Expected Number of Distinct Subsequences in Randomly Generated Binary Strings

Yonah Biers-Ariel, Anant Godbole, Elizabeth Kelley

May 14, 2017

When considering binary strings, it's natural to wonder how many distinct subsequences might exist in a given string. For a fixed string, there's an existent algorithm which provides a straightforward way to compute the number of distinct subsequences. The natural extension to this question, then, is to consider random strings. To frame our discussion, we first need to establish some notation. Using the definitions established in [**?**], a binary string of length n is some $A = a_1a_2 \dots a_n \in \{0,1\}^n$ and another string B of length $m \leq n$ is a subsequence of A if there exist indices $i_1 < i_2 < ... < i_m$ such that

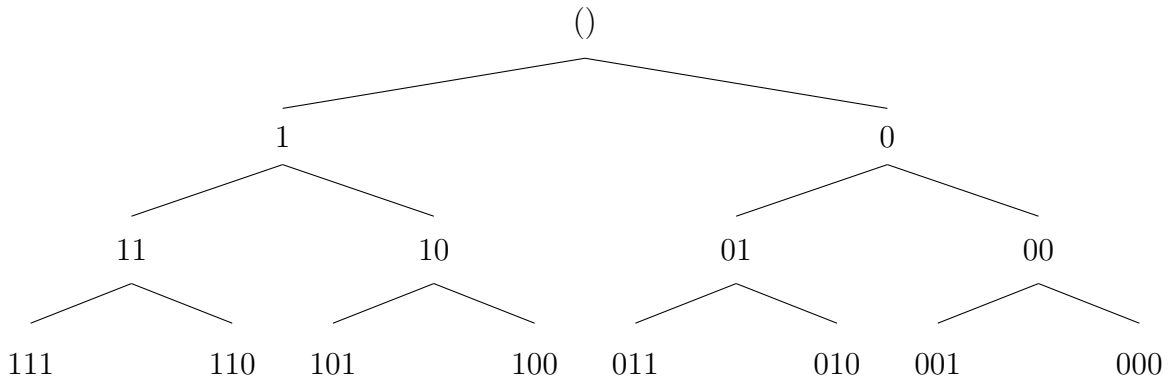
$$B = a_{i_1}a_{i_2}...a_{i_m}$$

We use the notation $B \preceq A$ when B is a subsequence of A.

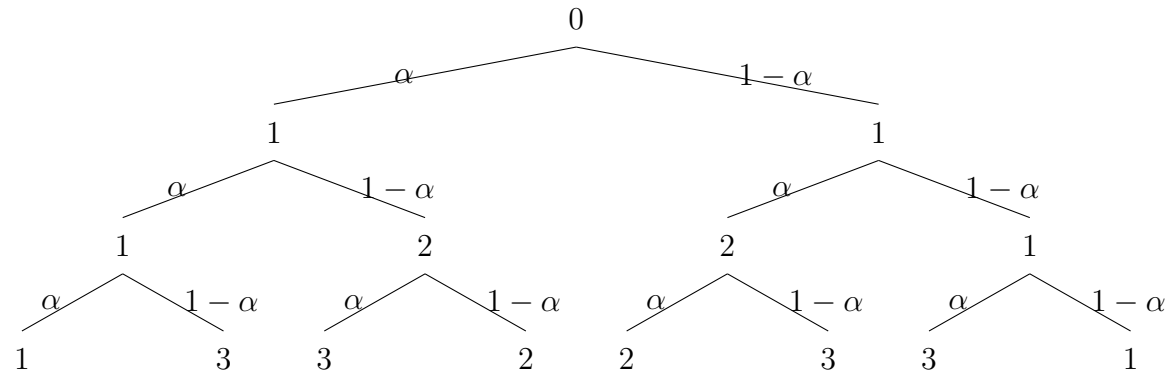
Let T_n be a binary string of length n, t_i be the i^{th} letter of T_n, T_i be the string formed by truncating T_n after the i^{th} letter, and $\phi(T_n)$ be the number of distinct subsequences in T_n. If we let S_n be a random binary string of length n, it was shown in [**?**] that when $\Pr[s_i = 1] = .5$ (that is, when each letter in S_n is equally likely to be a 0 or a 1), then $E[\phi(S_n)] \sim k(\frac{3}{2})^n$ for a constant k. Collins [**?**] later improved this result by determining that $E[\phi(S_n)] = 2(\frac{3}{2})^n - 1$ under these conditions.

