PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

rsta.royalsocietypublishing.org

Research

Cite this article: Vitányi PMB. 2013 Similarity and denoising. Phil Trans R Soc A 371: 20120091. http://dx.doi.org/10.1098/rsta.2012.0091

One contribution of 17 to a Discussion Meeting Issue 'Signal processing and inference for the physical sciences'.

Subject Areas:
algorithmic information theory, bioinformatics, pattern recognition

Keywords:
similarity, denoising, individual data, Kolmogorov complexity, information distance, lossy compression

Author for correspondence:
Paul M. B. Vitányi
e-mail: paul.vitanyi@cwi.nl

Similarity and denoising

Paul M. B. Vitányi

National Research Center for Mathematics and Computer Science in the Netherlands (CWI), Science Park 123, 1098XG Amsterdam, The Netherlands

We can discover the effective similarity among pairs of finite objects and denoise a finite object using the Kolmogorov complexity of these objects. The drawback is that the Kolmogorov complexity is not computable. If we approximate it, using a good real-world compressor, then it turns out that on natural data the processes give adequate results in practice. The methodology is parameter-free, alignment-free and works on individual data. We illustrate both methods with examples.

1. Introduction

In recent studies [1–3], we and others have developed theoretical approaches to (i) similarity of finite objects, and (ii) denoising of the same. We proved that these theories based on Kolmogorov complexity are perfect. By approximating the Kolmogorov complexities involved by real-world compressors, we transformed these theoretical notions into applications that work better than we could expect [4,5]. The purpose of this study is to review these results and give some possible reasons why they are so good.

(a) Similarity

In pattern recognition, learning and data mining, one obtains information from information-carrying objects. The notion of Kolmogorov complexity [6] is an objective measure for the information in a *single* object, and information distance measures the information between a *pair* of objects [1]; see appendix A.3. The information distance (in normalized form) can be used to find the similarity between a pair of objects. We give a computable approximation using real-world compressors. In many situations, this approximation is argued below to be sufficient in the case of natural data and good compressors. The resulting similarity measure

Royal Society Publishing
Informing the science of the future

is a parameter-free and alignment-free method.[1] One of the advantages of this method is that it is very fast and also works for noisy objects [7].

(b) Denoising

If, in observations or signal analysis, one wants to get rid of the noise in the observed object or in the signal, then we use a denoising method. We show a method based on Kolmogorov complexity. Again, we approximate the Kolmogorov complexities involved by a real-world compressor. A disadvantage of the method is that it is currently slow.

(c) Natural data and Kolmogorov complexity

The Kolmogorov complexity (appendix A.2) of a file is a lower bound on the length of the ultimate compressed version of that file. In both cases mentioned earlier, we approximate the Kolmogorov complexities involved by a real-world compressor. Because the Kolmogorov complexity is incomputable, in the approximation we never know how close we are to it. However, we assume that the natural data we are dealing with contain no complicated mathematical constructs such as $\pi = 3.1415\ldots$ or universal Turing machines. In fact, we assume that the natural data we are dealing with contain mostly effective regularities that a good compressor finds. Under those assumptions, the Kolmogorov complexity of the object is not much smaller than the length of the compressed version of the object.

Remark 1.1. As an aside, in many applications we are interested in shared information between *certain* objects instead of just a pair of objects. For example, in customer reviews of gadgets, in blogs about public happenings, in newspaper articles about the same occurrence, we are interested in the most comprehensive one or the most specialized one. Thus, we want to extend the information distance measure from pairs to multiples. This approach was introduced in Li *et al.* [8], whereas much of the theory is developed in Vitányi [9].

2. Similarity

In many situations, one is interested in how much a pair of data is alike, and whether one pair of data is more alike than another pair. In the following, we give a method to quantify this alikeness (or similarity) of a pair of data in terms of a distance between them. This distance is a quantity between 0 (identical) and 1 (completely dissimilar). To visualize the n^2 pairwise distances between n data, we make use of the following technique: a hierarchical clustering in a dendrogram (a tree where each internal node has three edges incident on it). To construct this dendrogram, we use the so-called quartet method from computational biology. We developed a new fast heuristic for the quartet method [10]. There is nothing prescribed (such as the number of clusters); we let the data decide for themselves.

The method takes the $n \times n$ distance matrix as input, and yields a dendrogram with the n objects as leaves (so the dendrogram contains n external nodes or leaves and $n-2$ internal nodes such as in the figures below). We assume $n \geq 4$. The resulting dendrogram models the distance matrix as well as possible qualitatively. If the distance between object o_1 and object o_2 is smaller than that between o_1 and o_3, then the shortest path in the dendrogram between o_1 and o_2 has at most as many edges as the shortest path between o_1 and o_3 (equal if o_2 and o_3 are sibling nodes). Thus, the edges themselves have no length, and the dendrogram represents the partial order induced by the distance matrix. For details, see the cited reference [10]. The method is available as an open-source software tool [11].

[1]In bioinformatics, the computation of the similarity between genetic strings commonly involves the so-called 'alignment method'. This method has high or even forbidding computational costs. See the comments in the example of phylogeny in §2*b*. For certain problems, biologists look for alignment-free methods.

Commonly, the data are of a certain type: music files, transaction records of ATMs, credit card applications, genomic data. In these data, there are common features that can be quantified by giving a number to each data item. Such methods use *feature-based similarities*. For example, from genomic data, one can extract letter or block frequencies (the blocks are over the four-letter alphabet); from music files, one can extract various specific numerical features, related to pitch, rhythm, harmony and so on. Examples are given in [12–16]. However, these methods require specific and detailed knowledge of the problem area, because one needs to know what features to look for.

By contrast, we aim at *non-feature-based similarity*. That is, capturing, in a single similarity metric, *every effective distance*: effective versions of Hamming distance, Euclidean distance, edit distances, alignment distance, Lempel–Ziv distance and so on. The metric should be so general that it works in every domain, music, text, literature, programs, genomes, executables, natural language determination, equally and simultaneously. It should be able to simultaneously detect *all* similarities between pieces that other effective distances can detect separately. Such a metric that we called the *normalized information distance* (NID) was exhibited in Li *et al.* [2]. A brief formal introduction is given in appendix A.3.

Let us give an intuitive explanation. Two objects are deemed close if we can significantly 'compress' one given the information in the other, the intuition being that, if two pieces are more similar, then we can more succinctly describe one given the other. The NID discovers all effective similarities in the sense that if two objects are close according to *some* effective similarity then they are also close according to the NID (appendix A.3).

Put differently, the NID represents similarity according to the dominating shared feature between the two objects being compared. In comparisons of more than two objects, different pairs may have different dominating features. For every two objects, this NID zooms in on the dominant similarity between those two objects out of a wide class of admissible similarity features. It may be called *'the' similarity metric*. For the original theoretical derivation, see [1,2], and for a comprehensive review, see Vitányi *et al.* [17].

(a) Normalized compression distance

Unfortunately, the universality of the NID comes at the price of incomputability. In fact, it is not even semi-computable (this is weaker than computable, appendix A.1) and there is no semi-computable function at a computable distance from it [18].

