Combinatorics: The Fine Art of Counting

Chinese Dice Group Activity

Overview

In an ancient Chinese game, players roll six 6-sided dice in a single throw. A throw is awarded points by categorizing the results according to the number of dice that show the same value, e.g the throw (3,3,3,3,3,3) has all six values the same, the throw (3,4,3,3,5,4) has three values the same plus two values the same plus one other value. Our goal is to analyze the probabilities of the different types of throws.

Step 1) Invent a notation for describing the different types of throws and construct a list of all the possible types.

Step 2) Perform a sequence of 36 throws (use teamwork!) and classify them using your list. Note frequency of each throw type.

Step 3) Based on your results and/or intuition, construct a provisional ranking of the throw types according by probability, from lowest to highest.

Step 4) Compute the actual probability of each type of throw and rank the throw types by probability. Check your computations by verifying that the sum of all the probabilities is 1.

Step 5) Compare the results with your provisional ranking. Were there any surprises?

Areas for further investigation

In order to complete step 1 you need to determine the number of different ways that 6 could be written as an unordered sum of positive integers; e.g. 4+1+1=6 was one type of throw 3+2+1=6 was another. Such an unordered sum is called a **partition of an integer**. We have looked at ordered partitions in class, but unordered partitions present a number of new and interesting challenges and there are many unsolved problems in this area. Can you determine a rule or a pattern for counting the number of partitions of a given integer?

"Craps" Group Activity

Overview

Craps is a popular casino dice game which uses two six-sided dice. A turn of the game begins with a player (called the "shooter") making an initial roll of the dice (called the "come-out" roll). If the initial roll is a 7 or an 11 they win and get to take another turn. If they roll a 2, 3, or 12 (called "craps"), they lose but get to take another turn. Any other roll is called a "point" and the shooter must then repeatedly roll the dice until they either roll a 7, in which case they lose and must pass the dice to the next player, or they roll the same "point" as their initial roll in which case they win and get to take another turn. We want to determine the probability that the shooter wins on a given turn.

Step 1) Play the game craps for several rounds.

Step 2) Compute the probability of winning (7 or 11) or losing (2, 3, or 12) on the initial turn. Then compute the probability of getting each of the six "point" values (4, 5, 6, 8, 9, 10).

Step 3) For a particular point value, compute the probability that if you roll the dice repeatedly you will get that point value before you roll a 7. Note that this may take an arbitrarily long sequence of rolls.

Step 4) Repeat the analysis in step 3 for the other distinct point value probabilities.

Step 5) Using the Law of Alternatives, combine your results to determine the overall probability that the shooter wins by combining conditional probabilities based on the outcome of the first roll. Your answer should be close to ½. Is this a fair game? If not, what is the casino's advantage?

Areas for further investigation

The game craps can be modeled as a simple finite state machine with two different states (the initial roll and all subsequent rolls) where the behavior of the game is different in each state. The outcome of the game depends on the sequence of transitions through these states. Craps is a very simple example - can you think of other games that can be modeled in this way? Can you come up with a general method for computing probabilities for such games?

Combinatorics: The Fine Art of Counting

"Set" Group Activity

Overview

The game Set is based on an unusual deck of cards in which each card has four different properties: Color, Number, Shape, and Texture. Each property can have one of three distinct values. Each of the 81 possible combinations of these properties is represented by one card in the deck. The game is based on finding "Sets" of three cards which have the following characteristic: for each of the four properties, every card in the set must either have the same value, or they must all have different values. 12 cards are dealt face up and all players simultaneously try to find a Set of three cards among the 12. Once a Set is found it is removed and replaced by three new cards and the game continues.

Step 1) Play the game Set long enough to become familiar with the different types of Sets that may be found.

Step 2) Compute the probability that three randomly selected cards form a set. How many sets are there?

Step 3) Sets may be categorized based on the number of properties which the 3 cards have in common (0, 1, 2, 3, or 4) versus the number of properties in which they differ. One of these categories can't occur in a standard deck, but the others do. Compute the probability of each of the four types of Sets which can occur.

Step 4) Play the game Set again and note how many sets of each category are found. Does the distribution agree with your probabilities computed in step 3? If not, why do think this might be?

Step 5) Compute the probability that four randomly selected cards contain a set, and then do the same thing for five and then six randomly selected cards. Note that while four cards can only contain one set (why?) a group of 5 cards may contain more than one set. Be careful not to overcount when you compute the probabilities.

Areas for further investigation

Can you compute the probability that 12 randomly selected cards contain a set? How many cards are needed to make this probability 1?

Combinatorics: The Fine Art of Counting

Winning Streak Group Activity

Overview

Winning streaks grab a lot of attention. The length of the current winning streak (if any) is often listed along-side the win/loss columns in the sports pages. In this activity you will compute the probability of winning streaks for a particular team during a hypothetical 10 game season. To simplify the analysis, we will assume that ties don't occur (simply count them as non-wins), and that the probability that the team wins a given game is always the same value, ***p***, independent of the opponent or the results of any previous games. Our sample space will be the Bernoulli sample space with 1's indicating wins and 0's indicating losses.

