A Similarity Measure using Smallest Context-free Grammars

Daniele Cerra[1] and Mihai Datcu[1,2]
[1] German Aerospace Center (DLR), Remote Sensing Technology Institute, 82234 Wessling, Germany
[2] Télécom Paris, 46 rue Barrault, 75634 Paris, France
Email: {daniele.cerra, mihai.datcu}@dlr.de

Abstract – This work presents a new approximation for the Kolmogorov complexity of strings based on compression with smallest Context Free Grammars (CFG). If, for a given string, a dictionary containing its relevant patterns may be regarded as a model, a Context-Free Grammar may represent a generative model, with all of its rules (and as a consequence its own size) being meaningful. Thus, we define a new complexity approximation which takes into account the size of the string model, in a representation similar to the Minimum Description Length. These considerations result in the definition of a new compression-based similarity measure: its novelty lies in the fact that the impact of complexity overestimations, due to the limits that a real compressor has, can be accounted for and decreased.

1. Introduction

The Kolmogorov complexity $K(x)$ of an object x, defined as the length of the shortest description of x, represents an ideal quantification of the intrinsic information content of a string [1]. This idea, being uncomputable, could be labelled as exquisitely theoretical: yet, practical uses relying on its approximations have been described. Li and Vitányi proposed the most widely used computable approximation of $K(x)$, the size $C(x)$ of the compressed version of x, obtained by means of any off-the-shelf compressor, considering the shortest description of a string as the lower bound for what a real compressor can achieve [2]. This allowed the definition of the Normalized Compression Distance (NCD), a similarity metric successfully applied to diverse datasets [3]. Nevertheless, other compression-based techniques had been described before the definition of NCD and successfully employed to perform unsupervised clustering and classification. The first important work in this field was the definition by Benedetto et al. in [4] of an intuitive relative entropy distance between two isolated strings, with the link between this notion and algorithmic information theory being established in [3]. A few months later, Watanabe et al. used a different approach based on the direct extraction of dictionaries from representative objects, the Pattern Representation using Data Compression [6]. These dictionaries are used in a second step to compress general data, previously encoded into strings, and estimate the amount of information shared by the chosen objects. The dictionary containing the relevant information of an object, or class of objects, may be in turn regarded as a model for that object [7]: we propose then a new estimation for $K(x)$ which takes into account not only the size of the compressed file $C(x)$, but also the complexity of the dictionary $D(x)$ used to compress x, in a two-part representation with a formalism similar to Rissanen's concept of Minimum Description Length (MDL) [8]. To estimate the complexity of the model we use as dictionary an approximation of the smallest Context Free Grammars (CFG), regarded as a generative model of the string: this allows defining a new compression-based similarity measure, in which the typical complexity overestimations introduced by real compressors are accounted for and decreased. Furthermore, the separation between the complexities of model and data given the model allows tuning the complexity estimation, to focus on the data structure represented by the model rather than on the full data, and may be adopted to separate meaningful information from noise.

The paper is structured as follows. Section 2 introduces the ideas of Kolmogorov complexity and compression-based similarity measures. The proposed complexity estimation using CFG is introduced in section 3, while section 4 presents some applications to DNA genomes and satellite images. We conclude in section 5.

2. Preliminaries

2.1. Kolmogorov Complexity and Normalized Information Distance

The Kolmogorov complexity of a binary string is the size in bits (binary digits) of the shortest self-delimiting program *q* used as input by a universal Turing machine to compute *x* and halt:

$$K(x) = \min_{q \in Qx} |q|, \tag{1}$$

with *Qx* being the set of instantaneous codes that generate *x* [1]. One interpretation of *K*(*x*) is as the quantity of information needed to recover *x* from scratch. So, strings presenting recurring patterns have low complexity, whereas the complexity of random strings is high and almost equals their own length. It is important to remark that *K*(*x*) is not a computable function of *x*. The *joint complexity* *K*(*x*,*y*) is defined as the length of the shortest program which outputs *x* followed by *y*. An important application of these notions is the ultimate estimation of shared information between two objects: the Normalized Information Distance (*NID*) [2]. The *NID* minimizes any admissible metric, and is proportional to the length of the shortest program that computes *x* given *y*, as well as computing *y* given *x*. The distance computed on the basis of these considerations is, after normalization,

$$NID(x,y) = \frac{K(x,y) - \min\{K(x),K(y)\}}{\max\{K(x),K(y)\}}. \tag{2}$$

The *NID* is a metric, so its result is a positive quantity *r* in the domain $0 \le r \le 1$, with $r = 0$ *iff* the objects are identical and $r = 1$ representing maximum distance between them.

