Mutual Information and Algorithmic Information Transfer as Ideal Undirected and Directed Independence Tests

Bruno Bauwens
Department EESA, machinelearning
Ghent University
Ghent, Belgium

Bart Wyns
Department EESA, machinelearning
Ghent University
Ghent, Belgium

Dieter Devlaeminck
Department EESA, machinelearning
Ghent University
Ghent, Belgium

Luc Boullart
Department EESA, machinelearning
Ghent University
Ghent, Belgium

Georges Otte
Department of Neurophysiology
P.C.Dr. Guislain Institute
Ghent, Belgium

Patrick.Santens@ugent.be
Department of Neurology
Ghent University
Ghent, Belgium

Abstract

In this paper ideal undirected independence tests are defined and it is investigated whether optimal ones exist in the arithmetical hierarchy of functions. It is shown that there are no non-constant universal independence tests in the classes of recursive and co-enumerable functions. However, there exists one in the classes Σ^0_n, $n \geq 2$ which equals algorithmic mutual information up to a small term. Mutual information can be decomposed as a sum of three parts: information transfer from x to y, information transfer from y to x and common simultaneous information. We investigate the objectivity of both tests. A small amount of prior information can only result in a decrease of the test strength, but it can change its decomposition in algorithmic information transfer arbitrarily.

Keywords: Algorithmic Complexity, Independence Tests, Causality, Recursion Theory.

1 Introduction

Much literature is available on randomness of infinite sequences and randomness of infinite sequences relative to other infinite sequences. In the last decade an impressive amount of new material has been published and has been organized in [1]. Randomness relative to a string answers the question: is x random given y? While an independence test answers the question: becomes x more random given y? Little attention has been drawn to measures for the last question as well for measures that study causal or directed independence.

Independence tests play a crucial role in many engineering applications. A popular application is the construction of contrast functions in ICA algorithms for solving blind source separation problems [2, 3, 4, 5]. Directed independence tests are important to determine how information is flowing in complex systems if little knowledge on the system is available except for the activity of some components. Examples of applications of directed independence tests can be found in the analysis of EEG signals for estimating the influence of the activity of different brain areas to others [6, 7].

In this paper we investigate from a theoretical point of view the existence and the use of ideal directed and non-directed independence tests. In [8] there is a short reference to independence measures for the universal distribution. In [9, 10] the information distance and information metric are described. The information distance is the shortest

program that reproduces x from y and y from x. The similarity metric is a normalization of the information distance. Independent strings have an information distance close to 1. However, to us, it is not clear how one can make this property into a statistical test for independence with confidence bounds. To call a test a 'statistical' test, confidence bounds are necessary. Here we study the existence of universal statistical independence tests relative to the classes of the arithmetical hierarchy. We show that we can calculate algorithmic information transfer with high probability given a short program.

Section 2 to 4 describe some basic concepts such as: algorithmic prefix complexity, the arithmetical hierarchy for total functions, and the universal probability measure, necessary for proving the theorems in Sections 5 and 6. In Section 5 general purpose dependence tests are introduced and characterized in terms of the universal distribution. We prove theorems regarding the existence and the calculability of universal independence tests. In Section 6 algorithmic information transfer is defined and it is shown that it decomposes mutual information. Calculability properties are proved and the objectivity of the measures is investigated.

2 Algorithmic complexity

Algorithmic complexity $K(x)$ is the accepted absolute measure of information content in a binary string x [9]. $K(x)$ can be viewed as the data compression limit of x. In this section we present the theory of algorithmic (Kolmogorov) complexity as treated in the standard work [11] in which all proofs of the theorems can be found.

