COMPRESSION-BASED NORMAL SIMILARITY MEASURES FOR DNA SEQUENCES

Paulo J. S. G. Ferreira, Armando J. Pinho

Dept. Electronica, Telecom. e Informatica / IEETA
Universidade de Aveiro, Portugal
pjf@ua.pt / ap@ua.pt

ABSTRACT

Similarity measures based on compression assess the distance between two objects based on the number of bits needed to describe one, given a description of the other. Theoretically, compression-based similarity depends on the concept of Kolmogorov complexity, which is non-computable. The implementations require compression algorithms that are approximately normal. The approach has important advantages (no signal features to identify and extract, for example) but the compression method must be normal. This paper proposes normal algorithms based on mixtures of finite context models. Normality is attained by combining two new ideas: the use of least-recently-used caching in the context models, to allow deeper contexts, and data interleaving, to better explore that cache. Examples for DNA sequences are given (at the human genome scale).

Index Terms— Normalized compression distance, finite context models, LRU cache, interleaving, DNA sequences

1. INTRODUCTION

Measuring the similarity between two numeric signals or images or between two symbolic signals such as DNA sequences is an important problem. For numeric signals, the simplest approach is to fix a norm $\|\cdot\|$ and measure the size d of the difference of two signals using that norm, $d = \|x - y\|$. In this context, the L^2-norm is probably the one most often used. This approach works reasonably well when, for example, one signal is a noisy version of the other. Under more general circumstances, the direct application of a norm may yield meaningless results (imagine that one signal is a scaled version of the other, for example). Using norms in this way is of course also out of the question for symbolic sequences (in which the concept of "difference" is not even defined).

The ideal comparison method would give a meaningful indication of how similar two objects are, regardless of their geometry, orientation, scale, alphabet and other similar characteristics. A common approach is to extract a set of features from the objects and compare them. The difficulty associated with that approach is how to choose the set of features to extract.

There has been interest in similarity measures based on compression methods [1–8]. They rely on Kolmogorov complexity, or algorithmic entropy. The Kolmogorov complexity of a string of bits A is the minimum size of a program that produces A and stops. A repetitive structure can be described by a small program ("print zero, repeat for 10,000 times"), the size of which scales like the logarithm of the size of A, that is, $\log |A|$. By contrast, for a complex pattern A there might be no better program than "print A", which scales with $|A|$, indicating high complexity. The drawback of Kolmogorov complexity, usually denoted by $K(A)$, is its non-computable character, which forces the use of approximations that set only upper bounds on the complexity.

Lossless compression algorithms provide natural ways to approximate the Kolmogorov complexity. Given an encoder and the appropriate decoder, the bitstream produced by a lossless compression algorithm determines the original object. The number of bits required to represent the decoder and the bitstream itself can be viewed as an estimate of the Kolmogorov complexity of the object.

Any compression-based similarity measure (for DNA sequences or for any other data) is associated with a compression method. In a sense, the selection of the compression method resembles the norm selection step in the naive method $d = \|x - y\|$. Different compression methods yield different similarity indexes. Nevertheless, all of them can be regarded as approximations to the Kolmogorov complexity of the object, and all of them yield upper bounds for it.

These ideas lead to the normalized information distance

$$\mathrm{NID}(A, B) = \frac{\max\{K(A|B), K(B|A)\}}{\max\{K(A), K(B)\}},$$

where $K(A|B)$ denotes the conditional Kolmogorov complexity of A given B. This is the size of the shortest program that prints A and halts, when it is fed the input B. The NID is

Work partially supported by the European Fund for Regional Development (FEDER) through the Operational Program Competitiveness Factors (COMPETE) and by the Portuguese Foundation for Science and Technology (FCT), in the context of projects FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011) and Incentivo/EEI/UI0127/2013.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

related to the normalized compression distance [9, 10]

$$\text{NCD}(A,B) = \frac{C(AB) - \min\{C(A), C(B)\}}{\max\{C(A), C(B)\}},$$

where $C(A)$ denotes the number of bits required by the compressor C to describe A, and AB denotes the concatenation of A and B. The NCD yields a number in the interval $[0, 1]$, with values close to zero indicating strong similarity and values close to one strong dissimilarity. Note that

$$\text{NCD}(A,A) = \frac{C(AA) - C(A)}{C(A)}.$$

The compression method is called **normal** if it generates essentially the same number of bits when compressing AA (the concatenation of A with A) and when compressing A alone (see also [7, 11]). Lempel-Ziv based compressors are approximately normal but do not perform well on DNA sequences or images. On the other hand, the best performing image or DNA compression algorithms are not normal. Under the normality assumption, $\text{NCD}(A, A)$ should tend to zero as the size of A increases.

