1 Other Counting Strategies

In the previous lecture, we saw a huge variety of different counting scenarios: with and without replacement, order matters and does not matter, and so on. We saw that the First and Second Rules of Counting formed the basis of many counting strategies. Here, we highlight a few more commonly seen counting tools; we'll encounter most of these tools again when studying discrete probability.

While these strategies help solve many kinds of counting problems, **this is by no means an exhaustive list**. It is good to be familiar with these strategies, but most counting problems in the wild may involve combining a few of these at a time (with the First and Second rules).

1.1 Casework

For some problems, it may be useful to split the objects that we want to count into a small number of (non-overlapping) cases. Ideally, each individual case is easy to count.

Say we want to count the number of n-bit strings that either start with a 0 or start with two 1's. Counting strings that start with a 0 is a straightforward application of the First Rule of Counting. The first bit is fixed, and the remaining $(n-1)$ bits each have 2 choices, giving us 2^{n-1} strings. Counting strings that start with two 1's is also a single application of the First Rule of Counting. We fix the first two bits, and the remaining $(n-2)$ bits each have 2 choices, giving us 2^{n-2} strings. Since these two sets of strings can't overlap (why?), we do not over-count any strings. Our final count will be $2^{n-1}+2^{n-2}$.

1.2 The Complement

Say we have a set of objects S, and we want to count a subset of them, A. If A itself is difficult to count, but its complement $S \setminus A$ (i.e. elements of S not in A) is easy to count, we can instead count the total number of objects in S, and subtract the number in $S \setminus A$. In other words,

$$|A| = |S| - |S \setminus A|$$

Say we want to count the number of 6-bit strings with at least one 0. Our initial hunch may be to split this up into disjoint cases. Using the Second Rule of Counting, we can count strings with exactly one 0: there are $\binom{6}{1} = 6$. We can also do this for strings with exactly two 0's ($\binom{6}{2} = 15$), three 0's ($\binom{6}{3} = 20$), and so on. Combining these cases gives us a grand total of:

$$\binom{n}{1}+\binom{n}{2}+\binom{n}{3}+\binom{n}{4}+\binom{n}{5}+\binom{n}{6} = 6+15+20+15+6+1 = 63$$

For 6-bit strings, this isn't too bad. What happens though, when we want to do this for 10-bit strings? 30-bit strings? This method won't scale.

However, notice that if a string doesn't have at least one 0, then it must be the all 1's string! The total number of n-bit strings is 2^n. By counting the complement, we get that the number of n-bit strings with at least one zero is $2^n - 1$.

1.3 Symmetry

Counting the complement involved surveying a larger space of objects, S, and seeing if the set we want to count, A, had some nice relationship to S (i.e. $S \setminus A$ is easy to count). The idea of symmetry generalizes this strategy of looking at A's relationship to S. Often, we can find other sets living in S that are similar to A.

We can illustrate the idea behind symmetry by once again using bit strings. Say we want to find the number of 9-bit strings with more 0's than 1's. We can again attempt casework, but as with the previous example, this strategy won't scale.

However, there is an easy way to relate the set of strings with more 0's than 1's to the set of strings with more 1's than 0's! If we start with a string in the former, we can flip each bit to get a string in the latter. This bit-flipping operation is actually a bijection between these two sets, so we can conclude:

$$\#\{\text{strings with more 0's than 1's}\} = \#\{\text{strings with more 1's than 0's}\}$$

Sanity check! Verify that the bit flip operation is a bijection.

Since we are working with odd-length strings, every string must belong to exactly one of the two sets we described! Therefore, $\#\{\text{strings with more 0's than 1's}\}$ must be exactly *half* of the total number of 9-bit strings. This gives us an answer of 2^8.

Sanity check! Using symmetry, solve the above problem for even n.

1.4 Set Unions

Sometimes, we are tempted to use casework, but the cases are not disjoint. To get around this, we'll use set theory! Say we want to count a set of objects that all fall into either (finite) set A, (finite) set B, or both. Then, we can describe our desired set of objects by $A \cup B$, their union.

Let's try find $|A \cup B|$. If we naively add $|A|$ and $|B|$, we overcount objects that belong to the intersection of the two sets, $A \cap B$. However, each one of those objects is only overcounted *exactly once*. Therefore, we can handle the overcounting by subtracting $|A \cap B|$.

$$|A \cup B| = |A| + |B| - |A \cap B|$$

To see this in action, we again use our bit string example. Say we want to count the number of n-bit strings that either start with a 0 or end with a 1. These two conditions are not mutually exclusive: the string $00 \cdots 01$ satisfies both.

