IPN Progress Report 42-186 • August 15, 2011

Twenty Questions Games Always End With Yes

William Wu* and John T. Gill III†

Huffman coding is often presented as the optimal solution to Twenty Questions. However, a caveat is that Twenty Questions games always end with a reply of "Yes," whereas Huffman codewords need not obey this constraint. We bring resolution to this issue by showing that the average number of questions still lies between $H(X)$ and $H(X) + 1$.

I. Introduction

Twenty Questions is a classic parlour game involving an answerer and a questioner. The questioner must guess what object the answerer is thinking of, but is only allowed to ask questions whose answers are either "Yes" or "No". Popular initial questions include: "Is it an animal? Is it a vegetable? Is it a mineral?" The name of the game arises from the fact that if one bit of information could be acquired from each question, then twenty questions can distinguish between 2^{20} different objects, which should be more than sufficient.

Twenty Questions games are also related to some communication theory problems where the communicating nodes have very disparate resources. These problems are referred to as asymmetric communications, and Twenty Questions strategies are appropriate for sending information efficiently in these cases. A very low power spacecraft in deep space with a powerful uplink signal is such an example.

Courses in information theory often cast Huffman coding as the optimal approach to Twenty Questions. Given the set of possible objects and their probabilities, the questioner associates a Huffman codeword with each object, and then inquires about each bit of the codeword that the answerer is thinking of. The average number of questions is the Huffman tree's average depth, which is greater than or equal to $H(X)$, and less than $H(X) + 1$, where X is the non-degenerate random variable indicating which of n objects

---

*Communications Architectures and Research Section

†Stanford University Electrical Engineering Department

The research described in this article was performed at Stanford University. The writing and publication was supported by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

the answerer is thinking of, and $H(X)$ is its Shannon entropy.

However, upon further thought, there is a disparity between Huffman coding and how Twenty Questions games are played. Namely, real-world Twenty Questions games always terminate with the questioner pinpointing a specific object (e.g., "Is it a *tank*?" [1]), to which the answerer replies, "Yes!" In terms of source coding, this is equivalent to enforcing what we call the *terminating yes constraint*: all codewords must terminate with "1". Yet Huffman codes do not satisfy this constraint! In short, Huffman trees determine X, but do not specify X.

In this article, we first provide a simple example showing that simply appending branches to a Huffman tree may not produce the optimal Twenty Questions tree. We then prove that even under the terminating yes constraint, the average number of questions lies strictly between $H(X)$ and $H(X) + 1$.

It should be pointed out that all the conclusions in this article follow from the analyses in [2] and [3], which approached this problem from the perspective of finding optimum "1"-ended binary prefix codes, and also provided a lower bound on the average number of questions that is tight, but more complicated than $H(X)$. Here we provide a short exposition that emphasizes the Twenty Questions perspective, and aims to be as accessible as possible.

II. Bar Bet: Guessing One of Four Objects

Since Huffman coding solves Twenty Questions without a terminating yes, a natural idea is to first build the Huffman tree, and then append branches to it so the terminating yes constraint is satisfied. Call the result an *augmented Huffman tree*. In the following example, we show that augmented Huffman trees may not be optimal Twenty Questions trees.

Suppose there are only four objects the answerer could be thinking of. Denote them by x_1, x_2, x_3, x_4, with corresponding probabilities $p_1 \geq p_2 \geq p_3 \geq p_4$. Figure (1) shows the only two four-leaf questioning trees possible up to graph isomorphism, where the dashed edges have been added to accommodate the terminating yes constraint. Although there are many possible assignments of objects to leaves, the assignments shown in Figure (1) are the only reasonable candidates which place higher probability objects at shallower depths.

One naturally imagines that the choice of a questioning tree should depend on the probability distribution. For instance, if the probabilities are close to uniform, we would guess that the balanced tree is better. However, if we let Q_1 and Q_2 denote the average number of questions used by the unary and balanced trees, respectively, then

$$\begin{aligned} Q_1 &= p_1 + 2p_2 + 3p_3 + 4p_4 = 1 + p_2 + 2p_3 + 3p_4, \\ Q_2 &= 2(p_1 + p_2) + 3p_3 + 3p_4 = 2 + p_3 + p_4, \end{aligned}$$

