Authorship attribution using relative compression

Armando J. Pinho, Diogo Pratas, and Paulo J. S. G. Ferreira

IEETA - Institute of Electronics and Informatics Engineering of Aveiro
DETI - Department of Electronics, Telecommunications and Informatics
University of Aveiro, 3810-193 Aveiro, Portugal
{ap,pratas,pjf}@ua.pt

Abstract

Authorship attribution is a classical classification problem. We use it here to illustrate the performance of a compression-based measure that relies on the notion of *relative compression*. Besides comparing with recent approaches that use multiple discriminant analysis and support vector machines, we compare it with the Normalized Conditional Compression Distance (a direct approximation of the Normalized Information Distance) and the popular Normalized Compression Distance. The Normalized Relative Compression (NRC) attained 100% correct classification in the data set used, showing consistency between the compression ratio and the classification performance, a characteristic not always present in other compression-based measures.

Introduction

We consider the problem of authorship attribution to assess the performance of three compression-based measures, namely, the Normalized Compression Distance (NCD), the Normalized Conditional Compression Distance (NCCD), and a new approach, the Normalized Relative Compression (NRC). For comparison with other non compression-based approaches, we rely on recent benchmarking results available for the data set used in [1]. To attenuate the potential impact of using different compressors for each measure, we implemented a generic compressor, based on mixtures of finite-context models (FCMs), that is able to operate in several modes: the usual non-referential compression mode, denoted $C(x)$, the (referential) conditional compression mode, denoted $C(x|y)$, and the (referential) relative compression mode, denoted $C(x\|y)$, where, without loss of generality, x and y can be considered binary strings.

In the conditional compression mode, the compressor starts by building an internal model of y, using a combination of FCMs of several orders (see below for more details). After processing y, these models are kept fixed. In the second phase, x is compressed using the (fixed) models of y and another set of FCMs that learn the statistics of x as it is processed. Each symbol of x is encoded using a probability estimate, resulting from a mixture of the probabilities produced by each of the FCMs (those modeling y and those modeling x). This implements the $C(x|y)$, required to compute the NCCD.

The relative compression mode, $C(x\|y)$, differs in how the internal models of the encoder are built. In this case, as in $C(x|y)$, a set of FCMs are loaded with the information of y and kept fixed afterwards. However, contrarily to $C(x|y)$, there is no modeling of x during the encoding phase, i.e., x is encoded *exclusively* using the models built from y.

Some compression-based measures

Almost two decades ago, Bennett *et al.* proposed an information distance that minorizes, in an appropriate sense, every effective metric [2], i.e., if two strings are closely related according to any "admissible distance", then they will also be close to each other according to the Information Distance (ID) [3]. The ID and its normalized version, the Normalized Information Distance (NID) [3], are defined in terms of the Kolmogorov complexity of the involved strings, $K(x)$ and $K(y)$, as well as of the complexity of one of them when the other is provided, i.e., $K(x|y)$ and $K(y|x)$,

$$\mathrm{NID}(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \tag{1}$$

Because the Kolmogorov complexity is noncomputable, alternatives to (1) have been proposed to render it practical. Its compression-based direct counterpart depends on $C(x|y)$, i.e., the number of bits of the compressed version of x when y is given as additional input to the compressor, leading to the Normalized Conditional Compression Distance (NCCD)

$$\mathrm{NCCD}(x,y) = \frac{\max\{C(x|y), C(y|x)\}}{\max\{C(x), C(y)\}}. \tag{2}$$

This measure was used, e.g., by Nikvand *et al.* to estimate image distortion [4, 5].

A well known and popular alternative to the NCCD is the Normalized Compression Distance (NCD) [3, 6], defined as

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{3}$$

where $C(x)$ and $C(y)$ represent, respectively, the number of bits of a compressed version of x and y, and $C(xy)$ the number of bits of a compressed version of x and y concatenated. The main advantage of the NCD is not requiring special purpose compressors for computing the conditional compression, $C(x|y)$.

