Study of event relevance, rapport mi-parcours

Petru Adrian Dimulescu

May 26, 2010

1 Introduction

This work is placed withing a framework of artificial intelligence that sees understanding as compression. For an observer that receives data from the world, understanding the world means building a relatively simple model of the world that can grasp its regularities and ignore its randomness.

Narrative relevance and its correlate, unexpectedness (or surprise) looks likes an important point of access to cognition. The fact that human communication, at least, is structured around relevant events which are received as unexpected makes us wonder if there is a possibility of framing this general phenomenon in a theory which might eventually throw some light on further aspects of cognition.

1.1 Algorithmic complexity

The incoming data for such an observer can be seen as a symbol sequence (binary sequence for example). Indeed, any data, visual images, auditive or other can be brought down to a sequence of symbols, or binary sequence. We ask the question if there is a way to assess the regularity of such a piece of data, in an observer-independent way (some piece of data should be, in general, regular or not regular (random), independently of the observer – of course, up to some specific particularities of the latter).

The basic mathematical concept of compression is the algorithmic complexity, also known as Kolmogorov complexity of a string. This complexity is operational on a Turing machine (a theoretical model for a computer): the complexity of a string is the size of the smallest program that, run on the given machine, can generate that string.

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

where $U(p) = x$ means that program p generates x when run on machine U.

For example, suppose the observer receives the description '`HHHHHHHH`' from the world. What is the complexity of this description? A way to assess it would be to take a program '`8*H`' meaning "eight times the litteral `H`", and run it on the machine. We can see that the size of the 'compressed' description is significantly smaller than the original string, which makes the original string 'simple', i.e. having a low Kolmogorov complexity.

What does a complex string look like ? It might look like this : '`HHTHTTTH`'. Depending on the machine, the shortest description of this string may be the string itself (no compression would be possible). Of course we might have a machine that 'overlearned' exactly this pattern, knows it by heart, a machine for which this specific sequence is simple. But as string length increases, the observer's particularities tend to disappear, regularity and randomness becoming absolute.

A mathematical theory of induction based on the algorithmic complexity has been set up by [Solomonoff, 1964]; this has given some developments recently into a theory of general artificial intelligence in [Hutter, 2004] which establishes theoretical mathematical basis of artificial intelligence.

We might see a program that generates a sequence as a *cause* of that sequence.

1.2 Algorithmic probability

A parallel and interesting notion is the algorithmic probability of a string. On a fair coin, the chances to get heads, `H`, is 1/2. The chances to get two succeeding heads, `HH` is 1/4, and so on. The chances to have the sequence `HHHHHHHH` is 2^{-8} that is two to the power of the negative length of the string. So if the mechanisms of the world are random, the probability of getting `HHHHHHHH` or `HHTHTTTH` are the same. But the world has regularities, we try to infer its inner workings, so this definition is not sufficient. Can we have a definition of probability that would render simple descriptions more probable than complicated ones?

The algorithmic probability of a string does just that. The simpler and the more numerous the causes of a string are, the higher its algorithmic probability. $Pr(x) = \sum_{p:U(p)=x} 2^{-l(p)}$ where $U(p) = x$ means that program p generates x when run on machine U. $l(p)$ is the length of program p. So in this case it is not the length of string that is taken into account, but the length of the causes of the string.

1.3 Encoding

If we have a program, we just need to feed it to a Turing machine, which will deterministically compute its output. The operation is theoretically trivial. The opposite, encoding, is inferring the program (cause) that might generate a piece of data. This is called the inversion problem, is incomputable in the general case.

Still, making sense of the world means encoding data that occurs, which is exactly the inversion problem. Not only finding the smallest program p that generates x but, if possible, using some pre-existing 'programs' that the observer might have acquired in its previous experience of the world.

1.4 Event relevance

The relevance of an event, or the relevance of an incoming piece of data, can be articulated in this frameset. The goal of this report is to review some work that has been aimed in this direction and to hopefully propose novel ways of calculating relevance and event surprise. If a relevant event is an event that generates interest, it matters to propose a model for what an interesting event is, or equivalently, to model surprise in a complexity-based framework.

2 Information theory primer

When concerned by the complexity of data sequences, we have first to note that there is a bijection between data structure and numbers. Indeed, any symbol sequence, finite or not, can be associated, in lexicographical order, to a corresponding real number. Studying data complexity can be brought down to the study of complexity of numbers. What is a simple and what is a complex number?

We will introduce the problem of complexity by introducing the prime numbers following [Chaitin, 2005]. Natural numbers can be subjected to the operation of prime factors decomposition. It has been noticed since the ancient times that some numbers cannnot be decomposed in prime factors other than 1 and themselves. One could ask himself, then: is there a finite or an infinite number of prime numbers?

Decomposing a natural number into prime factors can be seen as an operation of compression. We describe a possibly large number using powers of numbers less than itself. We use then less information in order to describe it than if we actually wrote its decimal configuration. A simple example is the number $2^{999} * 3^{999}$ which, if developed, need hundreds of decimal digits to be written.

The question of the finitude of the prime number list can be seen of asking if all natural numbers are compressible. The answer to this question is negative and several demonstrations can be brought. Not only the prime numbers set is infinite but most natural numbers cannot be written in a shorter form than their textual representation.

The same question can be asked of real numbers. On real numbers, the question bears even more philosophical sense as it brings together the notions of compression and naming. A compressible real number is a nameable one, and the negation holds, too: an uncompressible real number is a number that cannot be defined nor named. Let us formulate the question first. Are there "irrational" real numbers, that is, real numbers that cannot be written as a basic arithmetic operation of natural numbers? The history of geometry starts with one, it is $\sqrt{2}$, the length of the diagonal of a square. This number cannot be written as a fraction of natural numbers $\frac{n}{m}$ because if it could, then $\left(\frac{n}{m}\right)^2 = 2 \Rightarrow n^2 = 2m^2$. But then, decomposing both n^2 and $2m^2$ in prime factors, we get an even exponent for 2 in the first decomposition and an odd exponent in the second, which would imply that several prime factor decompositions are possible for a given number. We cannot thus express $\sqrt{2}$ as a ratio of natural numbers.