We generalize Collin's result, finding a formula for the expected value of $\phi(S_n)$ when $\Pr[s_i = 1] = \alpha \in (0,1)$. Because the cases when $\Pr[s_i = 1]$ is 0 or 1 are trivial, this gives us $E[\phi(S_n)]$ when $\Pr[s_i = 1] = \alpha \in [0,1]$. We use a very different method than Collins [**?**]. We define a new property of a string - the number of new distinct subsequences - and then use these numbers as the entries in a binary tree. Our formula is then given as a weighted sum of the entries in this tree.

Let B be a binary tree whose entries are binary strings and let $B_{n,m}$ be the m^{th} entry in the n^{th} row of B. The root of B is the empty string, each left child is its parent with a 1 appended, and each right child is its parent with a 0 appended. If we call the first row "row 0", then row n of this tree contains all length n binary strings. Rows 0-3 of this tree are shown below:



Clearly, there's a one-to-one correspondence between binary strings and entries in this tree. In order to make use of this tree, we form the binary tree B' with $B'_{n,m}$ denoting the m^{th} entry in the n^{th} row of B'. Letting $\nu(T_n)$ denote the number of new distinct subsequences in a string (i.e., the number of distinct subsequences that do not exist in the truncation T_{n-1}), we can define each $B'_{n,m}$ as $\nu(B_{n,m})$, the number of new distinct subsequences introduced in the entry $B_{n,m}$. Finally, for each child $B'_{n,m}$ we assign the edge between it and its parent $B'_{n-1,\lceil \frac{m}{2} \rceil}$ a weight equal to $\Pr[S_n = B_{n,m} | S_{n-1} = B_{n-1,\lceil \frac{m}{2} \rceil}]$. Thus we give each edge going to a left child the weight α and each edge going to a right child the weight $1-\alpha$. Rows 0-3 of B' are shown below:



Use of such trees provides proof for the following theorem:

Theorem 0.1. *Suppose* $Pr[s_i = 1] = \alpha \in [0,1]$ *for all* $1 \leq i \leq n$. *Then we have*

$$\phi(S_n) = \begin{cases} n & \text{if } \alpha = 0, 1 \\ \frac{\left(1-2\sqrt{\alpha(1-\alpha)}\right)\left(1-\left(1-\sqrt{\alpha(1-\alpha)}\right)^n\right)+\left(1+2\sqrt{\alpha(1-\alpha)}\right)\left(\left(1+\sqrt{\alpha(1-\alpha)}\right)^n-1\right)}{2\sqrt{\alpha(1-\alpha)}} & \text{if } \alpha \neq 0, 1 \end{cases}$$

This has the natural asymptotic corrollary,

Corollary 0.2. *Suppose* $Pr[s_i = 1] = \alpha \in (0,1)$ *for all* $1 < i < n$. *Then there exists a constant* k *such that*

$$\phi(S_n) \sim k\left(1 + \sqrt{\alpha(1-\alpha)}\right)^n$$

Having exhausted the possible cases for a binary string, we then proceed to look at strings on the extended alphabet $\{1, 2, ..., d\} = [d]$ where each letter is j with probability α_j for all $j \in [d]$. For such strings, we have the following way to compute the expected value of $\phi(S_n)$:

Theorem 0.3. *Let S_n be a random length-n string on the alphabet $[d]$ where $Pr[s_i = j] = \alpha_j$ for all i, j. Then,*

$$E[\phi(S_n)] = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix} \left(\sum_{i=0}^{n-1} \begin{bmatrix} 1 & \alpha_1 & \alpha_1 & \dots & \alpha_1 \\ \alpha_2 & 1 & \alpha_2 & \dots & \alpha_2 \\ \alpha_3 & \alpha_3 & 1 & \dots & a_3 \\ & & & \ddots & \\ \alpha_d & \alpha_d & \alpha_d & \dots & 1 \end{bmatrix}^i \right) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_d \end{bmatrix}.$$