We shall use real data-compression programs to approximate the Kolmogorov complexity. The length of the compressed version of a finite object is obviously computable. Usually, the computation process is fast. For the natural data we are dealing with, we assume that the length of the compressed version is not too far from its Kolmogorov complexity (see the discussion in §1). We substitute the Kolmogorov complexity in the NID, appendix A.3, by its approximation. If Z is a compressor and we use $Z(x)$ to denote the length of the compressed version of a string x, then we arrive at the *normalized compression distance* (NCD):

$$\mathrm{NCD}_Z(x,y) = \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}, \tag{2.1}$$

where we have replaced the pair (x,y) in the formula by the concatenation xy (file y appended to file x) and we ignore logarithmic terms in the numerator and denominator; see Cilibrasi & Vitányi [4].[2] Examples of the use of the NCD are given in [19–21], among others.

Remark 2.1. In reference [4], we propose axioms to capture the real-world setting, and show that (2.1) approximates optimality. Actually, the NCD is a family of compression functions parametrized by the given data compressor Z. As compressors we used gzip (a Lempel–Ziv compressor with small window size of 32 kB), bzip2 (a block-sorting compressor based on the Burrows–Wheeler transform with a larger window size of 256 kB) and PPMZ (prediction by

[2]Here and elsewhere in this study 'logarithm' or 'log' refers to the binary logarithm.

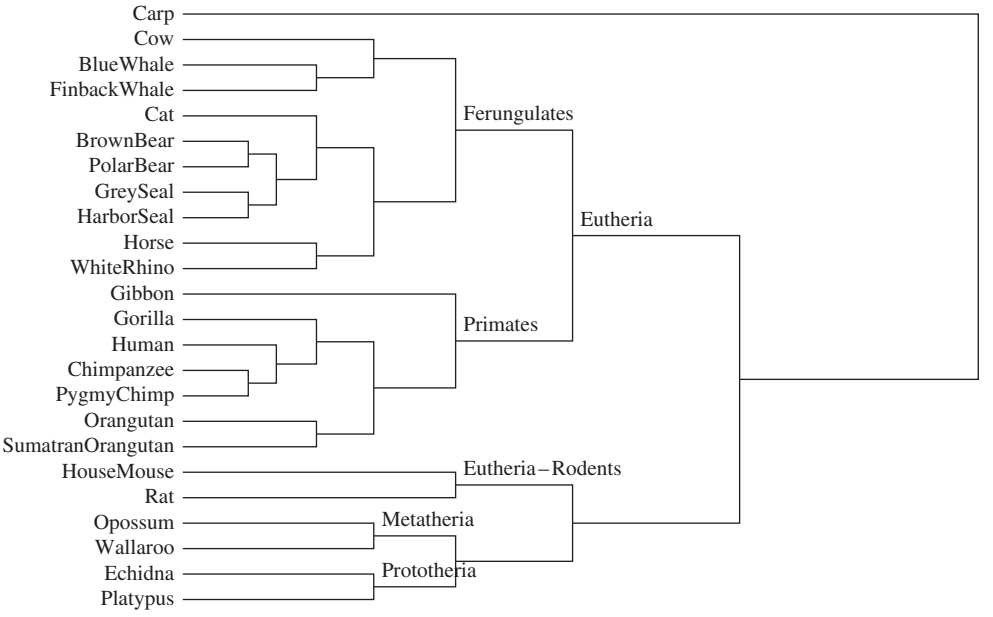

Figure 1. The evolutionary tree built from complete mammalian mtDNA sequences. $S(T) = 0.996$.

partial matching (PPM) is an adaptive statistical data compression technique based on context modelling and prediction [22]; PPM models use a set of previous symbols in the uncompressed symbol stream to predict the next symbol in the stream; it has infinite window size; we used variant Z). The objects being compared for similarity must fit in, say, one-half the window size. By far the poorest compressor is gzip, whereas PPMZ is the best (although slowest) in this line-up. For example, the ideal compressor Z takes care that $\mathrm{NCD}_Z(x,x)$ equals 0. With $Z =$ gzip, usually it is between $\frac{1}{2}$ and 1 (very bad). With $Z =$ bzip2, it is lower but nowhere near 0, and $\mathrm{NCD}_{\mathrm{PPMZ}}(x,x)$ in the genomic experiment of figure 1 was between 0.002 and 0.006. For more experimental evidence, see Cebrian *et al.* [23]. So when one obtains poor results in NCD experiments using gzip, it pays to use a better compressor such as one from the PPM family. Because it is not our intention here to give a data compression course, we refer the interested reader to Wikipedia for more details.

Remark 2.2. Because of the normalization it does not matter for the NCD whether the length of dataset x is different from the length of y. In practice, this difference should not be too great.

Remark 2.3. The $S(T)$ value states how well the quartet tree represents the underlying $n \times n$ distance matrix. It is defined in recent studies [4,10]. See appendix B for an explanation. The examples of the use of the NCD are from Cilibrasi & Vitányi [4].

(b) Phylogeny

A DNA sequence is a finite string over a four-letter alphabet $\{A, C, G, T\}$. We used the mitochondrial genomes (mtDNA) of 24 mammals, each of at most 18 000 base pairs, obtained from the GenBank Database on the Internet. Hence, the mtDNA of every species involved is a string of at most 36 000 bits. Because we use the entire mtDNA of every species involved, we do 'whole-genome' phylogeny.

Whole-genome phylogeny is usually feasible only with alignment-free methods, such as the NCD method. This type of phylogeny is often computationally forbidding for the usual alignment methods used in bioinformatics. Moreover, gene areas move easily over the genome to other places again, making the use of these methods impossible or hard. Hence, it is more usual in

bioinformatics to select a particular gene from the genome of each species. This particular gene should not evolve too fast, such as the gene coding for insulin. Mutations here are usually fatal for the individual concerned. Thus, biologists feel that comparing these genes of species gives trustworthy information about the evolution of species. This may be called 'gene tree phylogeny'; see Rannalal & Yang [24]. However, using different genes may result in different trees [25]. But using the whole genome gives a single tree.

We use the common carp (*Cyprinus carpio*) as the 'outgroup'. The outgroup is used in biology because we want a directed tree with a root. The outgroup is a species so far removed from the other species investigated (according to biology and common sense) that where the branch of the outgroup joins the tree is where the root is. That is, the outgroup species split off earliest. For the other 23 species used, we do not give the Latin names; they can be found in Cilibrasi & Vitányi [4].

The first question we test is a hypothesis about the Eutherian orders, which is viewed as the most important question of mammalian evolution. It has been hotly debated in biology which two of the three main placental mammalian groups, primates, ferungulates and rodents, are more closely related. One cause of the debate is that in the analysis of the genomics the standard maximum-likelihood method, which depends on the multiple alignment of sequences corresponding to an individual protein, gives (rodents, (ferungulates, primates)) for half of the proteins in the mitochondrial genome, and (ferungulates, (primates, rodents)) for the other half.

The second question we test is related to the extant monophyletic divisions of the class Mammalia: the Prototheria (monotremes: mammals that procreate using eggs), the Metatheria (marsupials: mammals that procreate using pouches) and the Eutheria (placental mammals: mammals that procreate using placentas). The sister relationships between these groups is viewed as the most fundamental question in mammalian evolution [24]. Phylogenetic comparison by either anatomy or mitochondrial genome has resulted in two conflicting hypotheses: the gene-isolation-supported Marsupionta hypothesis ((Prototheria, Metatheria), Eutheria) versus the morphology-supported Theria hypothesis (Prototheria, (Methateria, Eutheria)).

