AARHUS UNIVERSITY, COMPUTER SCIENCE

Crypthology Handin 2

Peter Burgaard - 201209175

September 14, 2016

1 PROBLEM TITLE

You are shown N cards, each of which cover one letter. Each letter has been independently chosen from the same distribution, and you are given the distribution $(p_0, p_1, ..., p_{25})$. You get to choose one letter from the alphabet, say you choose letter number i. Now every position in the hidden string where letter i occurs (if any) are uncovered. Your goal is to learn (on average) as much information as possible on the hidden string.
People tend to choose the most frequent letter as their guesses. Let's try to see what information theory has to say about this. Suppose we adopt the convention that Shannon used when defining Entropy: if you know that some event occurs with probability p, and you learn that this event did indeed occur, you have learnt log(1/p) bits of information.

QUESTION 1: Now, if your guess is letter nr. i, how many bits of information will you learn on average from playing the game?

From each guess, we can learn one of two things, either letter i's position(s), or that it's not present among the cards. This can be modeled as a random variable $\mathbf{X} \in \{True, False\}$, with $True$ being a correctly guessed letter i, and $False$ being an incorrect guess. This gives the simple distribution function:

$$P_r[\mathbf{X}] = \begin{cases} p_i & \text{if } True \\ 1 - p_i & \text{if } False \end{cases}$$

By definition 2.4 Stinson, we can calculate the entropy of $\mathbf{X}$ as

$$H(\mathbf{X}) = -\sum_{x \in \mathbf{X}} Pr[x] \cdot log_2(Pr[x]) = \sum_{x \in \mathbf{X}} Pr[x] \cdot log_2\left(\frac{1}{Pr[x]}\right)$$

By inserting the destribution of **X**

$$H(\mathbf{X}) = p_i \cdot log_2\left(\frac{1}{p_i}\right) + (1-p_i)\cdot log_2\left(\frac{1}{1-p_i}\right)$$

This is the entropy function if we only had one card, so we need to multiply by N, to get our entropy function f

$$f(\mathbf{X}, N) = N \cdot (p_i \cdot log_2\left(\frac{1}{p_i}\right) + (1-p_i)\cdot log_2\left(\frac{1}{1-p_i}\right))$$

QUESTION 2: What strategy does your result suggest for choosing your guess, given frequencies $p_0, .., p_{25}$ as in English?

If we assume $p_0, p_1, ..., p_{25}$ have the same probability distribution of the english language, we see:[1]

$$f(N,E) = N \cdot (0.12702 \cdot log_2(\frac{1}{0.12702}) + (1-0.12702)\cdot log_2\left(\frac{1}{1-0.12702}\right)) \approx 0.16532N$$
$$f(N,Z) = N \cdot (0.00074 \cdot log_2(\frac{1}{0.00074}) + (1-0.00074)\cdot log_2\left(\frac{1}{1-0.12702}\right)) \approx 0.0.00263N$$

which implies guession on the most frequently used letters gets the most information.

QUESTION 3: Based on this, does it make sense that players in real life choose the most frequent letter(s)? why or why not?

Based on the above example, it makes sense, if the N letters based like the english language. So in a case where these players are trying to guess words, it makes sense to choose the most used letters in the language of the word.

QUESTION 4: Would this be a good strategy no matter what the frequencies were?
No, e.g.

$$f(N, 0.5) = N \cdot (0.5 \cdot log_2\left(\frac{1}{0.5}\right) + (1-0.5)\cdot log_2\left(\frac{1}{1-0.5}\right) = 0.301N$$
$$f(N, 0.9) = N \cdot (0.9 \cdot log_2\left(\frac{1}{0.9}\right) + (1-0.9)\cdot log_2\left(\frac{1}{1-0.9}\right) \approx 0.1411N$$

The information pr bit we gather is highest at 50% chance of appreance, and decreases when more frequent than that. Therefore we might gather more information by guessing on more unfrequent letters, if there is e.g. one dominant letter which appears 90% times or more.

[1] Statistics are gathered from https://en.wikipedia.org/wiki/Letter_frequency