Returning to binary strings, we then consider strings where the probability of seeing a particular letter will depend on the letter before (we assume that strings are generated from left to right). In the random string S_n, $\Pr[s_i = 1|s_{i-1} = 1] = \alpha$ and $\Pr[s_i = 1|s_{i-1} = 0] = \beta]$. Of course, we will need some other rule for $\Pr[s_1 = 1]$; one logical choice is to take $\Pr[s_1 = 1] = \gamma$ where γ is the steady-state probability of a 1 occurring, which in this case gives $\gamma = \frac{\beta}{1+\beta-\alpha}$. Using the same techniques as for the extended alphabet strings, we obtain the recurrences

$$\begin{aligned} a_{n+1} &= \alpha(a_n + c_n); \\ b_{n+1} &= (1-\alpha)(a_n + c_n) + \alpha b_n + \frac{\beta(1-\alpha)}{1-\beta} d_n; \\ c_{n+1} &= \beta(b_n + d_n) + (1-\beta)c_n + \frac{(1-\alpha)\beta}{\alpha} a_n; \\ d_{n+1} &= (1-\beta)(b_n + d_n). \end{aligned}$$

which give rise to the matrix equations

$$\begin{bmatrix} a_n \\ b_n \\ c_n \\ d_n \end{bmatrix} = \begin{bmatrix} \alpha & 0 & \alpha & 0 \\ 1-\alpha & \alpha & 1-\alpha & \frac{\beta(1-\alpha)}{1-\beta} \\ \frac{(1-\alpha)\beta}{\alpha} & \beta & 1-\beta & \beta \\ 0 & 1-\beta & 0 & 1-\beta \end{bmatrix}^{n-1} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix},$$

and

$$E[\phi(S_n)] = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix} \left(\sum_{i=0}^{n-1} \begin{bmatrix} \alpha & 0 & \alpha & 0 \\ 1-\alpha & \alpha & 1-\alpha & \frac{\beta(1-\alpha)}{1-\beta} \\ \frac{(1-\alpha)\beta}{\alpha} & \beta & 1-\beta & \beta \\ 0 & 1-\beta & 0 & 1-\beta \end{bmatrix}^{i-1} \right) \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix}.$$

We have also obtained similar recurrences in the case of letters generated by a two-state Markov chain. It remains to be seen if extraction of eigenvalues can aid in the identification of the growth rate in any of these cases, but in the case of independent non-uniform letter generation, we have shown using subadditivity arguments that the limiting expected value is asymptotic to C^n for some $C > 1$.

References

[1] Arratia, Richard. "On the Stanley-Wilf Conjecture for the Number of Permutations Avoiding a Given Pattern." *Electronic Journal of Combinatorics* 6 (1999).

[2] Athreya, Jayadev and Lucas Fidkowski. "Number Theory, Balls in Boxes, and the Asymptotic Uniqueness of Maximal Discrete Order Statistics." *Integers: Electronic Journal of Combinatorial Number Theory*, Paper A3 0 (2000).

[3] Biers-Ariel, Yonah, Anant Godbole, and Yiguang Zhang. "Some Results on Superpatterns for Preferential Arrangements," *Advances in Applied Mathematics* 81 (2016): 202–211.

[4] Burstein, Alexander, Peter Hästö, and Toufik Mansour. "Packing Patterns into Words." *Electronic Journal of Combinatorics* 9.2 (2002-2003).

[5] Collins, Michael. "The Number of Distinct Subsequences of a Random Binary String." Unpublished. The *ArXiv.org* reference is `https://arxiv.org/pdf/1310.7288.pdf`.

[6] Elzinga, Cees, Sven Rahmann, and Hui Wang. "Algorithms for Subsequence Combinatorics." *Theoretical Computer Science* 409.3 (2008): 394–404.

[7] Flaxman, Abraham, Aram W. Harrow, and Gregory B. Sorkin. "Strings with Maximally Many Distinct Subsequences and Substrings." *Electronic Journal of Combinatorics* 11.1 (2004).