In recent years, as a result of more sophisticated methods, together with biological evidence, with respect to our first question it is believed that (rodents, (ferungulates, primates)) reflects the true evolutionary history. With respect to our second question, we find support for the Marsupionta hypothesis.

For every pair of mitochondrial genome sequences x and y, we evaluated the formula in equation (2.1) using a good compressor such as PPMZ. The resulting distances are the entries in an $n \times n$ distance matrix. Constructing a phylogeny tree from the distance matrix, using our quartet tree method [10] as tree-reconstruction software, gives the tree in figure 1. The resulting phylogeny, with an almost maximal $S(T)$ score (see appendix B) of 0.996, supports anew the currently accepted grouping (rodents, (primates, ferungulates)) of the Eutherian orders, and additionally the Marsupionta hypothesis ((Prototheria, Metatheria), Eutheria) (figure 1). That is, our whole-mitochondrial NCD analysis supports the following hypothesis.

$$\overbrace{(\underbrace{(\text{primates}, \text{ferungulates})}_{\text{Eutheria}}(\text{rodents}, (\text{Metatheria}, \text{Prototheria})))}^{\text{Mammalia}}$$

This indicates that the rodents, and the branch leading to the Metatheria and Prototheria, split off early from the branch that led to the primates and ferungulates. This tree confirms the accepted hypothesis of (rodents, (primates, ferungulates)), and every single branch of the tree agrees with the current biological classification.

(c) Hierarchical clustering

In the following, we test gross classification of files based on heterogeneous data of markedly different file types: (i) four mitochondrial gene sequences, from a black bear, polar bear, fox and rat obtained from the GenBank Database on the Internet; (ii) four excerpts from the novel *The*

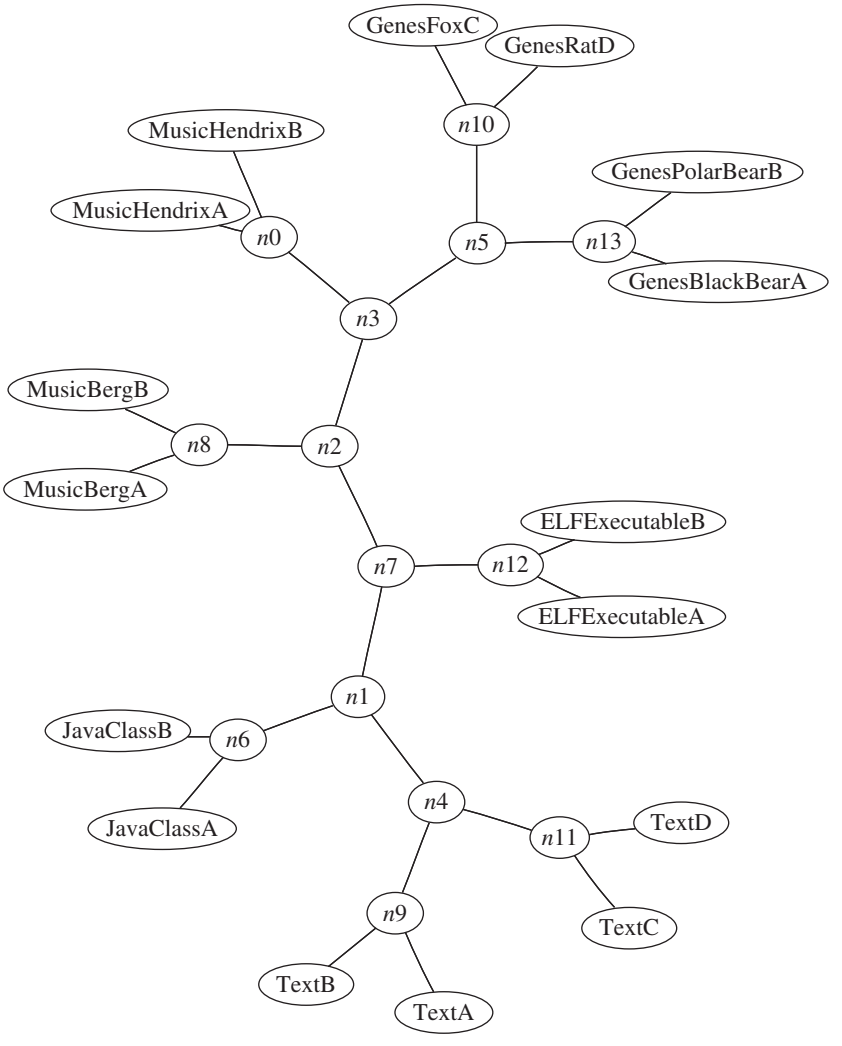

Figure 2. Clustering of heterogeneous file types. $S(T) = 0.984$.

Zeppelin's Passenger by E. Phillips Oppenheim, obtained from the Project Gutenberg edition on the Internet; (iii) four MIDI files without further processing, two works by Jimi Hendrix and two movements from Debussy's 'Suite Bergamasque', downloaded from various repositories on the Internet; (iv) two Linux x86 ELF executables (the *cp* and *rm* commands), copied directly from the RedHat 9.0 Linux distribution; and (v) two compiled Java class files, generated directly. The program correctly classifies each of the different types of files together with like near like. The result is reported in figure 2. This experiment shows the power and universality of the method: no features of any specific domain of application are used. We believe that there is no other method known that can cluster data that are so heterogeneous this reliably. Researchers from the data-mining community noticed that this methodology is in fact a parameter-free, feature-free, data-mining tool. They have experimentally tested a closely related distance on a large variety of sequence benchmarks. Comparing the compression-based method with 51 major parameter-loaded methods found in the eight major data-mining conferences (SIGKDD, SIGMOD, ICDM, ICDE, SSDB, VLDB, PKDD and PAKDD) in 1994–2004, on every database of time sequences used, ranging from heartbeat signals to stock market curves, they established clear superiority of the compression-based method for clustering heterogeneous data and for anomaly detection and competitiveness in clustering domain data [26].

