1. Counting

Chris Piech and Mehran Sahami

Oct 2017

1 Introduction

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.

The ideas presented in this chapter 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 probability for computer scientists, 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. While this chapter goes over how to count for many foreseeable circumstances, we place special emphasis on learning how to count *distinct* objects. This will prove useful when we transition into probability.

2 Basic Building Blocks

Counting with STEPS

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 $|A||B| = mn$.

Example 1

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

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. Think of the overall "experiment" of rolling the two dice as having two parts: in the first part we roll the first die[1], in the second part we roll the second die. Since the first die can come up with 6 possible

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

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 2

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?

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.

Counting with OR

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 $A \cap B = \emptyset$, then the number of outcomes of the experiment is $|A| + |B| = m + n$.

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 $A \cap B = \emptyset$), 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 $A \cap B = \emptyset$, the Inclusion-Exclusion Principle gives the same result as the Sum Rule of Counting since $|\emptyset| = 0$.

Example 3

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?

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.

Double Counting and Constraints

There are many reasons for having counted some elements more than once (aka "double counted"). One common case, is that there is a constraint in the problem that you must contend with. It goes without saying that if you over-count, then you have to subtract off the number of elements that were double counted. If you did something along the lines of: count every element some multiple, then you can divide your total number of elements by that multiple to get the correct final answer.

3 Combinatorics

Counting problems can be approached from the basic building blocks described in the first section. However some counting problems are so ubiquitous in the world of probability that it is worth knowing a few higher level counting abstractions. When solving problems, if you can find the analogy from these canonical examples you can build off of the corresponding combinatorics formulas:

Permutations of Distinct Objects

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 4

iPhones have 4-digit passcodes. Suppose there are 4 smudges over 4 digits on the screen. How many distinct passcodes are possible?

Since the order of digits in the code 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.

What if there are 3 smudges over 3 digits on screen? One of 3 digits is repeated, but we don't know which one. We can solve this by making three cases, one for each digit that could be repeated (each with the same number of permutations). Let A, B, C represent the 3 digits, with C repeated twice. We can initially pretend the two C's are distinct. Then each case will have 4! permutations:

$$A\ B\ C_1\ C_2$$

However, then we need to eliminate the double-counting of the permutations of the identical digits (one A, one B, and two C's):

$$\frac{4!}{2! \cdot 1! \cdot 1!}$$

Adding up the three cases for the different repeated digits gives

$$3 \cdot \frac{4!}{2! \cdot 1! \cdot 1!} = 3 \cdot 12 = 36$$

What if there are 2 smudges over 2 digits on the screen? There are two possibilities: 2 digits used twice each, or 1 digit used 3 times, and other digit used once.

$$\frac{4!}{2! \cdot 2!} + 2 \cdot \frac{4!}{3! \cdot 1!} = 6 + (2 \cdot 4) = 6 + 8 = 14$$

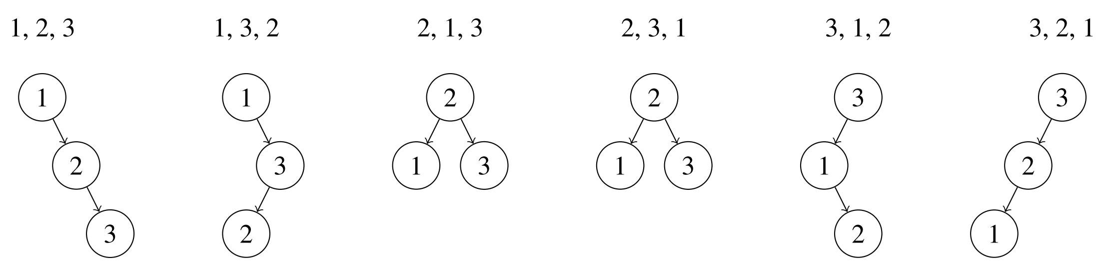
Example 5

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.

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)?

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).

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 $\frac{n!}{n_1!n_2!\ldots n_r!}$ permutations

Example 6

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

5 total digits would give 5! permutations. But that is assuming the 0's and 1's are distinguishable (to make that explicit, let's give each one a subscript). Here is a subset of the permutations.

$$\begin{array}{ccccc} 0_1 & 1_1 & 1_2 & 0_2 & 0_3 \\ 0_1 & 1_1 & 1_2 & 0_3 & 0_2 \\ 0_2 & 1_1 & 1_2 & 0_1 & 0_3 \\ 0_2 & 1_1 & 1_2 & 0_3 & 0_1 \\ 0_3 & 1_1 & 1_2 & 0_1 & 0_2 \\ 0_3 & 1_1 & 1_2 & 0_2 & 0_1 \end{array}$$

If identical digits are indistinguishable, then all the listed permutations are the same. For any given permutation, there are 3! ways of rearranging the 0's and 2! ways of rearranging the 1's (resulting in indistinguishable strings). We have over-counted. Using the formula for permutations of indistinct objects, we can correct for the over-counting:

$$\text{Total} = \frac{5!}{3! \cdot 2!} = \frac{160}{6 \cdot 2} = \frac{120}{12} = 10.$$

Combinations of Distinct Objects

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 distinct, unordered objects from a set of n distinct 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.

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

The total ways to chose 3 objects from a set of 7 distinct objects is:

$$\text{total} = \binom{7}{3} = \frac{7!}{3!(7-3)!} = 35$$

Example 7

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

This is a straightforward combinations problem. $\binom{8000}{2} = 31{,}996{,}000$.

Example 8

How many ways are there to select 3 books from a set of 6?

If each of the books are distinct, then this is another straightforward combination problem. There are $\binom{6}{3} = \frac{6!}{3!3!} = 20$ ways.

How many ways are there to select 3 books if there are two books that should not both be chosen together (for example, don't choose both the 8th and 9th edition of the Ross textbook)? This problem is easier to solve if we split it up into cases. Consider the following three different cases:

Case 1: Select the 8th Ed. and 2 other non-9th Ed.: There are $\binom{4}{2}$ ways of doing so.

Case 2: Select the 9th Ed. and 2 other non-8th Ed.: There are $\binom{4}{2}$ ways of doing so.

Case 3: Select 3 from the books that are neither the 8th nor the 8th edition: There are $\binom{4}{3}$ ways of doing so.

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

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

Alternatively, we could have calculated all the ways of selecting 3 books from 4, and then subtract the "forbidden" ones (i.e., the selections that break the constraint). Chris Piech calls this the Forbidden City method.

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

$$\text{Total} = \text{All possibilities} - \text{forbidden} = 20 - 4 = 16.$$

Two different ways to get the same right answer!

4 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. I looked up why people called their containers urns. 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.

Note that there are many flavors of group assignment (eg with replacement, without replacement).

Assignment of Distinct Objects

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 Product Rule of Counting this comes out to r^n.

Assignment of Indistinct Objects

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 9

Say you are a startup incubator and you have \$10 million to invest in 4 companies (in \$1 million increments). How many ways can you allocate this money?

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

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

What if you don't have to invest all \$10 M? (The economy is tight, say, and you might want to save your money.)

Imagine that you have an extra company: yourself. Now you are investing \$10 million in 5 companies. Thus, the answer is the same as putting 10 balls into 5 urns.

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

What if you know you want to invest at least \$3 million in Company 1?

There is one way to give \$3 million to Company 1. The number of ways of investing the remaining money is the same as putting 7 balls into 4 urns.

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

---

5 Computer Science Case Studies

Counting is important in the world of computer science for a few reasons. Primarily (1) in order to understand probability on a fundamental level, it is useful to first understand counting (2) while computers are fast, some problems require so much work that they would take an unreasonable amount of time to complete. Using counting we can better estimate how much work we have to do (3) some counting problems are so large and complex that we benefit from computation to solve them.

Example: Atoms in the Universe

Peter Norvig, the current director of research at Google made a great argument about the number of atoms in the universe and how it relates to computer science: The number of atoms in the observable universe is about 10 to the 80th power (10^{80}). This measure is frequently used as a canonical really big number. There certainly are a lot of atoms in the universe. As a leading expert said,

> "Space is big. Really big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist, but that's just peanuts to space." - Douglas Adams

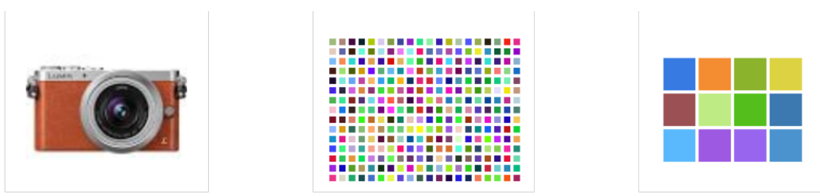
This number is often used to demonstrate tasks that computers will never be able to solve. Problems can quickly grow to such an absurd size through the product rule of counting. Let's look at a few examples (again, thanks to Peter).

A Go board has 19×19 points where a user can place a stone. Each of the points can be empty or occupied by black or white stone. By the product rule of counting, we can compute the number of unique board configurations. Each board point is a unique choice where you can decide to have one of the three options in the set $\{$Black, White, No Stone$\}$ so there are $3^{(19\times19)} \approx 10^{172}$ possible board positions, but "only" about 10^{170} of those positions are legal. That is about the square of the number of atoms in the universe. In otherwords: if there was another universe of atoms for every single atom, only then would there be as many atoms in the universe as there are unique configurations of a Go board.

Example: The Number of Digital Pictures

Let's switch from Go positions to digital pictures. There is an art project to display every possible picture. Surely that would take a long time, because there must be many possible pictures. But how many? We will assume the color model known as True Color, in which each pixel can be one of $2^{24} \approx 17$ million distinct colors.

How many distinct pictures can you generate from (a) a digital camera shown with 12 million pixels, (b) a grid with 300 pixels, and (c) a grid with just 12 pixels?



(a) 12 million pixels (b) 300 pixels (c) 12 pixels

Answer: An array of n pixels produces (17 million)^n different pictures. (17 million)$^{12} \approx 10^{86}$, so the tiny 12-pixel grid produces a million times more pictures than the number of atoms in the universe! How about the 300 pixel array? It can produce 10^{2167} pictures. You may think the number of atoms in the universe is big, but that's just peanuts to the number of pictures in a 300-pixel array. And 12M pixels? $10^{86696638}$ pictures.

So the number of possible pictures is really, really, really big. And the number of atoms in the universe is looking relatively small, at least as a number of combinations. The crucial idea is, that as a number of physical things, 10^{80} is a really big number. But as a number of combinations of things, 10^{80} is a rather small number. It doesn't take a universe of stuff to get up to 10^{80} combinations.

Example: Leveraging Exponential Growth

The above argument might leave you feeling like some problems are incredibly hard as a result of the product rule of counting. Let's take a moment to talk about how the product rule of counting can help! Most logrithmic time algorithms leverage this principle. If 2^n becomes incredibly large fast, it also holds that $\log_2(n)$ stays small even for enormous n.

For a real world example: recently a colleague and I were doing a machine learning project that required a lot of data. In fact we needed around 10 million unique solutions to an assignment given to intro to programming students (which we didn't have). My colleague and I decided to generate the data ourselves. How could we create 10 million unique answers to a programming problem? It was easy! The rubric for the assignment articulated 24 binary decisions that students could make while writing their solutions. We encoded all 24 decision points and ended up with $2^{24} \approx 1.6$million unique solutions!

Example: Clustering

At some point in all of your careers as computer scientist, you will probably want to do something along the lines of: given n objects, run an algorithm that does a comparison between every pair of objects. One concrete example of this comes from clustering n cancer cells from a single person's body, based on the DNA of the cells. As a first step we need to run a "similarity" metric to calculate how similar the DNA of each cell is to the other. How many times do you need to run the pairwise comparison?

The simplest answer is that you can think of choosing two objects from a set of n where the first choice is distinct from the second. Thus there are $n \times n = n^2$ comparisons. But what if we don't consider the ordering as important? In other-words, if the similarity of A to B is equal to the similarity of B to A, we don't need to compute both. You may be tempted to divide n^2 by two to fix the double counting, however there is a small oversight in this thinking. Consider the following concrete example with 14 unique species. The number of unique pairwise comparisons, where order doesn't matter, if you do not include comparing a species to itself, can be counted from the grid of all 14^2 comparisons: there are 91. If you include comparing elements to themselves there are 105 pairwise comparisons. Unfortunately $\frac{14^2}{2} = 98$. The oversight is that we were double counting every comparison *except* comparisons of elements to themselves.

Another way to approach this problem is to start by thinking of the number of unique comparisons as an unordered selection. We have a rule for how many ways you can chose two elements from n where order does not matter. That is simply $\binom{n}{2}$. In this case $\binom{14}{2} = 91$, which is the right answer.

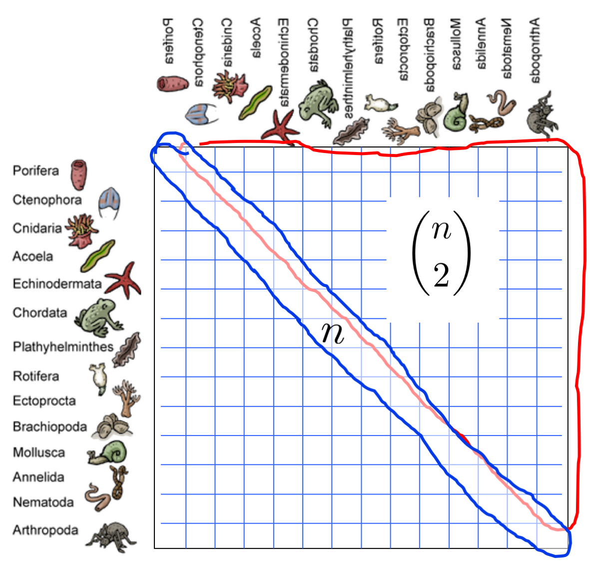


Figure 1: Number of Comparisons of $n = 14$ objects

If you apply the combinations formula, you are getting the number of ways of choosing two items, without replacement! In other words it does not include comparisons of elements to themselves. If we want to include such comparisons, then there are $\binom{n}{2} + n = \binom{14}{2} + 14 = 105$ unique comparisons.