Algorithmic Complexity

Peter Sunehag and Marcus Hutter
Research School of Computer Science
Australian National University
ACT 0200 Australia
Email: {peter.sunehag, marcus.hutter}@anu.edu.au

June 13, 2014

Abstract

Algorithmic complexity provides a mathematical formal notion of string complexity. Building on this, one arrives at mathematical "gold standard" (though incomputable) definitions of randomness, induction, similarity and even intelligence. These definitions can be turned into practical algorithms by using common compressors to approximate the universal solutions. One can consider the theories as idealized cognition with respect to which one can aim to describe actual biological cognition by listing biases and limitations that need to be defined relative to some normative reference.

Keywords: Kolmogorov Complexity, Algorithmic Information Theory, Cognition, Rationality, Simplicity, Optimism, Induction, Similarity, Clustering, Prediction, Agents, Learning, Reinforcement

1 Introduction

Our world contains ubiquitous examples of phenomena relying on compact representations. Some examples are the genome (Krakauer, 2002), human cognition (Chater, 1996, 1999; Chater and Vitányi, 2003) and natural languages (Zipf, 1949; Piantadosi et al., 2011) which all involve efficient use of resources. They all involve a compact code and a method of translating the code into a result. The field of algorithmic complexity (Solomonoff, 1964; Kolmogorov, 1965; Chaitin, 1969), sometimes called Kolmogorov complexity (Li and Vitányi, 2008) or Algorithmic Information Theory (Hutter, 2007a), provides a mathematical theory for such phenomena and also answers some fundamental theoretical questions about randomness (Martin-Löf, 1966) as well as telling us how to ideally make predictions about the future (Solomonoff, 1964; Hutter, 2007b; Rathmanner and Hutter, 2011). The translation mechanism mentioned above is called the reference machine and is formalized as a universal Turing machine (UTM) (Turing,

1936). The codes are programs for this machine in the form of a binary string. The result is an output of another binary string. The length of the shortest program that leads to a certain output string is called the Kolmogorov complexity of this string and can be interpreted as a measure of information content as well as of simplicity. A string with lower Kolmogorov complexity than another is considered simpler with respect to the used reference machine since it can be described more succinctly. If a string can be represented in a substantially shorter form than its naive representation, it is called compressible, otherwise incompressible. Incompressible strings look random and incompressibility is now the established definition for what it means for an individual sequence to be random. Furthermore, algorithmic complexity provides a way of defining a priori probabilities (Solomonoff, 1964) for different strings, where simpler strings are deemed more likely. From these a priori probabilities, one can derive conditional probabilities for the future as observations are made. These predictions will converge towards the true probabilities under mild assumptions on the generating process (Solomonoff, 1978; Hutter, 2007b). The assigned probabilities depend on the choice of reference machine since the Kolmogorov complexity differs by up to an additive constant for any pair of machines. This constant can make a big difference initially while the effect vanishes asymptotically. Algorithmic probability has also been combined with reinforcement learning and sequential decision theory to define an intelligent agent denoted AIXI, thereby introducing the field of universal artificial intelligence (Hutter, 2005). The agent follows a policy that maximizes expected utility with respect to the algorithmic probabilities and the utilities are defined as a sum of discounted rewards.

Algorithmic complexity has the drawback of being incomputable, though successful algorithms have been devised using practical compressors to compute crude approximations of the true complexity (Li et al., 2004; Cilibrasi and Vitányi, 2005; Veness et al., 2011). We will mention these as we outline the basic theory in the next section and then we discuss the possibility of using algorithmic complexity to provide an idealized theory of cognition. It has been argued that this model of learning resolves the "poverty of stimulus" problem with human language acquisition (Solomonoff, 1964; Perfors et al., 2006; Hsu et al., 2013). The article ends with a summary in Section 4.

2 Basic Definitions and Results

In this section, we present the formal definitions and basic results in algorithmic complexity, algorithmic randomness, algorithmic probability and universal artificial intelligence. We refer to Hutter (2005); Li and Vitányi (2008) for comprehensive studies.