A prefix-free Turing machine U is a Turing machine with a read only input tape, a read and write work tape and a write only output-tape. A program p is a halting program if the input reading head reads all bits of p and halts before reading the next bit. Denote x and y as finite binary strings, ϵ as the empty string. $l(x)$ is the length of x. The finite strings can be identified with the natural numbers by the bijection $1 \leftrightarrow \epsilon$, $2 \leftrightarrow 0$, $3 \leftrightarrow 1$, $4 \leftrightarrow 00$, $5 \leftrightarrow 01$, ... In this way U defines a partial function $U : \mathbb{N} \to \mathbb{N}$. Algorithmic complexity is defined as:

Definition 2.1.

$$K_U(x) = min\{l(p)|U(p) \downarrow = x\}$$

This definition of information content seems to depend on the type of machine one uses, but one can show that this independence is bounded by a constant for universal Turing machines (TM).

Definition 2.2. *A function f (additively) dominates a function g, (notation: $g <^c f$) if there exists a constant c such that $\forall x \in dom(f) : g(x) < f(x) + c$.*

Definition 2.3. *A function f equals g up to an (additive) constant (notation: $g =^c f$) if $g <^c f$ and $f <^c g$.*

Theorem 2.4. *If U and U' are universal prefix-free Turing-machines, $K_U =^c K_{U'}$*

From now on we use a fixed TM U and write $K(x)$ for $K_U(x)$.

Every string has a trivial representation and therefore:

Theorem 2.5. $K(x) <^c l(x) + 2 \log l(x)$

One can equip a TM with extra input-tapes and extra output tapes. This enables us to define:

$$\begin{aligned} K(x,y) &= min\{l(p)|U(p) = (x,y)\} \\ K(x|y) &= min\{l(p)|U(p,y) = x\} \end{aligned}$$

Denote y^* as the shortest program such that $U(y^*) \downarrow = y$.

Theorem 2.6.

$$K(x,y) =^c K(x|y^*) + K(y)$$

Definition 2.7. *The mutual algorithmic information $I(x;y)$ is:*

$$I(x;y) = K(x) + K(y) - K(x,y)$$

One has:

$$I(x;y) =^c K(x) - K(x|y) =^c K(y) - K(y|x)$$

3 The arithmetical hierarchy

The arithmetic hierarchy represents the 'hardness' of evaluating a function. This section gives the definitions and theorems we will use in Section 5. They can be found in [12] and [11]. Generalized Kolmogorov complexities are studied in [13].

Definition 3.1. *Total, recursive, enummerable and co-enumerable functions.*

- *A function is* total *if it is defined in it's whole domain.*
- *A total function f is* computable *if there is a program that calculates $f(x)$ and halts for all x.*
- *A total function $f : \mathbb{N} \to \mathbb{N}$ is* enumerable *resp.* co-enumerable *if there is a computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\forall x, \forall t, t' \in \mathbb{N} : g(t, x) \leq g(t+t', x)$ resp $g(t, x) \geq g(t+t', x)$ and $\lim_{t=\infty} g(t, x) = f(x)$.*

$K(x)$ is an example of a function that is co-enumerable but not computable.

The halting sequence ξ is defined as: $\xi_n = 1$ if $U(n) \downarrow$ and $\xi_n = 0$ else. There is no program that calculates ξ_n for all n. This is the famous halting problem. We can equip a TM with an extra infinite input tape containing the halting sequence (notation: $U^{(2)}$). The algorithmic complexity of x relative to $U^{(2)}$ is $K^{(2)}(x)$:

Definition 3.2.

$$K^{(2)}(x) = min\{l(p)|U^{(2)}(p) \downarrow = x\}$$

K_U is computable relative to $U^{(2)}$. It is possible to go further and ask if a program x on such a $U^{(2)}$ halts. The halting sequence relative to the halting problem is: $\xi_n^{(2)} = 1$ if $U^{(2)}(n) \downarrow$. A program for $U^{(2)}$ that calculates $\xi_n^{(2)}$ for all n does not exist. We can equip a Turing machine $U^{(2)}$ with an extra input tape containing $\xi^{(2)}$ and denote it by $U^{(3)}$, then $K^{(3)}(x)$ is defined in a similar way as $K^{(2)}(x)$.

