Partitioning information in 3B1B Wordle video

Eric Marcon

March 10, 2022

This is a comment on 3Blue1Brown's video about solving Wordle using information theory. Please watch it before reading what follows for clarity.

1 Rationale

The video is "an excuse to teach a lesson on information theory and entropy". As usual on 3B1B's channel, it is excellent but the way the total information of a set of words is split into the information of the color pattern and that of the remaining set was not obvious for me. That's why I explicit it here and show that it is a bit more complicated when word probabilities are not equal.

2 The simple case

In the simple case, words are equally probable. In the video at time 15:43, all the 12972 possible words (those allowed by the rules of the game) are considered equally probable. Their average information, i.e. their entropy, is $\log_2(12972)$, i.e. 13.66 bits. This is explained in the "Information theory basics" part of the video.

The word "SLATE" has been proposed and produced a color pattern (yellow, grey, yellow, grey, grey) compatible with 578 words. The information of this possible set is $\log_2(578)$, i.e. 9.17 bits.

When the word "SLATE" is chosen, this color pattern is obtained if the hidden word is one of the 578 words that are compatible with it. The probability to obtain it is simply 578/12972 since all words have the same probability to be the hidden one. The information brought by the color pattern is thus $\log_2(12972/578)$.

The important result at this stage is that the total entropy (9.17 bits) can be partitioned between the entropy of the possible set of words (9.17 bits) and the information brought by the knowledge of the color pattern of the tentative word (4.49 bits). The latter is not an entropy as defined for the two sets of words: it is not the average information of 12972/578 equally probable sets.

That said, the proof of the validity of the partitioning is straightforward:

$$578 * 12972/578 = 12972$$

so

$$\log_2(578) + \log_2(12972/578) = \log_2(12972).$$

3 Unequal weights

This proof does not hold when words have unequal weights. Actually, the partitioning is not exact. At 24:03, the 12.54 bits of the whole set of words (with unequal probabilities) is not the sum of the entropy of the possible set (8.02 bits) and that brought by the color pattern (4.42 bits): 0.10 bits are missing.

The partitioning of entropy has been derived by Rao and Nayak (1985). It is widely used in the measurement of biodiversity (e.g. Marcon et al., 2012).

The whole set of words must be split into two subsets when the color pattern is known: the possible words and the impossible ones. The total entropy (called γ entropy after Whittaker, 1960) is the sum of the average entropy of the two subsets (called α entropy) and their β entropy, i.e. the relative entropy that describes how different they are from the whole set.

Note p_w the probability of the word w, $w_+ = \sum_+ p_w$ the sum of the probabilities of the words of the possible subset and $w_- = \sum_- p_w$ that of the impossible subset.

The entropy of the possible subset is:

$$H_+ = \sum_+ (p_w/w_+) log_2(w_+/p_w),$$

and H_- is entropy of the impossible subset. Since the probabilities are considered in each subset, they are divided by the weight of their subset in order to sum to 1.

α entropy is the weighted average entropy of the subsets:

$$w_+ H_+ + w_- H_-.$$

Since no word is shared between subsets, β entropy is simply (Marcon et al., 2012):

$$w_+ log_2(1/w_+) + w_- log_2(1/w_-).$$

Their sum is γ entropy, i.e. that of the whole set of words:

$$\sum p_w log_2(1/p_w)$$

The relation α entropy plus β entropy equals γ entropy can be arranged considering that $w_+ = 1 - w_-$ to obtain:

$$\sum p_w log_2(1/p_w) = H_+ + log_2(1/w_+) + w_-[H_- - H_+ + log_2 w_+/w_-]. \quad (1)$$

The left side of the equality is the total (γ) entropy, 12.54 bits in the example. The first two terms on the right side are the entropy of the possible group (8.02 bits) and the information brought by the color pattern (4.42 bits). The last term contain the 0.10 bit approximation:

$$w_-[H_- - H_+ + log_2 w_+/w_-]$$

Now, the first two terms of the sum are the difference between the entropies of the impossible and the possible subsets. If the probability distributions are similar (the rarity of words is not related to the subset they belong to), then the difference between entropies is roughly the difference between the logarithms of the sizes of the subsets, i.e. the opposite of the last term of the sum. In the simple case where words are equally probable, the first terms are exactly $log_2 w_-$ and $log_2 w_+$ so the whole sum equals zero. In the general case, the small difference is multiplied by w_-, making it yet smaller.

In conclusion, the entropy partitioning proposed in the video is not exact when word weights vary but the error is small as long as the distribution of word probabilities in the possible group are similar to that of the words that do not match the color pattern.

4 Simulation

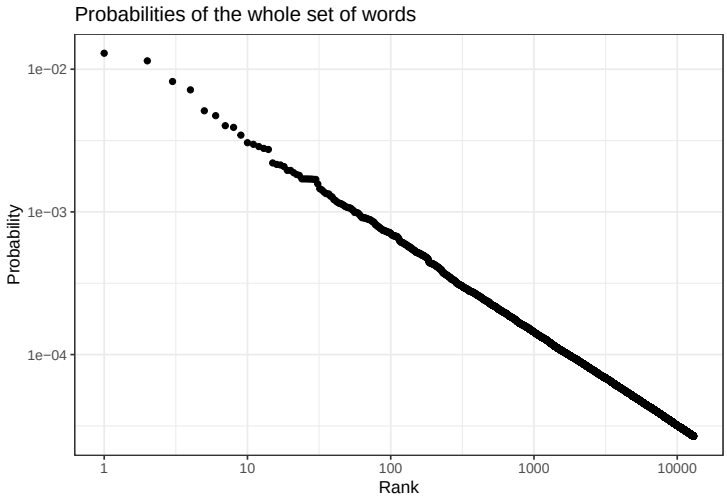
4.1 Data

The function `ent()` returns the entropy of a distribution of probabilities.

```
# Shannon's entropy in bits
ent <- function(x) sum(x * log2(1/x))
```