A more interesting division of real numbers is between algebraic and trancendant numbers. Algebraic real numbers can be written as the solution of an algebraic equation. Transcendant real numbers cannot. Following this definition, algebraic number have a short, finite definition (that of the algebraic formulas whose solution they are) while transcendant real numbers do not. It follows that trancendant numbers are not easily nameable, at least not in a concise algebraic fashion.

The algorithmic complexity (also known as the Kolmogorov complexity) of a number is the size of the shortest program that, when run on an universal Turing machine, prints out that number and stops. It follows that "regular" strings have low algorithmic complexity, while string that have no such regularity must be created by a program that contains the string itself. The complexity is then closely related to the notion of compression and regularity detection.

The notion of randomness is simple to define once we have hold of the algorithmic complexity. A string of size n is random if its algorithmic complexity is no less than n. It is regular, if its algorithmic complexity is less than n

We are talking here about the complexity of numbers (or of bit strings because we can bring any number to its binary notation), still complexities we observe in nature are complexities of surrounding objects. To the extent that any object lends itself to a description, we define the complexity of that object as the complexity of its description, digitized and transformed into a sequence of bits. This sequence can be seen as a large binary number. We can encode in this way an encyclopaedia in a number, or, rather theoretically, even the whole of human knowledge (for instance, in Borel's number, which is a real number that contains, for its i^{th} binary digit, the answer "true" or "false" to the i^{th} question of all possible questions ordered in lexicographical order).

Are there more complex numbers or more non-complex numbers? In other words, which is more rare, the regular, or the random?

"Fermez les yeux et choisissez au hasard un nombre réel dans cet intervalle, en faisant en sorte que la probabilité de tomber sur l'un ou sur l'autre soit exactement la même: la probabilité que vous choisissiez un nombre réel calculable est égale à zéro" [Chaitin, 2005], p 140.

Computable real numbers are thus the exception rather than the rule; the computable is rare and rather unexpected.

2.1 Prefix-free codes

The mathematics of Kolmogorov complexity can be written in the simplest form when self-delimiting strings are used. A self-delimiting string, or a prefix-free code is a bit string that contains its own delimiting information. For example, given the binary string `111101110`, if we know it contains two strings, x and y, without a limiting convention, we cannot uniquely distinguish x from y. Still, if we make the convention that 0 is a stop sign and 1s are used in order to unary encode the string, we see that the string uniquely decodes to, in decimal, 4 and 3 (more efficient encodings exist). A machine can decode each string without having to look ahead or memorize. A prefix-free Turing machine can be seen as a regular Turing machine that has an read-only input tape, a write-only output tape, and a number of working tapes.

Prefix codes are characterized by Kraft's inequality (equation 1) which states that a set of natural numbers l_i are the length of a prefix-free set if the following inequality holds :

$$\sum_i 2^{-l_i} \leq 1 \tag{1}$$

When using prefix codes we can use event probabilities in order to approximate underlying information content in a simple manner. As the sum of probabilities is 1, taken into account Kraft's inequality, we can put in one-to-one relation probability and average code legth using the Shannon-Fano coding : $C = -\log P$. This is a raw approximation which makes use of the statistics contained in a probability distribution and not at all of one particular object's regularities.

The conditional complexity, $K(y|x)$ is the length of the shortest program to calculate y if x is also provided in input to the Turing machine. Kraft's inequality is verified by the conditional complexity, which can be interpreted as stating that in the “neighbourhood” of x, there is a limited number of “close” objects.

$$\sum_y 2^{-K(y|x)} \leq 1 \tag{2}$$

2.2 Information distance

To the same extent that Kolmogorov complexity is an absolute measure of information that an object description has, [Bennett *et al.*, 1998] proposes an information distance between two objects that is universal and optimal (it minimizes any other acceptable distance). The information distance is a metric, which means that it is symmetric, the distance from an object to itself is zero and respects the triangle inequality:

- $D(x, y) = 0$ if and only if $x = y$
- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$

The conditional complexity $K(y|x)$ cannot be used, as is, a information distance because of its lack of symmetry. The information distance between the strings x and y is defined as the length of the shortest program that can compute y from x as well as x from y, with F a prefix (self-delimiting) Turing machine. The minimal program that turns x into y overlaps to a large extent of the one that, inversely, turns y into x, instead it is shown to maximally overlap it.

$$E_0(x, y) = \min\{l(p) : F(p, x) = y, F(p, y) = x\} \tag{3}$$

Another expression of the information distance is written as the maximum of the two conditional complexities:

$$E_1(x, y) = \max\{K(x|y), K(y|x)\} \tag{4}$$

The two equations, 4 and 3 are proven to be equivalent up to an additive logarithmic term.

An *acceptable* information distance D itself must verify a normalization constraint $\sum_y 2^{-D(x,y)} \leq 1$. The notion of acceptable distance allows us to enforce the intuition that each object has only a limited number of neighbours and eliminates “useless” metrics like $D(x, y) = 1/2$ for $x \neq y$ and 0 otherwise and to formulate the intuition that there is a limited number of neighbours within a given range. $E(x, y) \overset{+}{=} E_0(x, y) \overset{\log}{=} E_1(x, y)$ minorizes any acceptable distance.

$K(x, y)$ is the length of the shortest program that prints x, y and a description of how to tell them apart [Li *et al.*, 2003].

If x^* is the program whose length is $K(x)$, the information about x contained in y is $I(x : y) = K(x) - K(y|x^*)$. Then $K(x, y) = K(x) + K(y|x^*) = K(y) + K(x|y^*)$, equalities up to a constant additive term.

The sheer information distance $E(x, y)$ cannot be applied properly when comparing string with unequal length. Two huge strings that only differ by a small amount d of information would be considered as distant from each other as two completely unrelated strings of size d. [Li *et al.*, 2003] addresses this issue by proposing a "normalized" metric that is related to $E(x, y)$, called the *normalized information distance* (NID):

$$d(x,y) = \frac{\max\{K(x|y^*), K(y|x^*)\}}{\max\{K(x), K(y)\}} \tag{5}$$

Equations look much simpler if we remove the max functions : suppose $K(x) < K(y)$. Then $E(x, y) = K(y|x)$ and $d(x, y) = \frac{K(y|x)}{K(y)}$. The actual approximation of these values can be done by using a compressor. Knowing that $K(x|y) = K(x, y) - K(y)$, we can express the conditional complexity in terms of the complexity of the two concateneted prefix-codes. Moreover, we can approximate $K(x, y)$ by $K(xy)$, that is, we relax the prefix-free condition, if we accept an additive logarithmic error.

