)[1-\sigma(z)] \frac{\partial z}{\partial \theta_j^{(\hat{y})}} \qquad \text{Using the formulate for derivative of sigmoid}$$
$$= \hat{y}[1-\hat{y}] \cdot h_j \qquad \text{Recognizing that } \hat{y} = \sigma(z)$$

The Future

Deep learning is a growing field. There is a lot of room for improvement. Can we think of better networks? Can we develop structures that do a better job of incorporating prior beliefs? These problems (and many others) are open. It is worth knowing the math of deep learning well because you may one day have to invent the next iteration.