2.2. Normalized Compression Distance

Since the complexity *K*(*x*) is not a computable function of *x*, Li and Vitányi define a suitable approximation by considering it as the size of the ultimate compressed version of *x*, and a lower bound for what a real compressor can achieve [3]. This allows approximating *K*(*x*) with *C*(*x*) = *K*(*x*) + *k*, i.e. the length of the compressed version of *x* obtained with any off-the-shelf lossless compressor *C*, plus an unknown quantity *k*: the presence of *k* is required by the fact that it is not possible to estimate how close to the lower bound represented by *K*(*x*) this approximation is. To clarify this consider two strings *b* and *p* having the same length *n*, where the former is the random output of a Bernoulli process, and the latter represents the first *n* digits of the number π. The quantity *K*(*p*) will be much smaller than *K*(*b*), since exists a program of length *K*(*p*) << *n* that outputs π, while a program that outputs *b* will have a length close to *n*: *K*(*p*) << *K*(*b*). Nevertheless, a standard compressor will not be effective in representing neither *b* nor *p* in a compact way, so $C(p) \cong C(b) \cong n$. This example shows how *k* ranges from a negligible value to a strong bias for the complexity estimation. The equation (2) can be estimated by the Normalized Compression Distance (NCD) as follows:

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

where $C(x,y)$ is an approximation of the joint Kolmogorov complexity $K(x,y)$ and represents the size of the file obtained by compressing the concatenation of x and y . The *NCD* can be explicitly computed between any two strings or files x and y with a basically parameter-free approach. It has to be remarked that the approximation of $K(x)$ with $C(x)$ is data dependant, being some compressors more efficient than others on certain data types. Therefore, the choice of the compressor is not a free parameter in itself: for each dataset a compression algorithm able to fully exploit the redundancies in that kind of data should be adopted [9]; better compression, in fact, means better approximation of the Kolmogorov complexity. Performance comparisons for general compression algorithms have shown that this dependence is generally loose [10], but increases when compressors for specific data types are employed: for example, good results have been achieved by using image compressors such as Jpeg2000 in applications to satellite images [11], by exploiting the vertical spatial information within the images intrinsically within the computation of the information distance, whereas a compressor for general data is limited since it linearly scans the data, failing at capturing the full information about the spatial distribution of the pixels.

About other fields of applications, the choice of ad hoc compression algorithms for sequences of DeoxyriboNucleic Acid (DNA) and RiboNucleic Acid (RNA) yields better results for unsupervised clustering of genomes belonging to different species [13] or for the estimation of the information content in the sequences [14].

2.3. Kolmogorov Complexity and Minimum Description Length

The Minimum Description Length (MDL) of an object x is the shortest combination of the description for one of the available models describing x and the description of x given that model [8]. Relations between MDL and algorithmic complexity have been considered in many works, see for example [15-17]; the Kolmogorov complexity of an object x may be regarded as the MDL consisting of the size of the representation of the best model Mx of x, plus the length of the description of x according to Mx, but in Kolmogorov's frame the model and the data given the model are indissolubly joint and the description of Mx is implicit. So, for every x, we have $K(x) \leq MDL(x)$. Recently Kolmogorov complexity has been also considered to be a way to formalize the celebre empiric rule known as Occam's razor [18], which basically states that, if many explanations exist for a given phenomenon, one should always choose the simplest one. An event x is likely to have been generated by its simplest model as well in MDL (if the size of the representation of x does not vary), which is in turn assimilated to the shortest program which outputs x in Kolmogorov's frame. A formal two-part model for complexity in terms of Kolmogorov bringing the notions of algorithmic complexity and MDL even closer is introduced by Guegen and Datcu in [19]. Connecting these topics to compression-based similarity measures is the MDL-Compress algorithm,

```
1. Identify symbols $a$ and $b$ such that $ab$ is the most frequent pair of adjacent symbols
   in the dataset previously encoded into a string. If no pair appears more than once, stop.
2. Introduce in $R$ a new rule $A \rightarrow ab$ .
3. Replace all occurrences of $ab$ with $A$ .
4. Repeat from step 1.
```

Fig. 1. Pseudo-code to generate the set of rules *R* constituting an approximation of the smallest Context-Free Grammar *G*(*x*) related to a string *x*.

applied to network security in [20].

3. Complexity Approximation with Grammars

3.1. Complexity Approximation using Smallest CFG

A two-part complexity approximation for a string *x* would consist of the complexity of a model for a string, plus the complexity of the data given the model. Considering then a dictionary *D*(*x*) as extracted in [6] as a data model for *x* [7], and keeping in mind the formalism of MDL, we could assimilate the complexity of *x* to the following two-part representation:

$$K(x) \approx C(x) + D(x) \ . \tag{4}$$

Nevertheless, it is hard to estimate (4) if we consider the technique described in [6], since dictionaries are extracted with a LZ-family compressor [12], and contain redundancies (having the prefix-closure property) and elements which are not relevant (never used in the compression step). From a completely random string *r*, with a large enough alphabet, in which a pattern is never repeated twice, we would extract a dictionary *D*(*r*), with a size |*D*(*r*)|>0 and a certain complexity, that would not be able to compress at all *r*, from which it was extracted. The estimated complexity would be then clearly overestimated, if we attain ourselves to (4). We are interested then in computing the smallest dictionary *D*(*x*) useful in compressing *x*, which as a consequence is empty if *x* cannot be compressed, and in estimating its complexity.