The NID verifies a normalized version of Kraft's inequality:

$$\sum_y 2^{-d(x,y)\times K(x)} \leq 1 \tag{6}$$

In practice, as the Kolmogorov complexity is uncomputable, one must approximate it using existing compressors. [Cilibrasi and Vitanyi, 2005] developed a tool named CompLearn which uses available open source compressors like `gzip`, `bzip2` or `ppmz` in order to achieve this approximation. On such tools, the calculation of $C(x|y)$ is achieved through the calculation of the equivalent $C(x, y) - C(y)$. Approximating $C(x, y)$ with $C(xy)$, we get from equation 5 the expression of the normalized compression distance, which is equivalent except that it uses a compressor C in order to approximate K:

$$\begin{aligned} NCD(x,y) &= \frac{\max\{C(x|y^*), C(y|x^*)\}}{\max\{C(x), C(y)\}} \qquad (7)\\ &= \frac{C(xy) - \min\{(C(x), C(y)\}}{\max\{C(x), C(y)\}} \qquad (8)\end{aligned}$$

Generally, if S is a finite alphabet, a compressor is defined as "a lossless encoder mapping S^* into $0, 1^*$ such that the resulting code is a prefix code". Interesting compressors are those whose resulting codes are shorter than the original string. In order for the NCD to be a similarity metric, the compressor C must be a *normal compressor*, that is, it must verify a number of properties:

1. Idempotency: $C(xx) = C(x)$
2. Monotonicity: $C(xy) \geq C(x)$
3. Symmetry: $C(xy) = C(yx)$
4. Distributivity: $C(xy) + C(z) \leq C(xz) + C(yz)$

Any compressor with these properties can be explored for interesting similarity distance detections. Choice of compressors can be of course done among the existing file-compression algorithms, but not only. A search engine [Cilibrasi and Vitányi, 2007] can be used as a compressor in order to calculate semantic distance between words.

How can a search engine be a compressor? Let us restate that a compressor is a function whose domain is a prefix code. If the search engine can return the number of hits for single terms x, y and for the combined $x \cap y$ (that is, the number of pages containing both terms), given the total number of documents M, we can calculate the probability of occurrence of either terms or their combination. This probability can be normalized to sum to 1, which allows us to infer the underlying information content of a corresponding prefix-code. In the sense of a function that generates a prefix code, a search engine can be seen as a compressor and we can apply the formula described above in order to calculate distances between words.

Note that the compression provided by a search engine using the Shannon-Fano coding is only the *expected* minimal code, not the *individual* calculated complexity. It is thus an approximation of NCD which itself is an approximation of NID.

The number $|x \cap y|/M$ cannot be interpreted as the probability $Pr(x \cap y)$ because, as search terms overlap, the summed probabilities over the entire set of terms S is bigger than 1. In order to bring this probability to 1, we use N, the total number of hits obtained by trying every combination of x and y (including those with $x = y$). In this case, $N = \sum_{x,y \in S} |x \cap y|$. Defining the $g(x) = g(x, x)$ and $g(x, y) = |x \cap y|/N$, we get a probability density distribution which sums up to 1. The implied prefix code is $G(x) = -\log(g(x))$ which can be seen a search engine compressor.

The resulting formula is called the *normalized google distance* (NGD), and can be used against any public search engine that provides hits counts (or for home-grown indexes):

$$NGD(x, y) = \frac{G(x \cap y) - \min\{(G(x), G(y)\}}{\max\{G(x), G(y)\}} \tag{9}$$

$$= \frac{\log \max\{f_x, f_y\} - \log f_{xy}}{\log N - \log \min\{f_x, f_y\}} \tag{10}$$

with f_x, f_y and f_{xy}, the number of hits for x, y, and $x \cap y$. The number N is difficult to calculate but it may be approximated by another figure, like M (the total number of documents).

The resulting number is not a mathematically pure distance. As shown even by its creators, it does not thoroughly observe the triangle inequality. In a practical setup, though, on the wikipedia corpus we used for other measurements, the triangle inequality is generally observed (we ran a script that chosses thousands of random groups of three terms and checks triangle inequality between them with close to zero cases of non-observance).

2.2.1 Normalized conditional complexity

If we stick with the unsymmetric conditional complexity. $G(y|x)$ returns often very frequent, small-coded, grammatical terms. There is a need to normalize, in order to retrieve the "closest" meanings.

$$\frac{G(y|x)}{G(x)} = \frac{G(x, y) - G(x)}{G(x)} = \frac{G(x, y)}{G(x)} - 1 \tag{11}$$

as $G(x,y) = \frac{1}{g(x,y)}$ with $g(x,y) = f_{xy}/M$. we get $G(x,y) = \log_2 M - \log_2 f_{xy}$ The equation becomes :

$$\frac{\log M - \log f_{xy}}{\log M - \log f_x} - 1 \tag{12}$$

3 State of the art on surprise

An event is relevant if it produces interest on the observer. It matters to define interest or, equivalently, surprise or unexpectedness. We review some work in what follows, work that is formulated in various frameworks, but which all have the common ground of analyzing surprise as a *difference* between two ways of describing the data.

3.1 Complexity-Drop Theory

The Complexity drop theory (CDT) [Dessalles, 2006; Dessalles, 2008a] proposes, together with Schmidhuber's work on novelty and low-complexity art, among the first predictive models of interestingness to be based on Kolmogorov complexity. The theory models cognition using two cognitive machines: a world (or generation) machine, which simulates the causal workings of the known world and a description machine which actually describes the event using language-level constructs. The CDT holds that events appear interesting for humans when the complexity of the event using the world machine is larger than that on the description machine. On this account, events are interesting when their description is simpler than their generation from the known workings of the world.

