LEMPEL-ZIV COMPLEXITY BASED ESTIMATORS

Most algorithms for mutual information focuses on random variables instead of stochastic processes. It can be partially justified whenever a process is formed by independent and identically distributed (iid) variables through time. More precisely, let $\{X(n)\}$ and $\{Y(n)\}$ be two discrete-time random processes, where $n \in \mathbb{Z}$ stands for the discrete-time counter. By defining $H(\{X(n)\})$ and $H(\{Y(n)\})$ as the entropy rate (or information rate, in bits per symbol) of $\{X(n)\}$ and $\{Y(n)\}$, respectively, and by defining $H(\{X(n)\},\{Y(n)\})$ as their joint entropy rate, we also define

$$I(\{X(n)\};\{Y(n)\}) = H(\{X(n)\}) + H(\{Y(n)\}) - H(\{X(n)\},\{Y(n)\})$$

Eq. (3.1)

as the MI between the processes $\{X(n)\}$ and $\{Y(n)\}$, i.e. a generalization of MI between random variables.

Clearly, if $\{X(n)\}$ is iid, then every random variable in it, $X(n)$, $\forall n \in \mathbb{Z}$ is associated to the very same amount of entropy $H_X = H(X(n))$, in bits. Consequently, the entropy rate of the i.i.d. process, $H(\{X(n)\})$, in bits per symbol, is numerically equal to H_X, and all usual methods for entropy and MI estimation for random variables are enough as tools.

By contrast, if a process is stationary [3.1] but not independent, then $H(\{X(n)\}) < H_X$, and its entropy rate per symbol is given by:

$$H(\{X(n)\}) = \lim_{N\to\infty} \frac{-1}{N} \sum_{i=1}^{K^N} p_i \log_K(p_i)$$ Eq. (3.2)

where K stands for the number of possible states/symbols X may assume, and p_i stands for the joint probability of $(X(n), X(n+1), \ldots, X(n+N-1))$ being equal to the $i-th$ sequence of states (out of K^N possible sequences).

The main drawback of the *so called* plug-in estimator suggested by Eq. 3.2. (with p_i replaced with relative frequencies, $\hat{p}_i$) is the huge amount of stationary data it demands, because the number of possible sequences, K^N, exponentially grows with the sequence length N.

Some alternatives to cope with it have appeared in literature. In the following, we briefly explain one of the most powerful of them, based on complexity analysis of finite sequences (instead of entropy of sources).

In 1976, A. Lempel and J. Ziv [3.2] proposed an approach for complexity analysis of symbol sequences. An important aspect of their approach is the lack of *a priori* with regard to the source of symbols, which clearly contrasts with the measurement of (source) Shannon entropy. Though they are conceptually different measures, it was shown that [3.1], under ergodicity conditions, Lempel-Ziv's complexity of increasingly long symbol sequences converges almost surely to the Shannon entropy of the source from which symbols are drawn.

Lempel-Ziv's (LZ) approach, latterly simplified for practical reasons, became widely known as the compression algorithm behind many computer programs for file compression - the "zip-like" programs. We should probably credit its success to its universality, in other words, to its lack of a priori. Nevertheless, it should be also highlighted that zip-like programs are just the "tip of the iceberg," for compression is just a single offspring of the elegant theory presented in [3.2.].

Complexity measure - Let x_1^N represent a sequence (e.g. a single instance of a random process), and let a *Minimal Length Block* (MLB) be a subsequence x_i^j of x_1^N ($1 \le i, j, \le N$) such that it does not occurs in x_1^{j-1}. Then, there is a unique decomposition of x_1^N into MLB, and the total number of these blocks, p, is the complexity measure of the sequence, denoted as:

$$C(x_1^N) = p$$

Illustration (from [3.2]): In this illustration, a sequence of $N = 16$ binary ($K = 2$) symbols is parsed as:

$$x_1^{16} = 0001101001000101$$

$$\downarrow \textit{parsing}$$

$$0 \cdot 001 \cdot 10 \cdot 100 \cdot 1000 \cdot 101$$

Please note that the last block may produce an exception to the parsing rule, since it may be not unique (i.e. not an MLB), as in this illustration. As a result, we have that the complexity of this specific sequence is 6, i.e.:

$$C(x_1^{16}) = 6$$

By comparing the complexity $C(x_1^N)$ to the maximum expected complexity of a hypothetical sequence of same length, which is given by $\frac{N}{\log_K N}$, we obtain the normalized complexity of the sequence, denoted by:

$$c(x_1^N) = \frac{C(x_1^N)}{N/\log_K N}$$ Eq.(3.3)

which almost surely [3.1] converges to the entropy rate given by Eq. 3.2., i.e. the entropy rate of the stochastic process of which x_1^N is likely to be an instance.

In order to illustrate the use of the Lempel-Ziv approach for entropy rate estimation, we reproduce here the experiment presented in [3.3], where a Markov Chain (whose true entropy rate can be analytically calculated) is used to generate random sequences of 0s and 1s.