2.1 Algorithmic Complexity

Given a Universal Turing Machine (UTM) U, we define the Kolmogorov complexity $K(x)$ of a string x to be the length of the shortest program that makes

U output x. Formally

$$K(x) := \min\{\ell(p) : U(p) = x\}$$

where $\ell(p)$ is the length of p and $U(p) = x$ means that U outputs x when given program p on the input tape. That U outputs x can be given several precise meanings. For example it can mean that x is printed on the output tape and then the machine halts or that x is first printed and then the machine continues. K is defined by demanding that U halts after x is printed but the other mentioned variant as well as several others differs from K by at most $O(\log K)$. We here state all properties (equalities and inequalities) only up to such a logarithmic term and they, therefore, are true for a whole range of variations on K.

We also define the conditional complexity $K(x|y)$ which is the length of the shortest program to output x given y as side information. For example, if y is a string that only differs from x by missing the last bit, then the program only has to add that bit and such a program is very short even if x is a long complex string by itself. $K(x|y)$ tells us how much information there is in x that is not in y. Furthermore, we define the joint complexity $K(x, y)$ as the length of the shortest program to output first x and then y. This can be much shorter than the sum of the shortest programs to output the individual strings since they can have much in common. It can, however, not be much longer than the sum of those since one can always create a program that first runs p_1 that outputs x and then runs p_2 that outputs y. The extra length is the length of the extra code needed for the command to first run one program and then another and is is at most a constant that does not depend on x, y, p_1 or p_2. A profound result that connects all the mentioned concepts is

$$K(x, y) = K(x) + K(y|x) = K(y) + K(x|y)$$

where $=$ should, as everywhere in this article, be interpreted as being up to a logarithm and not as exact equality. This result can be rewritten as a property that is called "symmetry of information", stating that

$$K(x) - K(x|y) = K(y) - K(y|x).$$

This relation informally means that the amount of information y has about x $(K(x) - K(x|y))$ is the same as what x has about y. The involved quantity provides one measure of how similar they are but it is not formally a distance. However, one can transform it to

$$d(x, y) := 1 - \frac{K(x) - K(x|y)}{K(x, y)}$$

which satisfies the distance measure properties and still says that two strings are similar if they have a large amount of mutual information. Another alternative to defining a distance is $D(x, y)$ which is the length of the shortest program that converts x to y and y to x. It can be proven (Bennett et al., 1998; Mahmud, 2009) that within a constant term

$$D(x, y) := \max\{K(x|y), K(y|x)\}$$

which means that it is about as hard converting both ways as the hardest of the two directions. Both d and D are universal distance functions in the sense that if two strings are close for any computable distance function (satisfying some mild properties) then they are also close as measured by d or D. In other words, d and D pick up any possible computable regularity between any two given strings.

A normalized version of D called the normalized compression distance

$$\tilde{D}(x,y) := \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$

was introduced by Li et al. (2004) where it was also proven to be universal. Furthermore, Li et al. (2004) introduced practical approximations using common compressors, namely they defined the normalized compression distance by

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

where $Z(x)$ is the size of the compression of x. By using NCD as a similarity measure the authors of Li et al. (2004) inferred an evolutionary tree from DNA sequences as well as the language tree from the text of the UN declaration of human rights in 52 languages. Other applications include clustering of music as well as finding the phylogenetic tree of chain letters (Bennett et al., 2003; Cilibrasi and Vitányi, 2005). This method of clustering is sometimes referred to as algorithmic clustering or clustering by compression.

2.2 Algorithmic Randomness

It is clear that $K(x) \leq \ell(x)$, where $\leq$ is up to a logarithm as always (though here true for a constant) and $\ell(x)$ is the length of the string, since one can write a program that says Print(x). The extra constant length is the Print command. If there is no shorter program than this, i.e. $K(x) \geq \ell(x)$ we say that x is incompressible and random. The notion of randomness based on incompressibility is also called Martin-Löf randomness (Martin-Löf, 1966) and has been found to be similar to human perception of randomness (Griffiths and Tenenbaum, 2003). Intuitively something is random if no regularities can be found in the string.