The goal of this paper is to describe normal compression-based similarity measures for DNA sequences, using compressors based on mixtures of finite-context models. To achieve normality, we implemented the finite-context models using least-recently-used caches. This leads to very good performance for DNA sequences the size of the human genome. It is also essential to allow deeper model depths and control memory usage. This was complemented with data interleaving, to allow the compressor to take full advantage of the cache. These ideas complement each other — one does not work well without the other. We present results for large DNA sequences (two human genomes).

2. MAIN IDEAS AND RATIONALE

The probabilistic models that we will use employ mixtures of finite context models. A finite context model assigns a probability estimate to the next symbol s, given the most recent past outcomes $c_1, c_2, \ldots, c_n$. The estimator is

$$P(X = s | c_1, c_2, \ldots, c_n) = \frac{N(s, c_1, c_2, \ldots, c_n) + \alpha}{N(c_1, c_2, \ldots, c_n) + \alpha k}, \quad (1)$$

where $N(s, c_1, c_2, \ldots, c_n)$ is the number of times that the symbol s has been seen having $c_1, c_2, \ldots, c_n$ as the conditioning context, $N(c_1, c_2, \ldots, c_n)$ is the total number of times that the context has been seen and k is the size of the alphabet. The parameter α allows balancing between the maximum likelihood estimator and an uniform distribution. The value $\alpha = 1$ corresponds to the well-known Laplace estimator. The impact of α is usually very small unless n, the context depth, is large (this will be discussed later).

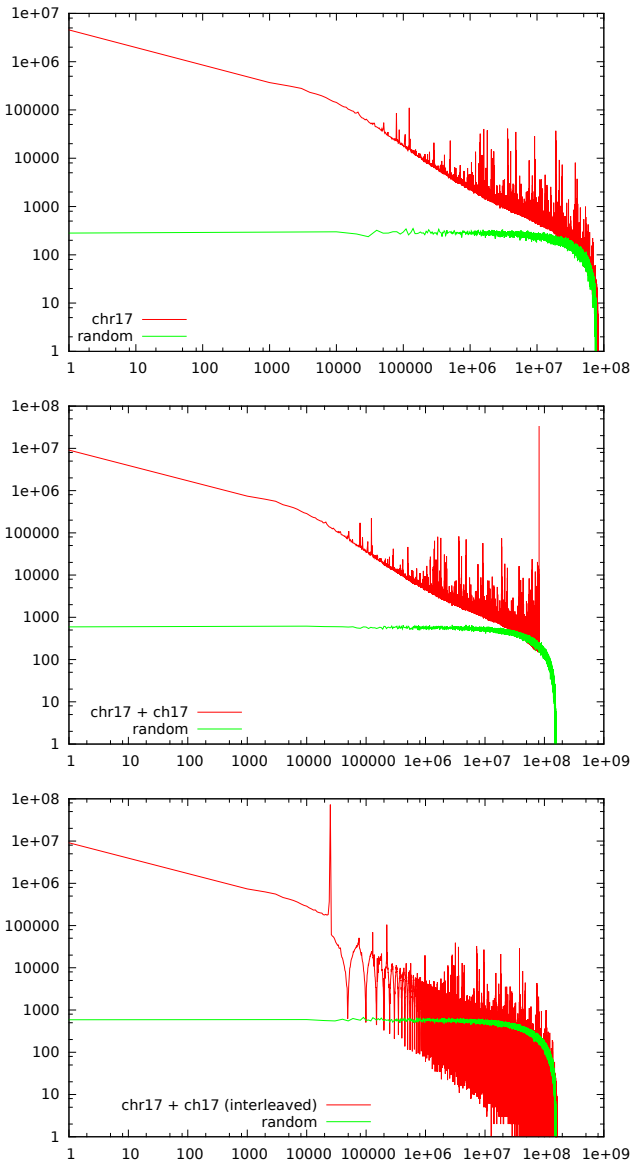

Fig. 1. Top: Histogram of the distances between successive repetitions for all strings of length 14 in the human chromosome 17. Middle: Same, for the human chromosome 17 concatenated with itself. Bottom: Same, for chromosome 17 interleaved with itself. Also shown are similar histograms for random strings of the same length.

