Playing DnD with a Yahtzee Set

Stephen Roller
roller@cs.utexas.edu

March 7, 2011

DRAFT

Abstract

We describe a procedure for emulating any discreet uniform probability distribution using another. For example, using two six-sided dice in order to emulate the twenty-sided die found in Dungeons and Dragons. We also perform a cost analysis of this procedure, as well as note two special cases when constant runtime is achieved.

1 Introduction

Dungeons and Dragons, an infamous tabletop role-playing game, uses a variety of unusual dice in order to determine outcomes of the game. The most iconic of these dice is the twenty-sided die. In DnD, a roll of the twenty-sided die may vastly determine the outcome of the game; for example, when a character performs an attack, the 20-sided die is used to determine how much damage is incurred to the victim.

Wizards of the Coast calls this style of gameplay the "d20 system" [1]. Since its first introduction, this notation has caught on and gained some notoriety, especially in the larger geek community.

Unfortunately, this style of gameplay requires the use of a twenty-sided dice. In many areas, such dice may be difficult or inconvenient to obtain. As such, we aim to answer the question, "How may one emulate a twenty-sided die using only a set of six-sided dice?"

Definition 1 *For any nonzero, positive integer n, d_n is the uniform, discreet probability distribution whose possible outcomes range from 0 to $n-1$, inclusive.*

Informally, we may say d_n is the distribution representing a fair, n-sided die with sides labeled $0, 1, \ldots, n-1$. For example, d_2 may represent to a coin flip, d_6 may represent a commonplace six-sided die, and d_{20} represents our coveted 20-sided die.

Using this notation, we can formalize our goal as a procedure which uses a distribution d_n in order to simulate a distribution d_m.

2 Combinatoric Intuition

2.1 Using decision trees

The general idea behind emulating d_m distribution will require that we use the d_n distribution to produce $k \geq m$ unique possibilities to choose from. Of these k possibilities, we will assign exactly $k - m$ of these options a probability of 0, and the remaining m possibilities will all have a probability of exactly $1/m$.

In order to use the d_n distribution to produce k possibilities, we will sample the d_n distribution multiple times, using a decision tree to decide which of the k possibilities to choose from. The simplest way to do this is to keep track of ordered rolls. If we roll x times, then we will have n^x different possibilities for ordered outcomes. Let us choose x such that,

$$k = n^x \geq m.$$

Clearly, the minimal integer x for which this inequality holds is

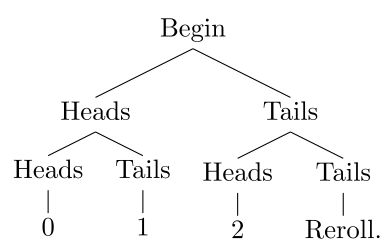
$$x = \lceil \log_n m \rceil.$$

Of course, our decision tree currently has k possibilities, each with a probability of $1/k$. We will solve this by rereolling whenever we choose one of the rightmost $k - m$ branches. In this way, these rightmost branches will have a probability of exactly 0 (since they will never be chosen), and the leftover probability will be equally distributed among the other m branches.

It is important that whenever we reach a branch with $P = 0$, we must restart at the very top of the decision tree. Otherwise, we will have a situation very similar to the Monty Hall Problem, resulting in a nonuniform distribution.

Example 1 *Create a decision tree for emulating a three-sided die using coin flips.*

In this example, we have $m = 3$ and $n = 2$. Using the equations above, $x = 2$ and $k = 4$. Thus, one possible decision tree for our procedure is



That is, to emulate a three-sided die, we flip a coin twice. If both our coins are Heads, we will return 0. If we get a Heads, then a Tails, we will return 1. If we get a Tails, and then a Heads, we will return 2. Finally, if both of our flips are Tails, we must retry by flipping both coins again until we get something other than Tails–Tails.

2.2 Using base-n numbers

Next, we observe that it is unnecessary to actually completely construct a decision tree. Rather, it is possible to interpret each progressive roll as a digit in a base-n number.

