Some answers to student questions

Trevor K. Karn

February 14, 2023

Contents

Noiseless coding

Question 1. *Does there exist an algorithm to determine if a code is uniquely decipherable?*

Answer. *Yes! It is called the Sardinas-Patterson algorithm.*[1]

Question 2. *What happens when equality is achieved in Kraft-McMillan inequality? In other words, given a sequence of potential lengths oof code-words $(\ell_1, \ell_2, \ldots, \ell_m)$, in an n-ary alphabet, what can we say when*

$$\sum_{i=1}^{m} \frac{1}{n^{\ell_i}} = 1?$$

Answer. *In the case equality is achieved, the code is* exhaustive.[2] *An exhaustive code $f : W \to \Sigma^*$ is one in which any sequence of letters is either a message $f^*(w_1 w_2 \cdots w_n)$ or the prefix of a message.*[3]

Question 3. *If $X : \Omega_1 \to \mathbb{R}$ and $Y : \Omega_2 \to \mathbb{R}$ are random variables, define $X * Y$ to be the convolution, a random variable $X * Y : \Omega_1 \times \Omega_2 \to \mathbb{R}$ defined as $X * Y(\omega_1, \omega_2) = X(\omega_1) + Y(\omega_2)$. Is $\mathbb{E}(X * Y) = \mathbb{E}(X) + \mathbb{E}(Y)$?*

Answer. *Yes. It is a distinct statement from the classic "linearity of expectation," which discusses the relationship between the sum of two random variables from the same probability space, while here we are dealing with two probability spaces. In the setting of a continuous random variable, it requires Fubini's theorm.*[4] *Here is the proof for finite, discrete probability spaces: let $X : A \to \mathbb{R}$ and $Y : B \to \mathbb{R}$ be two random variables, where $A = \{\alpha_1, \ldots, \alpha_m\}$ and $B = \{\beta_1, \ldots, \beta_n\}$. Let $\Omega = A \times B$. Then define the convolution of X with Y to be $X * Y : \Omega \to \mathbb{R}$ as $X * Y(\alpha_i, \beta_j) = X(\alpha_i) + Y(\beta_j)$. Assume that X, Y are independent (but they need not be identically distributed). Independence means that*

[1] Wikipedia. Sardinas–Patterson algorithm — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Sardinas%E2%80%93Patterson%20algorithm&oldid=1128726851`, 2023. [Online; accessed 24-January-2023]

[2] Mordecai J. Golin and Hyeon-Suk Na. Generalizing the Kraft-McMillan Inequality to Restricted Languages. In *Proceedings of the Data Compression Conference*, DCC '05, page 163–172, USA, 2005. IEEE Computer Society

[3] L.S. Bobrow and S.L. Hakimi. Graph theoretic prefix codes and their synchronizing properties. *Information and control*, 15(1):70–94, 1969

[4] Carl (https://stats.stackexchange.com/users/99274/carl). Prove that the mean value of a convolution is the sum of the mean values of its individual parts. Cross Validated. URL:https://stats.stackexchange.com/q/342023 (version: 2018-04-22)

$$\begin{aligned}
&P(\omega_{ij}) = P(\alpha_i)P(\beta_j) \textit{ when } \omega_{ij} = (\alpha_i, \beta_j) \\
&\mathbb{E}(X * Y) = \sum_{\omega_{ij}\in\Omega} P(\omega_{ij}) X * Y(\omega_{ij}) \\
&\quad = \sum_{\omega_{ij}} P(\alpha_i)P(\beta_j)(X(\alpha_i) + Y(\beta_j)) && \textit{since } X, Y \textit{ indpt.} \\
&\quad = \sum_{\alpha_i\in A}\sum_{\beta_j\in B} P(\alpha_i)P(\beta_j)(X(\alpha_i) + Y(\beta_j)) \\
&\quad = \sum_{\alpha_i\in A} P(\alpha_i)\left(\sum_{\beta_j\in B} P(\beta_j)(X(\alpha_i) + Y(\beta_j))\right) \\
&\quad = \sum_{\alpha_i\in A} P(\alpha_i)\left(X(\alpha_i) + \sum_{\beta_j\in B} P(\beta_j)Y(\beta_j)\right) && \textit{since } \sum P(\beta_j) = 1 \\
&\quad = \sum_{\alpha_i} P(\alpha_i)X(\alpha_i) + P(\alpha_i)\mathbb{E}(Y) && \textit{def. of } \mathbb{E}(Y) \\
&\quad = \sum_{\alpha_i} P(\alpha_i)X(\alpha_i) + \sum_{\alpha_i} P(\alpha_i)\mathbb{E}(Y) \\
&\quad = \mathbb{E}(X) + \mathbb{E}(Y) && \textit{def. of } \mathbb{E}(X) \textit{ and since } \sum P(\alpha_i) = 1.
\end{aligned}$$

