Compression-based Similarity

Paul M.B. Vitányi

CWI, Amsterdam, The Netherlands

(*Invited Lecture*)

Abstract

First we consider pair-wise distances for literal objects consisting of finite binary files. These files are taken to contain all of their meaning, like genomes or books. The distances are based on compression of the objects concerned, normalized, and can be viewed as similarity distances. Second, we consider pair-wise distances between names of objects, like "red" or "christianity." In this case the distances are based on searches of the Internet. Such a search can be performed by any search engine that returns aggregate page counts. We can extract a code length from the numbers returned, use the same formula as before, and derive a similarity or relative semantics between names for objects. The theory is based on Kolmogorov complexity. We test both similarities extensively experimentally.

I. Introduction

In pattern recognition, learning, and data mining one obtains information from information-carrying objects. This involves an objective definition of the information in a single object, the information to go from one object to another object in a pair of objects, the information to go from one object to any other object in a multiple of objects, and the shared information between objects.

The notion of Kolmogorov complexity [15] is an objective measure for the information in an a *single* object, and information distance measures the information between a *pair* of objects [1]. This leads to the notion of similarity we shall explore below.

Objects can be given literally, like the literal four-letter genome of a mouse, or the literal text of *War and Peace*

Affiliation: National Research Center for Mathematics and Computer Science in the Netherlands (CWI). Address: CWI, Science Park 123, 1098XG Amsterdam, The Netherlands. Email: Paul.Vitanyi@cwi.nl

by Tolstoy. For simplicity we take it that all meaning of the object is represented by the literal object itself. Objects can also be given by name, like "the four-letter genome of a mouse," or "the text of *War and Peace* by Tolstoy." There are also objects that cannot be given literally, but only by name and acquire their meaning from their contexts in background common knowledge in humankind, like "home" or "red." In the literal setting, similarity of objects can be established by feature analysis, one type of similarity per feature. In the abstract "name" setting, all similarity must depend on background knowledge and common semantics relations, which is inherently subjective and "in the mind of the beholder."

REMARK 1. As an aside, in many applications we are interested in shared information between *many* 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 [20] while most of the theory is developed in [25]. ◇

II. Similarity

All data are created equal but some data are more alike than others. We have proposed methods expressing this alikeness, using a new similarity metric based on compression. It is parameter-free in that it doesn't use any features or background knowledge about the data, and can without changes be applied to different areas and across area boundaries. It is universal in that it approximates the parameter expressing similarity of the dominant feature in all pairwise comparisons. It is robust in the sense that its success appears independent from the type of compressor used (among equally good compressors). The clustering

we use is hierarchical clustering in dendrograms based on a new fast heuristic for the quartet method [11]. If we consider n objects, then we find n^2 pairwise distances. These distances are between natural data. We let the data decide for themselves, and construct a hierarchical clustering of the n objects concerned. For details see the cited reference. 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. We assume $n \geq 4$. The method is available as an open-source software tool, [6].

A. Feature-Based Similarities

We are presented with unknown data and the question is to determine the similarities among them and group like with like together. Commonly, the data are of a certain type: music files, transaction records of ATM machines, credit card applications, genomic data. In these data there are hidden relations that we would like to get out in the open. 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 etc. One can extract such features using for instance Fourier transforms [23] or wavelet transforms [13], to quantify parameters expressing similarity. The resulting vectors corresponding to the various files are then classified or clustered using existing classification software, based on various standard statistical pattern recognition classifiers [23], Bayesian classifiers [12], hidden Markov models [7], ensembles of nearest-neighbor classifiers [13] or neural networks [12], [21]. For example, in music one feature would be to look for rhythm in the sense of beats per minute. One can make a histogram where each histogram bin corresponds to a particular tempo in beats-per-minute and the associated peak shows how frequent and strong that particular periodicity was over the entire piece. In [23] we see a gradual change from a few high peaks to many low and spread-out ones going from hip-hip, rock, jazz, to classical. One can use this similarity type to try to cluster pieces in these categories. However, such a method requires specific and detailed knowledge of the problem area, since one needs to know what features to look for.

B. Non-feature Similarities

Our aim is to capture, 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. This metric should be so general that it works in every domain: music, text, literature, programs, genomes, executables, natural language determination, equally and simultaneously. It would be able to simultaneously detect *all* similarities between pieces that other effective distances can detect seperately.

Such a "universal" metric was co-developed by us as a normalized version of the "information metric" of [19], [1]. There it was shown that the information metric minorizes up to a constant all effective distances satisfying a mild density requirement (excluding for example distances that are 1 for every pair x,y such that $x \neq y$). This justifies the notion that the information distance is universal.

We may be interested what happens in terms of properties or features of the pair of objects analyzed, say x and y. It can be shown that the information distance captures every property of which the Kolmogorov complexity is logarithmic in the length of $\min\{|x|,|y|\}$. If those lengths go to infinity, then logarithm of those lengths go to infinity too. In this case the information distance captures every property.

This information distance (actually a metric) is normalized so that the resulting distances are in $[0,1]$ and can be shown to retain the metric property, [18]. (A nonoptimal precursor was given in [17].) The result is the "normalized information distance" (actually a metric). All this is in terms of Kolmogorov complexity [19].

Intuitively, 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 normalized information distance 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 normalized information distance.

Put differently, the normalized information distance 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 normalized information metric distance zooms in on the dominant similarity between those two objects out of a wide class of admissible similarity features. Since the normalized information distance also satisfies the metric (in)equalities, and takes values in $[0,1]$, it may be called *"the" similarity metric*.

C. Normalized Compression Distance

Unfortunately, the universality of the normalized information distance comes at the price of noncomputability. In fact, the normalized information distance is not even semicomputable (this is weaker than computable) and there

is no semicomputable function at a computable distance of it [24]. But since the Kolmogorov complexity of a string or file is the length of the ultimate compressed version of that file, we can use real data compression programs to approximate the Kolmogorov complexity. Therefore, to apply this ideal precise mathematical theory in real life, we have to replace the use of the noncomputable Kolmogorov complexity by an approximation using a standard real-world compressor. Starting from the normalized information distance, 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*:

$$NCD(x,y) = \frac{Z(xy) - \min(Z(x), Z(y))}{\max(Z(x), Z(y))}, \qquad (1)$$

where for convenience we have replaced the pair (x,y) in the formula by the concatenation xy, and we ignore logarithmic terms in the numerator and denominator, see [18], [9]. In [9] we propose axioms to capture the real-world setting, and show that (1) approximates optimality. Actually, the NCD is a family of compression functions parameterized by the given data compressor Z.

EXAMPLE 1. (PHYLOGENY) One cannot find more appropriate data than DNA sequences to test our theory. A DNA sequence is a finite string over a 4-letter alphabet $\{A, C, G, T\}$. We used the entire mitochondrial genomes of 20 mammals, each of about 18,000 base pairs, to test a hypothesis about the Eutherian orders. 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 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.

In recent years, as a result of more sophisticated methods, together with biological evidence, it is believed that (rodents, (ferungulates, primates)) reflects the true evolutionary history. We confirm this from the whole-genome perspective using the NCD distance. We use the complete mitochondrial genome sequences from the following 20 species: rat (*Rattus norvegicus*), house mouse (*Mus musculus*), gray (or grey) seal (*Halichoerus grypus*), harbor seal (*Phoca vitulina*), cat (*Felis catus*), white rhino (*Ceratotherium simum*), horse (*Equus caballus*), finback whale (*Balaenoptera physalus*), blue whale (*Balaenoptera musculus*), cow (*Bos taurus*), gibbon (*Hylobates lar*), gorilla (*Gorilla gorilla*), human (*Homo sapiens*), chimpanzee (*Pan*
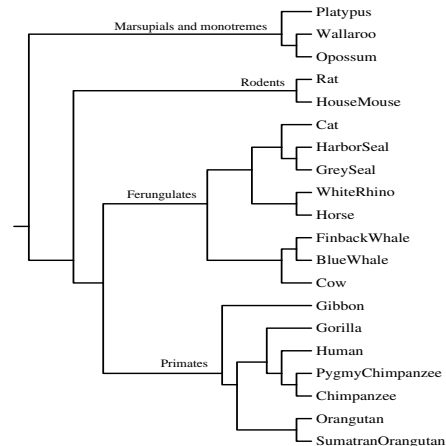

Fig. 1. The evolutionary tree built from complete mammalian mtDNA sequences

troglodytes), pygmy chimpanzee (*Pan paniscus*), orangutan (*Pongo pygmaeus*), Sumatran orangutan (*Pongo pygmaeus abelii*), with opossum (*Didelphis virginiana*), wallaroo (*Macropus robustus*), and platypus (*Ornithorhynchus anatinus*) as the outgroup.

For every pair of mitochondrial genome sequences x and y, evaluate the formula in Equation 1 using a special-purpose DNA sequence compressor DNACompress, or a good general-purpose 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 common tree-reconstruction software, gives the tree in Figure 1. This tree confirms the accepted hypothesis of (rodents, (primates, ferungulates)), and every single branch of the tree agrees with the current biological classification. ◊

EXAMPLE 2. (HIERARCHICAL CLUSTERING) The normalized compression distance has been used to fully automatically reconstruct language and phylogenetic trees as above. It can, and has, also be used for a plethora of new applications of general clustering and classification of natural data in arbitrary domains, for clustering of heterogeneous data, and for anomaly detection across domains. It has further been applied to authorship attribution, stemmatology, music classification, Internet knowledge discovery, to analyze network traffic and cluster computer worms and viruses, software metrics and obfuscation, web page authorship, topic and domain identification, hurricane risk assessment, ortholog detection, and clustering fetal heart rate tracings. 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 worldwide web; (ii) four excerpts from the novel *The Zeppelin's Passenger* by E. Phillips Oppenheim, obtained from the Project Gutenberg Edition on the worldwide web; (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 worldwide web; (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
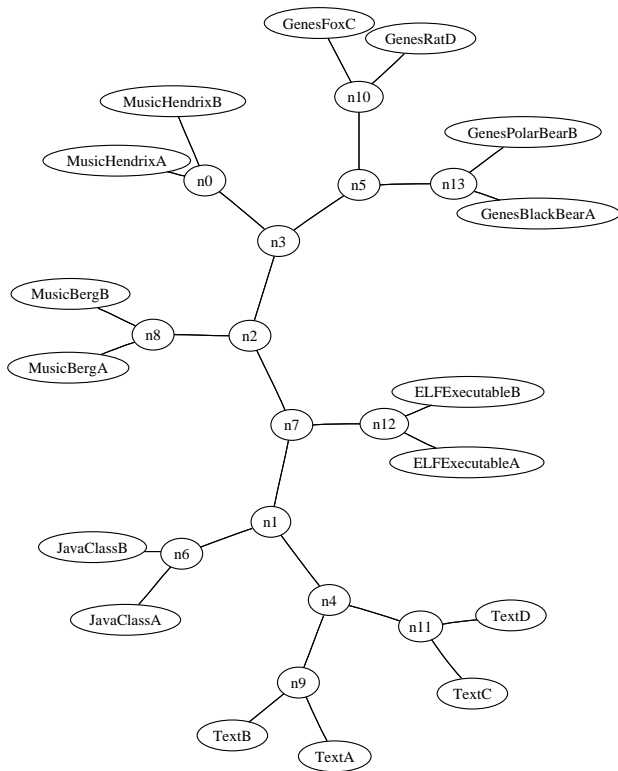

Fig. 2. Clustering of heterogeneous file types

this methodology is in fact a parameter-free, feature-free, data-mining tool. They have experimentally tested a closely related metric on a large variety of sequence benchmarks. Comparing the compression-based method

with 51 major parameter-loaded methods found in the 7 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, for anomaly detection, and competitiveness in clustering domain data [14]. ◊

D. NCD in Retrospect

In [9] it is proved that the NCD is a metric. The compression-based NCD method establishes a similarity metric (1) among objects given as finite binary strings. It has been applied to objects like genomes, music pieces in MIDI format, computer programs in Ruby or C, pictures in simple bitmap formats, or time sequences such as heart rhythm data, heterogenous data and anomaly detection. This method is feature-free in the sense that it doesn't analyze the files looking for particular features; rather it analyzes all features simultaneously and determines the similarity between every pair of objects according to the most dominant shared feature.

The crucial point is that the method analyzes the objects themselves. This precludes comparison of abstract notions or other objects that don't lend themselves to direct analysis, like emotions, colors, Socrates, Plato, Mike Bonanno and Albert Einstein.

III. Web-based Similarity

To make computers more intelligent one would like to represent meaning in computer-digestable form. Long-term and labor-intensive efforts like the *Cyc* project [16] and the *WordNet* project [22] try to establish semantic relations between common objects, or, more precisely, *names* for those objects. The idea is to create a semantic web of such vast proportions that rudimentary intelligence and knowledge about the real world spontaneously emerges. This comes at the great cost of designing structures capable of manipulating knowledge, and entering high quality contents in these structures by knowledgeable human experts. While the efforts are long-running and large scale, the overall information entered is minute compared to what is available on the world-wide-web.

The rise of the world-wide-web has enticed millions of users to type in trillions of characters to create billions of web pages of on average low quality contents. The sheer mass of the information available about almost every conceivable topic makes it likely that extremes will cancel and the majority or average is meaningful in a low-quality approximate sense. Below, we give a general method to tap the amorphous low-grade knowledge available for free

on the world-wide-web, typed in by local users aiming at personal gratification of diverse objectives, and yet globally achieving what is effectively the largest semantic electronic database in the world. Moreover, this database is available for all by using any search engine that can return aggregate page-count estimates like Google for a large range of search-queries.

While the previous NCD method that compares the objects themselves using (1) is particularly suited to obtain knowledge about the similarity of objects themselves, irrespective of common beliefs about such similarities, we now develop a method that uses only the name of an object and obtains knowledge about the similarity of objects by tapping available information generated by multitudes of web users. The new method is useful to extract knowledge from a given corpus of knowledge, in this case the world-wide-web accessed by a search engine returning aggregate page counts, but not to obtain true facts that are not common knowledge in that database. For example, common viewpoints on the creation myths in different religions may be extracted by the web-based method, but contentious questions of fact concerning the phylogeny of species can be better approached by using the genomes of these species, rather than by opinion. The approach below was proposed by [10].

A. The Search Distribution

Let the set of singleton *search terms* be denoted by $\mathcal{S}$. In the sequel we use both singleton search terms and doubleton search terms $\{\{x,y\} : x, y \in \mathcal{S}\}$. Let the set of web pages indexed (possible of being returned) by search engine be Ω. The cardinality of Ω is denoted by $M = |\Omega|$. Assume that a priori all web pages are equiprobable, with the probability of being returned being $1/M$. A subset of Ω is called an *event*. Every *search term* x defines a *singleton event* $\mathbf{x} \subseteq \Omega$ of web pages that contain an occurrence of x and are returned by the search engine if we do a search for x. Let $L : \Omega \to [0,1]$ be the uniform mass probability function. The probability of such an event $\mathbf{x}$ is $L(\mathbf{x}) = |\mathbf{x}|/M$. Similarly, the *doubleton event* $\mathbf{x} \cap \mathbf{y} \subseteq \Omega$ is the set of web pages returned if we do a search for pages containing both search term x and search term y. The probability of this event is $L(\mathbf{x} \cap \mathbf{y}) = |\mathbf{x} \cap \mathbf{y}|/M$. We can also define the other Boolean combinations: $\neg\mathbf{x} = \Omega \backslash \mathbf{x}$ and $\mathbf{x} \cup \mathbf{y} = \Omega \backslash (\neg\mathbf{x} \cap \neg\mathbf{y})$, each such event having a probability equal to its cardinality divided by M. If $\mathbf{e}$ is an event obtained from the basic events $\mathbf{x}, \mathbf{y}, \ldots$, corresponding to basic search terms $x, y, \ldots$, by finitely many applications of the Boolean operations, then the probability $L(\mathbf{e}) = |\mathbf{e}|/M$.

These events capture in a particular sense all background knowledge about the search terms concerned available (to search engine) on the web. Therefore, it is nat-

ural to consider code words for those events as coding[5] this background knowledge. However, we cannot use the probability of the events directly to determine a prefix code such as the Shannon-Fano code [19]. The reason is that the events overlap and hence the summed probability exceeds 1. By the Kraft inequality [19] this prevents a corresponding Shannon-Fano code. The solution is to normalize: We use the probability of the events to define a probability mass function over the set $\{\{x,y\} : x, y \in \mathcal{S}\}$ of search terms, both singleton and doubleton. Define

$$N = \sum_{\{x,y\} \subseteq \mathcal{S}} |\mathbf{x} \cap \mathbf{y}|,$$

counting each singleton set and each doubleton set (by definition unordered) once in the summation. Since every web page that is indexed by contains at least one occurrence of a search term, we have $N \geq M$. On the other hand, web pages contain on average not more than a certain constant α search terms. Therefore, $N \leq \alpha M$. Define

$$g(x) = L(\mathbf{x})M/N = |\mathbf{x}|/N \qquad (2)$$
$$g(x,y) = L(\mathbf{x} \cap \mathbf{y})M/N = |\mathbf{x} \cap \mathbf{y}|/N.$$

Then, $\sum_{x \in \mathcal{S}} g(x) + \sum_{x,y \in \mathcal{S}} g(x,y) = 1$. Note that $g(x,y)$ is not a conventional joint distribution since possibly $g(x) \neq \sum_{y \in \mathcal{S}} g(x,y)$. Rather, we consider g to be a probability mass function over the sample space $\{\{x,y\} : x, y \in \mathcal{S}\}$. This g-distribution changes over time, and between different samplings from the distribution. But let us imagine that g holds in the sense of an instantaneous snapshot. The real situation will be an approximation of this. Given the search machinery, these are absolute probabilities which allow us to define the associated Shannon-Fano code for both the singletons and the doubletons.

B. Normalized Web Distance

The *web code* length G is defined by

$$G(x) = \log 1/g(x) \qquad (3)$$
$$G(x,y) = \log 1/g(x,y).$$

In contrast to strings x where the complexity $Z(x)$ represents the length of the compressed version of x using compressor Z, for a search term x (just the name for an object rather than the object itself), the code of length $G(x)$ represents the shortest expected prefix-code word length of the associated event $\mathbf{x}$. The expectation is taken over the distribution p. In this sense we can use the distribution as a compressor for "meaning" associated with the search terms. The associated *normalized web distance* (NWD) is defined just as (1) with the search engine in the role of compressor yielding code lengths $G(x), G(y)$ for the

singleton search terms x, y being compaired and a code length $G(x, y)$ for the doubleton pair (x, y), by

$$NWD(x, y) = \frac{G(x, y) - \min(G(x), G(y))}{\max(G(x), G(y))}. \quad (4)$$

This *NWD* uses the background knowledge on the web as viewed by the search engine as conditional information.

C. Searching the Web for Knowledge

Many web search engines index more than twenty-five billion pages on the web. Each such page can be viewed as a set of index terms. A search for a particular index term (in 2004) say "horse", returned a certain number of hits (web pages where this term occurred), say 46,700,000. The number of hits for the search term "rider" is, say, 12,200,000. It is also possible to search for the pages where both "horse" and "rider" occur. This gives, say, 2,630,000 hits. The search engine searched at that time, say, 8,058,044,651 web pages. The same formula as (4) can be written in terms of frequencies as

$$NWD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}, \quad (5)$$

and if $f(x), f(y) > 0$ and $f(x, y) = 0$ then $NWD(x, y) = \infty$. It is easy to see that

1) $NWD(x, y)$ is undefined for $f(x) = f(y) = 0$;
2) $NWD(x, y) = \infty$ for $f(x, y) = 0$ and either or both $f(x) > 0$ and $f(y) > 0$; and
3) $NWD(x, y) \geq 0$ otherwise.

Our experimental results suggest that every reasonable (greater than any $f(x)$) value can be used for the normalizing factor N, and our results seem in general insensitive to this choice. In our software, this parameter N can be adjusted as appropriate, and we often use M for N. In the [10] we analyze the mathematical properties of NWD, and prove the universality of the search engine distribution. We show that the NWD is not a metric, in contrast to the NCD. The generic example showing the nonmetricity of semantics (and therefore the NWD) is that a man is close to a centaur, and a centaur is close to a horse, but a man is very different from a horse.

With the hit numbers above, using M for N, we can compute from (4) that

$$NWD(horse, rider) \approx 0.443.$$

The NWD formula itself (5) is *scale-invariant* in the sense that if M doubles and so do the f-frequencies then the result stays the same.

EXAMPLE 3. (CLASSIFICATION) In cases in which the set of objects can be large, in the millions, clustering cannot do us much good. We may also want to do definite classification, rather than the more fuzzy clustering. One can use the NCD/NWD distances as an oblivious feature-extraction technique to convert generic objects into finite-dimensional vectors.

(We have used this technique to train a support vector machine (SVM) based OCR system to classify handwritten digits by extracting 80 distinct, ordered NCD features from each input image in the manner explained below in the context of the NWD experiments. For details about the SVM see [3]. We achieved a handwritten single decimal digit recognition accuracy of 87%. The current state of the art for this problem, after half a century of interactive feature-driven classification research, is in the upper ninety percent level. These experiments were benchmarked on the standard NIST Special Data Base 19.)

For classification using the NWD distance, the setting is, say, a binary classification problem on examples represented by search terms. In this experiment, we require a human expert to provide a list of at least 40 *training words*, consisting of at least 20 positive examples and 20 negative examples, to illustrate the contemplated concept class. The expert also provides, say, six *anchor words* $a_1, \ldots, a_6$, of which half are in some way related to the concept under consideration. Then, we use the anchor words to convert each of the 40 training words $w_1, \ldots, w_{40}$ to 6-dimensional *training vectors* $\bar{v}_1, \ldots, \bar{v}_{40}$. The entry $v_{j,i}$ of $\bar{v}_j = (v_{j,1}, \ldots, v_{j,6})$ is defined as $v_{j,i} = NWD(w_j, a_i)$ $(1 \leq j \leq 40, 1 \leq i \leq 6)$. 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 [4].

In an experiment to learn prime numbers, we used the literal search terms below (digital numbers and alphabetical words) in the Google search engine.
Positive training examples: 11, 13, 17, 19, 2, 23, 29, 3, 31, 37, 41, 43, 47, 5, 53, 59, 61, 67, 7, 71, 73.
Negative training examples: 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 4, 6, 8, 9.
Anchor words: composite, number, orange, prime, record.
Unseen test examples: The numbers 101, 103, 107, 109, 79, 83, 89, 97 were correctly classified as primes. The numbers 36, 38, 40, 42, 44, 45, 46, 48, 49 were correctly classified as nonprimes. The numbers 91 and 110 were false positives, since they were incorrectly classified as primes. There were no false negatives. The accuracy on the test set is $17/19 = 89.47\%$. Thus, the method automatically learns to distinguish prime numbers from nonprime numbers by example, using a search engine that knows nothing about mathematics. Note that in this example we have to keep the numbers under, say, 200, since larger numbers do not necessarily occur on the web in the required multitude.

The NWD has been used for clustering, classification,

and translation of small samples [10]. That reference reports a massive experiment comparing the performance of the NWD–SVM method with the human-expert-entered information in the WordNet database [22]. They showed a mean accuracy of agreement of 87.25% of the NWD–SVM method with the WordNet semantic concordance.   ◇

IV. Conclusion

By now applications abound. See the many references to the papers [18], [9], [10] in Google Scholar. Below we treat some favorite applications we are fimiliar with out of this multitude.

APPLICATIONS OF NCD: We and others performed experiments in vastly different application fields to test the quality and universality of the method. The success of the method as reported below depends strongly on the judicious use of encoding of the objects compared. Here one should use common sense on what a real world compressor can do. There are situations where our approach fails if applied in a straightforward way. For example: comparing text files by the same authors in different encodings (say, Unicode and 8-bit version) is bound to fail. For the ideal similarity metric based on Kolmogorov complexity as defined in [18] this does not matter at all, but for practical compressors used in the experiments it will be fatal. Similarly, in the music experiments we use symbolic MIDI music file format rather than wave format music files. The reason is that the strings resulting from straightforward discretizing the wave form files may be too sensitive to how we discretize. Further research may overcome this problem.

The method is implemented and available as public software [6]. This approach gives the first completely automatic construction of the phylogeny tree based on whole mitochondrial genomes (above), and a completely automatic construction of a language tree for over 50 Euro-Asian languages [18], detects plagiarism in student programming assignments [5], gives phylogeny of chain letters [2], and clusters music [8]. Moreover, the method turns out to be more-or-less robust under change of the underlying compressor-types: statistical (PPMZ), Lempel-Ziv based dictionary (gzip), block based (bzip2), or special purpose (Gencompress). Obviously the window size matters, as well as how good the compressor is. For example, PPMZ gives for mtDNA of the investigated species diagonal elements ($NCD(x,x)$) between 0.002 and 0.006. The compressor bzip2 does considerably worse, and gzip gives something in between 0.5 and 1 on the diagonal elements. Nonetheless, for texts like books gzip does fine in our experiments; the window size is sufficient and we do not use the diagonal elements. But for genomics gzip is no good.

In [9] we report evidence of successful application in areas as diverse as genomics, virology, languages, literature, music, handwritten digits, astronomy, and combinations of objects from completely different domains (see example above), using statistical, dictionary, and block sorting compressors. In genomics we presented new evidence for major questions in Mammalian evolution, based on whole-mitochondrial genomic analysis: the Eutherian orders (we gave this example above) and the Marsupionta hypothesis against the Theria hypothesis. Apart from the experiments reported in [9], the clustering by compression method reported in this paper has recently been used in many different areas all over the world. For example, how to analyze network traffic and cluster computer worms and virusses [26].

APPLICATIONS OF NWD: This method is proposed in [10] to extract semantic knowledge from the world-wide-web for both supervised and unsupervised learning using a search engine in an unconventional manner. The approach is novel in its unrestricted problem domain, simplicity of implementation, and manifestly ontological underpinnings.

Evidence is given of elementary learning of the semantics of concepts, in contrast to most prior approaches (outside of Knowledge Representation research) that have neither the appearance nor the aim of dealing with ideas, instead using abstract symbols that remain permanently ungrounded throughout the machine learning application.

The world-wide-web is the largest database on earth, and it induces a probability mass function via page counts for search queries. This distribution allows us to tap the latent semantic knowledge on the web. While in the NCD compression-based method one deals with the objects themselves, in the NWD method we deal with just names for the objects.

In [10], as proof of principle, we demonstrate positive correlations, evidencing an underlying semantic structure, in both numerical symbol notations and number-name words in a variety of natural languages and contexts. Next, we give applications in (i) unsupervised hierarchical clustering, demonstrating the ability to distinguish between colors and numbers, and to distinguish between 17th century Dutch painters; (ii) supervised concept-learning by example, using support vector machines, demonstrating the ability to understand electrical terms, religious terms, emergency incidents, and by conducting a massive experiment in understanding WordNet categories; and (iii) matching of meaning, in an example of automatic English-Spanish translation.