Consider a model that includes several finite context models and let the maximum depth be n. If the alphabet has size $k = 4$ (as in many problems involving DNA sequences), the deepest model alone requires a fixed size table with 4^n entries, indexed by the context itself. The exponential memory required to implement very deep models renders the approach based on fixed size tables unfeasible.

A better solution is to use hash tables. A hash table uses

the context as the key and stores only contexts that have occurred. If the sequence is short, the hash table will work well even for very deep contexts, because only a small fraction of the 4^n possible contexts are likely to occur. Thus, the hash table will only need to store a small fraction of the 4^n possible contexts, a feasible task.

Unfortunately, in many applications the data sequences are not short. In bioinformatics, sequences of size above 10^9 are common. In that case, the hash table will slowly expand as the sequence is read, leading to poor performance and ultimately exhausting the available memory.

Since finite context models explore repetitions of patterns found in the data, it is of interest to investigate how the repetitions are distributed along the sequences. Fig. 1 (top) shows the histogram of the distances between repetitions for all strings of length 14 found in the human chromosome 17. It is seen that a large fraction of the repetitions occur at comparatively small distances (thousands or tens of thousands in a sequence with tens of millions of symbols). A similar behavior was observed in many other cases.

Our observations suggest the use of least-recently-used (LRU) caches for the context models. An LRU cache works in a way that resembles a hash table, but has a limit on the number of entries. When the limit is reached, the oldest item is discarded to make room for the new item. In DNA data, most repetitions occur at comparatively short distances. Thus, an LRU cache can provide good probability estimates *and* a hard limit on memory usage. This makes very deep contexts accessible for experiment.

The LRU cache by itself will not lead to a normal compression method. To reach that goal we need to understand the problem in more depth.

Since our goal is to measure distances using information theoretic tools, it is necessary to examine the distribution of repetitions in pairs of sequences. The usual way of combining pairs of sequences, say A and B, is to concatenate them — see the definition of NCD(A, B). To obtain normal methods it is necessary to keep NCD(A, A) small. This depends on the term $C(AA)$, and so we need to consider the concatenation of a sequence with itself.

Fig. 1 (middle) shows the repetition histogram for chromosome 17 concatenated with itself. The result, in fact, could have been predicted: the concatenation gives rise to a new set of repetitions which occur at a very long distance (the length of the chromosome itself). The compression of the concatenated sequence, and therefore the property of normality, will be difficult to achieve because a very important fraction of the repeats found in the data will occur very far from each other.

Our solution is the following: instead of combining A and B using concatenation, we interleave them. This is still consistent with the term $K(A|B)$ in the NID and, as seen on Fig. 1, it leads to a strikingly different and much more favorable distance distribution. Interleaving is appropriate for processing with short-term memory. It is impossible to explore the redundancy present in concatenated sequences unless the encoders have very long-term memory.

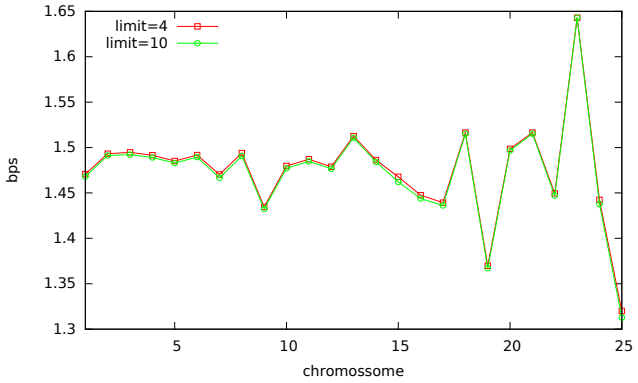

Fig. 2. Cache size impact for a mixture of models of depth 3, 7, 11 and 15, with cache limits of 6 and 10 million entries, tested on the entire human genome (including mitochondrial DNA). The performance is very little affected by the cache size, due to the characteristics of the data.