We also consider a measure based on the notion of *relative compression*, denoted by $C(x\|y)$, representing the compression of x relatively to y. This measure obeys to

1. $C(x\|y) \approx 0$ iff string x can be built efficiently from y;
2. $C(x\|y) \approx |x|$ iff $K(x|y) \approx K(x)$,

based on which we define the Normalized Relative Compression (NRC) of string x given string y as

$$\mathrm{NRC}(x,y) = \frac{C(x\|y)}{|x|}, \tag{4}$$

where $|x|$ denotes the length of x. Notice that in the more common interpretation of conditional compression, denoted by $C(x|y)$, we have $C(x|y) \approx C(x)$ iff $K(x|y) \approx K(x)$ (i.e., when x and y are totally unrelated), and $C(x|y) \approx |x|$ iff $K(x|y) \approx |x|$ (i.e., when x and y are totally unrelated and x is incompressible).

The idea of relative compression was used before in several different forms, for various purposes. For example, Watanabe *et al.* [7, 8] proposed the pattern representation scheme using data compression (PRDC) for media data analysis, in which we can find ideas close to those of the relative compression concept. Similar ideas were used to estimate the relative entropy from the data contained in the x and y strings, for posterior estimation of the distance between them [9–11]. Also, Cao *et al.* [11] proposed estimators for the relative entropy based on the Burrows-Wheeler block sorting transform [12] and on the context-tree weighting data compression method [13], revisiting the key idea of model-freezing, initially proposed by Dawy *et al.* [14]. In fact, model-freezing is a central idea in relative compression.

A generic encoder based on mixtures of finite-context models

We developed an encoder based on mixtures of FCMs, whose mixture weights are continuously adapted during compression, according to the performance of each individual probabilistic model. After seeing the first n symbols of x, denoted x_1^n, the average number of bits generated by an order-k FCM is (logarithms are base-2)

$$H_{k,n} = -\frac{1}{n}\sum_{i=1}^{n} \log P(x_i|x_{i-k}^{i-1}), \tag{5}$$

where we assume the convention that x_{1-k}^0 is known to both the encoder and decoder. $H_{k,n}$ can be viewed as a measure of the average performance of model k until position n. Therefore, the overall probability estimate for position $n+1$ can be given by the weighted average of the probabilities provided by each model, according to their individual performance, i.e.,

$$P(x_{n+1}) = \sum_{k\in\mathcal{K}} P(x_{n+1}|x_{n-k+1}^n)\; w_{k,n}, \tag{6}$$

where $\mathcal{K}$ denotes the set of $|\mathcal{K}|$ models involved in the mixture, and

$$w_{k,n} = P(k|x_1^n), \tag{7}$$

i.e., the weights correspond to the probabilities that each model has generated x_1^n. Hence, we have

$$w_{k,n} = P(k|x_1^n) \propto P(x_1^n|k)P(k), \tag{8}$$

where $P(x_1^n|k)$ denotes the likelihood of sequence x_1^n being generated by model k and $P(k)$ denotes the prior probability of model k. Assuming $P(k) = 1/|\mathcal{K}|$, we also obtain

$$w_{k,n} \propto P(x_1^n|k). \tag{9}$$

Calculating the logarithm of this probability we get

$$\log P(x_1^n|k) = \log\prod_{i=1}^{n} P(x_i|k, x_1^{i-1}) = \log\prod_{i=1}^{n} P(x_i|x_{i-k}^{i-1}) = \sum_{i=1}^{n} \log P(x_i|x_{i-k}^{i-1}), \tag{10}$$

which is related to the number of bits that would be required by model k to represent the sequence x_1^n. Therefore, it is related to the accumulated measure of the performance of model k until position n.

To facilitate faster adaptation to non-stationarities of the data, instead of using the whole accumulated performance of the model, we adopt a progressive forgetting mechanism. The idea is to let each model to progressively forget the distant past and, consequently, to give more importance to recent performance results. To accommodate this, we first rewrite (10) as

$$\log P(x_1^n|k) = \sum_{i=1}^{n-1} \log P(x_i|x_{i-k}^{i-1}) + \log P(x_n|x_{n-k}^{n-1}) \tag{11}$$

and then

$$\log p_{k,n} = \gamma \log p_{k,n-1} + \log P(x_n|x_{n-k}^{n-1}), \tag{12}$$

where $\gamma \in [0, 1)$ is the forgetting factor and $(-\log p_{k,n})$ represents the estimated number of bits that would be required by model k to represent the sequence x_1^n (we set $p_{k,0} = 1$), taking into account the forgetting mechanism. Removing logarithms, we rewrite (12) as

$$p_{k,n} = p_{k,n-1}^{\gamma} P(x_n|x_{n-k}^{n-1}) \tag{13}$$

and, finally, we set the weights to

$$w_{k,n} = \frac{p_{k,n}}{\sum_{k \in \mathcal{K}} p_{k,n}}. \tag{14}$$

The probability estimates $P(x_{n+1}|x_{n-k+1}^n)$ are calculated using symbol counts, according to

