Exact Expression For Information Distance [1]

Paul M.B. Vitányi

Abstract

Information distance can be defined not only between two strings but also in a finite multiset of strings of cardinality greater than two. We give an elementary proof for expressing the information distance. It is exact since for each cardinality of the multiset the lower bound for some multiset equals the upper bound for all multisets up to a constant additive term. We discuss overlap.

Index Terms— Information distance, multiset, Kolmogorov complexity, similarity, pattern recognition, data mining.

I. Introduction

The length of a shortest binary program to compute from one object to another object and vice versa expresses the amount of information that separates the objects. This is a proper distance [6, p. 205], is (almost) a metric and spawned theoretic issues. Normalized in the appropriate manner it quantifies a similarity between objects [10], [4], [5] and is now widely used in pattern recognition [2], learning [3], and data mining [8]. Extending this approach we can ask how much the objects in a set of objects are alike, that is, the common information they share. All objects we discuss are represented as finite binary strings. We use Kolmogorov complexity [9]. Informally, the Kolmogorov complexity of a string is the length of a shortest binary program from which the string can be computed. Therefore it is a lower bound on the length of a compressed version of that string for any current or future computer. The text [12] introduces the notions, develops the theory, and presents applications.

We write *string* to denote a finite binary string. Other finite objects, such as multisets of strings (a generalization of the notion of a set where each member can occur more than once), may be encoded into single strings in natural ways. The length of a string x is denoted by $|x|$. Let X be a finite multiset of strings ordered length-increasing lexicographic. In this paper $|X| \geq 2$. Examples are $X = \{x, x\}$ and $X = \{x, y\}$ with $x \neq y$. The logarithms are binary throughout.

Paul Vitányi is with the Center for Mathematics and Computer Science (CWI), and the University of Amsterdam. Address: CWI, Science Park 123, 1098XG Amsterdam, The Netherlands. Email: Paul.Vitanyi@cwi.nl.

Let U be a fixed *optimal universal prefix Turing machine* for which the programs are binary. The set of programs for such a machine is a prefix code (no program is a proper prefix of another program). Since computability is involved, such a program is called *self-delimiting*. The minimal length of a self-delimiting program computing a string x is the *prefix Kolmogorov complexity* $K(x)$ of that string. We can define $K(X)$ as the length of a shortest self-delimiting program p computing all the members of X and a means to tell them apart. Similarly we define $K(X|x)$.

A. Related Work

In the seminal [1] the information distance $ID(x,y)$ between pairs of strings x and y was introduced as the length of a shortest binary program p for the reference optimal universal prefix Turing machine U such that $U(p,x)=y$ and $U(p,y)=x$. It was shown that $ID(x,y)=\max\{K(x|y),K(y|x)\}+O(\log\max\{K(x|y),K(y|x)\})$. In [13] it was proven how to reduce the $O(\log\max\{K(x|y),K(y|x)\})$ additive term to $O(1)$. In [11] the information distance $ID(x_1,\ldots,x_n)$ between a multiset of strings $(x_1,\ldots,x_n)$ was introduced as the length of a shortest binary program p for U such that $U(p,x_i,j)=x_j$ for all $1\leq i,j\leq n$. It was shown that $ID(x_1,\ldots,x_n)$ requires at most $\max_{1\leq i\leq n}\{K(x_1,\ldots,x_n|x_i)\}$ plus $2\log n$ bits of information. Coding this self-delimiting uses an additional logarithmic number of bits. However, more precise reasoning allows to dispense with this. Note that this also reduces the $O(\log\max\{K(x|y),K(y|x)\})$ additive term to $O(1)$ for $n=2$. In [18] information distance is made uniform by denoting $X=\{x_1,\ldots,x_n\}$ and defining $ID(X)$ as the length of a shortest program to compute X from any $x\in X$. If a program computes from every $x\in X$ to any $y\in X$ then it must compute X on the way and specify additionally only the index of $y\in X$. (The proof in [11] ignores this additional item anyway.) The essence is to compute X. Formally,