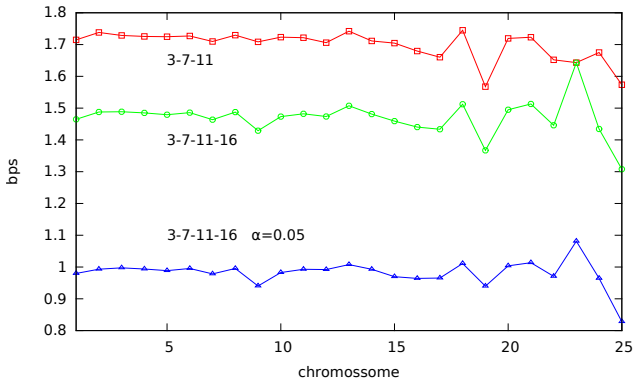

Fig. 3. Comparison of models that use only shallower (3-7-11) or shallower and deeper contexts (3-7-11-16) on the human genome. The addition of the model of depth 16 greatly improves performance. For deep contexts, reducing α also improves performance. Thanks to the cached model, there is no need to store the 4^{16} table entries of the deeper model (in fact, deeper models could have been used, even in portable computers).

3. RESULTS AND CONCLUSION

Since we have decided to base the deeper models in LRU caches, it is important to understand how the cache size impacts the compression results. This, in turn, determines the ability of the models to discriminate distances between sequences. We have found that for DNA data, in which most

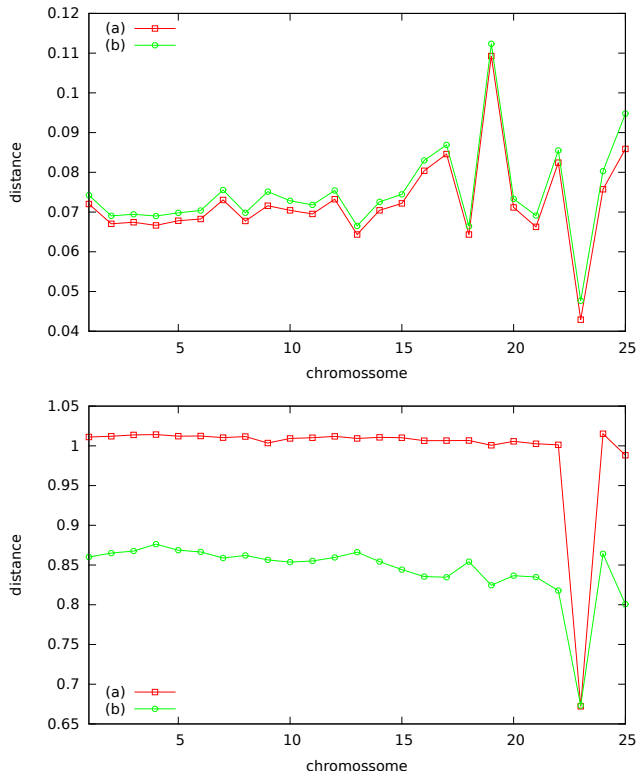

Fig. 4. Top: normalized compression distances obtained using mixtures of finite context models with LRU caching and interleaving. The curve (a) corresponds to the distances NCD(A, A) between each chromossome and itself, and the curve (b) to the distances between two chromossomes in two different assemblies of the human genome. We are comparing close data, but even so the distances reveal the difference. Models used: 4, 8, 12 and 18, with α set to 1, 1, 0.05 and 0.02, respectively. Bottom: distances obtained using models 4, 8 and 12 only, without interleaving. In this case the results are meaningless. The method is far from normal, since $C(AA)$ is very different from $C(A)$.

repetitions occur at short distances (as seen in Fig. 1), the cache limit has very little impact on the performance. Cache limits of 4 million entries or so are more than enough to obtain consistent performance. Fig. 2 shows results for cache limits of 4 and 10 million entries, respectively. It is clear that increasing the cache size has negligible positive impact.

Our cache-based implementation renders much deeper contexts possible, and these deeper contexts do improve performance, as shown in Fig. 3. The importance of the parameter α of the estimator (1) grows with the model depth. When the context depth is small, the counts in (1) are large and dominate the numerator and denominator. For deeper contexts, α has a noticeable impact, as Fig. 3 clearly shows.