2.3 Algorithmic Probability

Based on a chosen reference machine U, the algorithmic a priori probability of a string x is

$$M(x) := \sum_{p:U(p)=x} 2^{-l(p)}$$

which can be interpreted as the probability that the string x is the output from U if zeros and ones are placed by independent balanced coin flips on the input tape. The sum of terms $2^{-l(p)}$ for all programs p (in a prefix free set of

programs) is upper bounded by 1, according to Kraft's inequality. They do not sum to exactly one and M is actually just a semi-measure though it can be renormalized to a proper probability measure. Given any string x, one would then say that the probability of the next bit being a one is

$$M(1|x) := \frac{M(x1)}{M(x0) + M(x1)}.$$

$M(x)$ is closely approximated by $2^{-K(x)}$. This is because much of the mass of $M(x)$ comes from the shortest program. Furthermore, one can prove (Li and Vitányi, 2008) that if one samples a string from M, one will with high probability get a string of low complexity. This is in accordance with the principle of favoring simplicity, which is often called Occam's razor. Algorithmic complexity and algorithmic probability provide a mathematical formalization of this principle.

Sequence prediction. Ray Solomonoff developed a theory of sequence prediction using M, including his "Prediction Error Theorem" (Solomonoff, 1978) stating that

$$\sum_{x \in \{0,1\}^*} \mu(x)(M(0|x) - \mu(0|x))^2 \leq K(\mu)\frac{\ln(2)}{2}$$

for any computable measure μ. This implies, by rewriting the left-hand-side as $\sum_{j=0}^{\infty} \sum_{\ell(x)=j} \mu(x)(M(0|x) - \mu(0|x))^2$, that with μ-probability one

$$M(x_n|x_1x_2...x_{n-1}) \to \mu(x_n|x_1x_2...x_{n-1})$$

as $n \to \infty$ and that using M we can learn any computable measure. How long it takes depends on the complexity of the true measure.

Induction. If one is interested in finding the explanation for the data and not only about predicting the future one needs to choose a program. The measure M can be understood as a mixture of all computable sequences by choosing prior probability $Pr(p) = 2^{-l(p)}$ for the program p generating the sequence. The minimum message length (Wallace and Boulton, 1968) (or minimum description length (Rissanen, 1978)) approach chooses the program

$$p^* := \arg\min\{\ell(p) : U(p) = x\}$$

This is the least complex hypothesis that fits the observed data. The choice can also be understood as maximizing the posterior probability $Pr(p|d) = Pr(d|p)Pr(p)/Pr(d)$. This latter expression also extends to having a prior over classes of stochastic hypothesis where a trade-off between how well data fits the hypothesis (i.e. $\log Pr(d|\nu)$) and the prior probability $\log Pr(\nu)$ (that can be based on algorithmic complexity (Hutter, 2005; Li and Vitányi, 2008)) is made by choosing

$$\nu^* := \arg\min_{\nu}\{-(\log Pr(d|\nu) + \log Pr(\nu))\}.$$

2.4 Intelligent Agents

In this section, we consider an agent (Russell and Norvig, 2010; Hutter, 2005) that interacts with an environment through performing actions a_t from a finite set $\mathcal{A}$ and receives observations o_t from a finite set $\mathcal{O}$ and rewards r_t from a finite set $\mathcal{R} \subset [0,1]$. Let $\mathcal{H} = (\mathcal{A} \times \mathcal{O} \times \mathcal{R})^*$ be the set of histories. A function $\nu : \mathcal{H} \times \mathcal{A} \to \mathcal{O} \times \mathcal{R}$ is called a deterministic environment. It is called a stochastic environment if instead the resulting (o, r) is stochastic. A function $\pi : \mathcal{H} \to \mathcal{A}$ is called a (deterministic) policy or an agent. The value function V is defined based on geometric discounting by $V_\nu^\pi(h_{t-1}) = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$ where the sequence r_i are the rewards achieved by following π from time step t onwards in the environment ν after having seen $h_{t-1} \in \mathcal{H}$. If the environment is stochastic we define V using an expectation.