$$P(0|x_{n-k+1}^n) = \frac{N(0|x_{n-k+1}^n) + \alpha}{N(0|x_{n-k+1}^n) + N(1|x_{n-k+1}^n) + 2\alpha}, \tag{15}$$

where $N(0|x_{n-k+1}^n)$ represents the number of times that, in the past, the sequence $x_{n-k+1}^n 0$ was found. Parameter α allows balancing between the maximum likelihood estimator and a uniform distribution (when the total number of events, n, is large, it behaves as a maximum likelihood estimator). For $\alpha = 1$, (15) reduces to the Laplace estimator.

We consider up to two sets of FCMs—those belonging to what we call the reference set, $\mathcal{R}$, and those in the target set, $\mathcal{T}$. The reference set contains the FCMs responsible for modeling the conditioning string, i.e., the y of $C(x|y)$ or of $C(x\|y)$, whereas the target set of FCMs is used to represent x, when required.

Basically, the probability of the next symbol, x_{n+1}, is given by

$$P(x_{n+1}) = \sum_{k \in \mathcal{R}} P_r(x_{n+1}|x_{n-k+1}^n)\ w_{k,n}^r + \sum_{k \in \mathcal{T}} P_t(x_{n+1}|x_{n-k+1}^n)\ w_{k,n}^t, \tag{16}$$

where $P_r(x_{n+1}|x_{n-k+1}^n)$ and $P_t(x_{n+1}|x_{n-k+1}^n)$ are, respectively, the probability assigned to the next symbol by a FCM from the reference set and from the target set, and

where $w_{k,n}^r$ and $w_{k,n}^t$ denote the corresponding weighting factors, with

$$w_{k,n}^r \propto (w_{k,n-1}^r)^\gamma P_r(x_n|x_{n-k}^{n-1}) \quad \text{and} \quad w_{k,n}^t \propto (w_{k,n-1}^t)^\gamma P_t(x_n|x_{n-k}^{n-1}), \tag{17}$$

constrained to

$$\sum_{k\in\mathcal{R}} w_{k,n}^r + \sum_{k\in\mathcal{T}} w_{k,n}^t = 1. \tag{18}$$

Experimental results on authorship attribution

Authorship attribution is a classical classification problem and we use it here to illustrate the performance of the proposed approach. We use the same corpus of 168 English texts of known authorship described in [1]. The corpus was built from original texts obtained from Project Gutenberg (`http://www.gutenberg.org/`) and truncated to approximately the first 5,000 words [1]. The seven authors are: Sir Arthur Conan Doyle (26 texts), Andrew Lang (14 texts), B. M. Bower (25 texts), Charles Dickens (25 texts), Henry James (26 texts), Richard Harding Davis (26 texts), and Zane Grey (26 texts).

We replaced single or multiple line breaks with a single space and multiple spaces with a single space. The experiments intended to simulate the case where a text with unknown authorship is compared against known works of several authors. Therefore, for each target text, $t_i, i = 1, \ldots, 168$, the compression-based measures to the seven references, $r_j, j = 1, \ldots, 7$, were computed and the author corresponding to the smallest one was assigned.

In principle, the references should be built using all information available (although obviously excluding the text under classification). However, because in this data set one of the authors (Andrew Lang) has only about half of the texts of the other authors, and to avoid biasing this author negatively, we forced each reference to have approximately 390,000 characters. Apart from Andrew Lang, this was attained by using only about the first half of the characters available in the texts of all other authors.

Results using the NCD

We computed the NCD using several off-the-shelf, general purpose, compression tools, namely gzip, bzip2, lzma, ppmd and zpaq. The worst performance was attained by gzip, with 41 out of the 168 texts misclassified, whereas the best performance—only one misclassified text—was provided by lzma and zpaq (see Tables 1 and 2).

Table 1 shows, for each compressor, the number of misclassifications and the size of the compressed targets (each target compressed separately from the others). Hence, the values presented correspond to

$$S_{\mathrm{CT}} = \sum_{i=1}^{168} C(t_i),$$

where $C(t_i)$ denotes the number of bytes required by compressor C (gzip, ppmd, lzma, ...) to compress text t_i.

