10 The Inner Coincidence Index of a Text

Definition

Let $a \in \Sigma^r$ ($r \geq 2$) be a text, and $(\kappa_1(a), \ldots, \kappa_{r-1}(a))$ be its autocoincidence spectrum. Then the mean value

$$\varphi(a) := \frac{1}{r-1}\left[\kappa_1(a) + \cdots + \kappa_{r-1}(a)\right]$$

is called the **(inner) coincidence index** of a.

It defines a map

$$\varphi\colon \Sigma^{(\geq 2)} \longrightarrow \mathbb{Q}.$$

See the Perl program `phi.pl` from `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`.

Another description

Pick up the letters from two random positions of a text a. How many "twins" will you find? That means the same letter $s \in \Sigma$ at the two positions, or a "coincidence"?

Let $m_s = m_s(a) = \#\{j \mid a_j = s\}$ be the number of occurrences of s in a. Then the answer is

$$\frac{m_s \cdot (m_s - 1)}{2}$$

times. Therefore the total number of coincidences is

$$\sum_{s\in\Sigma} \frac{m_s \cdot (m_s-1)}{2} = \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s^2 - \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s = \frac{1}{2} \cdot \sum_{s\in\Sigma} m_s^2 - \frac{r}{2}$$

We count these coincidences in another way by the following algorithm: Let z_q be the number of already found coincidences with a distance of q for $q = 1, \ldots, r-1$, and initialize it as $z_q := 0$. Then execute the nested loops

for $i = 0, \ldots, r-2$	[loop through the text a]
for $j = i+1, \ldots, r-1$	[loop through the remaining text]
if $a_i = a_j$	[coincidence detected]
increment z_{j-i}	[with distance $j - i$]
increment z_{r+i-j}	[and with distance $r + i - j$]

After running through these loops the variables $z_1, \ldots, z_{r-1}$ have values such that

Lemma 1 (i) $z_1 + \cdots + z_{r-1} = \sum_{s\in\Sigma} m_s \cdot (m_s - 1)$.

(ii) $\kappa_q(a) = \frac{z_q}{r}$ *for* $q = 1, \ldots, r-1$.

Proof. (i) We count all coincidences twice.

(ii) $\kappa_q(a) = \frac{1}{r} \cdot \#\{j \mid a_{j+q} = a_j\}$ by definition (where the indices are taken mod r). $\diamond$

The Kappa-Phi Theorem

Theorem 1 (Kappa-Phi Theorem) *The inner coincidence index of a text* $a \in \Sigma^*$ *of length* $r \geq 2$ *is the proportion of coincidences among all letter pairs of* a*.*

Proof. The last term of the equation

$$\begin{aligned} \varphi(a) &= \frac{\kappa_1(a) + \cdots \kappa_{r-1}(a)}{r-1} = \frac{z_1 + \cdots + z_{r-1}}{r \cdot (r-1)} \\ &= \frac{\sum_{s \in \Sigma} m_s \cdot (m_s - 1)}{r \cdot (r-1)} = \frac{\sum_{s \in \Sigma} \frac{m_s \cdot (m_s - 1)}{2}}{\frac{r \cdot (r-1)}{2}} \end{aligned}$$

has the total number of coincidences in its numerator, and the total number of letter pairs in its denominator. $\diamond$

Corollary 1 *The inner coincidence index may be expressed as*

$$\varphi(a) = \frac{r}{r-1} \cdot \sum_{s \in \Sigma} \left(\frac{m_s}{r}\right)^2 - \frac{1}{r-1}$$

Proof. This follows via the intermediate step

$$\varphi(a) = \frac{\sum_{s \in \Sigma} m_s^2 - r}{r \cdot (r-1)}$$

$\diamond$

Note that this corollary provides a much faster algorithm for determining $\varphi(a)$. The definition formula needs $r - 1$ runs through a text of length r, making $r \cdot (r-1)$ comparisons. The above algorithm reduces the costs to $\frac{r \cdot (r-1)}{2}$ comparisons. Using the formula of the corollary we need only one pass through the text, the complexity is linear in r. For a Perl program implementing this algorithm see the Perl script `coinc.pl` from the web page `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/`

Corollary 2 *The inner coincidence index of a text is invariant under* **mono***alphabetic substitution.*

Proof. The number of letter pairs is unchanged. $\diamond$