Counting II

Margaret M. Fleck

9 April 2010

This lecture covers more examples of permutations and combinations, from section 5.3 of Rosen plus part of section 5.4. It also introduces "combinatorial" proofs.

1 Recap: basic counting methods

Last class, we covered several basic counting rules.

The product rule: if you have p choices for one part of a task, then q choices for a second part, and your options for the second part don't depend on what you chose for the first part, then you have pq options for the whole task.

The sum rule: suppose your task can be done in one of two ways, which are mutually exclusive. If the first way has p choices and the second way has q choices, then you have $p + q$ choices for how to do the task.

Inclusion-exclusion principle: if the two sets of choices do overlap, you have to subtract the overlap amount so you don't double-count those options.

Permutations: if S is a set of n objects (all different), then a k-permutation of S is a way to put k of the objects from S into an ordered list. There are $P(n, k) = n(n-1)\ldots(n-k+1) = \frac{n!}{(n-k)!}$ different k-permutations.

For example, suppose I own ten differently-decorated coffee mugs but I only want to put six of them out on the table. Then I have 10 choices for which to put at the first place, 9 for what to put at the second, and so on down to 5 choices for the last mug to put out. That is, $P(10, 6) = \frac{10!}{4!}$.

Notice that these formulas only work if all items in the set are distinguishable and we don't get to pick duplicates of the same item. When those assumptions fail, we have to restructure the problem or use a more complex formula.

2 Combinations

The permutations formula applies when we care about the order in which we are selecting the objects, e.g. we are putting them into an arrangement or choosing them for a series of different roles. If we simply want to select a subset of the objects, we need a different formula. An unordered set of k elements is called a k-combination.

Example: How many ways can I select a 7-card hand from a 60-card deck of Magic cards (assuming no two cards are identical)?[1]

One way to analyze this problem is to figure out how many ways we can select an ordered list of 7 cards, which is $P(60, 7)$. This over-counts the number of possibilities, so we have to divide by the number of different orders in which the same 7-cards might appear. That's just 7!. So our total number of hands is $\frac{P(60,7)}{7!}$ This is $\frac{60\cdot59\cdot58\cdot57\cdot56\cdot55\cdot54}{7\cdot6\cdot5\cdot4\cdot3\cdot2}$. Probably not worth simplifying or multiplying this out unless you really have to. (Get a computer to do it.)

In general, suppose that we have a set S with n elements and we want to choose an unordered subset of k elements. We have $\frac{n!}{(n-k)!}$ ways to choose k elements in some particular order. Since there are $k!$ ways to put each subset into an order, we need to divide by $k!$ so that we will only count each subset once. So the general formula for the number of possible subsets is $\frac{n!}{k!(n-k)!}$.

The expression $\frac{n!}{k!(n-k)!}$ is often written $C(n, k)$ or $\binom{n}{k}$. This is pronounced "n choose r." It is also sometimes called a "binomial coefficient," for reasons that will become obvious shortly. So the shorthand answer to our question about magic cards would be $\binom{60}{7}$.

[1] Ok, ok, for those of you who play Magic, it's not that each to make the land cards all different. But it should be possible in theory, right?

Notice that $\binom{n}{r}$ is only defined when $n \geq r \geq 0$. What is $\binom{0}{0}$? This is $\frac{0!}{0!0!} = \frac{1}{1\cdot 1} = 1$.

3 The bit string viewpoint

Applying the combinations formula can also require reworking how you think about the problem. In particular, it often helps to see placing objects in certain positions of an arrangement as choosing a subset of the positions.

Example: How many 16-digit bit strings contain exactly 5 zeros?

Solution: The string contains 16 positions. We need to pick 5 of these to be the ones with the zeros. So we have $\binom{16}{5}$ ways to do this.

More complex example: How many 10-digit strings from the 26-letter ASCII alphabet contain exactly 3 A's?

Solution: We need to pick a subset of the 10 positions in which to put the three A's. There are $\binom{10}{3}$ ways to do this. After we've done that, we have seven positions to fill with our choice of any character except A. We have 25^7 ways to do that. So our total number of strings is $\binom{10}{3}25^7$.

Solutions like this are somewhat custom, and not easy to generalize to apparently similar problems. You have to think about each situation carefully when your problem involves multiple identical objects.

Modified example: How many 10-character strings from the 26-letter ASCII alphabet contain no more than 3 A's?

Solution: We do the above analysis to count the number of strings with exactly 3 A's, exactly 2 A's, exactly 1 A, and no A's. Then add up these four results. So the total number of strings is

$$\binom{10}{3}25^7 + \binom{10}{2}25^8 + \binom{10}{1}25^9 + 25^{10}$$

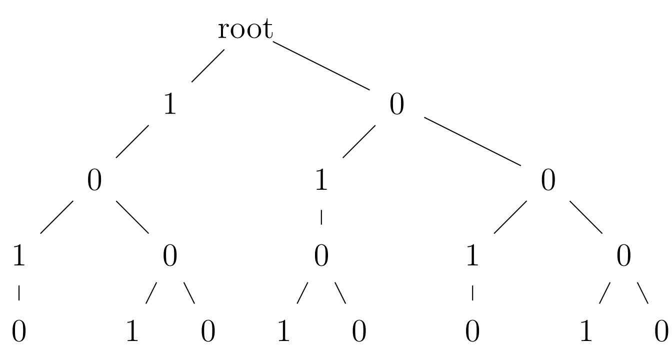
4 Tackling less obvious problems

The formulas for counting are not especially difficult to use. The biggest problem is knowing how to describe your problem in terms of a known formula. This sometimes takes some fiddling around.