The CDT formula of unexpectedness, $U = C_w - C$ permits approximative quantitative calculation of interest levels of events [Dimulescu and Dessalles, 2009], or see `http://www.unexpectedness.eu`.

The Complexity-Drop Theory or Simplicity theory focuses on the complexity of the piece of data. No explicit observer model intervenes, yet two machines are required in order to account for the surprise generated by one given piece of data.

Surprise is defined as the difference between the *generation complexity* and the *description complexity*, measured on two distinct machines. The generation machine contains a model of the world and its input program is a code containing parameters that need to be set in order for this model of possible worlds to generate the exact state contained by the event.

The description complexity is "the length of the smallest available description". It makes use of all the cognitive abilities of the observer.

An example is a lottery drawing: if the sequence `111111` comes out at a lottery drawing, its generation complexity is $6 \times \log 10$ if we supppose the representation of a single decimal digit to take $\log 10$ binary bits, while its description may use a very light copy operator in order to describe the whole sequence in pretty much one copied digit, close to $\log 10$. The surprise of this event would be then around $5 \times \log 10$. If the representation of one digit requires more than $\log 10$ bits, which is plausible given the difficulty of human brains to manipulate numbers, than the computed surprise is even more ample.

When two analogous situations s_1 and s_2 occur independently, their generation complexity $C_w(s_1\&s_2)$ is close to $2 \times C(s_1)$, whereas their description requires only $C(s_1) + C(s_2|s_1)$, which is smaller if the analogy is close. Individuals experience such situations as coincidences and consider them worth telling. CDT's scope includes other important aspects such as a new approach to subjective probability [Dessalles, 2008b].

Most situations are not unexpected, which means that their generation and their description require approximately the same amount of information. There are exceptions, however, and these exceptions make the topic of conversational narratives.

3.2 Surprise in a bayesian framework

[Itti and Baldi, 2009] follow the problem of surprise in a bayesian framework. [Baldi, 2002] notes that "while eminently successful for the transmission of data, Shannon's theory of information does not address semantic and subjective dimensions of data such as relevance and surprise". The surprise is the difference between the prior and the posterior probability distributions that a bayesian model represents. The probability distributions are modelled as bayesian networks and the distance between the two distributions is the Kullback-Leibler distance [Kullback and Leibler, 1951].

Bayesian networks model probability distributions using nodes that factor out common "sources" of probability.

This work makes an important assumption: that surprise is essentially *subjective* and that it can only be measured when a model of an observer is explicitely indicated. A stimulus, or a piece of data, can is thus surprising relatively to that observer's view of the world, according to its particular expectations:

"Surprise, no matter how one defines it, is obviously related to Shannon's information: a rare event is in general surprising and ought to carry a great deal of Shannon information due to its low probability. But beyond this obvious relationship, a theory of surprise should to be able to measure information surprise that is contained in data (1) in an observer-dependent way (2) related to his changes in expectation; (3) through a definition that clearly establishes a connection with the foundations of probability theory and (4) clarifies the "white snow" paradox and related concerns" [Baldi, 2002]

The "white snow" paradox mentioned above is the exception to the rule "rare is interesting". Indeed, if we watch a TV screen with random "no channel" black and white pixels, one given snapshot of the screen is extremely rare, yet not at all interesting. Whence the necessary distinction between subjective and objective probability.

The surprise produced by data D on model M is:

$$S(D, M) = KL(P(M|D), P(M)) = \int_M P(M|D) \log \frac{P(M|D)}{P(M)} dM$$

with $KL(P(M|D), P(M))$ being the Kullback-Leibler distance between the prior and the posterior distribution.

3.3 Beauty and surprise

But is it possible to articulate the study of surprise in a Kolmogorov complexity framework? [Schmidhuber, 2010] attempts to formalize subjective notions like beauty and surprise using just this mathematical foundation that we explore. A two-part code is "beautiful" if its regular part is overwhelming. In other words beauty is "already known". Surprise is then the first derivative of beauty:

$$I(D, O(t)) \sim \frac{\partial B(D, O(t))}{\partial t}$$

The evolution of a cognitive device is seen in this terms as the continuous improvement of an adaptive compressor that tries to predict the future based on history. If the size of the compressed predictor is significantly smaller than the actual history, than the world presents regularities and useful compression has been achieved.

This work is combined with an effort to build an implementation of a piece of software, named OOPS (The Optimal Ordered Problem Solver) [Schmidhuber, 2002] which solves some inversion problems using a modified version of Levin search, when possible reusing programs that were already acquired in solving previous tasks.

3.4 Integration hypothesis: surprise as sense making

A parallel theory of unexpectedness [Maguire and Maguire, 2009] relates it to sense-making. In this account, a cognitive device like the human brain constantly updates a working model of reality; when a stimulus is not coherent with the internal representation of the world, that stimulus is perceived as unexpected.

A series of expriments measures the level of surprise declared by participants showing that the more well-installed beliefs are in general rule-like (or program-like) causal explications, rather than specific ad-hoc not-yet integrated experience data, the stronger the perceived unexpectedness.

4 Experimental setup for the complexity-Drop Theory

A study was attempted to measure the variations of interest aroused by conversational narratives when definite dimensions of the reported events are manipulated. The results are compared with the predictions of the Complexity Drop Theory, which states that events are more interesting when they appear simpler, in the Kolmogorov sense, than anticipated.

We conducted thus an experiment [Dimulescu and Dessalles, 2009] to test CDT's predictions on interest. A corpus of 18 stories in French was established, using material from personal recorded sources and from a French community Web site (`viedemerde.fr`) where people describe short every-day life events, in an informal style, sometimes with a humorous touch. The stories were presented to 95 participants. The 14 most illustrative stories are listed below in their English version. Answers to the stories may seem *obvious*, as individuals have a clear intuition of what contributes to interest. What is less obvious is that the rich phenomenology of interest illustrated by the stories can be backed by a unified theory. The situation parallels the problem of syntax: people have a clear intuition of which sentences are syntactically correct in their mother language, but designing a predictive model of syntactic correctness remains a scientific challenge. Besides, experiment data offered some surprises, as explained in the discussion.

1. *I was walking quietly in the street when a total stranger stops before me, looks at me and* [...] *before continuing his walk.* ⓐ gives me a phenomenal slap; ⓑ gives me a slap; ⓒ asks me the time.
2. *This afternoon, the police of Antibes discovered in the Baie des Anges area the floating dead bodies of two elegantly-dressed women. Apparently, the two accidents happened almost at the same time. Moreover, both women were wearing* [...]. ⓐ a red tattoo on the right arm representing the Tsuba-Kasai dragon; ⓑ a red tattoo on the right arm; ⓒ a tattoo on the right arm.
3. *I'd just bought a small Peugeot 106 ColorLine for 2000 euros. I had tried it the day before and it was very good. I turned the key, I started, I left the property of the former owner of the car when, coming from the left without looking, another* [...] *crashed into me.* ⓐ Peugeot 106 ColorLine; ⓑ Peugeot 106; ⓒ Peugeot.
4. *For the birthday of my little sister, my parents got her an 82-cm black flat TV screen. A little while ago, for my own, they got me* [...] ⓐ a 82-cm black baseball bat; ⓑ a black baseball bat; ⓒ a baseball bat.
5. *Wednesday, the city of Amiens police seized* [...] *kg of heroin at number 13 rue Fafet.* ⓐ 10; ⓑ 5; ⓒ 2.
6. *For a year, I had been thinking of changing my mobile phone at SFR (mobile operator). I finally decided to do so even if I had to pay a part because I did not have enough Red Square Points. I bought the new phone at 13:00.* [...] *I got a message from SFR: "Change your mobile, SFR offers you 15 000 Red Square Points."* ⓐ At 13:10; ⓑ At 14:00; ⓒ Two weeks later.
7. *I was lying on the ground under the trees on an autumn afternoon, when a leaf fell exactly* [...]. ⓐ on my nose; ⓑ on my face; ⓒ on my body.
8. *Two weeks after my car had been stolen, the police informed me that a car that might be mine was for sale on the Internet. They showed me the ad. The phone number had been identified. It was the mobile phone number of*[...]. ⓐ my office colleague; ⓑ a colleague of my brother's; ⓒ someone of my neighbourhood.
9. *I was walking in a street in downtown Paris when I heard someone calling me: it was a guy who I'd been babysitting* [...] *when he was a child. We exchanged addresses, it was really nice.* ⓐ for 2 years; ⓑ for 2 months; ⓒ a few evenings.
10. *It's funny, I found this on the Internet: the town of St-Chéron has* [...] *inhabitants.* ⓐ 4444; ⓑ 4000; ⓒ 3856.
11. *I got the license plate of my new car; I looked at the number I got, it was* [...]. ⓐ 999 NNN 91; ⓑ 253 NNN 91; ⓒ 253 UPV 91.
12. *In front of the coffee machine, I was talking to a colleague about SpongeBob and his best friend Patrick, the sea star. At some point I said: "I don't like Patrick, he's too stupid". At this precise moment, my boss passed by, his name is* [...] ⓐ Patrick Star; ⓑ Patrick; ⓒ Christopher.

13. *You know what? My neighbor, downstairs, she gave birth to* [...] *a few days ago. It was a premature birth. She said she was surprised to see them so small.* ⓐ quadruplets; ⓑ triplets; ⓒ twins.

14. *I had to pay a one year old fine of 400 euros because of the Treasury, which didn't send reminders to the correct address. When the sum was debited, there was exactly* [...] *euros on my account.* ⓐ 400; ⓑ 401; ⓒ 419.

All stories were presented in random order to all participants. Most participants were engineers and students working in fields other than cognitive science. For each story we isolated a parameter that we considered important to the relevance of the story and defined three options that would gradually affect the overall interest. One given participant only got to mark a preference between two such options, so that the overall ranking purpose of the test be less transparent. The test was implemented as a Web application. Participants had to fill-in a missing excerpt by clicking on one of two randomly chosen options that were displayed below the main text. Participants had "to select the option that made the story more interesting". A reward (USB stick) was to be given to those whose overall choices were most consistent with the majority. Results are listed in Table 1.

4.1 Ranking

At the end of the process, each story produced a list of three paired comparisons between its fill-in options. We computed a ranking of story versions to see if it was congruent with the predictions of CDT. In a manner similar to [Saaty, 1994], we want to calculate the rank r_i of a version i, given pairwise relations between versions. Knowing the win/lose history of competition between version i and j, we define w_{ij} as the ratio $wins(i)/wins(j)$. In order to avoid division by zero, we accorded one vote by default to all versions. For example, if the comparison between two versions of a given story was presented to 30 participants of which 23 chose the first version while the remaining 7 participants chose the second, then $w_{ij} = 24/8$. Note that $w_{ji} = 1/w_{ij}$. We want a story version that performed well against other versions to have a rank that reflects its win, taking into account both the rank of the defeated version and the bluntness of the win. A possible calculation is $r_i = k \sum_j w_{ij} r_j$, rewritten as $\frac{1}{k} R = WR$ which amounts to finding an eigenvector of the symmetrically reciprocal matrix W [Farkas, 2007]. Ranks are then normalized so that each eigenvector sum to 100.

Table 1 shows the overall results and rankings. For each story, versions are ordered from most interesting to least interesting according to CDT. Most of the comparisons (those marked with *) were found statistically significant ($p < .05$) on a standard binomial test targetted at detecting a marked preference for one particular version. Participants' choices are highly congruent with the predictions. Rankings show that participants altogether never preferred option (c), which is excluded by the model. For 17 stories out of 18, option (a), which is the most congruent with the model, is significantly ranked best. In only one story, option (b) tended to be preferred to option (a).

Table 1: Comparison between paired options and the resulting option rankings for each story

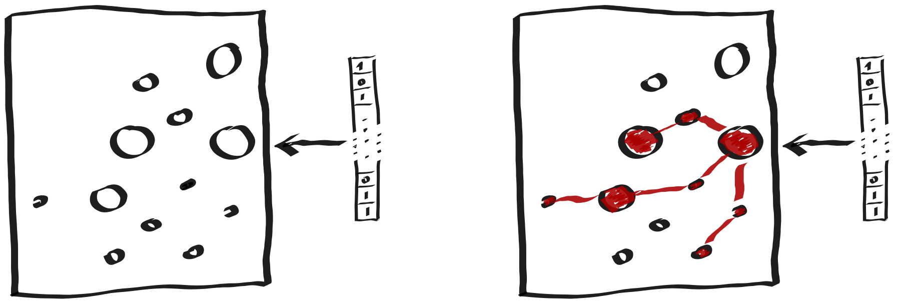
	comp.		rank
1	a-b	22/18	a 53
	b-c*	25/4	b 41
	a-c*	27/3	c 6
2	a-b	17/14	a 40
	b-c*	34/5	b 49
	a-c*	23/9	c 11
3	a-b*	24/8	a 61
	b-c*	32/4	b 32
	a-c*	28/5	c 7
4	a-b	18/16	a 38
	b-c*	22/11	b 39
	a-c*	20/14	c 23
5	a-b*	28/13	a 54
	b-c	20/11	b 28
	a-c*	21/8	c 18
6	a-b*	29/4	a 79
	b-c*	28/6	b 16
	a-c*	31/3	c 5
7	a-b*	26/4	a 68
	b-c*	36/4	b 25
	a-c*	24/6	c 7
8	a-b*	25/6	a 65
	b-c*	24/10	b 22
	a-c*	26/7	c 13
9	a-b*	25/11	a 58
	b-c	20/12	b 26
	a-c*	25/7	c 16
10	a-b*	35/1	a 92
	b-c*	21/11	b 4
	a-c*	30/2	c 3
11	a-b*	28/5	a 76
	b-c*	22/11	b 15
	a-c*	30/4	c 9
12	a-b	18/16	a 52
	b-c*	29/4	b 43
	a-c*	31/3	c 5
13	a-b*	24/11	a 56
	b-c	21/12	b 27
	a-c*	25/8	c 17
14	a-b	13/20	a 33
	b-c*	29/5	b 56
	a-c*	24/9	c 11
15	a-b*	24/9	a 56
	b-c*	28/6	b 33
	a-c*	26/8	c 11
16	a-b*	30/7	a 70
	b-c	17/15	b 16
	a-c*	25/5	c 14
17	a-b*	31/12	a 60
	b-c*	21/8	b 28
	a-c*	24/6	c 12
18	a-b*	32/5	a 66
	b-c*	30/1	b 31
	a-c*	28/4	c 3

5 Surprise on an encoding machine

Let us take a cognitive device, seen as a machine containing a set of programs that sequentially receives pieces of data from the outside world and tries to encode them. A prefix-free Turing machine has an input tape, an output tape and a number of work tapes. Such a machine receives an "input" program and generates an "output" sequence. The operation is theoretically trivial and deterministic. Not so concerning the opposite operation, that is finding the programs responsible for generating a given sequence. This is called the inversion problem [Solomonoff, 1997] and is in general incomputable.

Making sense of the world can be seen as an operation of encoding or compressing incoming data. Randomness or regularity judgments can be best understood as efforts of encoding presented data ([Falk and Konold, 1997]).

Encoding incoming data x means finding one or several programs p_i so that $M(p_i) = x$. The program p can use some preexisting subrouting that the cognitive device may have acquired in its previous experience. The program p_i



may be a two-part code, that is, containing a “regular” part and a “random” part. The regular part may be formed by reusing existing “concepts”, i.e. programs already contained, while the random part may supplement the regular one in order to account for the exact data presented.

The obtained encoding can be seen as the ‘meaning’ of the incoming piece of data. It is easy to note that one given piece of data can have several parallel encoding, not necessary of the same length, that all contribute to its algorithmic probability. This can be interpreted as a sign (incoming data) having several interpretations, or meanings. Within this mindset, we can equate encoding and meaning.

Such a machine could be seen as an adaptive compressor, in that it tries to find short codes for incoming data and also it aims at evolving these codes so that the overall size of the cognitive device remain within reasonable bounds.

The effort of encoding data can give several results then. One possibility is that the data be *expected*, in other words, an existing program is found that encodes the data completely. In this case the model is perfectly adapted to stimuli and we can consider the data *uninteresting*. What about the situation where there is no way for the existing model to encode data? A possibility is that this specific situation generates interest or surprise. In other words, when the encoding of the incoming data requires a change in the existing model, the *quantity of change* required is the level of surprised experienced by the encoding machine.

The analysis of surprise as the quantity of change in a set of programs, measurable in bits, is related to the bayesian analysis of surprise as the distance between the prior and the posterior probability distributions ([Baldi, 2002]) as well as to the view of surprise as the first derivative of beauty (improvement of compression adequacy) [Schmidhuber, 2010; Schmidhuber, 2008].

5.1 Relation to representativeness

[Tenenbaum and Griffiths, 2001] propose a formal measure of [Kahneman *et al.*, 1982]’s representativeness using what is called a *Bayes factor*: the evidence that a piece of data bring in favor of one particular hypothesis, in contrasts with alternative hypothesis.

On the relation between “bayesianism” and complexity : *“a key distinction between bayesianism and ideal MDL lies in the distinction between codes that are optimal on average and codes that are optimal in every individual case”* ([Vitanyi and Li, 2000]). Results coming from the bayesian research can be rearticulated in a complexity frame using work like this. The bayesian aspect that's interesting for us is the belief-update cognitive model which contains hypothesis and allows for matching between stimuli and hypothesis upon which belief strength may change; the question of the priority between complexity and probability is of no importance in what follows, although the in the following we focus on Kolmogorov complexity using Turing-machine programs for modelling hypothesis.

The Bayesian update

The well-known bayesian equation describes belief update. If we have a hypothesis h of data generation, and event x has occurred, the strength of our belief in h may change following the success or failure of the match between x and h. The equation is then :

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)}(1)$$

The word probability is not univocal (for a critical review of the frequentist, objectivist, subjectivist approaches, see [Hutter, 2004]). The subjectivist take on probability is the most adequate to our target. In the case of a hypothesis, $P(h)$ is the degree of belief held with respect to it. In the case of an outcome, $P(x|h)$ is the likelihood of having x produced by hypothesis h. When we refer to a ratio like $\frac{P(h_1)}{P(h_2)}$, this is called the odds of preferring h_1 over h_2 and can be seen a measure of preference of a theory over the other. The prior and the posterior probabilities can then be related with the term "availability" or "associative distance" from [Tversky and Kahneman, 1973].

From [Vitanyi and Li, 2000] : finding a good hypothesis h for data d means finding h with maximum probability of matching the data, i.e. with maximal $P(h|d)$. The rewrite of (1) in terms of complexity is $C(h|d) = C(d|h) + C(h) - C(d)$. Which makes the task of finding the best hypothesis h as the hypothesis which minimizes $C(h|d) = C(h) + C(d|h)$, the two-part code, consisting of a "model" regular part and of a meaningless random part. The goal is thus to minimize this sum.

Surprise as Bayes factor?

Let us take two hypothesis underlying observed data x and see how this piece of data favors one hypothesis over the other. In the case of random coin flipping, we may think of two major choices: fair coin (which essentially means no regulation whatsoever) or some unknown mechanism that generates structure. A comparison of hypothesis `rnd` and `reg`.

From (1), applied successively, follows :

$$\frac{P(rnd|x)}{P(reg|x)} = \frac{P(x|rnd)}{P(x|reg)}\frac{P(rnd)}{P(reg)}$$

This equation can be understood as describing a modification of a hypothesis "weight", or odds, after a stimulus x is perceived.

The left term is the posterior odds of the two hypothesis. The second factor of the right term is the prior odds of those hypothesis. The factor that influences the odds modification, the only one depending on the observation x, is the middle factor whose log is called the Bayes factor.

In order to explain human biases in estimating probabilities, [Kahneman *et al.*, 1982] propose that the cognition calculates the representativeness of a piece of data, relative to a hypothesis in order to estimate probability.

But representativeness as presented in [Kahneman *et al.*, 1982] still wants a formal expression. In a bayesian framework, the representativeness of x for the randomness hypothesis is proposed by [Tenenbaum and Griffiths, 2001] to be the log of the Bayes factor:

$$repr(x, rnd) = \log \frac{P(x|rnd)}{P(x|reg)}$$

The equation is then : `posterior_odds = repr(x, rnd) * prior_odds`. *The quantity represented by the middle term can be seen as the amount of belief change in order to account for stimulus x.*

In general terms, given a main hypothesis h and a number of alternative hypothesis alt_j, the definition of representativeness is :

$$repr(x, h) = \log \frac{P(x|h)}{\sum_j P(x|alt_j)P(alt_j|\bar{h})}$$

The denominator is a sum of the likelihoods of x appearing on all alternatives, on the condition that the main hypothesis h is false.

We can see that $repr(x, h)$ is an improper notation since this representativity also depends on the alternative hypothesis (for a psychological experiment which show the importance of contrast to alternative hypothesis see also [Teigen and Keren, 2003]). A more complete notation would be perhaps something like $repr(x \rightarrow [h|alt_1, alt_2])$ with the meaning of the representativeness of stimulus x over a hypothesis space in which h is the stronger hypothesis and the rest are its less strong siblings.

Some problems with probabilistic representativeness

[Tenenbaum and Griffiths, 2001] calculate representativeness using the Bayes factor and illustrate it on a coin flip. Given to data, $x_1 =$ `HHHHH` and $x_2 =$ `HTHHT`, and hypothesis of a fair coin ($h_{fair} : p(H) = 0.5$), biased coin or head-only coin, they show that $repr(x_2|h_{fair}) >> repr(x_1|h_{fair})$. There are two problems with this example:

a) The first problem is the origin of hypothesis: representativeness cannot be calculated in this account without a sum of alternatives. It matters to define where does the hypothesis space come from and how the selection of the main hypothesis is done, in order to define a piece of data as surprising. While a definitive answer is not our goal, this aspect is partially treated in what follows. Our proposal is that the hypothesis space is the set of programs that a cognitive box can present as possible alternative representations of a stimulus x, and the main hypothesis against which the representativeness is calculated is the most available one, or in complexity terms, the shortest program that represents x.

b) Lack of compression problem: if we take stimuli $x_1 =$ `HTHHT` and $x_2 =$ `HHHTT` and calculate its representativeness against the h_{fair} hypothesis (defined using probabilities as show above), we will find that the two values are equal. Intuitively, however, the “grouped” character of x_2 makes it seem subjectively less random than x_1. This is because the only aspect that $p(H) = 0.5$ formulation captures is counting the number of heads. While this is an important feature of a bit string is it by no means the only feature. Other features, like grouping or structure, escape such a probability-based formulation. Adding a compressor (or observation device) to the cognitive box, allows for the introduction of surprise in cases where the Bayes factor does not suffice.

Rewriting the Bayes factor

The relation between bayesian inference and Kolmogorov complexity-based models like ideal `MDL` are explored by [Vitanyi and Li, 2000]. The paper shows that, on restricting possible hypothesis to finite sets, the replacement of p with the universal prior distribution $m(x)$ is possible. The use of m basically means the assignment of a probability that is based on the quantity of information, in the Kolmogorov sense, that the object has, rather than on its statistics of apparition.

Let us rewrite this equation in the form of complexity, in the simplified case of a single alternative hypothesis, using complexity as the negative log of probability. (1) becomes:

$$\log \frac{P(h|x)}{P(alt|x)} = \log \frac{P(x|h)}{P(x|alt)} \frac{P(h)}{P(alt)} \Leftrightarrow$$
$$C(alt|x) - C(h|x) = C(alt) + C(x|alt) - C(h) - C(x|h) (3)$$

$C(h|x)$ is the a posteriori complexity of the h hypothesis (after the stimulus x occurred). $C(h)$ is the a priori complexity of the `reg` hypothesis.

Let's note $\Delta_{posterior} = C(alt|x) - C(h|x)$ the difference of complexity between the two posterior updated hypothesis. Let's note $\Delta_{prior} = C(alt) - C(h)$ the difference of complexity between the prior hypothesis. Then:

$$repr(x,h) = C(x|alt) - C(x|h) = \Delta_{post} - \Delta_{prior} (4)$$

Here $repr(x,h)$ is the belief change measured between h and alt. If x is representative for h than h gets shorter or remains about the same, resulting in a typical unsurprising experience, but if x is surprising for h, then the alternative hypothesis gets shorter (and therefore more probable) than the main hypothesis h.