The solution is to consider an approximation of the smallest grammar *G*(*x*) which contains all the relevant patterns within *x*. This idea comes from the link between Kolmogorov complexity and the smallest CFG generating a string. Both of them are not computable and represent the most compact representation of the object that can be achieved: the problem of finding the smallest grammar which generates a string is NP-hard, and this problem is assimilated to the computation of the Kolmogorov complexity of the string itself [21] [22]. The CFG generating *x* can be regarded as a generative model for *x*, representing a compact representation of the regularities within the data, and its set of production rules *R* may be regarded as a list of entries in a dictionary. In this work we use an approximation for smallest context-free grammars introduced by Larsson and Moffat in [23], described by the pseudo-code in Fig. 1. Finally, we can introduce our complexity approximation based on compression with smallest grammars *Cg*(*x*), defined as follows:

$$Cg(x) = \begin{cases} N \quad ,if \quad N \leq 1 \\ C_x + \left(1 - \dfrac{\log_2 N}{\log_2 C_x + |G(x)|}\right) |G(x)|, o.w. \end{cases} \tag{5}$$

where C_x is the number of elements of the object *x*, initially composed by *N* elements, after being compressed by *G*(*x*). The latter contains a set of production rules *R* which may be regarded as the smallest dictionary of *x*, and has a size |*G*(*x*)| equal to the total number of rules in *R*. It is important to notice that the complexity estimation in the second term of the equation decreases as the compression power of its production rules grows. Thus, the complexity overestimation due to the limits that a real compressor has is accounted for and decreased, when the possibility of compactly representing *x* is found. This approximation for complexity gives by definition $Cg(x) = 0$ if x is the empty string, and has the following characteristics.

Lemma 1. The second term of the sum, representing the complexity of the data model, is bounded between 0 and $|G(x)|$ - This corrects the overestimated size of $G(x)$ for a very simple object x, which we assume could be described in a more compact form than its compression with an approximated smallest grammar: in other words, this term accounts for complexity overestimations due to the limits that a real compressor has. When the grammar grows in size and complexity and is not very effective at compressing x, the second term approaches its limit $|G(x)|$.

Proof – It has to be shown that the factor in parentheses lies in the interval [0,1). To state that it is upper bounded by 1 it is sufficient to notice that all the values in the term $\frac{\log_2 N}{\log_2 C_x + |G(x)|}$ are positive, and when this term goes to 0 the upper bound is approached, but never reached since $\log_2 N$ is always strictly positive (note that we are in the case $N > 1$). Showing that the lower bound is 0 is equivalent to state that the following holds:

$$\log_2 N \le \log_2 C_x + |G(x)|. \tag{6}$$

In the limit case of $|G(x)| = 0$, we have that $C_x = N$, the equation above is true and the lower bound of 0 is reached. If we add any production rule to the grammar, the term to the right in (6) does not decrease; consider that the best compression that we can achieve, after the introduction of a single new rule in R, is reducing x to a size $C_x / 2$, in the limit case of a pattern composed of two symbols repeated for the whole length of the string, i.e. $N/2$ times: thus, after adding such rule to the grammar, the term to the right of the equation doesn't change, since $\log_2 C_x$ decreases by 1, while $|G(x)|$ increases by 1. If instead we add a rule which is not optimally compressing x, $\log_2 C_x$ decreases by a quantity $\Delta < 1$, while $|G(x)|$ still increases by 1. So the term to the right in (6) is lower bounded by $\log_2 N$. Note that, as $|G(x)|$ grows with rules that are not optimal in compressing x, the term in parenthesis in (5) approaches 1, avoiding to give a strong "discount" to the complexity of the grammar adopted. The complexity of the data model does not derive directly from the space needed to store the set of production rules contained in $G(x)$: in fact, it is always smaller.

The fact that a single rule which compresses x to a size of $N/2$ does not increase our complexity estimation is justified by another consideration: we could have reduced to $N/2$ the size of x by modifying our coding of the source, generating a string x' which would be as complex as x.

Lemma 2. Cg(x) is upper bounded by the size N of x - The size N of x is the same quantity bounding the Kolmogorov complexity $K(x)$ for a maximally random string.

Proof – Consider the limit case of x being maximally random and not compressible at all: it will produce an empty grammar, erasing the second term of the sum in (5), i. e. $C_x = N$ and $|G(x)| = 0$. If on the contrary x can be compressed, each rule added to the grammar will decrease C_x of at least two, and increase the second term of the equation of at most one. In any case, the sum will never exceed *N*.