$$ID(X)=\min\{|p|:U(p,x)=X \text{ for all } x\in X\}.$$

B. Results

Let X be a multiset of strings of finite cardinality greater or equal two. The information distance of X is $ID(X)$ and can be viewed as a *diameter* of X. For $|X|=2$ it is a conventional distance between the two members of X. Since it is a metric [18] the name "distance" seems justified. Since the 1990s it was perceived as a nuisance and a flaw that equality between $ID(X)$ and $\max_{x\in X}\{K(X|x)\}$ held only up to an $O(\log\max_{x\in X}\{K(X|x)\})$ additive term. We give an elementary proof that for all X holds $ID(X)\leq\max_{x\in X}\{K(X|x)\}+\log|X|$ up to a constant additive term, and for every cardinality of X

there are X's such that $ID(X) \geq \max_{x \in X}\{K(X|x)\} + \log(|X| - 1)$ provided the number of different halting self-delimiting programs (possibly with input) for U of length at most $\max_{x \in X}\{K(X|x)\}$ is at least $|X| - 1$. Throughout we assume the necessary rounding in case it is not given.

II. The Exact Expression

In the following the cardinality $|X|$ of a multiset X is the number of occurrences of (possibly the same) elements in X.

Theorem 2.1: Let X be a finite multiset, $\max_{x \in X}\{K(X|x)\} = k$, and $f(k)$ be the number of halting self-delimiting programs of length at most k for the reference optimal universal Turing machine U. For each integer $n \geq 2$ there exists a multiset X of cardinality n satisfying $ID(X) \geq k + \log n + \log(1 - (n-1)/(nf(k)))$ (with $f(k) \geq n - 1$ this implies $ID(X) \geq k + \log(n-1)$; $f(k) < n - 1$ is barely possible), and every multiset X of cardinality n satisfies $ID(X) \leq k + \log n + O(1)$.

Remark 2.2: The exact expression for $f(k)$ is $\log f(k) = k - K(k) + O(1)$ as seen from R. Solovay's results mentioned in [7] or [12, Exercise 4.3.11]. The expression for $ID(X)$ can be rewritten as $ID(X) = k + \log(n - (n-1)/f(k))$ but the form in the theorem stresses that $ID(X) \geq k + \log n$ minus a small quantity. Using the expression for $f(k)$ above one can for $k \geq 5$ rewrite $ID(X) \geq k + \log(n - (n-1)2^{K(k)}/2^k) > k + \log(n - (n-1)k^2/2^k) > k + \log n$ with the inequalities $>$ in the last expression holding up to an $O(1)$ additive term. $\diamondsuit$

PROOF. Let $n \geq 2$ be a finite integer. Enumerate (possibly incomputably) all Y of cardinality n without repetition such that $\max_{y \in Y}\{K(Y|y)\} \leq k$. That is, for every $y \in Y$ there is a self-delimiting program $p_{(Y,y)}$ of at most k bits such that $U(p_{(Y,y)}, y) = Y$. Let $\mathcal{Y}$ be the set of these Y, the set of natural numbers be $\mathcal{N}$, the set $P(k) = \{p : U(p,x) < \infty,\ x \in \{0,1\}^*\ 0 \leq |p| \leq k\}$, and the function $f : \mathcal{N} \to \mathcal{N}$ be defined by $f(k) = |P(k)|$. Since f is strictly increasing there is an inverse f^{-1} with as domain the range of f such that $f^{-1}f(k) = k$. The set $\mathcal{Y}$ is in general infinite since already for $n = 2$ and large enough k it contains $\{x, x\}$ for every string x. Define a bipartite graph $G = (V, E)$ with V the vertices and E the edges by

$$V_1 = \{Y : Y \in \mathcal{Y}\},$$

$$V_2 = \{y : y \in Y \in V_1\},$$

$$V = V_1 \bigcup V_2,$$

$$E = \{(y, Y) : y \in Y \in V_1\}.$$

We want to determine a labeling of every edge $(Y, y) \in E$ by a string q such that for each $Y \in \mathcal{Y}$ and $y \in Y$ the labeling satisfies:

(i) all edges incident with the same vertex in V_1 are labeled with identical labels; and

(ii) all edges incident with the same vertex in V_2 are labeled with different labels.

In this way all labels on edges incident on given vertex in V_1 are the same, and all labels on edges incident on a given vertex in V_2 and different vertices in V_1 are different. Therefore, a vertex in V_2 together with such a label determines a vertex in V_1. How many labels do we require? Each vertex $y \in V_2$ has degree $f(k)$ or less and is connected by an edge with a vertex $Y \in V_1$ for which holds $y \in Y$. Since $|Y| = n$ there are n or less different vertices in $V_2 \bigcap Y$. Each vertex in $V_2 \bigcap Y$ may be connected by an edge with $n-1$ different vertices in V_1 apart from Y. All edges incident on vertices in $V_2 \bigcap Y$ may therefore require different labels except the ones connected with Y (there are n or less of them) of which the labels are identical but different from the others. This results in $nf(k) - (n-1)$ different labels. Therefore we define $Q(k) = P(k) \times \{1, \ldots, n\}$ where each $q = (p, m) \in Q(k)$ with $p \in P(k)$, $|p| \leq k$, and $1 \leq m \leq n$, is represented by a binary string $p0^{k-|p|+\lfloor \log n \rfloor - \lfloor \log m \rfloor}\text{bin}(m)^R$ where $\text{bin}(m)$ is the standard binary notation of m and the superscript R means reversal. Then $|Q(k)| = nf(k)$ and every *label* $q \in Q(k)$ is represented by a self-delimiting string of length $k + \lfloor \log n \rfloor + 1$ from which k can be extracted.

Claim 2.3: For every finite integer $n \geq 2$ there are X's with $|X| = n$ such that $ID(X) \geq k + \log n + \log(1 - (n-1)/(nf(k)))$. For $f(k) \geq n - 1$ we have $ID(X) \geq k + \log(n-1)$.

PROOF. The vertices in V_1 are enumerated as $Y_1, Y_2, \ldots$. We proceed by induction on the mth enumerated vertex.

Base case ($m = 1$) Label all edges incident on Y_1 with the least label in $Q(k)$. This labeling satisfies condition (i), and condition (ii) is satisfied vacuously.

Induction ($m > 1$) Assume that all edges incident on vertices $Y_1, \ldots, Y_m$ have been labeled satisfying conditions (i) and (ii). Label the edges incident on Y_{m+1} by the least label in $Q(k) \setminus Q'(k)$ where $Q'(k)$ is defined below. Let $Y_{m+1} = \{y_1, \ldots, y_n\}$ and possibly $y_i \neq y_j$ $(1 \leq i \neq j \leq n)$. Every edge incident on a vertex $y \in Y_{m+1}$ and vertex Y_{m+1} must be labeled by the same label by condition (i). Every $y \in Y_{m+1}$ is connected by an edge with at most $f(k)$ vertices in V_1 (including Y_{m+1}). Hence Y_{m+1} is connected by a path of length 2 via some vertex $y \in Y_{m+1}$ (there are n such vertices) with at most $n(f(k) - 1)$ vertices in $Y_1, \ldots, Y_m$. Let $\mathcal{Z}$ be the set of these vertices and $Q'(k)$ be the set of labels of the edges in these paths incident on a vertex in the set $\mathcal{Z}$. By condition (ii) the label of an edge incident on Y_{m+1}

is not in $Q'(k)$. Since $|Q'(k)| + 1 \leq nf(k) - (n-1)$ and $|Q(k)| = nf(k)$ while $n \geq 2$ this is always possible. *End induction*

To prove the claim we count how many bits in self-delimiting format are required to index $nf(k) - (n-1) = nf(k)(1-(n-1)/(nf(k))$ labels. This is at least $f^{-1}f(k) + \log n + \log(1-(n-1)/(nf(k)))$ bits. Therefore $ID(k) \geq k + \log n + \log(1-(n-1)/(nf(k)))$. Provided $f(k) \geq n-1$ we have $\log n + \log(1-(n-1)/(nf(k))) = \log(n-((n-1)/f(k))) \geq \log(n-1)$ and $ID(X) \geq k + \log(n-1)$. Otherwise $f(k) < n-1$ which means that $f(\max_{x \in X} K(X|x)) < |X| - 1$ which is barely possible. □

Claim 2.4: For every finite integer $n \geq 2$ every multiset X of cardinality n satisfies $ID(X) \leq k + \log n + O(1)$.

PROOF. This is shown in [11, Theorem 2]. We give another proof of this fact. Enumerate *computably* all Y (multisets of n elements) without repetition such that $\max_{y \in Y}\{K(Y|y)\} \leq k$. Construct the graph $G = (V, E)$ as before. In the proof of Claim 2.3 it is shown that we need at most $nf(k) - (n-1)$ different labels for the edges satisfying conditions (i) and (ii). The labels in $Q(k)$ have precisely length $k + \lfloor \log n \rfloor + 1$. We can determine k from the length of such a label. Concatenate each label in front with a $O(1)$-length self-delimiting program r. This program r makes the reference universal Turing machine U generate graph G on input k and do the labeling process. Let the edge connecting $y \in Y$ with Y be labeled by $q \in Q(k)$. In this way $|rq| \leq k + \log n + O(1)$. Since all edges incident on vertex Y have the same label q, we can write the concatenation as $s_Y = rq$. Program r also tells U that s_Y is the concatenation of two strings as described, and to retrieve k from $|s_Y|$. Labeling each edge incident on vertex Y with s_Y is computable and satisfies conditions (i) and (ii).

The length of s_X is an upper bound on $ID(X)$. The string s_X computes output X on inputs consisting of every $x \in X$ as follows. The universal prefix Turing machine U unpacks first the self-delimiting program r from s_X. This r retrieves k from $|s_X|$ and computably enumerates $\mathcal{Y}$ and therefore G. The program r simultaneously labels the edges of G in a standardized manner satisfying conditions (i) and (ii) as described above, including the program r itself in front. It does so until it labels an edge by s_X itself incident on vertex x. Since the the label q_X of s_X is unique for edges (X, y) with $y \in X$ the program r using x finds edge (X, x) and therefore X. Since $|s_X| = k + \log n + O(1)$, this implies the claim. □ □

Corollary 2.5: For $|X| = 2$ the theorem shows the result of [1, Theorem 3.3] with error term $O(1)$ instead of $O(\log \max_{x \in X}\{K(X|x)\})$. That is, with $X = \{x, y\}$ the theorem computes x from y and y

from x with the same program of length $\max_{x\in X}\{K(X|x)\}+O(1)$. (One simply adds to program r the instruction "the other one" in $O(1)$ bits.) This result can also be derived from [11], [13].

Corollary 2.6: For every finite multiset X of cardinality at least 2 we have $ID(X) \geq \max_{x\in X}\{K(X|x)\}$. Hence $\max_{x\in X}\{K(X|x)\} \leq ID(X) \leq \max_{x\in X}\{K(X|x)\} + \log|X| + O(1)$.

III. Overlap

Programs p, q with p computing y from x and q computing x from y can overlap as follows. With x, y strings of length n, $K(x|y), K(y|x) \geq n$, and $p = x \oplus y$ with $\oplus$ the bitwise exclusive or yields $y = x \oplus p$ and $x = y \oplus p$. Therefore, the shortest program p that computes from x to y overlaps completely with a shortest program that computes from y to x.

Corollary 3.1: Let $|X| \geq 2$. In the previous section the programs s_X to compute X from any $y \in X$ with $|s_X| = \max_{x\in X}\{K(X|x)\} + O(1)$ are *maximally* overlapping in that these programs are the same for every $y \in X$.

It follows that for $X = \{x_1, \ldots, x_n\}$ and $K(X|x_i) = K(X|x_j)$ $(1 \leq i, j \leq n)$ there is a program p with $|p| = K(X|x_i) + O(1)$ such that $U(p, x_i) = X$ for all $1 \leq i \leq n$

If $X = \{x, y\}$ then p is a shortest program which computes x from y and y from x: complete overlap of shortest programs. For $K(x|y) < K(y|x)$ there are strings p, q, d with $p = qd$, $|p| = K(y|x) + K(K(x|y), K(y|x)) + O(1)$ and $|d| = K(y|x) - K(x|y)$ such that $U(p, x) = X$ and $U(q, y) = X$ [1, Theorems 3.3, 3.4, Remark 3.6].

For $X = \{x_1, \ldots, x_n\}$ the almost shortest programs p_i with $U(p_i, x_i) = X$ $(1 \leq i \leq n)$ have a common substring q of length $\min_{x\in X}\{K(X|x\}) + K(|X|, \max_{x\in X}\{K(X|x\}, \min_{x\in X}\{K(X|x\}) + \log|X| + O(1)$: the overlap of those programs ([18, Theorem 3.1]).

The *algorithmic mutual information* $I(x, y)$ between x and y is defined by $I(x : y) = K(x) + K(y) - K(x, y)$. For $X = \{x, y\}$ reference [1] asked whether we can find shortest programs p, q such that $U(p, y) = x$ and $U(q, x) = y$ that are *minimally* overlapping in the sense that for $x \neq y$ it holds that $I(p : q)$ is minimal? In [17] this question is resolved as follows. For all strings x, y there are binary programs p, q such that $U(p, x) = y$, $U(q, y) = x$, the length of p is $K(y|x)$, the length of q is $K(x|y)$, and $I(p : q) = 0$ where the last three inequalities hold up to an additive $O(\log K(x, y))$ term. In contrast, for some strings x, y this is not the case when we replace $O(\log K(x, y))$ with $O(\log(K(x|y)+K(y|x)))$.

Related is [14]. There the surprising fact is shown that there is a almost shortest p to compute x from y such that $K(p|x) = O(\log n)$ and $K(x|p, y) = O(\log n)$. That is, this almost shortest program depends

only on x and almost nothing on y. This is an analogue of the Slepian-Wolf result [16] in information theory.

ACKNOWLEDGMENT

Bruno Bauwens pointed out an error in an early version of this paper and the referees gave helpful comments.

REFERENCES

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

[2] M. Bailey, J. Oberheide, J. Andersen, Z.M. Mao, F. Jahanian, J. Nazario, Automated classification and analysis of internet malware. Pp 178–197 in: Recent Advances in Intrusion Detection, Lecture Notes in Computer Science Volume 4637, 2007.

[3] A.R. Cohen, F.L.A.F. Gomes, B. Roysam, M. Cayouette, Computational prediction of neural progenitor cell fates, *Nature Methods*, 7(2010), 213–218.

[4] R.L. Cilibrasi, P.M.B. Vitányi, Clustering by compression, *IEEE Trans. Inform. Theory*, 51:12(2005), 1523–1545.

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

[6] M.M. Deza, E Deza, *Encyclopedia of distances*, Springer, 2009.

[7] P. Gács, Lecture Notes on Descriptional Complexity and Randomness. Technical Report, Boston University, Computer Sci. Dept., Boston, MA 02215, 2009.

[8] E. Keogh, S. Lonardi, C.A. Rtanamahatana, Toward parameter-free data mining, In: *Proc. 10th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Seattle, Washington, USA, August 22–25, 2004, 206–215.

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

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

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

[12] M. Li, P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York, Third edition, 2008.

[13] M.M.H. Mahmud, On Universal Transfer Learning, *Theor. Comput. Sci.*, 410(2009), 1826–1846.

[14] An.A. Muchnik, Conditional complexity and codes, *Theor. Comput. Sci.*, 271(2002), 97–109.

[15] D. Musatov, A. Romashchenko, A. Shen, Variations on Muchnik's Conditional Complexity Theorem, *Theory Comput. Syst.*, 49(2011), 227–245.

[16] D. Slepian, J.K. Wolf, Noiseless coding of correlated information sources, *IEEE Trans. Inform. Theory*, 19(1973), 471–480.

[17] N.K. Vereshchagin, M.V. Vyugin, Independent minimum length programs to translate between given strings, *Theor. Comput. Sci.*, 271:1–2(2002), 131–143.

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