The two-state Markov process (in discrete time) used in this experiment has a stationary transition matrix given by:

$$P=\begin{bmatrix} 1-p_{10} & p_{01} \\ p_{10} & 1-p_{01} \end{bmatrix}$$

where $p_{01}=Prob(X(n+1)=0|X(n)=1)$ and $p_{10}=Prob(X(n+1)=1|X(n)=0)$ are transition probabilities. It can be demonstrated [3.1] that these two parameters, p_{01} and p_{10}, completely determine the entropy rate of the finite-length process $\{X(n)\}$, $n=1,2,\ldots,N$.

Moreover, it can be shown [3.1] that, as $N\to\infty$, the entropy rate of this Makovian source, in bits per symbol, is given by:

$$H=\frac{-p_{01}(p_{10}\log_2 p_{10}+(1-p_{10})\log_2(1-p_{10}))}{(p_{01}+p_{10})} \\ \frac{-p_{10}(p_{01}\log_2 p_{01}+(1-p_{01})\log_2(1-p_{01}))}{(p_{01}+p_{10})}$$

Eq(3.4)

For instance, if $p_{01}=0.8$ and $p_{10}=0.1$, we obtain that the resulting binary source asymptotically "produces" 0.497 bits of information per emitted binary symbol.

For all very specific cases where $p_{01}+p_{10}=1$, $H(\{X(n)\})$ does not depend on *N,* which makes even the plug-in method less inaccurate. However, these are rather rare cases of Markov processes and, in general, the plug-in method is to be avoided, unless a huge amount of data is available. This is evident if we keep in mind that this method relies upon relative frequency of occurrences of symbol sequences. Clearly, the number of possible sequences exponentially grows with its length, and so does the amount of necessary data to avoid statistical undersampling problems.

On the other hand, the astonishing 'simple to obtain' measure presented in Eq.(3.3) provides us with accurate estimates of $H(\{X(n)\})$ (although it is aimed at measuring complexity of specific sequences of symbols). To illustrate it, in Figures 3.1 and 3.2, we can observe how fast the normalized complexity measure converges to the true asymptotic information rate of corresponding processes, by using symbolic sequences of length up to 4000 symbols.

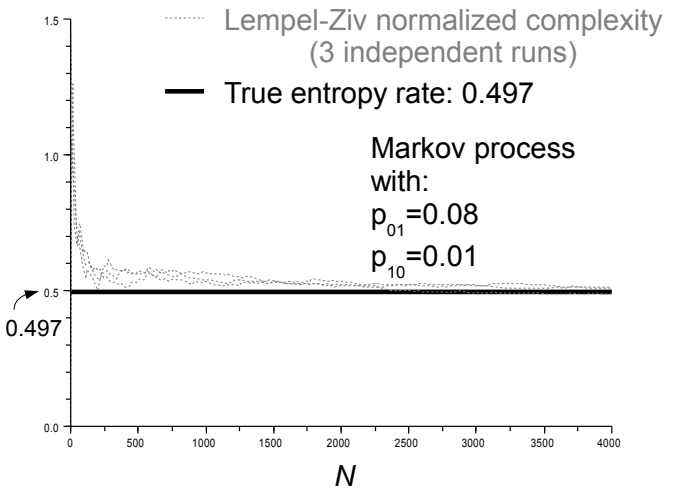

Fig. 3.1: Three independent runs of the normalized complexity measure, with $p_{01}=0.8$ and $p_{10}=0.1$.

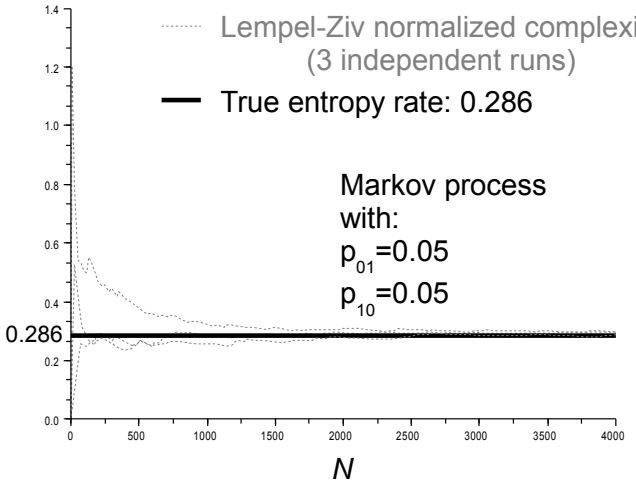

Fig. 3.2: Three independent runs of the normalized complexity measure, with $p_{01}=0.05$ and $p_{10}=0.05$.

Finally, if two random processes share some amount of information, it is also possible to measure it through Eq. (3.1), where $H(\{X(n)\},\{Y(n)\})$ can be easily computed as the entropy rate of a new "concatenated" process:

$$Z(n)=\begin{bmatrix} \{X(n)\} \\ \{Y(n)\} \end{bmatrix}$$

REFERENCES

[3.1] Cover, T. M., Thomas, J. A. (1991). *Elements of information theory.* Wiley, New York.

[3.2] Lempel, A., Ziv, J. (1976). On the complexity of an individual sequence. *IEEE Trans. Inform. Theory*, IT 22, pp. 75–88.

[3.3] Amigo, J. M., Szczepanski, J., Wajnryb, E. Sanchez-Vives, M. V. (2004). Estimating the entropy rate of spike trains via Lempel-Ziv complexity . *Neural Computation* , Vol.16, pp. 717–736.