Given a countable class of environments $\mathcal{M}$ and strictly positive prior weights w_ν for all $\nu \in \mathcal{M}$, the a-priori environment ξ is defined by letting $\xi(\cdot) = \sum w_\nu \nu(\cdot)$ and a rational agent (Hutter, 2005) is defined by following the policy

$$\pi^* := \arg\max_\pi V_\xi^\pi(\epsilon)$$

where ϵ is the empty initial history. In particular the AIXI agent (Hutter, 2005) is based on the same construction of a Solomonoff prior M as in the sequence prediction case, by weighting each computable environment based on how short implementations it has on the given reference machine. Though the agent is incomputable, practical approximations exist, e.g. MC-AIXI-CTW (Veness et al., 2011).

3 Cognition and Algorithmic Complexity

Human cognition is naturally very complex as it is an artifact of evolution. However, it is still meaningful to try to identify some of the fundamental principles that are involved with some consistency. After all, cognition operates with limited resources and has evolved to support survival. Cognitive science has discovered many systematic biases (Kahneman et al., 1982) but a question is biases from what ideal. The ideal is usually assumed to be rationality (Chater and Oaksford, 1999; Tenenbaum et al., 2011) in the decision theoretic sense of the word, i.e. expected utility maximization with respect to some a priori probabilities. Another general principle is a preference for simplicity (Chater, 1996, 1999; Chater and Vitányi, 2003). Starting with rationality and a preference for simplicity, algorithmic complexity, as we have described in Section 2, provides a formal "gold standard" theory for sequence prediction, choosing hypotheses, judging similarity, deciding what is random and acting rationally.

Language acquisition. Noah Chomsky has famously argued (Chomsky, 2005) that the "poverty of stimulus" makes the language learning observed in children impossible without an innate grammar. The argument is centered around the lack of negative examples. It has been argued that a rational simplicity-biased

approach, formalized using information theory, can in fact perform this task (Solomonoff, 1964; Perfors et al., 2006; Hsu et al., 2013). This simplicity bias is replacing the need for an innate grammar but can possibly be viewed as a soft version of the same, i.e. a preference for certain syntax, based on what is judged to be simple given the reference machine.

Sequential decision making. The AIXI agent (Hutter, 2005) is formalizing the same combination of simplicity and rationality as in the sequence prediction setting, for the sequential decision making setting. The sequential decision setting, where you make a sequence of decisions that does affect the environment, is far more complicated than performing inference from gathered data. In the sequential setting you are gathering data as you act in the environment and to gather useful data is important as well as achieving high immediate rewards. The data you gather at one time point allows you to make better decisions later on and receive more reward then. This is known as the exploration-exploitation dilemma. A common method to deal with this when building reinforcement learning agents is to introduce optimism into the agent (Szita and Lörincz, 2008; Sunehag and Hutter, 2012a). Optimism is also a principle that applies broadly in human decision making (Weinstein, 1980) and a case can be made that it should be considered another fundamental principle rather than a bias to overcome. The AIXI agent can be modified (Sunehag and Hutter, 2012b) to incorporate optimism in a manner consistent with multiple-prior expected utility (Gilboa and Schmeidler, 1989), though reversing pessimism to optimism. These agents have better guarantees for asymptotic optimality, consistent with the empirical observation that the practical MC-AIXI-CTW agent needs more exploration heuristically added for good performance.

Two other issues are the choice of discount function and the choice of reference machine. In this article we described AIXI based on geometric discounting while the fully general formulation can be found in Hutter (2005). Humans often display a more hyperbolic than geometric discount scheme (Laibson, 1997). The geometric discount is often considered preferable because it is time consistent (Lattimore and Hutter, 2014).