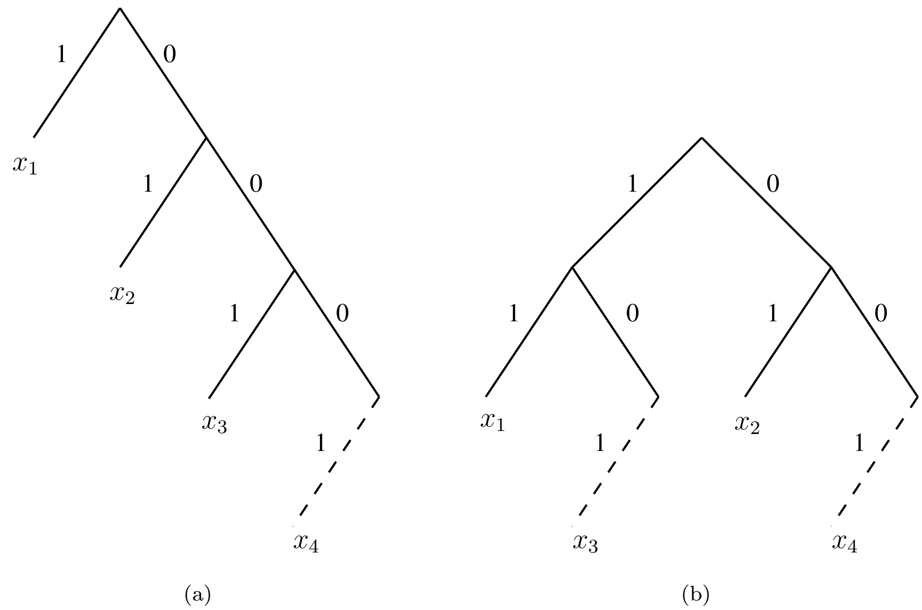

Figure 1. Four leaf trees. (a) Unary code. (b) Balanced code.

and the difference is

$$\begin{aligned} Q_2 - Q_1 &= 2 + p_3 + p_4 - 1 - p_2 - 2p_3 - 3p_4 \\ &= 1 - (p_2 + p_3 + 2p_4) \\ &= p_1 - p_4 \geq 0, \end{aligned}$$

with equality if and only if the distribution is uniform. Apparently the unary tree dominates the balanced tree, regardless of the probabilities! We think this makes for a good bar bet.

This example demonstrates that augmenting a Huffman tree does not necessarily produce the optimal Twenty Questions tree. For example, if the probabilities were $(3/10, 3/10, 2/10, 2/10)$, then the resulting augmented Huffman tree would yield the balanced tree, although the unary tree is better. In fact, using linear programming it can be easily shown that among all distributions for which the Huffman algorithm produces a balanced tree, the maximum difference in the average number of questions required by the balanced and unary trees approaches 1/3, and is achieved with the distribution $(\frac{1}{3} - \epsilon, \frac{1}{3} - \epsilon, \frac{1}{3} - \epsilon, 3\epsilon)$.

In general, the optimal Twenty Questions tree can be constructed using dynamic programming; see [4] for a simple $O(n^3)$ implementation. However, given examples such as the one above, it may be unclear where the average depth of the optimal Twenty Questions tree falls relative to the entropy of the source. The following section addresses this.

III. Entropy Bounds On The Average Number of Questions

Let L_H be the average depth of the Huffman tree, and let L_{yes} be the average depth of the optimal Twenty Questions tree. In this section, we prove

$$H(X) < L_{yes} < H(X) + 1.$$

Note that these are the same bounds satisfied by L_H, except for the strict inequality in the lower bound. We first require two Lemmas.

Lemma III.1 (Half-Bit Lemma)**.** *A binary tree that does not satisfy the terminating yes constraint can be modified to satisfy it while adding no more than 1/2 to the average depth.*

Proof. Let T be a tree that does not satisfy the terminating yes constraint. By appending a branch to all leaves whose codewords end with 0, we can construct an augmented tree T' that does satisfy it. (This forces all leaves to sway in the same direction.) To minimize the increase in average depth, interchange siblings in T as necessary so that the lower probability sibling is always the one that receives the appended branch. Consequently, if the average length of T is L, the average length of T' will be no more than $L + 1/2$. □

Lemma III.2 (Gallager's Redundancy Bound)**.** *For all finite distributions,* $L_H - H(X) \leq p_1 + \sigma$*, where* p_1 *is the largest probability, and* $\sigma := 1 - \log_2 e + \log_2(\log_2 e) \approx 0.086$.

Proof. See Gallager [5]. □

Theorem III.3. $H(X) < L_{yes} < H(X) + 1$.

Proof. We first establish the lower bound. By pruning the appended branches of the optimal Twenty Questions tree, we have a new tree of reduced average depth in which every internal node has two children. Amongst all such trees, the Huffman tree has lowest average depth, so $L_H < L_{yes}$. Lastly, $H(X) \leq L_H$ (see Cover and Thomas [6]).[1]

For the upper bound, we consider two cases. First, suppose $p_1 < 0.4$. From Lemma III.2,

$$L_H - H(X) \leq p_1 + \sigma < 1/2\,.$$

Adding 1/2 to both sides and rearranging,

$$L_H + 1/2 < H(X) + 1\,.$$

From Lemma III.1, $L_{yes} \leq L_H + 1/2$. Thus,

$$L_{yes} < H(X) + 1\,.$$

When $p_1 \geq 0.4$, we prove the upper bound by induction on the number of objects. For the base case, consider the case of two objects $\{x_1, x_2\}$, where without loss of generality x_1 is

[1] A tight lower bound is provided in [3].

chosen with probability $p_1 \in [0.5, 1)$, and x_2 is chosen with probability $1 - p_1$. The corresponding tree is shown in Figure (2). The expected number of questions is $L_{yes} = p_1 + 2(1 - p_1) = (1 - p_1) + 1$, which is upper bounded by $H(1 - p_1) + 1$ since $q < H(q)$ for $q \in (0, 0.5]$. This verifies the base case.

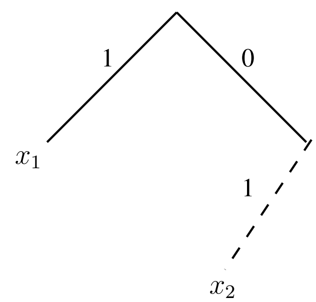

Figure 2. Tree corresponding to the induction base case when $p_1 \geq 0.4$.

For the induction step, let X be a random variable taking n possible values, and let $\hat{T}$ be the tree with minimum average depth under both the terminating yes constraint *and* the additional constraint that the most probable object has a codeword of length one. This tree $\hat{T}$ is illustrated in Figure (3). While this additional constraint may result in a suboptimal tree, we will show that $\hat{T}$ satisfies the desired upper bound regardless, and thus the optimal Twenty Questions tree does also.

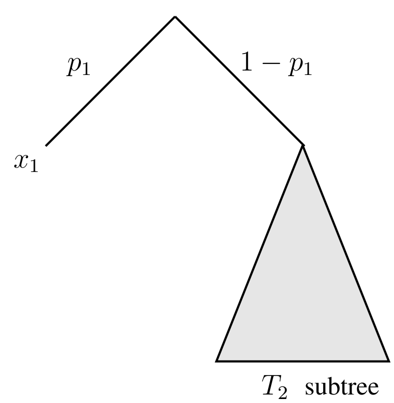

Figure 3. Tree $\hat{T}$ used in the induction argument when $p_1 \geq 0.4$.

Let $\hat{L}$ denote the average depth of $\hat{T}$, and let T_2 denote the right subtree containing $n - 1$ leaves. Then

$$\hat{L} = 1 + (1 - p_1)L(T_2)$$

where $L(T_2)$ denotes the average depth of T_2. Also, by the grouping law for entropy,

$$H(X) = H(p_1) + (1 - p_1)H(X_2)$$

where X_2 is a random variable with probability mass function given by the normalized $n-1$ remaining probabilities $\left(\frac{p_2}{1-p_1}, \frac{p_3}{1-p_1}, \ldots, \frac{p_n}{1-p_1}\right)$. Subtracting these equations,

$$\hat{L} - H(X) = 1 - H(p_1) + (1-p_1)(L(T_2) - H(X_2))$$

By construction of $\hat{T}$, it follows that T_2 is an optimal Twenty Questions tree for X_2. By the induction hypothesis, $L(T_2) - H(X_2) < 1$, and thus

$$\hat{L} - H(X) \leq 2 - (H(p_1) + p_1).$$

Since p_1 could be any value in $[0.4, 1]$, we want the largest upper bound, to cover all our bases. Setting $p_1 = 1$,

$$\hat{L} - H(X) \leq 1.$$

Thus,

$$L_{yes} \leq \hat{L} \leq H(X) + 1.$$

□

Lastly, by combining the bounds

$$H(X) \leq L_H < H(X) + 1 \tag{1}$$

$$H(X) < L_{yes} < H(X) + 1 \tag{2}$$

$$L_H < L_{yes} \tag{3}$$

we conclude that

$$H(X) \leq L_H < L_{yes} < H(X) + 1. \tag{4}$$

Since the classical bounds in Equation (1) are tight, it follows that the bounds in Theorem III.3 are also tight.

IV. Conclusion

Although Twenty Questions games always end with "Yes", thankfully the average number of questions they require is still within one of the entropy – a nice answer to a simple problem. As Forrest Gump would say, "One less thing to worry about."

Acknowledgments

The authors thank Thomas Cover for enlightening discussions, and Baris Erkmen for meticulously finding many typos in our write-up.

References

[1] *Bill and Ted's Bogus Journey*, Directed by Peter Hewitt, Los Angeles, CA: Interscope Communications, 1991.

[2] T. Berger and R.W. Yeung, "Optimum 1-ended Binary Prefix Codes," *IEEE Trans. Inf. Theory*, vol 36, no. 6, pp. 1435–1441, 1990.

[3] R. Capocelli, A. De Santis, and G. Persiano, "Binary Prefix Codes Ending In a 1," *IEEE Trans. Inf. Theory*, vol 36, no. 6, pp. 1435–1441, 1990.

[4] S.-L. Chan and M. J. Golin, "A Dynamic Programming Approach For Constructing Optimal 1-ended Binary Prefix-free Codes," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1637–1644, 2000.

[5] R. Gallager. "Variations on a Theme by Huffman," *IEEE Trans. Inf. Theory*, vol. 24, no. 6, pp. 668–674, 1978.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: Wiley-Interscience, second edition, 2006.