Lemma 3. Cg(x) is not a monotone function of x – It does not hold the property $Cg(xy) \ge Cg(x),\ \forall x, y$.

Proof – It is enough to provide a counterexample for which $Cg(xy) \ge Cg(x)$ is not true. Suppose to have a binary string $s = \{000\}$, with $|G(s)| = 0$, $|Cs| = 3$, and a complexity

$Cg(s) = 3$, another string $s' = \{0\}$, and the concatenation $ss' = \{0000\}$, with $|G(ss')| = 1$, $|Css'| = 2$, and $Cg(ss') = 2$. So the complexity decreases: this is because the size of ss' is now a power of 2, allowing better compressibility (see also lemma 4). Even if the monotonicity property is not respected, it may be argued that a very simple binary string with a size which is a power of 2 would be more easily built by a program running in a universal Turing machine. Also, this property could be satisfied by changing the way in which the grammar is built, allowing for a dynamic representation of patterns of different sizes, or different encodings for long runs of the same symbols-sequences: this would require a more complex approximation of the CFG that is outside the scope of this paper. This example also indicates that our complexity approximation works better on long strings, since on the long run these differences become negligible.

Lemma 4. Complexity of a maximally redundant object x does not increase with its size N, if $\{\ \exists p \mid 2^p = N\ \}$ – When x is maximally redundant, $Cg(x)$ is constant if its size N is a power of two.

Proof – Consider, without loss of generality, a string $x = \{01\}$ with initial complexity $Cg(x) = 2$. If we append to x its copy, we obtain $x^2 = \{01\}^2$ which will generate a rule in the grammar $G(x)$ of the kind $A -> 01$: after compressing x^2 with $G(x)$ we still have $Cg(x^2) = 2$. Appending again x to x^2 will result in a new rule $B -> AA$, which again will yield $Cg(x^3) = 2$. This process can be repeated over and over. It has to be noticed that, if the size N of x doesn't satisfy $\{\ \exists p \mid 2^p = N\ \}$, $Cg(x)$ increases of a small quantity; it should be considered that a simpler algorithm is required in a low-level language to output a string which size is a power of 2.

Lemma 5. The estimated complexity Cg(x) of an object x almost equals the complexity Cg(xx) of the concatenation of x with itself - the complexity of the concatenation of an object x with itself, or with an identical object y, is very close to the complexity of x considered separately.

Proof – Merging two identical objects x and y will create a repeated sequence: so, after substitution of the most recurring patterns, each subset of x and y will have a counterpart in the other object and will be added as a pattern to the CFG. Substitution of each of these sequences, which occur only twice in the whole xy, will make in (5) decrease the first term of the sum by one for each substitution, and bring the second term close to $|G(xy)|$, at the same time increasing it by 1 for each rule, balancing the decrement of the first term. An important consequence of this property is that $Cg(xy) \cong Cg(x) \cong Cg(y)$, if $x = y$.

Verification – To confirm empirically the validity of this property we carried out some experiments on 200 different strings, considering their complexity after concatenation of the objects with themselves: the average absolute difference between $Cg(x)$ and $Cg(xx)$ was less than 0.1%, which confirmed our hypothesis.

3.2. NCD using Complexity Estimated with Smallest CFG

We define the Normalized Compression Distance using Grammars (*NCDG*) by using a modified version of (3), where $C(x)$ is substituted by $Cg(x)$ as defined in (5):