Let A be the strings that start with 0 and B be the strings that end with 1. Using the First Rule of Counting, we see that both $|A| = |B| = 2^{n-1}$, while $|A \cap B| = 2^{n-2}$. This gives us a total of $2^n - 2^{n-2}$ strings.

1.5 The Principle of Inclusion-Exclusion

The scenario we described above is a special case of the Principle of Inclusion-Exclusion. We'll give a more formal treatment of it later, when studying discrete probability. However, we'll explain it here at a high level (i.e. no crazy notation), and walk through the three-set case.

Let A,B,C be finite sets. How do we compute $|A \cup B \cup C|$? We try something similar to what we did for the two-set case. First, we can add $|A|+|B|+|C|$. However, for an element in *exactly two* of the three sets, we overcount them once. For elements in *all three* sets, we overcount them twice.

To deal with the overcounting, we can subtract $|A \cap B|$, $|A \cap C|$, and $|B \cap C|$. This fixes the overcounting of elements in *exactly two* of the three sets, but for elements in $A \cap B \cap C$, we are overall *undercounting* them once (why?). Thus, we need to also add back $|A \cap B \cap C|$. To summarize:

$$|A \cup B \cup C| = (|A|+|B|+|C|) - (|A \cap B| + |A \cap C| + |B \cap C|) + (|A \cap B \cap C|)$$

It turns out that a more general pattern holds for larger numbers of sets. Let $A_1, A_2, \ldots, A_n$ be finite sets. Then, the Principle of Inclusion-Exclusion tells us:

$$\left| \bigcup_{i=1}^{n} A_i \right| = (\text{size-1 intersections}) - (\text{size-2 intersections}) + (\text{size-3 intersections}) - \ldots$$

2 Combinatorial Proofs

Combinatorial arguments are interesting because they rely on intuitive counting arguments rather than algebraic manipulation. We can prove complex facts, such as $\binom{n}{k+1} = \binom{n-1}{k} + \binom{n-2}{k} + \cdots + \binom{k}{k}$. You can directly verify this identity by algebraic manipulation. But you can also do this by interpreting what each side means as a combinatorial process. The left hand side is just the number of ways of choosing a subset X with $k+1$ elements from a set $S = \{1,\ldots,n\}$. Let us think about a different process that results in the choice of a subset with $k+1$ elements. We start by picking the lowest-numbered element of X. If we picked 1, to finish constructing X, we must now choose k elements out of the remaining $n-1$ elements of S greater than 1; this can be done in $\binom{n-1}{k}$ ways. If instead the lowest-numbered element we picked is 2, then we have to choose k elements from the remaining $n-2$ elements of S greater than 2, which can be done in $\binom{n-2}{k}$ ways. Moreover all these subsets are distinct from those where the lowest-numbered element was 1. So we should add the number of ways of choosing each to the grand total. Proceeding in this way, we split up the process into cases according to the first (i.e., lowest-numbered) object we select, to obtain:

$$\textbf{First element selected is} \begin{cases} \text{element } 1, & \text{leading to } \binom{n-1}{k} \text{ remaining ways} \\ \text{element } 2, & \text{leading to } \binom{n-2}{k} \text{ remaining ways} \\ \text{element } 3, & \text{leading to } \binom{n-3}{k} \text{ remaining ways} \\ \vdots & \\ \text{element } (n-k), & \text{leading to } \binom{k}{k} \text{ remaining ways} \end{cases}$$

(Note that the lowest-numbered object we select cannot be higher than $n-k$ as we have to select $k+1$ distinct objects.)

The last combinatorial proof we will do is the following: $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n$. To see this, imagine that we have a set S with n elements. On the left hand side, the i-th term counts the number of ways of choosing a subset of S of size exactly i; so the sum on the left hand side counts the total number of subsets (of any size) of S.

We claim that the right hand side (2^n) does indeed also count the total number of subsets. To see this, just identify a subset with an n-bit vector, where in each position j we put a 1 if the j-th element is in the subset, and a 0 otherwise. So the number of subsets is equal to the number of n-bit vectors, which is 2^n (there are

2 options for each bit). Let us look at an example, where $S = \{1,2,3\}$ (so $n = 3$). Enumerate all $2^3 = 8$ possible subsets of S: $\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$. The term $\binom{3}{0}$ counts the number of ways to choose a subset of S with 0 elements; there is only one such subset, namely the empty set $\varnothing = \{\}$. There are $\binom{3}{1} = 3$ ways of choosing a subset with 1 element, $\binom{3}{2} = 3$ ways of choosing a subset with 2 elements, and $\binom{3}{3} = 1$ way of choosing a subset with 3 elements (namely, the subset consisting of every element of S). Summing, we get $1+3+3+1 = 8$, as expected.