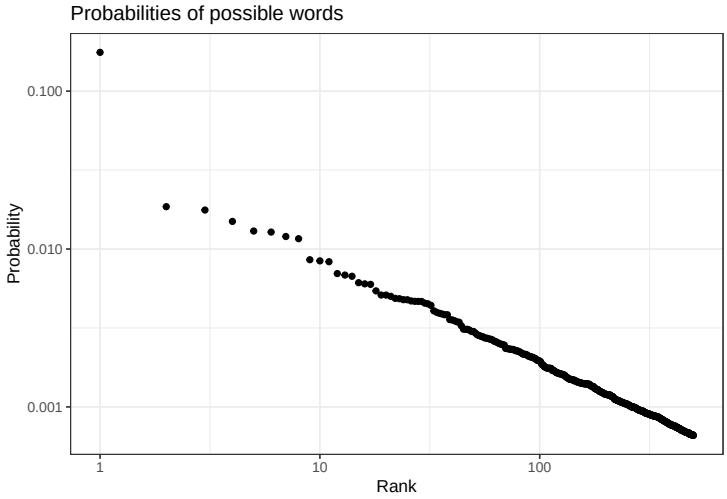
We draw a set of 13000 words in a pareto distribution. A rank-abundance curve shows the probabilities of words, by decreasing probability.

```
n_set_all <- 13000
# Random distribution
library("sads")
library("entropart")
p <- as.ProbaVector(rpareto(n_set_all, shape = 1.5))
autoplot(p, main = "Probabilities of the whole set of words") +
    scale_x_log10()
```

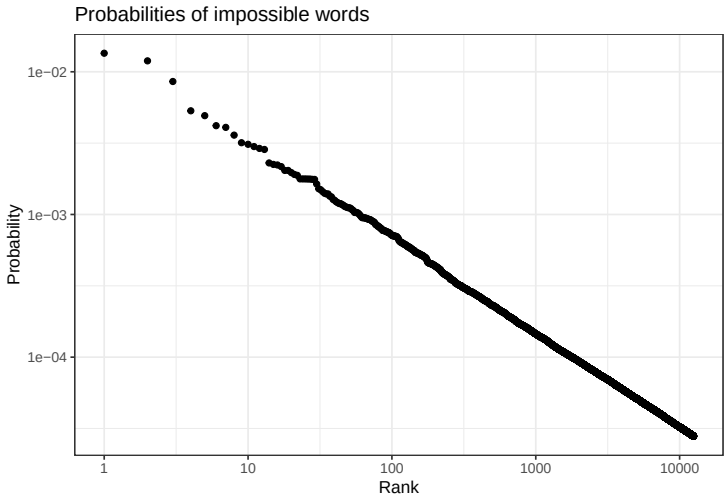


The possible word set contains 500 words.

```
n_plus <- 500
# Select the first n_plus words (they are not
# sorted) p_plus is the vector of their
# probabilities
p_plus <- p[1:n_plus]
autoplot(as.ProbaVector(p_plus), main = "Probabilities of possible words") +
    scale_x_log10()
```




```
# Impossible words
p_minus <- p[(n_plus + 1):n_set_all]
autoplot(as.ProbaVector(p_minus), main = "Probabilities of impossible words") +
    scale_x_log10()
```



4.2 Entropy partitionning

```
# Total entropy
gamma <- ent(p)
# Weights of groups
w_plus <- sum(p_plus)
w_minus <- sum(p_minus)
# Alpha entropy Probabilities in each group are
# global probabilities divided by the weight of
# the group
alpha <- w_plus * ent(p_plus/w_plus) + w_minus * ent(p_minus/w_minus)
# Beta entropy
beta <- ent(c(w_plus, w_minus))
# Check
gamma - alpha - beta  # Should be 0
```

```
## [1] 1.387779e-16
```

The entropy of the whole dataset is 12.65 bits. That brought by the color pattern is 4.62 bits. That of the possible word subset is 7.62 bits. The discrepancy is thus 0.41 bits.

4.3 Approximation

The derivation of **??**eq:partition) is detailed here, step by step. Each line of the code contains the total entropy, starting from α plus β entropy as defined before.

```
# Rearrange alpha and beta by group
w_plus * (ent(p_plus/w_plus) + log2(1/w_plus)) +
  w_minus * (ent(p_minus/w_minus) + log2(1/w_minus))
```

```
## [1] 12.65223
```

```r
# Replace w_plus by 1-w_minus in the first term
(1-w_minus) * (ent(p_plus/w_plus) + log2(1/w_plus)) +
  w_minus * (ent(p_minus/w_minus) + log2(1/w_minus))
```

```
## [1] 12.65223
```

```r
# Isolate the information of the video and the error term
ent(p_plus/w_plus) +
  log2(1/w_plus) +
  w_minus * (ent(p_minus/w_minus) - ent(p_plus/w_plus) + log2(1/w_minus) - log2(1/w_plus))
```

```
## [1] 12.65223
```

References

Marcon, E., B. Hérault, C. Baraloto, and G. Lang (2012). The decomposition of Shannon's entropy and a confidence interval for *beta* diversity. *Oikos 121*(4), 516–522.

Rao, C. R. and T. K. Nayak (1985). Cross entropy, dissimilarity measures, and characterizations of quadratic entropy. *IEEE Transactions on Information Theory 31*(5), 589–593.

Whittaker, R. H. (1960). Vegetation of the Siskiyou mountains, Oregon and California. *Ecological Monographs 30*(3), 279–338.