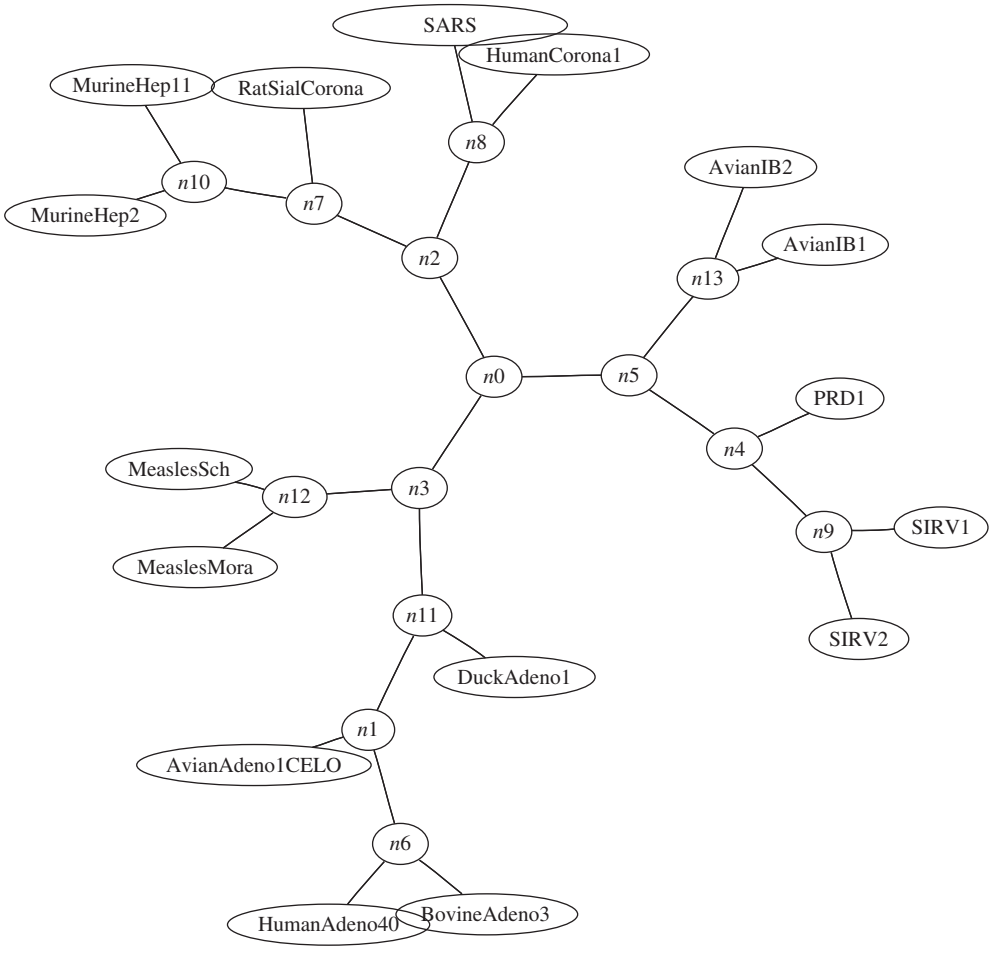

Figure 3. SARS virus among other viruses. $S(T) = 0.988$.[3]

(d) SARS virus

We clustered the SARS virus directly after its sequenced genome was made publicly available, in relation to potential similar viruses. The 15 virus genomes were downloaded from The Universal Virus Database of the International Committee on Taxonomy of Viruses, available on the Internet. The SARS virus was downloaded from Canada's Michael Smith Genome Sciences Centre, which had the first public SARS coronavirus draft whole-genome assembly available for download (SARS TOR2 draft genome assembly 120403). The NCD distance matrix was computed using the compressor bzip2. The entire computation took only a couple of minutes. The relations in figure 3 are very similar to the definitive tree based on medical–macrobio–genomics analysis, appearing later in the *New England Journal of Medicine* [27]. We depicted the figure in the ternary tree style, rather than the genomics–dendrogram style, because the former is more precise for visual inspection of proximity relations.[3]

[3]Legend for figure 3: AvianAdeno1CELO.inp: fowl adenovirus 1; AvianIB1.inp: avian infectious bronchitis virus (strain Beaudette US); AvianIB2.inp: avian infectious bronchitis virus (strain Beaudette CK); BovineAdeno3.inp: bovine adenovirus 3; DuckAdeno1.inp: duck adenovirus 1; HumanAdeno40.inp: human adenovirus type 40; HumanCorona1.inp: human coronavirus 229E; MeaslesMora.inp: measles virus strain Moraten; MeaslesSch.inp: measles virus strain Schwarz; MurineHep11.inp: murine hepatitis virus strain ML-11; MurineHep2.inp: murine hepatitis virus strain 2; PRD1.inp: enterobacteria phage PRD1; RatSialCorona.inp: rat sialodacryoadenitis coronavirus; SARS.inp: SARS TOR2v120403; SIRV1.inp: *Sulfolobus* virus SIRV-1; SIRV2.inp: *Sulfolobus* virus SIRV-2.

(e) Foetal heart rhythm

Another, even more clinical, experiment is reported in Costa Santos *et al.* [28]. Foetal heart rate (FHR) monitoring is widely used with the aim of detecting foetuses in danger of death or damage. The features determining these fates are poorly understood. The compression method is exploratory in the sense that the compressor determines similarity but does not give the grounds for this similarity. That burden may fall on the experts scrutinizing the objects to find the cause for this similarity or lack of it. Thirty-one FHR tracings acquired in the antepartum period were clustered using the NCD in order to identify the abnormal ones. At the highest level, the FHR tracings were clustered according to the technology used in signal acquisition (different machines, sampling at different rates and so on). At the lower levels, all tracings with abnormal or suspicious patterns were clustered together, independently of the technology used. The figure has a high $S(T)$ value of 0.944; however, it is too large to display here; interested readers are referred to the cited reference [28].

(f) Astronomy

In reference [29], observations of the microquasar GRS 1915 + 105 made with the *Rossi X-ray Timing Explorer* were analysed. The interest in this microquasar stems from the fact that it was the first Galactic object to show a certain behaviour (superluminal expansion in radio observations). Photonometric observation data from X-ray telescopes were divided into short time segments (usually of the order of 1 min), and these segments were classified into 15 different modes after considerable effort in Belloni *et al.* [29]. Briefly, spectrum hardness ratios (roughly, 'colour') and photon count sequences were used to classify a given interval into categories of variability modes. From this analysis, the extremely complex variability of this source was reduced to transitions between three basic states, which, interpreted in astronomical terms, gives rise to an explanation of this peculiar source in standard black-hole theory. The data we used in this experiment were made available to us by M. Klein-Wolt (co-author of the above paper) and T. Maccarone, at the time both researchers at the Astronomical Institute 'Anton Pannekoek' of the University of Amsterdam.

The observations are essentially time series. The task was to see whether the clustering would agree with the classification earlier. The NCD matrix was computed, using the compressor PPMZ. The results are in figure 4. In the figure, the initial capital letter indicates the class corresponding to Greek lower-case letters in Belloni *et al.* [29]. The remaining letters and digits identify the particular observation interval in terms of finer features and identity. The T-cluster is top left, the P-cluster is bottom left, the G-cluster is to the right and the D-cluster in the middle. This tree almost exactly represents the underlying NCD distance matrix: $S(T) = 0.994$. We clustered 12 objects, consisting of three intervals from four different categories denoted as $\delta, \gamma, \phi, \theta$ in table 1 of Belloni *et al.* [29]. In figure 4, we denote the categories by the corresponding roman letters D, G, P and T, respectively. The resulting tree groups these different modes together in a way that is consistent with the classification by experts for these observations. The oblivious compression clustering corresponds precisely with the laborious feature-driven classification in Belloni *et al.* [29].