$$NCDG(x,y) = \frac{Cg(xy) - min\{Cg(x),Cg(y)\}}{max\{Cg(x),Cg(y)\}}. \tag{7}$$

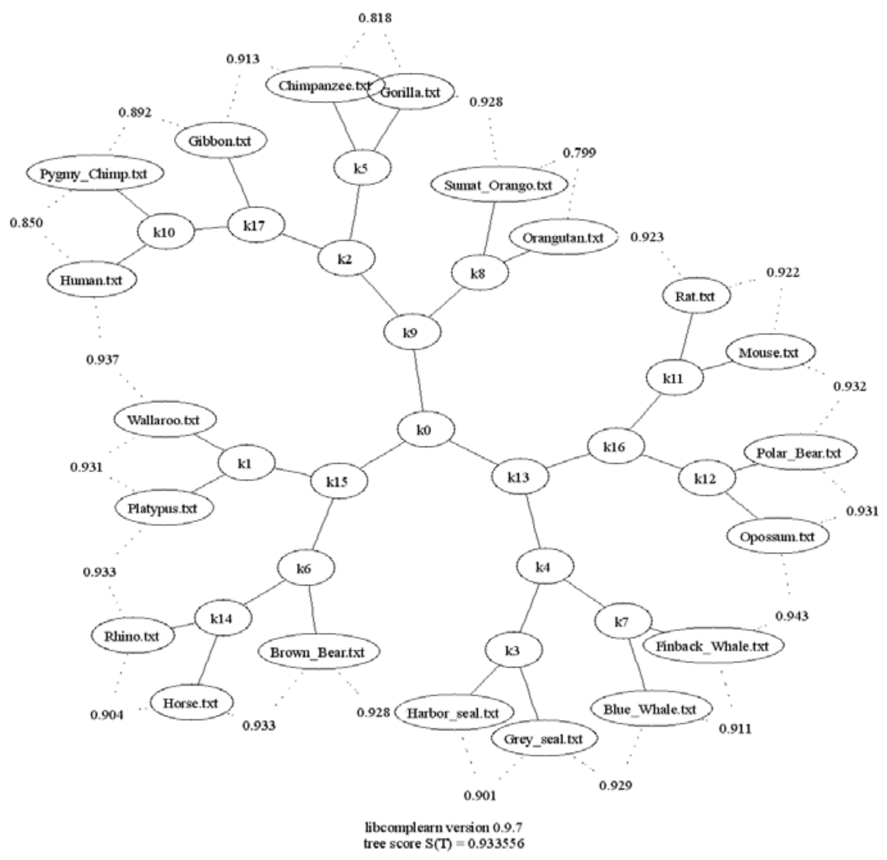

Fig. 2. Hierarchical clustering on DNA mitochondrial genomes using NCD. Polar Bear and Brown Bear genomes are not considered to be similar at all, with the former being placed among a group of genomes belonging to rodents.

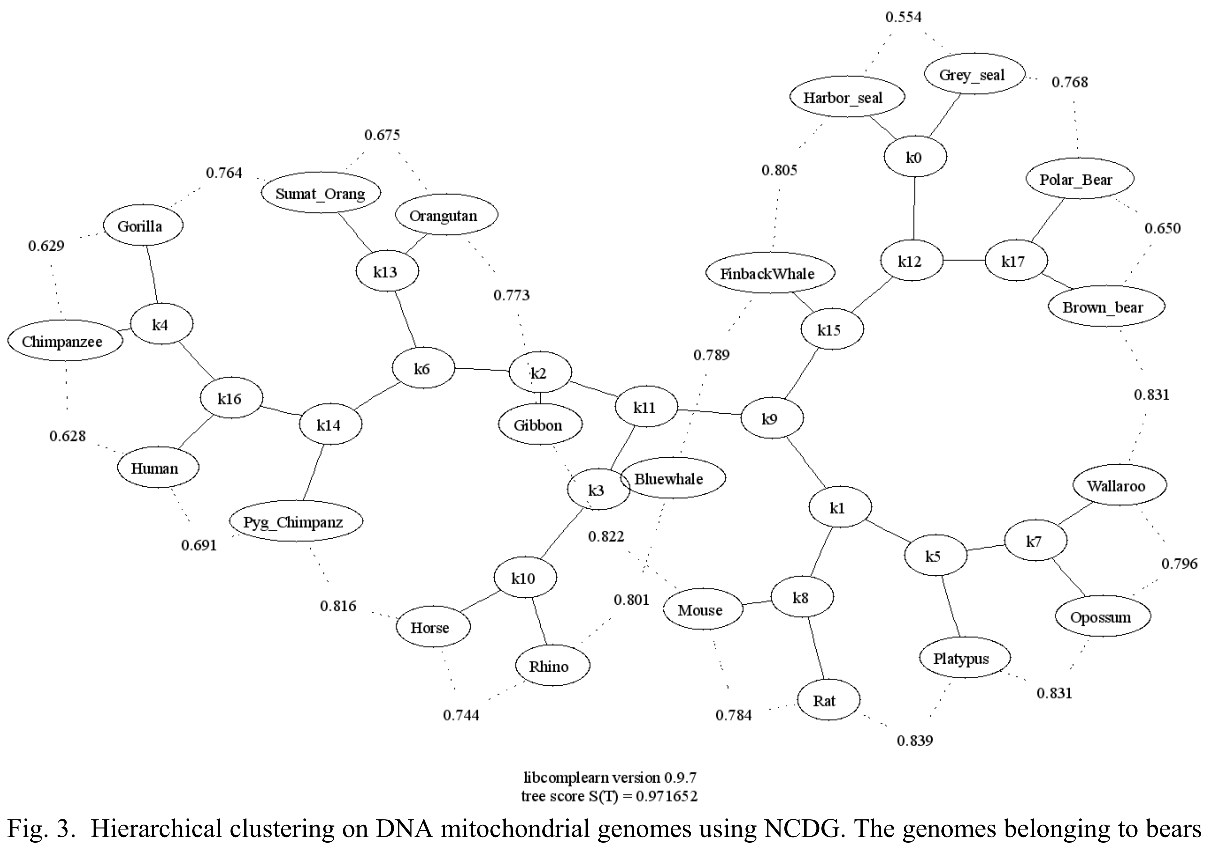

Fig. 3. Hierarchical clustering on DNA mitochondrial genomes using NCDG. The genomes belonging to bears are correctly found more similar to each other, and the group of rodents (Wallaroo, Opossum, Platypus, Rat, Mouse) is well separated in a cluster. Note that Blue and Finback Whales are still close, but not as much as in fig. 2.