The combination of caching, interleaving and deep context models yields a compression-based distance that is almost normal. This is demonstrated in Fig. 4 (top), which shows NCD(A, A) for each chromossome in the human genome (including mitochondrial DNA). As seen, most of the distances lie below 0.08, a value at least one order of magnitude smaller than the values obtained in Fig. 4 (bottom), i.e. without caching and interleaving. Also shown is NCD(A, B) where A and B are the same chromossomes in two different assemblies of the human genome. Such pairs are still very similar, but as seen in the figure the method is able to discriminate the differences between them clearly.

Fig. 4 (bottom) is an example of what happens when the ideas proposed in this work are not used. First, without LRU caches a context depth of 18 becomes unfeasible (it would require a table size of 2^{36}). Second, without interleaving, $C(AA)$ yields significantly more bits than $C(A)$ and NCD(A, A) is large (typically above 0.80). Recall that the NCD is supposed to yield a number in the range $[0, 1]$, with values close to 1 indicating strong unsimilarity. The result is clearly meaningless.

To conclude, we have carefully considered the design of normalized compression distances for DNA sequences of size above 10^9. Such measures are useless unless they are normal. This requires an underlying compressor, or statistical model, able to represent AA using about the same number of bits as A. We used probabilistic models based on mixtures of finite context models to explore the repetition-rich content of DNA sequences. An analysis of the distribution of distances between repetitions showed that most repeats lie at short distances. As a result, we implemented the finite models using LRU caches. To circumnvent the intrinsic short range memory of the LRU cache, we approached terms such as $K(X|Y)$ in a new way: sequences were combined using interleaving rather than concatenation. This lead to a normal compressor (to within 5-7%) able to discriminate even the small differences between two different assemblies of the same genome.

4. REFERENCES

[1] N. Tran, "The normalized compression distance and image distinguishability," in *Human Vision and Electronic Imaging XII — Proc. of the SPIE*, Jan. 2007, p. 64921D.

[2] I. Gondra and D. R. Heisterkamp, "Content-based image retrieval with the normalized information distance," *Computer Vision and Image Understanding*, vol. 111, pp. 219–228, 2008.

[3] J. Perkiö and A. Hyvärinen, "Modelling image complexity by independent component analysis, with application to content-based image retrieval," in *Proc. of the Int. Conf. on Artificial Neural Networks, ICANN 2009*, Limassol, Cyprus, 2009.

[4] J. Mortensen, J. J. Wu, J. Furst, J. Rogers, and D. Raicu, "Effect of image linearization on normalized compres-

sion distance," in *Signal Processing, Image Processing and Pattern Recognition*, D. Slezak, S. K. Pal, B.-H. Kang, J. Gu, H. Kuroda, and T.-H. Kim, Eds. 2009, vol. 61 of *Communications in Computer and Information Science*, pp. 106–116, Springer Berlin Heidelberg.

[5] A. R. Cohen, C. S. Bjornsson, S. Temple, G. Banker, and B. Roysam, "Automatic summarization of changes in biological image sequences using algorithmic information theory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1386–1403, Aug. 2009.

[6] D. Cerra, A. Mallet, L. Gueguen, and M. Datcu, "Algorithmic information theory-based analysis of earth observation images: an assessment," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp. 8–12, Jan. 2010.

[7] Armando J. Pinho and Paulo J. S. G. Ferreira, "Image similarity using the normalized compression distance based on finite context models," in *Proceedings of the IEEE International Conference on Image Processing, ICIP-2011*, Brussels, Belgium, Sept. 2011.

[8] S. P. Garcia, J. M. O. S. Rodrigues, S. Santos, D. Pratas, V. Afreixo, C. A. C. Bastos, Paulo J. S. G. Ferreira, and Armando J. Pinho, "A genomic distance for assembly comparison based on compressed maximal exact matches," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, July 2013.

[9] C. H. Bennett, Peter Gacs, Ming Li, P. M. B. Vitanyi, and W. H. Zurek, "Information distance," *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1407–1423, July 1998.

[10] Ming Li, Xin Chen, Xin Li, Bin Ma, and P. M. B. Vitanyi, "The similarity metric," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.

[11] Manuel Cebrián, Manuel Alfonseca, and Alfonso Ortega, "Common pitfalls using the normalized compression distance: What to watch out for in a compressor," *Communications in Information and Systems*, vol. 5, no. 4, pp. 367–384, 2005.