(g) Classification

For the experimental physical scientist who has lots of datasets and who wants to find ideas about the relations between them, the approach by hierarchical clustering through a dendrogram may not be useful. The sheer number of datasets may preclude a visual representation in a dendrogram. What is required is classification. (If necessary, one can vary the number of classes.) One can use the NCD distances as an oblivious, feature extraction technique to convert generic objects into finite-dimensional vectors. We have used this technique in reference [4] to train a support vector machine (SVM)-based optical character recognition (OCR) system to classify

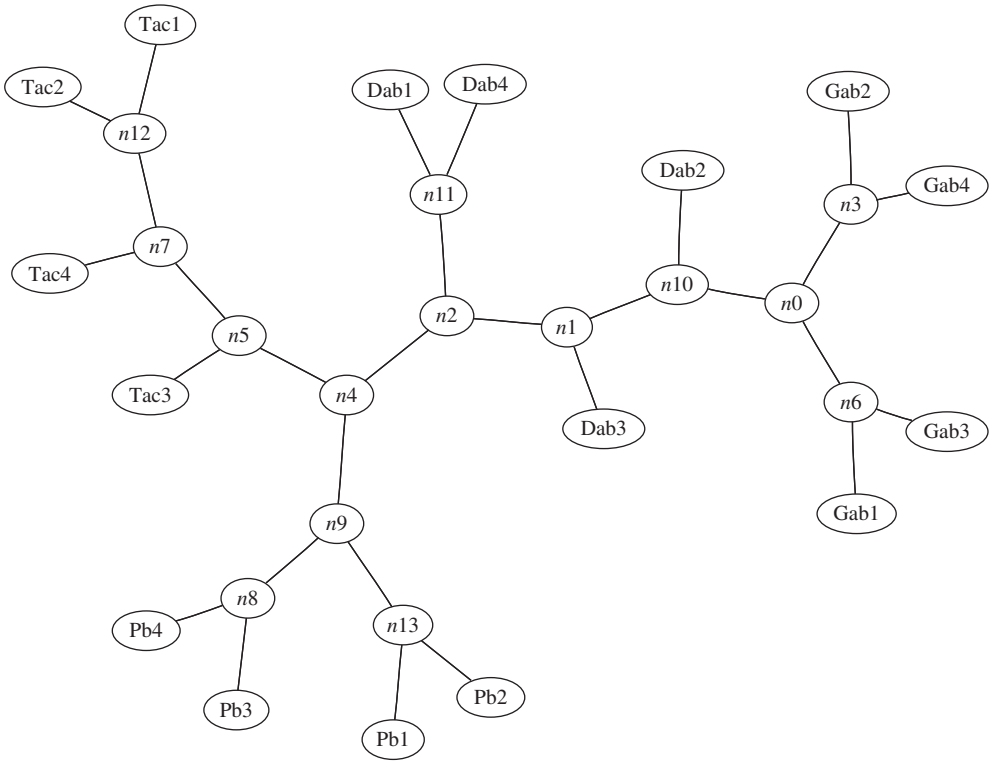

Figure 4. Sixteen observation intervals of GRS 1915 + 105 from four classes. $S(T) = 0.994$.

handwritten digits by extracting 80 distinct, ordered NCD features from each input image, in the manner explained below; see also [30]. For SVMs, see for example [31]. We achieved a handwritten single-decimal-digit recognition accuracy of 87 per cent. The current state of the art for this problem, after half a century of interactive feature-driven classification research, is in the upper 90 per cent level [32]. These experiments were benchmarked on the standard National Institute of Standards and Technology (NIST) Special Database 19.

We explain the method by a small binary classification problem. We require a list of at least, say, 40 *training examples*, consisting of at least 20 positive examples and 20 negative examples, to illustrate the contemplated concept class. We also provide, say, six *anchor* examples $a_1, \ldots, a_6$, of which half are in some way related to the concept under consideration. Then, we use the anchors to convert each of the 40 training examples $w_1, \ldots, w_{40}$ to six-dimensional *training vectors* $v_1, \ldots, v_{40}$. The entry $v_{j,i}$ of $v_j = (v_{j,1}, \ldots, v_{j,6})$ is defined as $v_{j,i} = \mathrm{NCD}_Z(w_j, a_i)$ $(1 \leq j \leq 40, 1 \leq i \leq 6$, where Z is the used compressor). The training vectors are then used to train an SVM to learn the concept. The test words are classified, using the same anchors and trained SVM model. The LIBSVM software was used for all SVM experiments [31].

3. Denoising

Commonly, the data in the computer that represent real objects are distorted, because the observation apparatus, the process of observing and other factors introduce additional errors. These errors are called *noise*. Eliminating the noise to obtain the pristine object is called *denoising*. We present a method of denoising of individual objects based on Kolmogorov complexity as in appendix A.2.

Rate–distortion theory analyses communication over a channel under a constraint on the number of transmitted bits, the 'rate'. It currently serves as the theoretical underpinning for lossy

compression and denoising. (As an aside, more than half of all files transmitted over the Internet consist of lossy compressed objects. The methods of lossy compression can be MP3, JPEG, MPEG.)

Rate–distortion theory was introduced by Shannon [33,34]. Classically, it is concerned with the trade-off between the rate and the achievable fidelity of the transmitted representation of an object under some distortion function in *expectation* under some source distribution (i.e. a random variable).

For large and complex objects, one instead relies on structural properties of the *individual objects*. A rate–distortion theory that allows analysis of individual objects has been developed [35] within the framework of Kolmogorov complexity (appendix A.2). The rate–distortion function is defined not with respect to some elusive source distribution, but with respect to an individual source word. Every source word thus obtains its own associated rate–distortion function. However, Kolmogorov complexity is not computable. We therefore, as before, approximate the Kolmogorov complexity by a real–world compressor. The theory hinted at is from reference [35] and the example is from reference [5].

(a) Algorithmic rate–distortion

Suppose we want to communicate objects x from a set of source words $\mathcal{X}(|\mathcal{X}| > 2^r)$ using at most r bits per object. We call r the *rate*. We locate a good *representation* of x within a finite set $\mathcal{Y}$, which may be different from $\mathcal{X}$ in general (but we use $\mathcal{X} = \mathcal{Y}$). The lack of fidelity of a representation y is quantified by a distortion function $d : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}^+$, where $\mathcal{R}^+$ denotes the set of non-negative real numbers.

We can transmit a representation y of x that has $K(y) \leq r$. Such a representation may induce a distortion $d(x, y)$, but the receiver can run the program to obtain y and is thus able to reconstruct x up to distortion $d(x, y)$.

Let $\mathcal{Q}$ denote the rational numbers. The *rate–distortion function* $r_x : \mathcal{Q} \to \mathcal{N}$ is the minimum number of bits in a representation y of x to obtain a distortion of at most δ defined by

$$r_x(\delta) = \min_{y \in \mathcal{Y}} \{K(y) : d(x, y) \leq \delta\}.$$

The 'inverse' of the above function is the *distortion–rate function* $d_x : \mathcal{N} \to \mathcal{R}$ and is defined by

$$d_x(r) = \min_{y \in \mathcal{Y}} \{d(x, y) : K(y) \leq r\}.$$

A representation y is said to *witness* the rate–distortion function of x if $r_x(d(x, y)) = K(y)$. These definitions are illustrated in figure 5. Algorithmic rate–distortion theory is developed and treated in much more detail in Vereshchagin & Vitányi [35]. It is a generalization of Kolmogorov's structure function theory [3].

(i) The noisy cross

In the example of the noisy cross, we used Hamming distortion. This function is a metric and $\mathcal{X} = \mathcal{Y}$. (In other experiments, we also used Euclidean distortion and edit distortion [36].)

Hamming distortion is perhaps the simplest distortion function that could be used. Let x and y be two objects of equal length n. The Hamming distortion $d(x, y)$ is equal to the number of symbols in x that do not match those in the corresponding positions in y.

The search space is very large: for an object of n bits, there are 2^n candidate representations of the same size, and objects that are typically subjected to lossy compression are often millions or billions of bits long. An exhaustive search is infeasible and a greedy search is likely to get stuck in a local optimum far from the global optimum. Because the structure of the search landscape is at present poorly understood and we do not want to make any unjustifiable assumptions, we use a genetic search algorithm, which performs well enough that interesting results can be obtained.