From *lemmas 3* and *5*, it derives that $NCDG(x,y) \cong 0$ if $x = y$ but, nevertheless, the conditions for *NCDG* to be a metric described in [2] do not hold, since our complexity approximation is not monotonic.

4. Experiments

4.1. DNA Samples

We test the validity of our complexity approximation on a set of mitochondrial genomes freely available on the web from the database GenBank [24]. A genome is a long sequence of just four elements (*adenine*, *cytosine*, *guanine* and *thymine*), and each of them has been encoded in a first step with an alphabet of 16 symbols, with each symbol representing the combination of any pair of basic components: so, each one is represented by a string with half the size of the original one. The DNA genomes used are the ones of 20 animal species divided in three categories: rodents, ferungulates, and primates, in a similar experiment to one contained in [3]. More specifically, the list of species used is as follows. Rodents: rat (*Rattus norvegicus*), house mouse (*Mus musculus*), opossum (*Didelphis virginiana*), wallaroo (*Macropus robustus*), and platypus (*Ornithorhynchus anatinus*); ferungulates: grey seal (*Halichoerus grypus*), harbor seal (*Phoca vitulina*), brown bear (*Ursus arctus*), polar bear (*Ursus thibetanus*), white rhino (*Ceratotherium simum*), horse (*Equus caballus*), finback whale (*Balaenoptera physalus*), and blue whale (*Balaenoptera musculus*); primates: gibbon (*Hylobates lar*), gorilla (*Gorilla gorilla*), human (*Homo sapiens*), chimpanzee (*Pan troglodytes*), pygmy chimpanzee (*Pan paniscus*), orangutan (*Pongo pygmaeus*), and Sumatran orangutan (*Pongo pygmaeus abelii*).

We generated the best fits of a binary tree to each distance matrix obtained applying the similarity measures *NCD* and *NCDG*. In this and the next experiment, the *NCD* distances were computed with the open-source utilities suite Complearn using default parameters [25]. Figs. 2 and 3 report the results of hierarchical clustering obtained with the two measures by means of the tool *maketree*, also provided by Complearn. In both cases the hierarchical clustering obtained looks accurate, since primates are correctly located in an independent branch of the tree, and the two species of seals are correctly considered close to each other. With *NCDG*, anyway, results improve: the genome related to the brown bear is correctly considered the closest to the one belonging to the polar bear, which is not the case for *NCD*; furthermore, for *NCDG* the class of rodents lies completely in a separated branch of the binary tree, while it is dislocated in two branches with the other method. It has to be remarked that the pertinence of the platypus to the family of rodents is discussed [26]. In [3] the authors obtain different results from the ones presented here, but in that case the ad hoc compressor for DNA sequences mentioned in section 2.2 was used, resulting in a better approximation of Kolmogorov complexity and better results. The tree score reached [3] is also higher for the *NCDG* distance matrix ($Ts \cong 0.97$) than for *NCD* ($Ts \cong 0.93$).

4.2. Satellite Imagery

In our second experiment we used a dataset consisting of 60 single band SPOT 5 images, all of them of size 64x64 and in byte format, divided into the classes *clouds*, *sea*, *desert*, *city*, *forest* and *fields*. Three compression-based clustering methods are compared with the *NCDG* as defined in (7): *NCD*, *PRDC*, and *McDCSM* [7]. The images are linearly scanned outputting sequences which are used in a subsequent step to compute the distances between each pair of objects. In Fig. 4 all the clusterings presents the same confusion in the separation of a sea image (*PRDC* yields an additional false alarm), but for *NCDG* the latter is brought closer to its class, with respect to the clustering obtained on the basis of the previous methods.