An application that uses both the NCD and the NWD (and some derived distance measures) is the competitive question-answer system in [27].

References

[1] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, W. Zurek, Information Distance, *IEEE Trans. Information Theory*, 44:4(1998), 1407–1423.

[2] C.H. Bennett, M. Li, B. Ma, Chain letters and evolutionary histories, *Scientific American*, June 2003, 76–81.

[3] C.J.C. Burges. A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2:2(1998),121–167.

[4] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[5] X. Chen, B. Francia, M. Li, B. McKinnon, A. Seker, Shared information and program plagiarism detection, *IEEE Trans. Inform. Th.*, 50:7(2004), 1545–1551.

[6] R.L. Cilibrasi, The CompLearn Toolkit, 2003–, www.complearn.org

[7] W. Chai and B. Vercoe. Folk music classification using hidden Markov models. *Proc. Intl Conf. Art. Intell.*, 2001.

[8] R.L. Cilibrasi, R. de Wolf, P.M.B. Vitányi. Algorithmic clustering of music based on string compression, *Computer Music J.*, 28:4(2004), 49-67.

[9] R.L. Cilibrasi, P.M.B. Vitányi, Clustering by compression, *IEEE Trans. Information Theory*, 51:4(2005), 1523–1545.

[10] R.L. Cilibrasi, P.M.B. Vitányi, *IEEE Trans. Knowledge and Data Engineering*, 19:3(2007), 370–383.

[11] R.L. Cilibrasi, P.M.B. Vitányi, A fast quartet tree heuristic for hierarchical clustering, *Pattern Recognition*, 44 (2011) 662-677

[12] R. Dannenberg, B. Thom, and D. Watson, A machine learning approach to musical style recognition, *Proc. Intl Comput. Music Conf.*, pp. 344–347, 1997.

[13] M. Grimaldi, A. Kokaram, P. Cunningham, Classifying music by genre using the wavelet packet transform and a round-robin ensemble. Technical report TCD-CS-2002-64, Trinity College Dublin, 2002.

[14] E. Keogh, S. Lonardi, and C.A. Rtanamahatana, Toward parameter-free data mining, In: *Proc. 10th ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining*, pp. 206–215, 2004.

[15] A.N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems Inform. Transmission*, 1:1(1965), 1–7.

[16] D.B. Lenat, Cyc: A large-scale investment in knowledge infrastructure, *Comm. ACM*, 38:11(1995),33–38.

[17] M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, 17:2(2001), 149–154.

[18] M. Li, X. Chen, X. Li, B. Ma, P.M.B. Vitányi. The similarity metric, *IEEE Trans. Information Theory*, 50:12(2004), 3250- 3264.

[19] M. Li, P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, 3nd Ed., Springer-Verlag, New York, 2008.

[20] M. Li, C. Long, B. Ma, X. Zhu, Information shared by many objects, Proc. 17th ACM Conf. Inform. Knowl. Management, 2008, 1213–1220.

[21] P. Scott. Music classification using neural networks, Manuscript Class ee373a, Stanford, 2001.

[22] G.A. Miller et.al, WordNet, A Lexical Database for the English Language, Cognitive Science Lab, Princeton University.

[23] G. Tzanetakis, P. Cook, Music genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing*, 10:5(2002), 293–302.

[24] S.A. Terwijn, L. Torenvliet, P.M.B. Vitányi, Nonapproximability of the Normalized Information Distance, *J. Comput. System Sciences*, 77:4(2011), 738–742.

[25] P.M.B. Vitányi, Information distance in multiples, *IEEE Trans. Inform. Theory*, 57:4(2011), 2451-2456.

[26] S. Wehner, Analyzing worms and network traffic using compression, *J. Comput. Security*, 15:3(2007), 303–320.

[27] X. Zhang, Y. Hao, X.-Y. Zhu, M. Li, New Information Distance Measure and Its Application in Question Answering System, *J. Comput. Sci. Techn.*, 23:4(2008), 557–572.