The Kolmogorov complexity is not computable and can be approximated by a computable process from above but not from below, whereas a real compressor is computable. Therefore, the approximation of the Kolmogorov complexity by a real compressor involves for some arguments

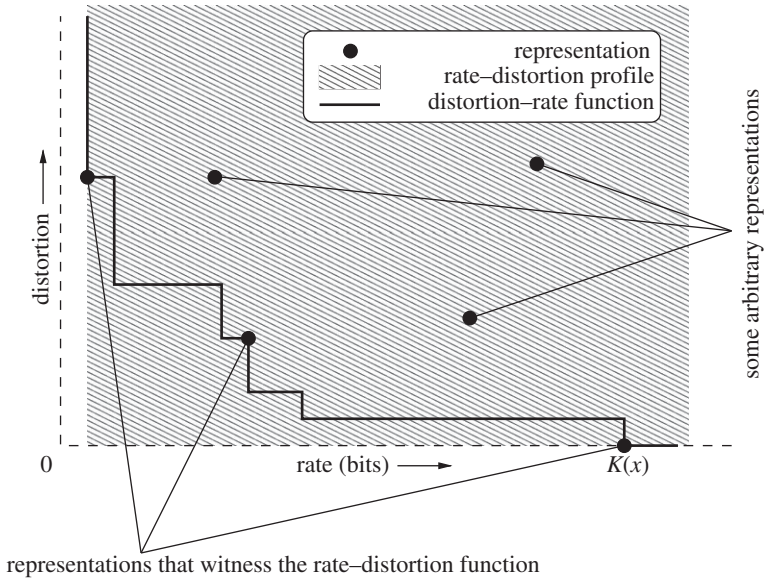

Figure 5. Rate–distortion profile and distortion–rate function.

errors that can be high and are in principle unknowable. Despite all these caveats, it turns out that the practical analogue of the theoretical method works surprisingly well in all the experiments we tried.

We approximated the distortion–rate function of a noiseless cross called the *target*. It consists of a monochrome image of 1188 black pixels together with 2908 surrounding white pixels, forming a plane of 64×64 black or white pixels. Added are 377 pixels of artificial noise inverting 109 black pixels and 268 white pixels. This way we obtain a noisy cross called the *input*. The input is in effect a pixelwise exclusive OR of the target and noise. The distortion used is Hamming distortion. At every rate r ($0 \leq r \leq 3500$), we compute a set $M(r)$ of candidates. Every candidate consists of the 64×64 pixel plane divided into black pixels and white pixels. Every candidate approximates the input in a certain sense and a compressed version requires at most r bits. For every (uncompressed) candidate in $M(r)$, the distortion to the input is computed. The candidate in $M(r)$ that minimizes the distortion is called the 'best' candidate at rate r.

Figure 6 shows two graphs. The first graph hits the horizontal axis at about 3178 bits. On the horizontal axis, it gives the rate, and on the vertical axis, it denotes the distortion to the input of the best candidate at every rate. The line hits zero distortion at rate about 3178, when the input is retrieved as the best candidate (attached to this point). The second graph hits the horizontal axis at about 260 bits. The horizontal axis denotes again the rate, but now the vertical axis denotes the distortion between the best candidate and the target. The line hits almost zero distortion (three bits flipped) at rate about 260. There an image that is almost the target is retrieved as the best candidate (attached to this point). The three wrong bits are two at the bottom left corner and one in the upper right armpit. The hitting of the horizontal axis by the second graph coincides with a sharp slowing of the rate of decrease of the first graph. Subsequently, the second graph rises again, because the best candidate at that rate starts to model more of the noise present in the input. Thus, the second graph shows us the denoising of the input, underfitting left of the point of contact with the horizontal axis, and overfitting right of that point.

However, in denoising an object, we do not know the target (in this case the noiseless cross), but only the input (in this case the noisy cross) and the first graph. The point of best denoising can be deduced from the first graph, where it is the point where the distortion–rate curve sharply levels off. Because this point has distortion of only 3 to the target, the distortion–rate function separates structure and noise very well in this example.

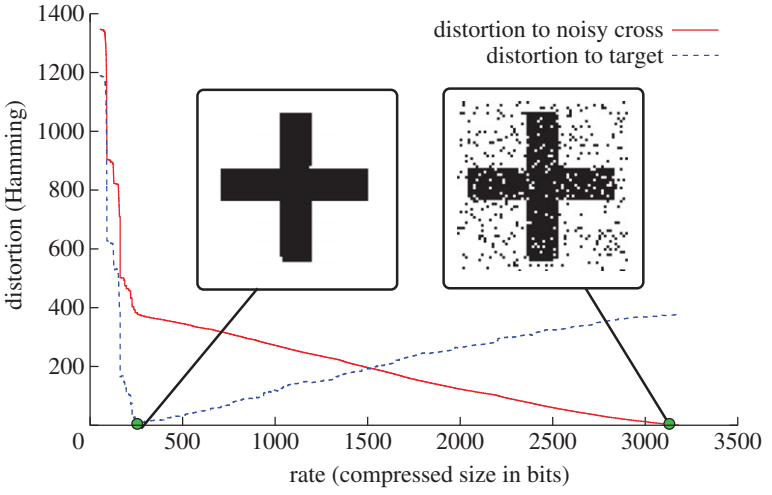

Figure 6. Denoising of the noisy cross. (Online version in colour.)

Appendix A. Computability, Kolmogorov complexity and information distance

A.1. Computability

Turing [37] defined the hypothetical 'Turing machine' whose computations are intended to give an operational and formal definition of the intuitive notion of computability in the discrete domain. These Turing machines compute integer functions, the *computable* functions. By using pairs of integers for the arguments and values, we can extend computable functions to functions with rational arguments and/or values. The notion of computability can be further extended [38]: a function f with rational arguments and real values is *upper semi-computable* if there is a computable function $\phi(x,k)$ with x a rational number and k a non-negative integer such that $\phi(x,k+1) \leq \phi(x,k)$ for every k and $\lim_{k\to\infty} \phi(x,k) = f(x)$. This means that f can be computably approximated from above. A function f is *lower semi-computable* if $-f$ is upper semi-computable. A function is called *semi-computable* if it is either upper semi-computable or lower semi-computable or both. If a function f is both upper semi-computable and lower semi-computable, then f is *computable*. A countable set S is *computably* (or *recursively*) *enumerable* if there is a Turing machine T that outputs all and only the elements of S in some order and does not halt. A countable set S is *decidable* (or *recursive*) if there is a Turing machine T that decides for every candidate a whether $a \in S$ and halts.

An example of a computable function is $f(n)$ defined as the nth prime number; an example of a function that is upper semi-computable but not computable is the Kolmogorov complexity function K in appendix A.2. An example of a recursive set is the set of prime numbers; an example of a recursively enumerable set that is not recursive is $\{x \in \mathcal{N} : K(x) < |x|\}$.