Noisy coding

Question 4. *If we use the Hamming distance, to measure the difference between words of the same length, how do we measure the distance between two words of differing lengths?*

Answer. *There is a different metric, called the* Levenshtein distance *which is defined recursively for words $x = x_1x_2\dots x_n$ and $y = y_1y_2\dots y_n$ as follows:*

$$d(x,y) = \begin{cases} n & m = 0 \\ m & n = 0 \\ d(x_2x_3\dots x_n, y_2y_3\dots y_m) & x_1 = y_1 \\ 1 + \min\{d(x_2\dots x_n, y_1\dots y_m), d(x_1\dots x_n, y_2\dots y_m), d(x_2\dots x_n, y_2\dots y_m)\} & o/w \end{cases}$$

[5] *For example, suppose we want to compare the distances between* 010 *and* 1001*. Since the first bits are different Then*

$$d(010, 1001) = 1 + \min\{d(10, 1001), d(010, 001), d(10, 001)\}$$

so we compute

Distance	*value*
$d(10, 1001)$	$d(0, 001)$
$d(010, 001)$	$d(10, 01)$
$d(10, 001)$	$1 + \min\{d(0, 001), d(10, 01), d(0, 01)\}$

[5] Wikipedia. Levenshtein distance — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Levenshtein%20distance&oldid=1135561718`, 2023. [Online; accessed 14-February-2023]

So now we compute

Distance	*value*
$d(0,001)$	$d(-,01) = 2$
$d(10,01)$	$1 + \min\{d(0,01), d(10,1), d(0,1)\}$
$d(0,01)$	$d(-,1) = 1$

and again,

Distance	*value*
$d(0,01)$	$d(-,1) = 1$
$d(10,1)$	$d(0,-) = 1$
$d(0,1)$	$1 + \min\{d(-,1), d(0,-), d(-,-)\} = 1 + 0 = 1$

So now we see that $\min\{d(0,01), d(10,1), d(0,1)\} = 1$*, and so* $1 + \min\{d(0,001), d(10,01), d(0,01)\} = 2$*. Then* $\min\{d(10,1001), d(010,001), d(10,001) = \min\{2,2,2\} = 2$*, so we conclude*

$$d(010,1001) = 3.$$

This makes sense because there is no way to shift the indices of 010 *to make it line up with a subword of* 1001*.*

References

[BH69] L.S. Bobrow and S.L. Hakimi. Graph theoretic prefix codes and their synchronizing properties. *Information and control*, 15(1):70–94, 1969.

[GN05] Mordecai J. Golin and Hyeon-Suk Na. Generalizing the Kraft-McMillan Inequality to Restricted Languages. In *Proceedings of the Data Compression Conference*, DCC '05, page 163–172, USA, 2005. IEEE Computer Society.

[3] Carl (https://stats.stackexchange.com/users/99274/carl). Prove that the mean value of a convolution is the sum of the mean values of its individual parts. Cross Validated. URL:https://stats.stackexchange.com/q/342023 (version: 2018-04-22).

[Wik23a] Wikipedia. Levenshtein distance — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Levenshtein%20distance&oldid=1135561718`, 2023. [Online; accessed 14-February-2023].

[Wik23b] Wikipedia. Sardinas–Patterson algorithm — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Sardinas%E2%80%93Patterson%20algorithm&oldid=1128726851`, 2023. [Online; accessed 24-January-2023].