Applications of information theory

Daniel Bosk

Department of Information and Communication Systems,
Mid Sweden University, Sundsvall.

6th April 2020

1 Applications
- Information density and redundancy
- Passwords
- Identifying information

1 Applications

- Information density and redundancy
- Passwords
- Identifying information

Definition

- Natural language L.
- Stochastic variable $\mathbf{P}_L^n$ of strings of length n.
- (Alphabet P_L.)
- Entropy of L defined as

$$H_L = \lim_{n \to \infty} \frac{H(\mathbf{P}_L^n)}{n}.$$

- Redundancy in L is

$$R_L = 1 - \frac{H_L}{\log |P_L|}.$$

Remark

Meaning we have H_L bits per character in L.

Example ([Sha48])

- Entropy of 1–1.5 bits per character in English.
- Redundancy of approximately $1 - \frac{1.25}{\log 26} \approx 0.73$.

Example ([Sha48])

Two-dimensional cross-word puzzles requires redundancy of approximately 0.5.

Example

- Redundancy of ‘SMS languages’ is lower than for ‘non-SMS languages’.
- Compare ‘wait’ and ‘w8’.

Remark

- Lower redundancy is more space-efficient.
- Incurs more errors.

Idea [Kom+11]

- Look at different aspects of passwords individually, then summarize.
- Can use $H(x_1, x_2, \ldots, x_n) \leq H(x_1) + H(x_2) + \cdots + H(x_n)$.
- This allows us to reason about bounds.

Example

- We can look at properties such as:
  - length,
  - number of and placement of character classes,
  - the actual characters,
  - . . .

Remark

- These are *not independent*.
- The sum will be an *upper bound*.

Example

- We can look at properties such as:
  - length,
  - number of and placement of character classes,
  - the actual characters,
  - . . .

Remark

- These are *not independent*.
- The sum will be an *upper bound*.

Remark

- With an upper bound we know it's not possible to do better.
- With an average we know how well most users will do.
- With a lower bound we have a guarantee — not possible!

Remark

- If a password policy yields low entropy, it implies it's bad.
- If a password policy yields high entropy, it *doesn't* imply that it's good.

Exercise

Why?

Remark

- If a password policy yields low entropy, it implies it's bad.
- If a password policy yields high entropy, it *doesn't* imply that it's good.

Exercise

Why?

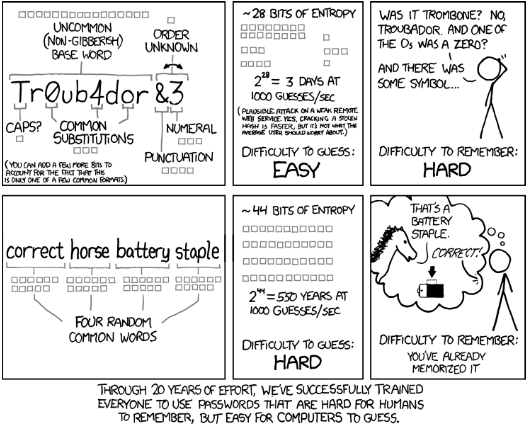

Figure: xkcd's strip on password strength. Picture: xkcd [xkc].

Example (Standard password)

- We have
  - 26 alphabetic characters,
  - 10 numbers,
  - 10 special characters (approximately).
- This yields $\log(2 \times 26 + 10 + 10) = \log 72 \approx 6$ bit per password character.
- A 10-character *uniformly randomly* generated password contains 60 bit.

Remark

What happens when we require two upper and two lower-case characters, two numbers must be included?

Example (Standard password)

- We have
  - 26 alphabetic characters,
  - 10 numbers,
  - 10 special characters (approximately).
- This yields $\log(2 \times 26 + 10 + 10) = \log 72 \approx 6$ bit per password character.
- A 10-character *uniformly randomly* generated password contains 60 bit.

Remark

What happens when we require two upper and two lower-case characters, two numbers must be included?

Example (Four-word passphrase)

- We have 125 000 words in the standard Swedish dictionary.
- This yields $\log 125\,000 \approx 17$ bit per word.
- A four-word *uniformly randomly* generated passphrase contains 68 bit.

Example (Random sentence)

- We estimated the entropy per character in a language.
- It was approximately 1.25 bit for English.
- A 20-character *uniformly randomly* generated sentence would yield 25 bit.

Remark

- All these require uniform randomness.
- Humans are bad at remembering random things.
- Thus they will choose non-randomly.
- The entropy will thus be (possibly much) lower.

Example

Do we get more information from zodiac signs or birthdays?

$$
\begin{aligned}
-\sum_{\text{zodiacs}} \frac{1}{12} \log \frac{1}{12} &= \log 12 \approx 3.58 \\
&< -\sum_{\text{days of year}} \frac{1}{365} \log \frac{1}{365} = \log 365 \approx 8.51.
\end{aligned}
$$

Exercise

How much information do we need to uniquely identify an individual?

Example

- Sometime during 2011 there were $n = 6\,973\,738\,433$[1] people on earth.
- To give everyone a unique identifier we need $\log n \approx 32.7 \approx 33$ bits of information.

[1] According to the World Bank.

Identifying information in browsers

- Electronic Frontier Foundation (EFF) studied [Eck10] how much information a web-browser shares.
- You can try your browser in
  - `http://panopticlick.eff.org/`, and
  - `https://amiunique.org/`.

Example (My browser)

- My Firefox-browser with all addons gave 21.45 bits of entropy.
- Then the number of tested users were 2 860 696.

Identifying information in browsers

- Electronic Frontier Foundation (EFF) studied [Eck10] how much information a web-browser shares.
- You can try your browser in
  - `http://panopticlick.eff.org/`, and
  - `https://amiunique.org/`.

Example (My browser)

- My Firefox-browser with all addons gave 21.45 bits of entropy.
- Then the number of tested users were 2 860 696.

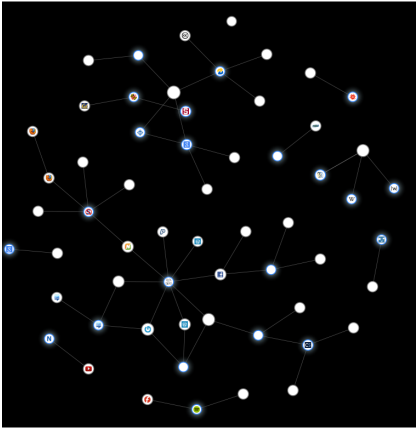

Figure: Screenshot from Collusion (now Lightbeam) for Firefox. Map over all pages that track me using this information.

[Eck10] Peter Eckersley. 'How Unique Is Your Browser?' In: *Privacy Enhancing Technologies*. Springer. 2010, pp. 1–18. URL: https://panopticlick.eff.org/static/browser-uniqueness.pdf.

[Kom+11] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Christin Nicolas, Lorrie Faith Cranor and Serge Egelman. 'Of passwords and people: Measuring the effect of password-composition policies'. In: *CHI*. 2011. URL: http://cups.cs.cmu.edu/rshay/pubs/passwords_and_people2011.pdf.

[Sha48] C. E. Shannon. 'A Mathematical Theory of Communication'. In: *The Bell System Technical Journal* 27 (July 1948), pp. 379–423, 623–656.

References

[xkc] xkcd. *Password Strength*. URL: https://xkcd.com/936/.