Finally, it has been discussed (Li and Vitányi, 2008) what a natural choice of reference machine would be. The choice will matter as long as we are considering a limited amount of data while asymptotically it does not. An agent based on cellular automata might have a very different sense of what is simple than a human does. The concept of a natural Turing machine could be interpreted as a reference machine that provides a notion of simplicity that agrees with human judgement. The computer programming languages we create to make it easier for us to program computers, capture to some extent a human notion of simplicity. The attempts at finding a more objective notion of a natural Turing machine have so far failed and such a choice might not exist (Müller, 2010). A recently suggested alternative (Sunehag and Hutter, 2014) is to learn a reference machine for which Occams razor is true as a proposition about the relevant world.

Defining and measuring intelligence. Given a reference machine and the

corresponding universal mixture environment ξ, the quantity V_ξ^π has been considered as a universal intelligence measure (Legg and Hutter, 2007). It has much in common with more narrow standard human I.Q. tests where a person's task is to see patterns in e.g. a sequences of numbers and guess their continuation. In principle, one can argue for any next number and find intricate mathematical patterns motivating very different answers. What makes the answers well-defined is that there is an implicit simplicity bias. The continuation should be based on the simplest pattern. If one can find the simplest program explaining what has so far been observed from an unknown environment and take actions as if that is the true environment, one receives high reward in expectation and hence, a high intelligence score when measured by V_ξ^π. This universal measure can be seen as a more complex generalization of human I.Q. tests which have actually been successfully attacked by machine intelligence (Sanghi and Dowe, 2003). This achievement has lead to the conclusion that current I.Q. tests are too narrow to measure general machine intelligence which can be taylored to the task (Dowe and Hernndez-Orallo, 2012). An approximation to the universal test has recently been introduced (Legg and Veness, 2011) and applied to test a number of agents and even more recently a large class of challenging environments has been created using Atari games (Bellemare et al., 2013).

4 Summary

Algorithmic complexity provides a formalization of the notions of simplicity and complexity. It leads up to a formal universal theory of sequence prediction, induction, similarity and randomness based on simplicity-biased rationality, principles suggested as idealized human cognition. Further, for the more difficult sequential decision making setting algorithmic complexity can again together with rationality be used to define a universal intelligent agent as well as a universal measure and definition of intelligence. For the sequential decision making setting, there are indications that further principles like optimism should be considered fundamental and not unwanted biases. Practical approximations of the otherwise incomputable theories discussed have been built based on common compression algorithms and have been successfully used for a range of practical applications including clustering of sequence data like DNA, text and music. Further, algorithmic complexity has also simplified proofs in mathematics and resolved philosophical problems including "problems of induction" in the philosophy of science and the problem with"Maxwell's demon" in statistical mechanics.

References

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The Arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.

Bennett, C., Gacs, P., Li, M., Vitányi, P. M., and Zurek, W. (1998). Information distance. *Information Theory, IEEE Transactions on*, 44(4):1407 –1423.

Bennett, C., Li, M., and Ma, B. (2003). Linking chain letters. *Scientific American*, (June 2003).

Chaitin, G. (1969). On the length of programs for computing finite binary sequences: Statistical considerations. *Journal of the ACM*, 13:547–569.

Chater, N. (1996). Reconciling simplicity and likelihood principles in perceptual organization. *Psychological Review*, 103:566–581.

Chater, N. (1999). The search for simplicity: A fundamental cognitive principle? *The Quarterly Journal of Experimental Psychology Section A*, 52(2):273–302.

Chater, N. and Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends Cogn Sci*, 3(2):57–65.

Chater, N. and Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends Cogn. Sci.*, 7(1):19–22.

Chomsky, N. (2005). *Rules and representations*. Columbia classics in philosophy. Columbia University Press.

Cilibrasi, R. and Vitányi, P. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545.

Dowe, D. and Hernndez-Orallo, J. (2012). IQ tests are not for machines, yet. *Intelligence*, 40(2):77 – 81.

Gilboa, I. and Schmeidler, D. (1989). Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153.