Table 1: Authorship misclassifications using the NCD and several data compressors. It is also shown the compressed size of all preprocessed texts (no linebreaks and only single spaces) using those data compressors. The "mxfcm(tar)" compressor uses an optimized mixture of FCMs over all the targets and over a parameter search space explained in the text. The "mxfcm(best)" compressor was obtained after testing the classification performance of several configurations and choosing the best one.

	Size (bytes)	Misclassifications
uncompressed	6,461,205	—
mxfcm(best)	2,965,030	1
gzip	2,635,064	41
mxfcm(tar)	2,600,534	8
lzma	2,480,973	1
bzip2	2,266,568	2
zpaq	2,079,993	1
ppmd	2,071,301	3
zpaq(max)	2,018,470	1

Table 2 shows the number of misclassifications and the size of the compressed references, for each compressor. More precisely, the compression values correspond to

$$S_{\mathrm{CR}} = \sum_{i=1}^{7} C(r_i),$$

where the r_i are the references.

It is interesting to observe that, although lzma was considerably worse than zpaq in compressing both the individual targets and the references, both compression algorithms attained the same classification performance. On the contrary, ppmd, which was the second best in terms of compression, did not behave as well in classifying the texts.

The poor performance of gzip is clearly due to the relatively small sliding window of its dictionary (only 32 Kbytes), which is insufficient for this problem. On the contrary, the default block size of bzip2 (900 Kbytes) was appropriate and hence its relatively good performance.

We also include results provided by three versions of our compressor based on mixtures of FCMs. Both "mxfcm(tar)" and "mxfcm(ref)" were found by searching all possible combinations of model orders ranging from zero to eight, and picking the combinations maximizing the compression of the targets and references, respectively. Version "mxfcm(best)" was obtained after testing the classification performance of those 511 configurations and choosing the best one.

Figure 1 shows a scatter plot of the number of misclassifications versus the compressed size of the targets. It shows that for the same compressed size the number of misclassifications may vary over a wide range. Figure 2 shows the number of misclassifications versus the compressed size of the references. As in the case of Fig. 1,

Table 2: Authorship misclassifications using the NCD and several data compressors. It is also shown the compressed size of all preprocessed references (no linebreaks and only single spaces) using those data compressors. The "mxfcm(ref)" compressor uses an optimized mixture of FCMs over all references and over a parameter search space explained in the text. The "mxfcm(best)" compressor was obtained after testing the classification performance of several configurations and choosing the best one.

	Size (bytes)	Misclassifications
uncompressed	2,727,843	—
mxfcm(best)	1,096,238	1
gzip	1,062,552	41
lzma	907,553	1
mxfcm(ref)	893,549	5
bzip2	818,456	2
ppmd	762,662	3
zpaq	730,414	1
zpaq(max)	703,525	1

an increase in compression ratio does not necessarily lead to better classification performance.

Results using the NCCD

The results in Tables 1 and 2 and the plots in Figs. 1 and 2 clearly show that the Normalized Compression Distance, even when combined with some of the best currently available text compressors, is not able to provide 100% correct classification. The results also show how difficult is to understand the impact of the compression gain in the number of misclassifications.

The NCCD also suffers from this same problem, although because it is derived directly from the NID, it might not be so severe. In fact, although we did not succeed in achieving 100% correct classification using our FCM-based encoder associated to the NCD, with the NCCD we were able to find some configurations of models leading to error free authorship attribution in the data set considered. Note that, in this case, the search space is much larger, because combinations of models for both the target and reference components have to be assessed, rendering exhaustive search much more demanding. Hence, since we were able to find some successful combinations by trial and error, in this case we did not perform a systematic search. One of such configurations comprised five FCMs for modeling the reference—orders one ($\alpha = 1/10$), two ($\alpha = 1/100$), four ($\alpha = 1/1000$), five ($\alpha = 1/1000$) and six ($\alpha = 1/10000$)—and seven for modeling the target—orders zero ($\alpha = 1$), one ($\alpha = 1/10$), two ($\alpha = 1/100$), three ($\alpha = 1/500$), four ($\alpha = 1/1000$), five ($\alpha = 1/1000$) and ($\alpha = 1/10000$). The mixture parameter was $\gamma = 0.1$.