Recall that a base-n number, A, of length x, is equivalent to the following base-10 formula:

$$A = a_0 + na_1 + \ldots + n^{x-2}a_{x-2} + n^{x-1}a_{x-1}, \tag{1}$$

where a_i is the ith digit of the A counting from the right. In this way, possible values of A ranges from 0 to $n^x - 1$, inclusive. After some algebraic manipulation, may rewrite (1) as

$$A = a_0 + n(a_1 + n(a_2 + \ldots)). \tag{2}$$

Equation (2) is key to the notion of using base-n numbers in order to implicitly construct our decision tree. We will treat each successive d_n roll as a digit in a base-n number of length x, thus giving us $k = n^x$ different possible results. To throw out the $k - m$ rightmost branches, we will simply reroll whenever $A \geq m$, thus creating a uniform distribution over the numbers 0 to $m - 1$.

3 Formal Procedure

We are now ready to formalize our procedure. Note that in this pseudocode, we use the notation $d_n()$ to represent taking a sample from the d_n distribution; in other words, $d_n()$ represents rolling an n-sided die.

Algorithm 1 Emulating a m-sided die using many n-sided rolls.

```
x ← ⌈log_n m⌉
loop
  A ← 0
  for i = 0 to x − 1 do
    A ← A ∗ n + d_n()
  end for
  if A < m then
    return A
  end if
end loop
```

In theory, this algorithm may possibly run forever, though with an infinitely small probability. In the next section, we aim to analyze how many times we will need to roll on average.

4 Cost Analysis

In each iteration of the loop, we have m possible ways of stopping (success) and $k - m$ possible ways of needing to reiterate (failure). That is, we have a geometric distribution with a $P = m/k$ probability of getting a success. The mean of a geometric distribution is well-known [2] to be $1/P$, and so the average number of trials (loop iterations) before a success (termination) is

$$\frac{n^x}{m},$$

where $x = \lceil \log_n m \rceil$. Therefore, the average number of samples taken from d_n (rolls of the n-sided die), will be

$$x\frac{n^x}{m}.$$

Example 2 *Emulating a six-sided die using coin flips will require 4 flips on average.*

Example 3 *Emulating a 20-sided die using only a single die will require 3.6 rolls on average. Using a set of two colored dice, we will need 1.8 rolls on average.*

Example 4 *Emulating a 100-sided die with two 10-sided die will never require a reroll.*

5 Perfect emulation

A modification to Algorithm 1 can allow for perfect emulation (that is, constant runtime) whenever $0 \equiv n^x \bmod m$. In this case, instead of rerolling, we can simply assign multiple branches of the decision tree the same value; because n^x is evenly divisible by m, we will have an equal number of branches for each possible value 0 to $m - 1$.

Algorithm 2 Simulating d_m using many d_n samples when $0 \equiv n^x \bmod m$.

$x \leftarrow \lceil \log_n m \rceil$
$A \leftarrow 0$
for $i = 0$ to $x - 1$ **do**
  $A \leftarrow A * n + d_n()$
end for
return $A \bmod m$

Example 5 *Simulating a 20-sided die with two 10-sided die. We roll the dice, concatenating the values in order to produce a number between 00 and 99. We return the* mod 20 *of this number.*

There exist alternative encodings, as well. For example, consider $m = 8$ and $n = 4$. Here, we may use the first dice roll to determine whether our answer should be "high" or "low", and the second roll to select a value from the sets $\{0, 1, 2, 3\}$ or $\{4, 5, 6, 7\}$. We leave the generalization of this case, and others, as exercises for the reader.

6 Conclusion

We have shown a procedure for simulating any discreet, uniform distribution d_m using a fixed number of samples from another discreet, uniform d_n distribution. One real world example is emulating the popular 20-sided die found in Dungeons and Dragons using only a six-sided die.

We also have shown a cost analysis for such a procedure, as well as pointed out two special cases where perfect emulation is achieved.

7 Acknowledgements

Special thanks to Michael Lee, Karl Pichota and Jeff Picton for the many helpful discussions regarding this problem.

References

[1] Wikipedia. d20 system — Wikipedia, the free encyclopedia, 2011. [Online; accessed 03-March-2011].

[2] Wikipedia. Geometric distribution — Wikipedia, the free encyclopedia, 2011. [Online; accessed 06-March-2011].