A.2. Kolmogorov complexity

For details, see reference [38]. Informally, the Kolmogorov complexity of a string is the length of the shortest string from which the original string can be losslessly reconstructed by an effective general-purpose computer such as a particular universal Turing machine U. Hence, it constitutes a lower bound on how far a lossless compression program can compress. In this study, we require that the set of programs of U is prefix-free (no program is a proper prefix of another program), that is, we deal with the *prefix Kolmogorov complexity*. (But for the results in this study, it does not matter

whether we use the plain Kolmogorov complexity or the prefix Kolmogorov complexity.) We call U the *reference universal prefix machine*. Formally, the *conditional prefix Kolmogorov complexity* $K(x \mid y)$ is the length of the shortest input z such that the reference universal prefix machine U on input z with auxiliary information y outputs x. The *unconditional prefix Kolmogorov complexity* $K(x)$ is defined by $K(x \mid \epsilon)$. The functions $K(\cdot)$ and $K(\cdot \mid \cdot)$, though defined in terms of a particular machine model, are machine-independent up to an additive constant and acquire an asymptotically universal and absolute character through Church's thesis, and from the ability of universal machines to simulate one another and execute any effective process.

The Kolmogorov complexity of an individual finite object was introduced by Kolmogorov [6] as an absolute and objective quantification of the amount of information in it. The information theory of Shannon [33], on the other hand, deals with *average* information to *communicate* objects produced by a *random source*. They are quite different.

Because the former theory is much more precise, it is surprising that analogues of theorems in information theory hold for Kolmogorov complexity, be it in somewhat weaker form. An interesting similarity is the following: $I(X;Y) = H(Y) - H(Y \mid X)$ is the (probabilistic) *information in random variable X about random variable Y*. Here, $H(Y \mid X)$ is the conditional entropy of Y given X. Because $I(X;Y) = I(Y;X)$, we call this symmetric quantity the *mutual (probabilistic) information*. The *(algorithmic) information in x about y* is $I(x:y) = K(y) - K(y \mid x)$, where x, y are finite objects such as finite strings or finite sets of finite strings. Remarkably, $I(x:y) = I(y:x)$ up to an additive term logarithmic in $K(x) + K(y)$. Namely

$$\begin{aligned} K(x,y) &= K(x) + K(y \mid x) + O(\log(K(x) + K(y))) \\ &= K(y) + K(x \mid y) + O(\log(K(x) + K(y))). \end{aligned} \tag{A 1}$$

A.3. Information distance

The *information distance* $D(x,y)$ between strings x and y is defined as

$$D(x,y) = \min_p \{|p| : U(p,x) = y \wedge U(p,y) = x\},$$

where U is the reference universal prefix machine above. Like the Kolmogorov complexity K, the distance function D is upper semi-computable. Define

$$E(x,y) = \max\{K(x \mid y), K(y \mid x)\}.$$

In reference [1], it is shown that the function E is upper semi-computable, $D(x,y) = E(x,y) + O(\log E(x,y))$, that the function E is a metric (more precisely, it satisfies the metric (in)equalities up to a constant), and that E is minimal (up to a constant) among all upper semi-computable distance functions D' satisfying the normalization conditions $\sum_{y:y\neq x} 2^{-D'(x,y)} \leq 1$ and $\sum_{x:x\neq y} 2^{-D'(x,y)} \leq 1$ (to exclude bogus distances, which state, for example, that every y is within distance $\frac{1}{2}$ of a given x). We call this metric E *universal*.

Thus, for every pair of finite files x, y, we have that $E(x,y)$ is at least as small as the smallest $D'(x,y)$. That is to say, $E(x,y)$ is at least as small as the distance engendered by the dominant feature shared between x and y. The NID is defined by

$$\mathrm{NID}(x,y) = \frac{E(x,y)}{\max\{K(x), K(y)\}}. \tag{A 2}$$

It is straightforward that $0 \leq \mathrm{NID}(x,y) \leq 1$ (up to an $O(1/\max\{K(x), K(y)\})$ additive term). It is a metric [2] (and so is the NCD of (2.1), see [4]). (A non-optimal precursor was given in Li *et al.* [39].)

Rewriting the NID using (A 1) yields

$$\mathrm{NID}(x,y) = \frac{K(x,y) - \min\{K(x),K(y)\}}{\max\{K(x),K(y)\}}, \tag{A 3}$$

up to some terms that we ignore. See reference [38] or [17] for reviews.

Appendix B. The normalized tree benefit score

The $S(T)$ value of a tree T is computed as follows. There are $\binom{n}{4}$ choices of four elements out of n. These are the *quartets*. Every quartet u,v,w,x can form three different *quartet topologies*: $uv\,|\,wx$, $uw\,|\,vx$ and $ux\,|\,vw$. Each such topology is a dendrogram. For example, $uv\,|\,wx$ is a dendrogram with two unlabelled internal nodes and u,v,w,x external nodes with u a sibling of v and w a sibling of x. Each topology has a *cost*. For example, the cost of $uv\,|\,wx$ is $C(uv\,|\,wx)$. (We set this cost usually at $C(uv\,|\,wx) = d(u,v) + d(w,x)$, where $d(u,v)$ is the distance between u and v. For us, usually $d(u,w) = \mathrm{NCD}(u,w)$.) The total cost C_T of a tree T with a set N of leaves (external nodes of degree 1) is defined as $C_T = \sum_{\{u,v,w,x\}\subseteq N}\{C_{uv|wx} : T \text{ is consistent with } uv\,|\,wx\}$—the sum of the costs of all its consistent quartet topologies. First, we generate a list of all possible quartet topologies for all four-tuples of labels under consideration. For each group of three possible quartet topologies for a given set of four labels u,v,w,x, calculate a best (minimal) cost $m(u,v,w,x) = \min\{C_{uv|wx}, C_{uw|vx}, C_{ux|vw}\}$ and a worst (maximal) cost $M(u,v,w,x) = \max\{C_{uv|wx}, C_{uw|vx}, C_{ux|vw}\}$. Summing all best quartet topologies yields the best (minimal) cost $m = \sum_{\{u,v,w,x\}\subseteq N} m(u,v,w,x)$. Conversely, summing all worst quartet topologies yields the worst (maximal) cost $M = \sum_{\{u,v,w,x\}\subseteq N} M(u,v,w,x)$. For some distance matrices, these minimal and maximal values cannot be attained by actual trees; however, the score C_T of every tree T will lie between these two values. In order to be able to compare tree scores in a more uniform way, we now rescale the score linearly such that the worst score maps to 0, and the best score maps to 1, and term this the *normalized tree benefit score* $S(T) = (M - C_T)/(M - m)$. Our goal is to find a full tree with a maximum value of $S(T)$, which is to say the lowest total cost. The $S(T)$ value states how well the quartet tree represents the underlying $n \times n$ distance matrix [4,10]. The range is $0 \le S(T) \le 1$, with 1 the best value. A random quartet tree has expected $S(T) = \frac{1}{3}$.

References

1. Bennett CH, Gács P, Li M, Vitányi PMB, Zurek W. 1998 Information distance. *IEEE Trans. Inf. Theory* **44**, 1407–1423. (doi:10.1109/18.681318)
2. Li M, Chen X, Li X, Ma B, Vitányi PMB. 2004 The similarity metric. *IEEE Trans. Inf. Theory* **50**, 3250–3264. (doi:10.1109/TIT.2004.838101)
3. Vereshchagin NK, Vitányi PMB. 2004 Kolmogorov's structure functions and model selection. *IEEE Trans. Inf. Theory* **50**, 3265–3290. (doi:10.1109/TIT.2004.838346)
4. Cilibrasi RL, Vitányi PMB. 2005 Clustering by compression. *IEEE Trans. Inf. Theory* **51**, 1523–1545. (doi:10.1109/TIT.2005.844059)
5. de Rooij S, Vitányi PMB. 2012 Approximating rate–distortion graphs of individual data: experiments in lossy compression and denoising. *IEEE Trans. Comput.* **61**, 395–407. (doi:10.1109/TC.2011.25)
6. Kolmogorov AN. 1965 Three approaches to the quantitative definition of information. *Problems Inf. Transm.* **1**, 1–7.
7. Cebrian M, Alfonseca M, Ortega A. 2007 The normalized compression distance is resistant to noise. *IEEE Trans. Inf. Theory* **53**, 1895–1900. (doi:10.1109/TIT.2007.894669)
8. Long C, Zhu X, Li M, Ma B. 2008 Information shared by many objects. In *Proc. 17th ACM Conf. on Information and Knowledge Management, Napa Valley, CA, 26–30 October*, pp. 1213–1220. New York, NY: ACM. (doi:10.1145/1458082.1458242)
9. Vitányi PMB. 2011 Information distance in multiples. *IEEE Trans. Inf. Theory* **57**, 2451–2456. (doi:10.1109/TIT.2011.2110130)
10. RL Cilibrasi, Vitányi PMB. 2011 A fast quartet tree heuristic for hierarchical clustering. *Pattern Recognit.* **44**, 662–677. (doi:10.1016/j.patcog.2010.08.033)

11. Cilibrasi RL. 2003 The CompLearn toolkit. (www.complearn.org)
12. Tzanetakis G, Cook P. 2002 Music genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **10**, 293–302. (doi:10.1109/TSA.2002.800560)
13. Grimaldi M, Kokaram A, Cunningham P. 2002 Classifying music by genre using the wavelet packet transform and a round-robin ensemble. Technical Report no. TCD-CS-2002-64, Trinity College Dublin.
14. Dannenberg R, Thom B, Watson D. 1997 A machine learning approach to musical style recognition. *Proc. 1997 Int. Comput. Music Conf.*, pp. 344–347. Ann Arbor, MI: MPublishing, University of Michigan Library. (http://quod.lib.umich.edu/i/icmc/bbp2372.1997.090/–machine-learning-approach-to-musical-style-recognition)
15. Chai W, Vercoe B. 2001 Folk music classification using hidden Markov models. *Proc. Int. Conf. Artif. Intell, Las Vegas, NV*. (http://en.scientificcommons.org/43297168)
16. Scott P. 2001 *Music classification using neural networks*. Manuscript, EE 373B Project, Stanford University, CA. (http://cuip.net/~dloquinte/researchfiles/musicclassification.pdf)
17. Vitányi PMB, Balbach FJ, Cilibrasi RL, Li M. 2009 Normalized information distance. In *Information theory and statistical learning* (eds F Emmert-Streib, M Dehmer), pp. 45–82. New York, NY: Springer.
18. Terwijn SA, Torenvliet L, Vitányi PMB. 2011 Nonapproximability of the normalized information distance. *J. Comput. Syst. Sci.* **77**, 738–742. (doi:10.1016/j.jcss.2010.06.018)
19. Cilibrasi RL, de Wolf R, Vitányi PMB. 2004 Algorithmic clustering of music based on string compression. *Comput. Music J.* **28**, 49–67. (doi:10.1162/0148926042728449)
20. Wehner S. 2007 Analyzing worms and network traffic using compression. *J. Comput. Security* **15**, 303–320. (http://dl.acm.org/citation.cfm?id=1370630)
21. Zhang X, Hao Y, Zhu X-Y, Li M. 2008 New information distance measure and its application in question answering system. *J. Comput. Sci. Technol.* **23**, 557–572. (doi:10.1007/s11390-008-9152-9)
22. Cleary JG, Witten IH. 1984 Data compression using adaptive coding and partial string matching. *IEEE Trans. Commun.* **32**, 396–402. (doi:10.1109/TCOM.1984.1096090)
23. Cebrian M, Alfonseca M, Ortega A. 2005 Common pitfalls using the normalized compression distance: what to watch out for in a compressor. *Commun. Inf. Syst.* **5**, 367–384. (http://projecteuclid.org/euclid.cis/1175791028)
24. Rannalal B, Yang Z. 2008 Phylogenetic inference using whole genomes. *Annu. Rev. Genomics Hum. Genet.* **9**, 217–231. (doi:10.1146/annurev.genom.9.081307.164407)
25. Cao Y, Janke A, Waddell PJ, Westerman M, Takenaka O, Murata S, Okada N, Pbo S, Hasegawa M. 1998 Conflict among individual mitochondrial proteins in resolving the phylogeny of Eutherian orders. *J. Mol. Evol.* **47**, 307–322. (doi:10.1007/PL00006389)
26. Keogh E, Lonardi S, Ratanamahatana CA. 2004 Towards parameter-free data mining. In *Proc. 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Seattle, WA, 22–24 August*, pp. 206–215. New York, NY: ACM. (doi:10.1145/1014052.1014077)
27. Ksiazek TG, *et al.* 2003 A novel coronavirus associated with severe acute respiratory syndrome. *New Engl. J. Med.* **348**, 1953–1956. (doi:10.1056/NEJMoa030781)
28. Costa Santos C, Bernardes J, Vitányi PMB, Antunes L. 2006 Clustering fetal heart rate tracings by compression. In *Proc. 19th IEEE Int. Symp. on Computer-Based Medical Systems, Salt Lake City, UT, 22–23 June*, pp. 685–690. Los Alamitos, CA: IEEE Computer Society Press. (doi:10.1109/CBMS.2006.68)
29. Belloni T, Klein-Wolt M, Méndez M, van der Klis M, van Paradijs J. 2000 A model-independent analysis of the variability of GRS 1915+105. *Astron. Astrophys.* **355**, 271–290. (http://adsabs.harvard.edu/full/2000A%26A...355..271B)
30. Cilibrasi RL, Vitányi PMB. 2007 The Google similarity distance. *IEEE Trans. Knowledge Data Eng.* **19**, 370–383. (doi:10.1109/TKDE.2007.48)
31. Chang C-C, Lin C-J. 2011 LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2** (3), Article 27. (doi:10.1145/1961189.1961199)
32. Oliveira LS, Sabourin R, Bortolozzi F, Suen CY. 2002 Automatic recognition of handwritten numerical strings: a recognition and verification strategy. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 1438–1454. (doi:10.1109/TPAMI.2002.1046154)
33. Shannon CE. 1948 The mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 623–656.
34. Shannon CE. 1959 Coding theorems for a discrete source with a fidelity criterion. In *IRE National Convention Record, Part 4* (*Coding Theorems*), pp. 142–163. Institute of Radio Engineers.

35. Vereshchagin NK, Vitányi PMB. 2010 Rate distortion and denoising of individual data using Kolmogorov complexity. *IEEE Trans. Inf. Theory* **56**, 3438–3454. (doi:10.1109/TIT.2010.2048491)
36. Levenshtein VI. 1966 Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **10**, 707–710.
37. Turing AM. 1936 On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **42**, 230–265; 'Correction' **43** (1937), 544–546.
38. Li M, Vitányi PMB. 2008 *An introduction to Kolmogorov complexity and its applications*, 3rd edn. New York, NY: Springer.
39. Li M, Badger JH, Chen X, Kwong S, Kearney P, Zhang H. 2001 An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics* **17**, 149–154. (doi:10.1093/bioinformatics/17.2.149)