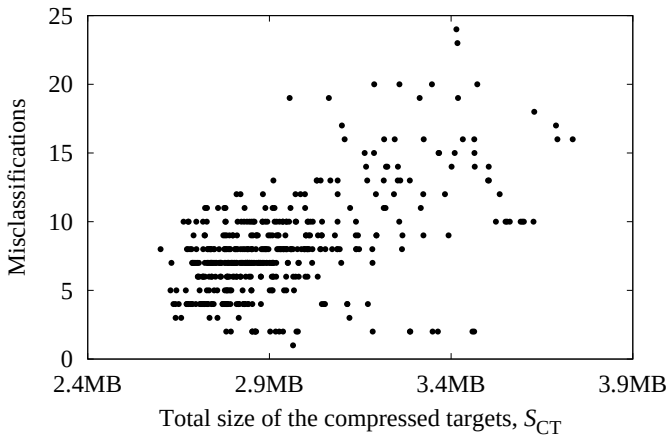

Figure 1: Number of wrong authorship attributions, using the NCD, as a function of the total compression size of all targets. Each dot in the graph represents a particular configuration of finite-context models in the encoder.

Results using the NRC

The NRC depends only on one term, $C(x\|y)$. Hence, in this case it was practical to perform a search over all possible combinations of models, ranging from order zero until order eight. Experimental evidence has been showing that deeper models generally benefit from a smaller α in the probability estimator defined in (15), hence giving more importance to past observations [15]. Therefore, whereas for models from zero-order until fourth-order we used $\alpha = 1$ (Laplace estimator), for order five we used $\alpha = 1/100$ and for orders six to eight $\alpha = 1/1000$. The mixture parameter was set to $\gamma = 0.1$.

Figure 3 shows how the total relative compression size of all references, given by

$$S_{\mathrm{RCR}} = \sum_{i=1}^{7} \sum_{j=1}^{7} C(r_i\|r_j), \tag{19}$$

correlates with the number of misclassifications in the authorship attribution problem. Note that S_{RCR} can be seen as an indicator of the modeling power of the particular configuration of FCMs used. Moreover, although it is straightforward to obtain such indicator for the NRC, since it depends only on a compression term, it is not so evident how to do it for measures that depend on several compression terms.

In the case addressed, for all 41 combinations of FCMs for which $S_{\mathrm{RCR}} \leq 5{,}892{,}933$ bytes, we achieved 100% correct authorship attribution. The combination having the second best value of S_{RCR} comprised just three FCMs (recall that in the NRC only the reference is modeled), with orders two, five and eight. In [1], the best result reported was 96.4% of correct classification (162 out of 168).

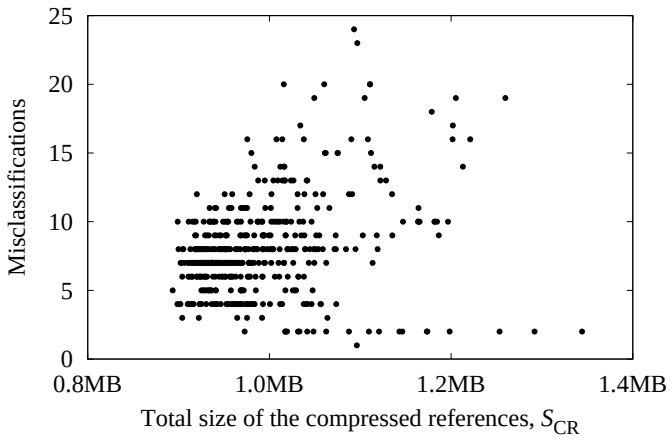

Figure 2: Number of wrong authorship attributions, using the NCD, as a function of the total compression size of all references. Each dot in the graph represents a particular configuration of finite-context models in the encoder.

Conclusion

Using the Normalized Relative Compression (NRC), we were able to obtain 100% correct classification performance in an authorship attribution problem. The NRC is also generally less computationally demanding than other compression-based measures. Notice that, to compute the NRC between several targets and a single reference, the workload can be even more reduced because it suffices to build each reference model once, and reuse it to compress each target in turn. Moreover, the NRC seems to be consistent regarding compression gain, i.e., improvements in the compression gain seem to provide consistent improvements in the classification performance, a characteristic not always present in other compression-based measures, and that might be related to the nonapproximability of the Normalized Information Distance [16].

Acknowledgments

This work was partially funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 305444 "RD-Connect: An integrated platform connecting registries, biobanks and clinical bioinformatics for rare disease research" and by National Funds through FCT - Foundation for Science and Technology, in the context of the project UID/CEC/00127/2013. The authors thank the authors of [1] for providing the corpus of English texts.