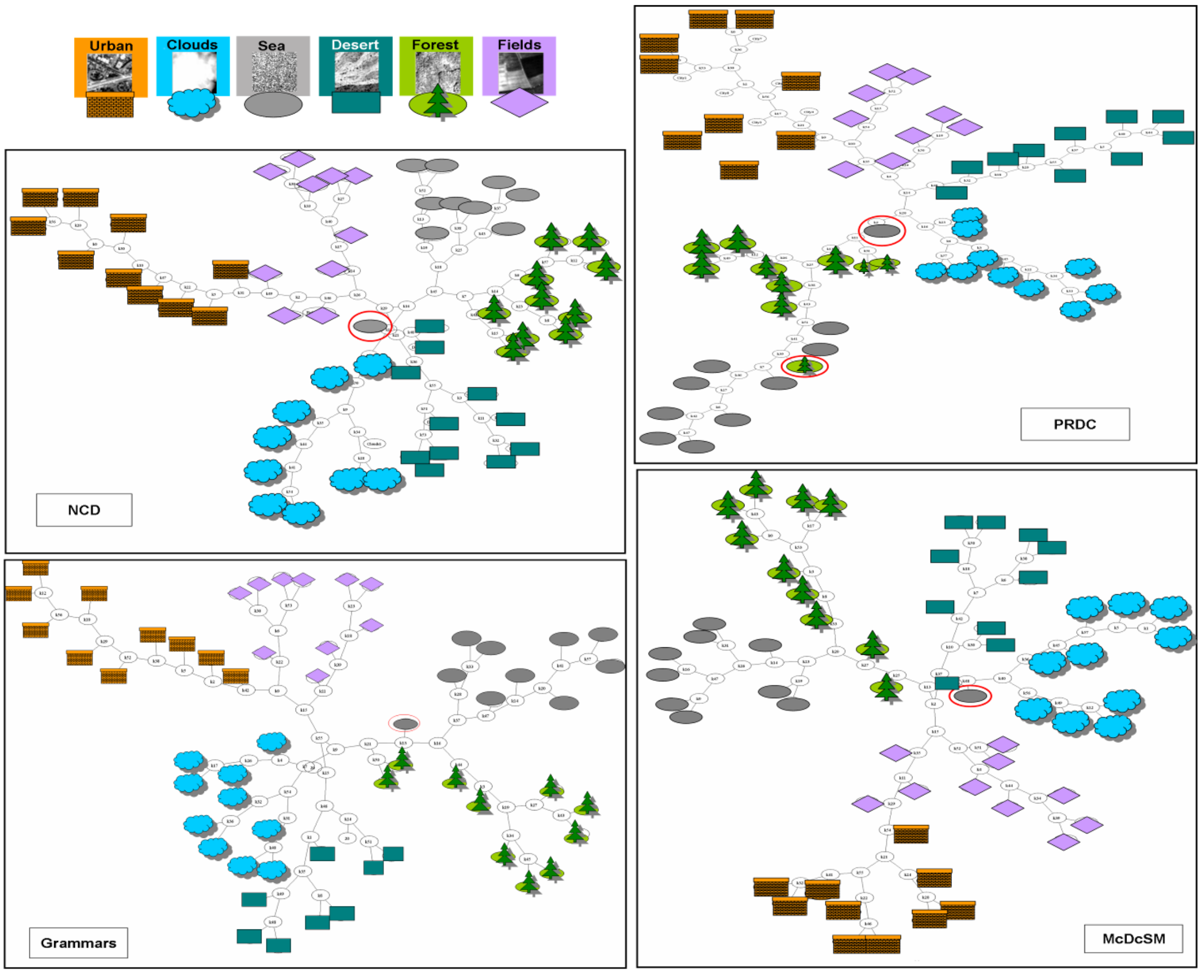

Figure 4. Visual description of the classes used (top left) and hierarchical clustering of similarity measures obtained with the following methods (starting from top right and in clockwise order): PRDC, McDcSM, Grammars, NCD. The dataset consists of 60 single band SPOT5 image subsets of size 64x64. False alarms are circled. Changing from PRDC to McDcSM allows to remove one false alarm. In compression with grammars we have the best clusters separation.

Table I reports the average interclass and intraclass distances obtained on with *NCD* and *NCDG* on an extended dataset of 200 objects, along with a "discrimination factor" which quantifies the separability between the classes as the average difference between interclass and intraclass distances: the latter is some 40% higher for *NCDG*.

5. Conclusions

We presented a new compression-based similarity measure based on Context-Free Grammars (CFG), relying on the concept of MDL and illustrating the relations that these ideas have with Kolmogorov complexity. The smallest CFG is assimilated to the smallest dictionary useful in compressing an object: since dictionaries capture the relevant patterns within the data, their use in the compression step allows considering separately their complexity, tuning the complexity estimation. This two-part representation of complexity results in the definition of a new similarity measure, the *NCDG*. The novelty of this approach lies in the fact that the impact of complexity overestimations, due to the limits that a real compressor has, is accounted for and decreased.

	Intraclass distance	Interclass distance	Discrimination
NCD	1.017	1.110	0.093
NCDG	0.828	0.961	0.133

Table 1. Average NCD and NCDG distances on a 200 image subsets dataset. The distances computed are 40000, 10000 intraclass and 30000 interclass.