This procedure can be iterated to define $U^{(n)}$, $\xi^{(n)}$ and $K^{(n)}$ for $n = 1, 2, ...$ Denote $U^{(1)} = U$,$\xi^{(1)} = \xi$ and $K^{(1)} = K$.

Definition 3.3. *Σ_n^0, Π_n^0 and Δ_n^0, $n \geq 1$:*
$f \in \Sigma_n^0 \iff$ f is total and co-enumerable relative to $U^{(n)}$.
$f \in \Pi_n^0 \iff$ f is total and enumerable relative to $U^{(n)}$.
$\Delta_n^0 = \Sigma_n^0 \cap \Pi_n^0$

The classes Σ_n^0, Π_n^0 and Δ_n^0 for $n \geq 1$ constitute the arithmetical hierarchy. It is shown that they form a strict hierarchy [12]: $\Delta_n^0 \subsetneq \Sigma_n^0$, $\Delta_n^0 \subsetneq \Pi_n^0$, and $\Sigma_n^0 \cup \Pi_n^0 \subsetneq \Delta_{n+1}^0$.

4 Algorithmic probability

The universal probability distribution is the central probability distribution for defining universal learning and proving many optimality claims and theorems. Here it is used as a mathematical tool. Definitions, theorems and proofs of this sections can be found in [11].

Definition 4.1. *A probability distribution or semi-measure P is a function $P : \mathbb{N} \to [0, 1]$ such that*

$$\sum_x P(x) \leq 1$$

The set of semi-measures is denoted as $\mathcal{P}$.

Definition 4.2. *A rational function $f : \mathbb{N} \to \mathbb{Q}$ is partial recursive if a program p exists such that $U(p, x) \downarrow = (r, s)$ and $f(x) = r/s$ each time $f(x)$ is defined.*

Definition 4.3. *A real function $f : \mathbb{N} \to \mathbb{R}$ is called enumerable if a recursive rational function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$ exists such that $\forall t, t', x \in \mathbb{N} : g(t, x) \leq g(t + t', x)$ and $\lim_{t\to\infty} g(t, x) = f(x)$.*

The set of semi-measures that are enumerable are denoted as $\mathcal{P}_1^0$.

The next theorem shows that $K(x)$ defines a probability distribution. Every other co-enumerable function defining in the same way a probability distribution dominates $K(x)$.

Theorem 4.4. *Let $f : \mathbb{N} \to \mathbb{N}$, $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $f, g \in \Pi_1^0$:*

- $\forall y : \sum_x 2^{-K(x|y)} \leq 1$
- $\sum_x 2^{-f(x)} < 1 \Rightarrow K(x) <^c f(x)$
- $\forall y : \sum_x 2^{-g(x,y)} < 1 \Rightarrow \exists c, \forall x, y : K(x|y) < g(x, y) + c$

The proof of the 3 equations can be found in [11]. We remark that the proof of this equations also hold if $U^{(n)}$ is used as reference machine:

Theorem 4.5. *Let $f : \mathbb{N} \to \mathbb{N}$, $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $f, g \in \Pi_n^0$:*

- $\forall y : \sum_x 2^{-K^{(n)}(x|y)} \leq 1$
- $\sum_x 2^{-f(x)} < 1 \Rightarrow K^{(n)}(x) <^c f(x)$
- $\forall y : \sum_x 2^{-g(x,y)} < 1 \Rightarrow \exists c, \forall x, y : K^{(n)}(x|y) < g(x, y) + c$

Definition 4.6. *The algorithmic probability is defined as:*

$$\begin{aligned} m(x) &= 2^{-K(x)} \\ m(x|y) &= 2^{-K(x|y)} \end{aligned}$$

It can be shown that $m, Q \in \mathcal{P}_1^0$. m and Q are the 'largest' elements of $\mathcal{P}_1^0$.