Step 1) Let the event F_k consist of sequences which have 1's in the first *k* positions (corresponding to winning the first k games in a row). Compute $P(F_k)$ for *k* = 1 to 10 in terms of *p*, and compute actual values for the two cases *p* = ½ (an average team) and *p* = 3/4 (a good team).

Step 2) Using dice and/or coins, simulate ten 10 game seasons for an average team (*p* = ½) and a good team (*p* = ¾). Write down a sequence of ten1's and 0's for each simulated season and note the length of the longest winning streak in each season. Based on this data and/or your intuition, determine two values of k such that a team which wins with probability ½ or ¾ has about a 50% chance of experiencing a winning streak of length k.

Step 3) Review the definitions on the following page and make sure you understand them (ask questions if you don't!). Using grid-paper, use the defined recurrence relations to compute a table for the coefficients *d(w,l)* for all *w* and *l* such that $w + l \leq 10$. Do this for both values of *k* you chose. Note that many of the entries will simply be binomial coefficients (for all $w < k$). Have two separate groups build the tables so you can check your work.

Step 4) Use your tables to compute the coefficients *c(w,l) = (w+l w) – d(w,l)* for all $w + l = 10$. Do this for each of the two values of k you chose. Once you have computed the coefficients, write down your polynomials $W_k(p,q)$. Check your work carefully.

Step 5) Use your polynomials to compute the probabilities for p = ½ and p = ¾ (use a calculator!) Do the results match your expectations?

Areas for further investigation

You will have noticed during your computations that computing the probabilities in this way is simple but time consuming. Could you program a computer to do it? Can you find a more efficient method that would reduce the work involved?

Let W_k be the event consisting of sequences which contain k consecutive W's. Note that such a sequence may contain more than k consecutive W's. We want to find $P(W_k)$ as a function of p. To do this we will construct a polynomial $W_k(p,q)$ with p and q as variables which we can write in the form $\boldsymbol{c(w,l)p^w q^l}$.

Note that all the terms of this polynomial will have the same degree, and since in our example $w + l = 10$, and there will be 11 terms in this polynomial. The coefficients $c(w,l)$ are constants equal to the number of binary strings with w 1's and l 0's which contain k consecutive 1's. Note that since $q = 1\text{-}p$, if we know p, we can compute the value of $W_k(p,q)$.

Rather than compute the coefficients $c(w,l)$ directly, we will compute them by counting their complements $d(w,l)$, i.e. the number of binary strings with w 1's and l 0's which *do not* contain k consecutive 1's. Note that since there are exactly (w+l w) strings binary strings with w 1's and l 0's, we have the equation:

$$c(w,l) = (w+l\ w) – d(w,l)$$

which we will use to determine $c(w,l)$ once we have found $d(w,l)$. The reason for using the $d(w,l)$ is that the satisfy a simple recurrence relationship:

$$d(w,l) = d(w,l-1) + d(w-1,l) – d(w-k,l-1) \qquad w,l,k \geq 1$$

To get started we will need the following base cases:

$$d(w,0) = 1 \quad \text{for } 0 \leq w < k$$
$$d(w,0) = 0 \quad \text{for } w \geq k$$
$$d(0,l) = 1 \quad \text{for } l \geq 0$$

And just to handle the boundaries correctly, we define:

d(w,l) = 0 whenever w or l is less than 0.

The base cases are easy to check (you should do so to confirm your understanding of the definitions). The main recurrence can be derived as follows:

Call a binary string good if it does not contain k consecutive 1's. $d(w,l)$ is the number of good strings with w 1's and l 0's. Consider one such string. If the last bit is a 0, then the preceding string is a good string with w 1's and l-1 0's, and conversely any good string with w 1's and l-1 0's can have a 0 tacked on the end and still be a good string. This gives us $d(w,l\text{-}1)$ as the first term on the RHS of the recurrence above. If the last bit is a 1, then the preceding string is a good string with w-1 1's and l 0's (the second term), BUT it is not the case than every good string with w-1 1's and l 0's can have a 1 tacked on the end and still be a good string.

There is precisely one type of exception: the case where the preceding good string ends with k-1 1's. In this case there must be a 0 immediately before the k-1 1's (since the string is good), and the rest of the string prior to that point is a good string which has l-1 0's and $w\text{-}1\text{-}(k\text{-}1) = w\text{-}k$ 1's. Moreover any good string with w-k 1's and l-1 0's could have a 0 and k-1 1's tacked on to result in the kind of exception we are considering. Thus the number of exceptions is exactly $d(w\text{-}k,l\text{-}1)$ which accounts for the negative third term in the recurrence.

Finally, it is worth noting that for $w < k$, the third term will be zero and the first two terms of the recurrence are equivalent to Pascal's identity, as are the base cases. Therefore:

$$d(w,l) = (w+l\ w) \qquad \text{whenever } w < k$$