The preliminary results presented in this section suggest that the proposed approach may improve result obtained in data compression based similarity measures by means of a standard compressor. This approximation may be tuned in order to focus on the properties of the data structure rather than on the data giving the model, and may be used to separate meaningful information inside the data from noise by adopting another selection of the rules constituting the grammar. On the other hand, computing an approximated smallest CFG requires polynomial time rather than the linear time needed by most compression algorithms.

References

[1] A. N. Kolmogorov, "Three Approaches to the Quantitative Definition of Information," *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[2] M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitányi, "The Similarity Metric", *IEEE Trans. Inf. Theory*, vol. 50, No. 12, pp. 3250-3264, 2004.

[3] R. Cilibrasi and P.M.B. Vitányi, "Clustering by Compression", *IEEE Trans. Inf. Theory*, 51|4:1523-1545, 2005.

[4] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping", *Phys. Rev. Lett.,* vol. 88, no.4, 2002.

[5] D. Cerra and M. Datcu, "Algorithmic Cross-complexity and Relative Complexity", *DCC 2009*, pp. 342-351.

[6] T. Watanabe, K. Sugawara, and H. Sugihara, "A New Pattern Representation Scheme Using Data Compression", *IEEE Trans. Pattern Anal. Mac. Intel.*, vol. 24, pp. 579-590, 2002.

[7] D. Cerra and M. Datcu, "A Model Conditioned Data Compression Based Similarity Measure", *DCC 2008*, p. 509.

[8] J. J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory*, vol. 29, no. 5, pp. 656–664, Sep. 1983.

[9] E. J. Keogh, S. Lonardi, C. Ratanamahatana, "Towards Parameter-Free Data Mining", *SIGKDD 2004*.

[10] M. Cebrian, M. Alfonseca, and A. Ortega, "Common Pitfalls Using the Normalized Compression Distance: What to Watch Out for in a Compressor", *Commun. Inform. Sys.* vol.5, no. 4, pp. 367-384, 2005.

[11] D. Cerra and M. Datcu, "Algorithmic Information Theory Based Analysis of Earth Observation Images: an Assessment", *IEEE Geoscience and Remote Sensing Letters*, in press, 2009.

[12] J. Ziv and A. Lempel, "Compression of Individual Sequences Via Variable-Rate Coding", *IEEE Trans. Inf. Theory*, 1978.

[13] M. Li et al., "An Information-based Sequence Distance and its Application to whole Mitochondrial Genome Phylogeny", *Bioinformatics* 2001, vol. 17, no. 2, pp. 149-154.

[14] Q. Liu et al., "RNACompress: Grammar-based compression and informational complexity measurement of RNA secondary structure", *BCM Bioinformatics*, vol. 9, no. 176, 2008.

[15] M. Li and P.M.B. Vitányi, "An Introduction to Kolmogorov Complexity and Its Applications", Chapter 5, *Springer*, 1997.

[16] N. K. Vereshchagin and P. Vitányi, "Kolmogorov's structure functions and model selection", *IEEE Trans. Inf. Theory,* vol. 50*,* pp. 3265-3290, 2004.

[17] C. S. Wallace and David L. Dowe*,* "Minimum Message Length and Kolmogorov Complexity", *The Computer Journal*, no. 42, pp. 270-283, 1999.

[18] A. N. Soklakov, "Occam's Razor as a Formal Basis for a Physical Theory", *Foundations of Physics Letters,* vol. 15, no. 2, pp. 107-135, 2002.

[19] L. Gueguen and M. Datcu, "A Similarity Metric for Retrieval of Compressed Objects: Application for Mining Satellite Image Time Series", *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 4, pp. 562-575, 2008.

[20] S. Evans et al., "MDLcompress for Intrusion Detection: Signature Inference and Masquerade Attack", *IEEE MILCOM 2007***,** pp. 1-7, 2007.

[21] M. Charikar et al., "The Smallest Grammar Problem", *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2554–2576, July 2005.

[22] E. Lehman and A. Shelat, "Approximation Algorithms for Grammar-Based Compression", *ACM-SIAM Symposium on Discrete Algorithms*, pp. 205-212, 2002.

[23] N. Jesper Larsson and A. Moffat, "Off-Line Dictionary-Based Compression", *Proc. IEEE,* vol. 88, no. 11*,* pp. 1722-1732, 2000.

[24] GenBank by NCBI, [Online], available at http://www.ncbi.nlm.nih.gov/Genbank/index.htm.

[25] Complearn tool by R. Cilibrasi, A. Cruz, S. de Rooij, and M. Keijzer, [Online], available at: http://www.complearn.org/index.html.

[26] John A. W. Kirsch and Gregory C. Mayer, "The Platypus is not a Rodent: DNA Hybridization, Amniote Phylogeny and the Palimpsest Theory". *Philosophical Transactions: Biological Sciences,* vol. 353, no. 1372, pp. 1221–1237, 1998.