References

[1] M. Ebrahimpour, T. J. Putniņš, M. J. Berryman, A. Allison, B. W.-H. Ng, and D. Abbott, "Automated authorship attribution using advanced signal classification techniques," *PLoS ONE*, vol. 8, no. 2, p. e54998, Feb 2013.

[2] C. H. Bennett, P. Gács, M. L. P. M. B. Vitányi, and W. H. Zurek, "Information distance," *IEEE T Inform Theory*, vol. 44, no. 4, pp. 1407–1423, Jul 1998.

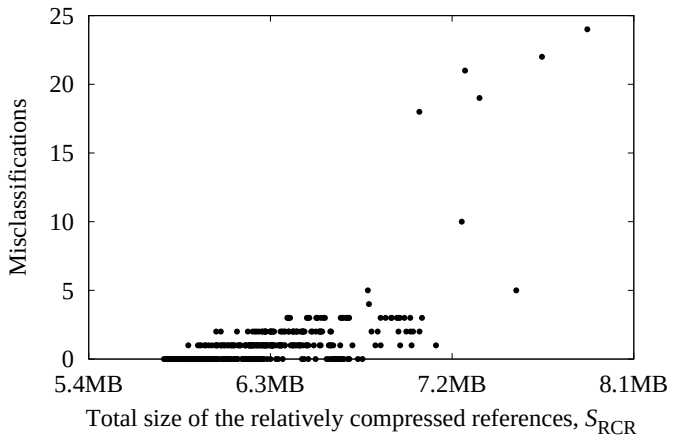

Figure 3: Number of wrong authorship attributions, using the NRC, as a function of the total relative compression size of all references. Each dot in the graph represents a particular configuration of FCMs in the encoder.

[3] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi, "The similarity metric," *IEEE T Inform Theory*, vol. 50, no. 12, pp. 3250–3264, Dec 2004.

[4] N. Nikvand and Z. Wang, "Generic image similarity based on Kolmogorov complexity," in *Proc of IEEE ICIP*, Hong Kong, Sep 2010, pp. 309–312.

[5] ——, "Image distortion analysis based on normalized perceptual information distance," *Signal, Image and Video Proc*, vol. 7, pp. 403–410, 2013.

[6] R. Cilibrasi and P. M. B. Vitányi, "Clustering by compression," *IEEE T Inform Theory*, vol. 51, no. 4, pp. 1523–1545, Apr 2005.

[7] T. Watanabe, K. Sugawara, and H. Sugihara, "A new pattern representation scheme using data compression," *IEEE T Pattern Analysis Machine Intelligence*, vol. 24, no. 5, pp. 579–590, May 2002.

[8] T. Watanabe, "Toward a compression-based self-organizing recognizer: preliminary implementation of PRDC-CSOR," *Pattern Recogn Lett*, vol. 34, pp. 1569–1576, 2013.

[9] J. Ziv and N. Merhav, "A measure of relative entropy between individual sequences with application to universal classification," *IEEE T Inform Theory*, vol. 39, no. 4, pp. 1270–1279, Jul 1993.

[10] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Rev Lett*, vol. 88, no. 4, pp. 048 702–1–048 702–4, Jan 2002.

[11] H. Cai, S. R. Kulkarni, and S. Verdú, "Universal divergence estimation for finite-alphabet sources," *IEEE T Inform Theory*, vol. 52, pp. 3456–3475, Aug 2006.

[12] M. Burrows and D. J. Wheeler, *A block-sorting lossless data compression algorithm*, Digital Systems Research Center, May 1994.

[13] F. M. J. Willems, Y. M. Shtarkov, T. J. Tjalkens, "The context-tree weighting method: basic principles," *IEEE T Inform Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[14] Z. Dawy, J. Hagenauer, and A. Hoffmann, "Implementing the context tree weighting method for content recognition," in *Proc of DCC*, Snowbird, Utah, 2004.

[15] A. J. Pinho, P. J. S. G. Ferreira, A. J. R. Neves, and C. A. C. Bastos, "On the representability of complete genomes by multiple competing finite-context (Markov) models," *PLoS ONE*, vol. 6, no. 6, p. e21588, 2011.

[16] S. A. Terwijn, L. Torenvliet, and P. M. B. Vitányi, "Nonapproximability of the normalized information distance," *J Computer and System Sci*, vol. 77, pp. 738–742, 2011.