Griffiths, T. and Tenenbaum, J. (2003). Probability, algorithmic complexity, and subjective randomness. In *In Proceedings of the 25th Annual Conference of the Cognitive Science Society*. Erlbaum.

Hsu, A., Chater, N., and Vitányi, P. (2013). Language learning from positive evidence, reconsidered: A simplicity-based approach. *Topics in Cognitive Science*, 5:35–55.

Hutter, M. (2005). *Universal Articial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin.

Hutter, M. (2007a). Algorithmic information theory. *Scholarpedia*, 2(3):2519.

Hutter, M. (2007b). On universal prediction and bayesian confirmation. *Theoretical Computer Science*, 384:33–48.

Kahneman, D., Slovic, P., and Tversky, A. (1982). *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11.

Krakauer, D. (2002). Evolutionary principles of genomic compression. *Comments on Theor. Biol.*, 7:215–236.

Laibson, D. (1997). Golden Eggs and Hyperbolic Discounting. *Quarterly Journal of Economics*, 112(2):443–477.

Lattimore, T. and Hutter, M. (2014). General time consistent discounting. *Theor. Comput. Sci.*, 519:140–154.

Legg, S. and Hutter, M. (2007). Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444.

Legg, S. and Veness, J. (2011). An approximation of the universal intelligence measure. In *Algorithmic Probability and Friends*, volume 7070 of *Lecture Notes in Computer Science*, pages 236–249. Springer.

Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. (2004). The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.

Li, M. and Vitányi, P. (2008). *An Introduction to Kolmogov Complexity and Its Applications*. Springer.

Mahmud, M. M. H. (2009). On universal transfer learning. *Theoretical Computer Science*, 410(19):1826–1846.

Martin-Löf, P. (1966). The definition of random sequences. *Information and Control*, 9:602619.

Müller, M. (2010). Stationary algorithmic probability. *Theoretical Computer Science*, 411:113–130.

Perfors, A., Regier, T., and Tenenbaum, J. (2006). Poverty of the stimulus? A rational approach. *Proc. of the Annual Conference of the Cognitive Science Society*, pages 663–668.

Piantadosi, S., Tily, H., and Gibson, E. (2011). Word lengths are optimized for efficient communication. *PNAS*, 108(9):3526–3529.

Rathmanner, S. and Hutter, M. (2011). A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 3^{nd} edition.

Sanghi, P. and Dowe, D. (2003). A computer program capable of passing I.Q. tests. In *Proceedings of the Joint International Conference on Cognitive Science (ICCS/ASCS-2003)*, pages 570–575.

Solomonoff, R. (1964). A formal theory of inductive inference. Part I and II. *Information and Control*, 7(1,2):1–22,224–254.

Solomonoff, R. (1978). Complexity-based induction systems: comparisons and convergence theorems. *IEEE Transactions on Information Theory*, 24:422–432.

Sunehag, P. and Hutter, M. (2012a). Optimistic agents are asymptotically optimal. In *Proceedings of the 25:th Australasian AI conference*. Springer.

Sunehag, P. and Hutter, M. (2012b). Optimistic AIXI. In *Proc. of the fifth conference on Artificial General Intelligence*, volume 7716 of *Lecture Notes in Computer Science*, pages 312–321. Springer.

Sunehag, P. and Hutter, M. (2014). Intelligence as inference or forcing Occam on the world. In *Proc. of the seventh conference on Artificial General Intelligence*, Lecture Notes in Computer Science. Springer.

Szita, I. and Lörincz, A. (2008). The many faces of optimism: a unifying approach. In *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1048–1055. ACM.

Tenenbaum, J., Kemp, C., Griffiths, T., and Goodman, N. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265.

Veness, J., Ng, K., Hutter, M., Uther, W., and Silver, D. (2011). A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40:95–142.

Wallace, C. and Boulton, D. (1968). An information measure for classification. *Computer Journal*, 11:185–194.

Weinstein, N. (1980). Unrealistic optimism about future life events. *Journal of Personality and Social Psychology*, 39(5):806–820.

Zipf, G. (1949). Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA.