Algorithmic Information Theory and Novelty Generation

Simon McGregor
Centre for Research in Cognitive Science
University of Sussex, UK
sm66@sussex.ac.uk

Abstract

This paper discusses some of the possible contributions of algorithmic information theory, and in particular the central notion of *data compression*, to a theoretical exposition of computational creativity and novelty generation. I note that the formalised concepts of pattern and randomness due to algorithmic information theory are relevant to computer creativity, briefly discuss the role of compression in machine learning theory and present a general model for generative algorithms which turns out to be instantiated by decompression in a lossy compression scheme. I also investigate the concept of novelty using information-theoretic tools and show that a purely "impersonal" formal notion of novelty is inadequate; novelty must be defined by reference to an observer with particular perceptual abilities.

1 Compression, Randomness and Pattern

The intuitive concepts of *pattern* and its converse, *randomness*, are of interest to those in the field of computer creativity. These concepts have been extensively explored in *statistical inference theory*: clearly, anything which has no pattern cannot be predicted; on the other hand, identifying a nonrandom pattern in data should allow us to predict it better than chance in future. Perhaps surprisingly, it turns out that the field of computer science known as *algorithmic information theory* has direct application to formalising the idea of randomness in observed data.

The concept of *algorithmic entropy* or *Kolmogorov* or *Kolmogorov-Chaitin* complexity is central to algorithmic information theory. The Kolmogorov complexity of a binary string is defined simply as the length of the shortest computer program which produces that string as an output. Some strings are *compressible*, i.e. there exists some computer program shorter than the string itself which produces that string as an output. For instance, the first 100,000 digits in the binary expansion of π can be generated by a program far shorter than 100,000 bits. A string consisting of the binary digit 1 repeated 1,000 times can be generated by a program shorter than 1,000 bits. However, it can be shown (Li and Vitanyi, 1997) that most strings are *incompressible*, i.e. they cannot be generated by a program shorter than themselves. Consequently, if you flip a perfectly random coin 100,000 times, the likelihood is that the sequence of heads and tails you obtain cannot be described by a program shorter than 100,000 bits. In algorithmic information theory a string is described as random if and only if it is incompressible.

Note that randomness in algorithmic information theory applies to *strings*, and not to the physical processes which generate those strings. A biased probabilistic random process such as radioactive decay could produce a sequence of 1s and 0s in which 1s were extremely common and 0s extremely rare; that sequence would be algorithmically nonrandom (because favouring 1s is a pattern) despite the fact that it was the product of a random process. Algorithmic randomness refers simply to the absence of pattern in a string.

Despite the best attempts of mathematicians to date, there are still some formal issues which restrict the usefulness of algorithmic entropy as an "objective" measure of randomness. Firstly, algorithmic entropy is provably uncomputable, so it cannot be used in practice. Secondly, in principle its exact value is dependent on the arbitrary reference machine on which programs are run, so that it is "non-arbitrarily" well-defined only in the asymptotic limit.

2 Lossy Compression

As mentioned above, most binary strings are incompressible. This means that theoretically, a compression program which allows objects to be reconstructed from their compressed representations cannot on average turn its inputs into shorter strings! Compression algorithms such as LZW[1] take advantage of the fact that some inputs (e.g. those containing many repeated substrings) are more likely in practice than others; for the majority of possible inputs (those not encountered in practice), the compressed representation will be longer than the original.

[1] Or the other algorithms used by compression utilities such as WinZip.

To make a compression program useful, one needs a *decompressor* (in practice, these two programs are usually bundled together). The decompressor takes the compressed representation of a string as its input and outputs the original string.

A *lossy compressor* is a program which destroys some information about its input in order to be able to produce (typically) shorter representations. For instance, the image compression standard JPEG is a lossy compression scheme. That is to say, when the output of a JPEG compression program is run through a JPEG decompressor, the result is typically not identical to the original input.

3 Generativity and Compression

Imagine a terminating computer program which is supposed to produce objects in our generative domain of interest (e.g. English poems, pictures of animals or rockabilly music). Due to the nature of digital computer programs, the objects generated must be encoded as binary strings (e.g. some ASCII text, or a PNG image, or an MP3 music file). Now, if the same program produces one object whenever it is run, and is capable of producing different objects, in formal terms the program can be considered as taking an input which determines its output. This conceptualisation is general enough to cover programs which operate in some random fashion (the random numbers can be provided as input). It is also general enough to encompass a non-terminating program which generates an infinite sequence of objects by changing its state: we can always write a terminating program which takes a number n as input and outputs the non-terminating program's nth generated object.

This leads to a view in which a generative computer model for a generative domain is seen as a program P which takes an arbitrary binary string as input and outputs a binary string encoding an object. Each bit of the input can be interpreted as a choice point in the process which generates the output. We'll presume that P is written in such a way that it cannot produce an "illegal" output no matter what the input is.

Let's additionally assume that it is possible to write an inverse program P', such that $P'(P(X)) = X$ always. The new program takes the encoding of an object Y and outputs a binary string X which can be fed into P (if there is such a string) to generate Y. If there is no input X which generates Y under P, P' finds the closest object Y^* to Y which can be produced by P, and outputs a binary string X^* which generates Y^* under P.

If its inputs are typically shorter than its outputs, the program P' as just defined is a standard *lossy compressor* program, and our generative model P is just the corresponding *decompressor*. In other words, a successful compression scheme which is computably decompressable yields a generative algorithm. Since optimal compression effectively abstracts away any pattern in data, this should not be surprising. The relation between lossy compression and generativity was noted as early as 1994 in the jokey paper Witten et al. (1994).

4 Learning and Compression

It is a well-known result in machine learning (Li and Vitanyi, 1997) that the shortest program which can produce observed data tends to generalise well to unseen data[2]. This is the centuries-old principle of *Occam's razor* - the simplest explanation is usually the best one. Any machine learning algorithm which is meant to generalise to unseen data from observed data must effectively perform some sort of compression.

Hence, a generative algorithm which is required to learn from its successes and failures can also be understood in terms of compression and algorithmic information theory. The most effective generalisation from past experience will in general be the one which compresses most, i.e. captures the pattern to the greatest possible extent.

5 Aesthetics and Compression

Unlike previous research, (e.g. Svangard and Nordin (2004); Schmidhuber (1997)) this paper does not consider the relation between compression and aesthetics. It focuses on learning, novelty and generativity, which are relevant both in artistic and non-artistic (for instance, engineering or mathematical) creative domains.

6 Novelty and Compression

6.1 The Problem of Novelty

Most theoretical accounts of creativity agree that creative products must be *novel*. Put simply, a product is novel if it is different from some set of already-observed things. Depending on the purpose, this reference set may be defined by what the originator has observed (what Boden (2003) calls personal- or p- creativity), or by what the entire historical community has observed (what Boden calls historical- or h- creativity). But we need to be careful here. "Different" does not merely mean non-identical. If I change one word of A. S. Byatt's "Possession", the resulting product is not novel[3] even though it is not identical to any pre-existing object. It is not "different enough" from prior works to qualify as "genuinely" novel.

That "different enough" is revealing: difference lies on a continuum, with identical objects being zero-different and other pairs of objects varying from hardly different to extremely different. Consequently, new products exhibit degrees of novelty, rather than falling into a binary novel / non-novel categorisation. The degree of novelty of a product depends on a (usually implicit) measure of *similarity* to a (usually implicit) reference class of pre-existing objects. For instance, in Saunders (2001), novelty is appraised using an implicit measure of similarity based on learning in unsupervised neural networks. In other words, novelty is relative not only to what has been seen before but also relative to how things are conceptually grouped together. For any formal version of novelty which relies

[2] Provided that the unobserved data comes from the same distribution as the observed data and that the distribution is computable.

[3] It is of course still *a* novel.

on similarity, it is necessary to specify what measure of similarity is being used.

6.2 Compression

There is a natural, impersonal formal sense in which two binary strings X and Y can be considered similar. The *information distance* (Bennett et al., 1998) tells us how close the algorithmic information in the two strings is. This distance, which is a metric up to an additive constant term, is defined as the length of the shortest program which produces Y given X as input and vice versa. In Bennett et al. (1998), it is described as a *universal cognitive similarity metric*. Formally,

$$E_1(X, Y) = \max\{K(X|Y), K(Y|X)\}$$

where $K(X|Y)$ is the conditional Kolmogorov complexity of X given Y (the length of the shortest progam which produces X given Y as input).

Although Kolmogorov complexity *per se* is uncomputable, an approximation to information distance has been successfully used in Cilibrasi and Vitanyi (2005) to identify similarity between sections of English text, similarity between DNA strings and similarity between musical melodies.

We could extend this formalism to give us measures of how "objectively" novel a binary string X_{n+1} is in comparison to previously known strings $X_1 \cdots X_n$. For instance,

$$\text{Nov}_1(X_{n+1}) = \min\{E_1(X_{n+1}, X_1), \cdots, E_1(X_{n+1}, X_n)\}$$

is the information distance from the new string to the most similar previously known string.

As we will see in the next section, however, the use of this "objective" similarity measure would be at fundamental odds with the goals and methods of computational creativity.

7 Tensions in "Objective" Novelty Generation

By definition, if novelty were held to be algorithmic randomness with respect to known previous examples, then there could not be a compact algorithm which generates maximum novelty. The reason for this is straightforward: when a compact algorithm generates strings, those strings are of a pattern with the other strings it generates.

Furthermore, if an algorithm learned from previous examples what is good and what is bad, and used this information to generate better objects, that would also defeat the end of producing "truly" novel objects. The very similarity which exists between known good objects and differentiates them from bad objects is a pattern which when identified can only be used to produce new good objects which are similar - in a precisely quantifiable sense - to the known ones. Maximally novel objects can in principle only be discovered using random search[4] or by already having a database of highly different objects and simply retrieving them from that database one by one.

7.1 Perceptual Novelty

The impersonal "objective" version of novelty described in the previous section does not correspond to how novel an object will seem to an intelligent observer. Two different clips of random audio white noise sound the same to the human ear, even though in information theoretic terms they are likely to be maximally different from one another (there will be no common pattern to them). As a consequence, a successful theory of creativity will probably need to be a theory of creativity *relative to* some observer whose perceptual and conceptual capacities determine the effective novelty of creative products. We will see shortly that a formal impersonal version of novelty leads to direct contradictions which may be resolved by a perceptually-based theory. For instance, it has been proposed by Schmidhuber (2006) that perceptual novelty is related to the degree to which a new stimulus is expected to improve the observer's predictive model (as his paper observes, a successful predictive model must compress historical data).

Does this mean that human creativity must rely on non-algorithmic processes? Certainly not. What really matters is the perception of novelty by an observer, rather than the "objective" novelty of information theory. A short program can in principle produce a sequence of objects which appear highly dissimilar to a human perceiver, and a series of mutually random objects can appear highly similar. In other words, endless apparent novelty could be generated by a compact program by exploiting the limitations of the perceiver's ability to detect patterns. For instance, a human being zooming into the Mandelbrot set sees novelty for quite a while, because our visual apparatus is unable to pick up the simple algorithm which generates it.

8 Conclusion

The theoretical tools of algorithmic information theory are valuable to researchers in the field of computer creativity, not only because of their potential relevance to formalising aesthetics, but because they formalise the crucial concepts of *pattern* and *randomness*. These concepts are central to learning and computer generativity, and relevant to evaluating the novelty of new generative products. Compression deserves more prominence as an organising idea. For instance, this paper has argued that all generative algorithms can be seen as decompressors for a lossy compression scheme. However, under the most general information theoretic measure of novelty, concise computer programs (and presumably human beings) must always be understood as generating patterns which *appear* novel to a perceptually limited observer, rather than being *objectively* novel in some observer-independent sense.

[4] Using a physical random number generator. The pseudorandom number generators used in typical "stochastic" computer programs do not have algorithmically random output.

Acknowledgements

I would like to thank Chris Thornton for helpful discussion and encouragement; my supervisor Inman Harvey, and an anonymous reviewer who pointed out some major omissions in my references. This research was conducted under a grant from the EPSRC.

References

Bennett, Gacs, Li, Vitanyi, and Zurek (1998). Information distance. *IEEETIT: IEEE Transactions on Information Theory*, 44.

Boden, M. (2003). *The Creative Mind; Myths and Mechanisms*. Routledge.

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

Li, M. and Vitanyi, P. M. B. (1997). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin.

Saunders, R. (2001). *Curious Design Agents and Artificial Creativity*. PhD thesis, Faculty of Architecture, University of Sydney.

Schmidhuber, J. (1997). Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, 30(2):97–103.

Schmidhuber, J. (2006). Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187.

Svangard, N. and Nordin, P. (2004). Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In *Applications of Evolutionary Computing*, pages 447–456. Springer.

Witten, I. H., Bell, T. C., Moffat, A., Nevill-Manning, C. G., Smith, T. C., and Thimbleby, H. (1994). Semantic and generative models for lossy text compression. *The Computer Journal*, 37(2):83–87.