For example, suppose we have a set of 7 adults and 3 kids. Let's call the kids A, B, and C. They need to stand in line to board an airplane and no two kids can stand next to each other because they will fight with one another and cause trouble. We have two choices for who is first. But then the later choices depend in a complex way on the earlier ones. This isn't going to work real well.

The trick for this problem is to place the 7 adults in line, with gaps between them. Each gap might be left empty or filled with one kid. There are 8 gaps, into which we have to put the 3 kids. So, we have 7! ways to assign adults to positions. Then we have 8 gaps in which we can put kid A, 7 for kid B, and 6 for kid C. That is $7! \cdot 8 \cdot 7 \cdot 6$ ways to line them all up.

When the problem is small and the dependencies among the decisions are complex, you can resort to drawing a tree diagram enumerating all the possibilities. For example, suppose we want to find all bit strings (strings of 0's and 1's) of length four which do not have two consecutive ones. We can draw the following tree, in which each branch represents one possible string. We can see that there are 8 strings with these properites.



5 Pigeonhole Principle

Pigeonhole principle: Suppose you have n objects and assign k labels to these objects. If $n > k$, then two objects must get the same label.

More specifically, there must be at least one group of $\lceil \frac{n}{k} \rceil$ objects sharing the same label. For example, if you have 47 playing cards and the cards come in five colors, then you must have at least 10 cards that are all the same color.

How to remember this? Suppose you want to assign the same label to no more than p objects. Then the most objects you can have is pk. That is, $p = \frac{n}{k}$. If n is any larger, then you'll need to assign (at least) $p + 1$ objects to one of the labels.

You'll often see the pigeonhole principle pulled out of nowhere as a clever trick in proofs. Such proofs are easy to read, but sometimes hard to come up with. For example, here's a fact that's less than obvious:

Claim 1 *Let S be a set of $n + 1$ natural numbers, all between 1 and $2n$ (inclusive). There must be a member of S that divides another member of S.*

> Proof: Suppose that the members of S are $a_1, a_2, \ldots, a_{n+1}$. We can write each integer as a power of two times an odd integer. That is, $a_i = 2^{k_i} q_i$ for each i.
>
> Consider the integers $q_1, \ldots, q_{n+1}$. They are all odd. But there are only n odd integers in the range 1 through $2n$. So two of them are the same. That is, there are two integers s and t such that $a_s = 2^{k_s} p$ and $a_t = 2^{k_t} p$.
>
> Suppose without loss of generality that $k_s \leq k_t$. (If that's not the case, switch which of the indices we've called s vs t.) Then $2^{k_s} p | 2^{k_t} p$, so $a_s | a_t$.

6 Binomial Theorem

A *binomial* is a sum of two terms, e.g. $(x + y)$. The *binomial theorem* shows how to raise a binomial to any integer power. Specifically

Claim 2 *(Binomial Theorem) Let x and y be variables and let n be any natural number. Then*

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Because of this application, the values $\binom{n}{k}$ are sometimes called **binomial coefficients**.

> Proof: If we were to expand the product $(x+y)^n$, each term is the product of n variables, some x's and the rest y's. For example, if $n = 6$, one term is $yxyyxx$. So each term is an ordered list of x's and y's.
>
> When we collect up terms, we group together the lists that have the same number of x's. To find the coefficient for $x^{n-k}y^k$, we need to count how many ways we can make a list of n elements that contains k y's and $n-k$ x's. This amounts to picking a subset of k elements from a set of n positions in the list. In other words, there are $\binom{n}{k}$ such terms. □

7 Corollaries of the Binomial Theorem

Suppose that we set $x = 1$ in the Binomial Theorem. Then we have

$$(1+y)^n = \sum_{k=0}^{n} \binom{n}{k} 1^{n-k} y^k = \sum_{k=0}^{n} \binom{n}{k} y^k$$

So we have the following corollary.

Claim 3 *For any variable y and any natural number n, $(1+y)^n = \sum_{k=0}^{n} \binom{n}{k} y^k$*

A corollary is a result that is very easy to prove, once you've proved some theorem (which often had a hard proof).

We could also state this as:

Claim 4 *For any variable y and any natural number n, $\sum_{k=0}^{n} \binom{n}{k} y^k = (1 + y)^n$.*

If we plug some specific values of y into this formula, we get some nice results about sums of binomial coefficients. For example, if $y = 1$, then we have $\sum_{k=0}^{n} \binom{n}{k} = 2^n$

One way to understand this equation is that we are counting all subsets of a set S that contains n elements. 2^n is the total number of subsets. The summation on the left considers each possible size of subset (k). For each size k, it computes the number of subsets of size k.

If $y = -1$, then we have

$\sum_{k=0}^{n} \binom{n}{k} (-1)^k = 0.$

If $y = 2$, then we have

$\sum_{k=0}^{n} \binom{n}{k} 2^k = 3^n.$

We don't expect that you'll remember all these random identities involving binomial coefficients. Rather, we're hoping that you remember the important named ones, and that you could figure out how to rederive the others (given some time to fiddle around).