wordlenatoR

Thomas Maierhofer

January 23, 2022

Abstract

This document explains the theoretical framework for wordlenatoR, the optimal strategy for solving the game wordle. At every stage of the game wordlenatoR proposes optimal guesses that will, on average, reduce the number of possible target words as much as possible. This is achieved by computing the number of words eliminated from the target word dictionary for all possible guesses given all possible target words. A software implementation is provided.

1 Introduction

Wordle is a puzzle game where 5 letter word from a dictionary has to be guessed in up to 6 attempts, with hints provided after every guess. Only 5 letter words from the dictionary are permissible guesses. Every letter that is in the correct location with respect to the target word is colored in greed, every letter that is contained in the target word, but in the wrong location is colored in yellow, and every letter not contained in the target word is colored grey.

2 Optimal Strategy

This strategy tries the reduce the number of possible target words as much as possible with every guess. Initially, the only information about the target word is that is a 5 letter word contained in a dictionary. After every guess, the feedback provided reduces the number of possible target words, by eliminating all words that are inconsistent with the feedback provided on prior guesses. The next guess is chosen to reduce the number of possible target words as much as possible. It is impossible to compute the reduction in possible target words for a next guess a priori, as it depends on the feedback received, which in turn depends on the true target word, which is unknown to the player. But, very importantly, the expected number of possible target words eliminated can be estimated a priori for every word, using a double-sum over the reduction in possible target words for every possible guess and every possible target word. Assuming that every word in the dictionary is equally likely, this is an unbiased estimator of the true reduction in possible target words.

3 Computation

Let $\mathcal{D}$ denote a known dictionary of all permissible 5 letter target words, where the true target word d drawn at random with equal probability. For our purpose, d is fixed but unknown. Let $g \in \mathcal{D}$ be a guess, and $f(g|d) \in \{\text{green}, \text{yellow}, \text{gray}\}^5$ the known feedback given for the five letters of guess g given the true but unknown target word d.

To propose an optimal guess g^*, find the guess that results on average in the smallest possible set of target words $\mathcal{D}'$ over all possible target words $\tilde{d} \in \mathcal{D}$,

$$g^* = \arg\min_{g \in \mathcal{D}} \sum_{d' \in \mathcal{D}} \sum_{\tilde{d} \in \mathcal{D}} \#\{d' \in D : f(g|d') = f(g|\tilde{d})\}.$$

After submitting a guess, the feedback received can be used to reduce the dictionary $\mathcal{D}$ to an updated dictionary $\mathcal{D}'$ containing only words possible given the received feedback, i.e. all target words that would have given the same feedback for our guess,

$$\mathcal{D}' = \{d' \in \mathcal{D} : f(g|d') = f(g|d)\}.$$

The process of proposing the next optimal guess follows analogously to above, uing the reduced target word dictionary D' instead of the original target word dictionary D.

4 Software

An online application implemented uisng R shiny is available at https://maierhofer.shinyapps.io/wordle_app/. Computation times after the first guess can be up to 5 minutes, but are usually much faster. Later guesses usually compute in seconds. The target dictionary containing all 2315 possible wordle words is copied from https://github.com/ggilestro/playground/tree/main/wordle_strategy/wordle_words.txt.