Theorem 4.7. *Domination* $\forall P \in \mathcal{P}_1^0, \exists c_P$ *such that* $\forall x$

$$P(x) \leq c_P m(x)$$

Theorem 4.8. *Coding theorem*

$$\begin{aligned} \log m(x) &=^c \log Q(x) \\ \log m(x|y) &=^c \log Q(x|y) \end{aligned}$$

c, c' *are independent form* x *and* y

5 Undirected independence tests

This section introduces statistical independence tests similar to sum-p tests in [11]. The differences with this approach are that we consider functions with 2 inputs and enforce no computability constraints.

Definition 5.1. *A total function* $d : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *is a* (P, Q)*-independence test for* $P, Q \in \mathcal{P}_1^0$ *iff:*

$$\sum_{x,y} P(x)Q(y)2^{d(x,y)} < c$$

This definition ensures that if x, y are drawn independently according to $P(x)$ and $Q(y)$ then $Prob\{d(x, y) - \log_2(c) > k\} < 2^{-k}$. An independence test has exponential increasing confidence bounds relative to P and Q.

We denote the set of (P, Q)-independence tests as $\mathcal{D}_{P,Q}$ and use the shorthand notation $\mathcal{D}$ for $\mathcal{D}_{m,m}$.

Lemma 5.2. $d \in \mathcal{D}$ *iff* $\forall P, Q \in \mathcal{P}_1^0 : d \in D_{P,Q}$

Proof. **part 1** : if: put $P = Q = m \in \mathcal{P}_1^0$
part 2 : only if:
$\forall P, \exists c, \forall x : P(m) < c_P m(x)$
Therefore

$$\begin{aligned} &\sum_{x,y} P(x)Q(y)2^{d(x,y)} \\ &< \sum_{x,y} c_P m(x) c_Q m(y) 2^{d(x,y)} \\ &< c_P c_Q c \end{aligned}$$

□

This lemma tells us that if a test works well for the universal measure, it works well for all enumerable measures.

Definition 5.3. *A function is universal in a set of functions if it dominates every other function in the set.*

Now the question rises: Is there a universal independence test in some class of functions? This question is answered by Theorems 5.4 and 5.5 for the classes Σ_n^0 and Δ_n^0, $n \in \mathbb{N}$.

Theorem 5.4. $\mathcal{D} \cap \Sigma_n^0$ *has a universal element for all* $n \geq 2$ *and* $\mathcal{D} \cap \Sigma_1^0$ *has the constant function as universal element. The universal elements for* $n \geq 2$ *are given by:*

$$d(x, y) = K(x) + K(y) - K^{(n)}(x, y) \qquad (1)$$

Theorem 5.5. $\mathcal{D} \cap \Delta_n^0$ *has no universal element for all* $n > 2$ *and* $\mathcal{D} \cap \Delta_1^0$ *has the constant function as universal element.*

The proofs are omitted because of space. The lowest class in the arithmetical hierarchy for which we found an independence test is Σ_2^0. The test equals Kolmogorov mutual information up to a term $I(x, y; \xi)$. Strings for which this term is high are seldom because it requires to much time for generating them [14].

Summarized one can see that these theorems prove or disprove the existence of a universal element in the arithmetical hierarchy for Σ_n^0 and Δ_n^0, $\forall n \in \mathbb{N}$. The lowest element in the hierarchy for which we have found a non-constant universal element is Σ_2^0. It equals the mutual information up to a term $I(x, y; \xi)$. It is believed that strings for which such term is high do not appear in nature because there is no time to generate them.

6 Algorithmic Information Transfer

The goal is to decompose mutual information of two strings x and y into three parts: information flowing from x to y, from y to x and information coming from both.

6.1 Definition

First we introduce a new kind of input tape on a Turing machine. Assume without loss of generality that the reading head of the input tape and the

writing head of the output tape of U are not allowed to return. A causal input tape is now defined as an input tape that is only allowed to read a new bit if a bit has been written to the output tape. Input that is given to the Turing machine on a causal input tape is denoted with an arrow. $U(p, y\uparrow) = z$ means that z_k is produced before reading $y_{k...l(y)}$, $k = 1...l(y)$.

Definition 6.1. *$K(x|y\uparrow)$:*

$$K(x|y\uparrow) = \min\{l(p)|U(p, y\uparrow) = x\}$$

Definition 6.2. *The information transfer is:*

$$IT(x \leftarrow y) = K(x) - K(x|y\uparrow)$$

The program that realizes the definition of 6.1 is denoted by $p_{x\leftarrow y}$. To decompose mutual information a third term is necessary representing a common source for both signals. Consider by example x and $y = x$: $I(x; y) = K(x)$ but $IT(x \leftarrow y) = IT(y \leftarrow x) = 0$.

Definition 6.3.

$$I(x = y) = I(p_{x\leftarrow y}; p_{y\leftarrow x})$$

One has that $K(x) >^c IT(x \leftarrow y) >^c 0$ and $K(x) >^c I(x = y) >^c 0$

Theorem 6.4. *The mutual information is split up in three parts up to an additive term $dIT(x, y) = K(K(x|y\uparrow), K(y|x\uparrow)|x, y) <^c l^*(l(x)) + l^*(l(y))$.*

$$\begin{aligned} I(x; y) =^{+c}\ & IT(x \leftarrow y) + IT(y \leftarrow x) \\ & +I(x = y) + dIT(x, y) \end{aligned}$$

Proof. The right side of equation 2 can be rewritten into:

$$I(x; y) + K(x, y) - K(p_{x\leftarrow y}, p_{y\leftarrow x}) + dIT(x, y)$$

It rests to show that:

$$\begin{aligned} & K(x, y) + K(K(x|y\uparrow), K(y|x\uparrow)|x, y) \\ =^c\ & K(p_{x\leftarrow y}, p_{y\leftarrow x}) \end{aligned}$$

x,y,$K(x|y\uparrow)$ and $K(y|x\uparrow)$ can be calculated from $p_{x\leftarrow y}$ and $p_{y\leftarrow x}$. The other way around, $p_{x\leftarrow y}$ and $p_{y\leftarrow x}$ can be calculated from x,y,$K(x|y\uparrow)$ and $K(y|x\uparrow)$ by devotailing all possible programs shorter than $\max(K(x|y\uparrow), K(y|x\uparrow))$. □

Consequently, $I(x; y) >^c IT(x \leftarrow y)$ and $I(x; y) >^c IT(x = y)$. Therefore $IT(x \leftarrow y), IT(x = y) \in \mathcal{D}$ and the tests will give us exponential confidence bounds for the conclusion of a directed dependence if x, y are drawn independently from P and P': $Prob\{IT(x \leftarrow y) - \log_2 c > k\} < 2^{-k}$.

6.2 Objectivity

We argue that mutual information is an objective test for independence and algorithmic information transfer is not. To define the effect of side information we calculate $I(x; y|z)$. However just giving a constant z is not general enough, instead we assume that the information is given in the form of a total function $\mathcal{M} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. For the undirected test the corresponding measure is $I(x; y|\mathcal{M}(x, y))$. Equation $I(x; y|\mathcal{M}(x, y)) <^c I(x; y)$ is false and we don't necessarily have that $I(x; y|\mathcal{M}(x, y)) \in \mathcal{D}$. But if it is in $\mathcal{D}$ we know by theorem 5.4 that it is dominated by $I(x; y)$ up to an additive term $I(x, y; \Omega)$ and this term is assumed to be small for real data. Adding information by a total function $\mathcal{M}$ in such a way that $I(x; y|\mathcal{M}(x, y))$ remains a statistical test, will only decrease the significance for the conclusion of a dependency by the test.

If we do the same for algorithmic information transfer we can change the partition of mutual information arbitrarily and this warns us for the subjectiveness of the algorithm. Assume the side information is available in terms of a total function as above:

$$\begin{aligned} & IT(x \leftarrow y|\mathcal{M}(x, y)) \\ = & K(x|\mathcal{M}(x, y)) - K(x|\mathcal{M}(x, y), y\uparrow) \end{aligned}$$

Take some u, v, w with $l(u) = l(v) = l(w) = n/2$ and $K(u), K(v), K(w) > n/2$. Denote $z = xor(u, v)$, the bitwise application of the xor-gate: $xor(1, 1) = xor(0, 0) = 0$ and 1 otherwise. Take $x = u^\wedge z$ denoting the concatenation of u and z, and take $y = 0^\wedge v^\wedge w_{1...n/2-1}$. We have that $I(x, y) = n/2$, $IT(x \leftarrow y) = n/2$ and $IT(y \leftarrow x) = 0$. However if we take the simple model $\mathcal{M} : (x, y) \to x_{1...n/2}$ we have that $I(x; y|\mathcal{M}(x, y)) = n/2$, $IT(x \leftarrow y) = 0$ and $IT(y \leftarrow x) = n/2$. For this reason a universal decomposition of I can not easily be defined using this framework.

7 Conclusion and further research

We investigated the existence of universal independence tests in the Σ_n^0 and Δ_n^0 classes in the arithmetic hierarchy. Σ_1^0 and Δ_1^0 have only constant dependence tests and Δ_n^0 has no universal dependence tests for $n \geq 2$. Non trivial universal independence tests exist in $\Sigma_n^0, n \geq 2$. The universal test in Σ_2^0 equals algorithmic (Kolmogorov) mutual

information up to a term $I(x, y; \Xi)$ which is small for real data. Mutual information can be decomposed into 3 parts that represent the flow of information in both directions and information from a direct common source. This partition is very sensitive for side information provided by total functions as it is able to flip the total information flow between some strings. One of the reasons for the loss of this 'objectivity' property is the generality of semi-measures under consideration. Further research will be carried out in defining suitable sets of semi-measures for which optimal directed independence tests can be constructed.

8 Acknowledgments

Bruno Bauwens was supported by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

[1] Rod Downey. Algorithmic randomness and complexity. http://www.mcs.vuw.ac.nz/~downey/, 2006.

[2] J. Karhunen A. Hyvarinen and H. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

[3] T.L. Fine C.J. Ku. A bayesian independence test for small datasets. *IEEE Transactions on Signal Processing*, 54(10), October 2006.

[4] J.F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE, Special Issue on Blind Identification and Estimation*, 90(10):2009–2025, October 1998.

[5] A. Hyvarinen. One-unit contrast functions for independent component analysis: Statistical analysis. In *Neural Networks for Signal Processing, Proceedings of the 1997 IEEE Workshop*, 1997.

[6] J. Bhattacharya U. Feldmann. Predictability improvement as an asymmetrical measure of interdependence in bivariate time series. *International Journal of Bifurcation and Chaos*, 14(2):505–514, 2004.

[7] M.L.V. Quyen M. Chavez, J. Martinerie. Statistical assessment of nonlineair causality: Application to epileptic eeg signals. *Journal of Neuroscience Methods*, 123:113–128, 2003.

[8] P. Gacs. Lecture notes on descriptional complexity and randomness. Unpublished, 1987.

[9] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi. The similarity metric. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 863–872, 2003.

[10] C. H. Bennett, P. Gacs, M. Li, P.M.B. Vitanyi, and W. H. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4), 1998.

[11] P.M.B. Vitanyi M. Li. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, 1993.

[12] H. Rogers Jr. *Theory of recursive functions and effective computability*. McGraw-Hill, 1967.

[13] Jurgen Schmidhuber. Hierarchies of generalized kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, 13(4):587–612, 2002.

[14] P.M.B. Vitanyi N.K. Vereshchagin. Kolmogorov's structure functions and model selection. *IEEE Transactions Information Theory*, 50(12):3265–3290, 2004.