Let us now see another interpretation of equation (3). Suppose we have a model which can produce a causal explanation of x using a combination of existing hypothesis (programs) of the world, as it was comprehended before stimulus x occurred. Let us call C_w (complexity using the "world" model) the complexity of this hypothesis (which we will name alt just in order to match the notation above). Then $C_w = C(x|alt)$. This world complexity is then a two-part code ([Vitanyi and Li, 2000]) that describes the stimulus using the existing world model.

Suppose the cognitive device contains a compression device responsible for encoding data x. This compression device can detect some regularities like, for example, repetitions. The operation of encoding data x generates a hypothesis, which we will identify with h in the notation above, a hypothesis whose length is the "observation" complexity $C_o(x) = C(x|h)$. Note that in the two cases, world and observation complexity are two part codes. (3) can be rewritten then as :

$$repr(x, h) = C_o(x) - C_w(x)(5)$$

By defining unexpectedness as the negative of representativeness, we retrieve the definition of unexpectedness in [Dessalles, 2008a]. The main difference relies in the fact that they are not measured on two distinct machines, but are length of hypothesis on the same cognitive machine, and that this unexpectedness in a measure of the change needed in the cognitive system in order to account for the unexpected data.

A simple machine that gets surprised

Let us take the classical example of a coin flip, and model a cognitive device that gets surprised. The cognitive device is a box of hypothesis (programs) equipped with a compression device for presented stimuli. For this example, an event is a group of 8 coin flips; like `HHHHHHHH`, or `HTTHTTHT`. The cognitive machine disposes of a simple compression language. In order to encode x = `HHHHHHHH` it may use the string `*8H` ("polish" notation for 8 `H`'s). The string `HHHHTTTTT` could then be encoded as `*4H*4T`. In order to be able to peacefully convert between probabilities and complexity we need to use prefix-free codes so we will reserve the character "." for marking the end of a string. The prefix-free code for code x will be noted $\bar{x}$ with $l(\bar{x}) = l(x) + 1$. As we use ASCII characters instead of binary, for readability reasons, the base of our encoding will be $b = 256$.

An encoding of exceptions will be provided reserving the character `X` in order to specify that what follows will replace the existing generated string on machine's output tape. For example, `*8HXT` means that the last character of the produced string will be overwritten with `T`, producing the string `HHHHHHHT`.

Finally, the cognitive device is subject to a constant constraint of overall minimal size. In order to achieve this, various factorization techniques may be employed, much like a software code base is factored out into procedures and functions, and every such basic unit of code is given a name, usually shorter than the actual code, in order to ensure minimum redundancy. The resulting representation structure could, for instance, resemble a Bayesian network graph in which nodes are subprograms and the dependency relations between them show how those subprograms can be put together. Every node might be interpreted as a more or less abstract concept.

We shall suppose that our new-born cognitive device does not have a developed world theory of how the sequence $x =$ "`HHHHHHHH`" was produced and it will only contain a null hypothesis : $alt_1 = \bar{\epsilon}$. alt_1 is the empty "anything goes" theory, $l(alt_1) = 1$.

The observation compressor containing a simple run-length encoding algorithm that only detects repetition, provides the following theory for the incoming data : $h =$ `*8H` with $l(h) = 3 + 1 = 4$. For now, the cognitive device has no other prior beliefs on the strength of the three hypothesis other than their algorithmic probability. Thus, $p(alt_1) = b^{-1}$, $p(h) = b^{-4}$. The "anything goes" theory alt_1 starts up in pole position as the most "available" (or probable in the algorithmic sense) hypothesis.

Let's calculate representativeness: $repr(x, alt_1) = C(x|h) - C(x|alt_1) = 0 - 0 = 0$ predicts no surprise, which essentially means that this definition of representativeness does not seek a minimum hypothesis length, and is not interested by a minimal encoding length.

So we cannot use this exact definition of representativeness in order to define surprise.

Let's see something else then. What happens upon reception of x ? As the most prevalent theory is the shortest, "anything goes" alt_1, let us see if this stimulus is perceived as surprising or not. As $C(x|alt1) = 0$; $C(x|h) = 0$, we have :

```
l(h) + C(x|h) = 4           <- 2nd-place hypothesis encodes better, surprise!
l(alt1) + C(x|alt1) = 9     <- "main" hypothesis
```

The surprise comes from the fact that an alternative, secondary hypothesis encodes better the stimulus than the currently most available one. Obviously such a world model is not the most economical one. A possibility is to discard the stimulus as non-relevant, as an "oddity". If it comes back, and here is the role of repetition, the model has to change.

It would be interesting to follow this path in order to show concept formation; concepts first originate in the fact that short compressed programs become more available (less complex) because of the power of compression of "observation"; then they become part of the "world" and observation needs to be better than the accumulated concepts in order to be able to bring surprise.

The fact that, before the "anything goes" hypothesis which basically states "nothing can surprise me", a simple structure is still surprising is remarkable it shows that when no bias exists, simplicity, purely syntactic simplicity, pattern repetition constitutes material for surprise and hypothesis formation.

The role of the repetition of a stimulus

At this point, the cognitive box contains two programs: $alt_1 = \bar{\epsilon}$ and, the old h, rebaptised now $alt_2 =$ `*8H` . If $x =$ `HHHHHHHT` comes, its observation encoding is `*7HT`. C(x—h) = 0; C(x—alt1) = 9; If alt2 is used for encoding, the resulting program would be `*8HXT` so $C(x|alt_2) = 2$; All existing hypothesis can encode x, alt2 is the favored smallest hypothesis:

```
l(h) + C(x|h) = 5 + 0  <- best encoding
l(alt1) + C(x|alt1)  = 0 + 9
l(alt2) + C(x|alt2) = 4 + 2 = 6 <- favorite hypothesis loses by 1 bit,
                                            so small surprise.
```

with surprise $U(x) = 6 - 5 = 1$ bit.

What if lots of `HHHHHHHH` come first and only then one `HHHHHHHT` ? Suppose there are lots of `HHHHHHHH` that are presented to the cognitive box. Each presentation of HHHHHHH is followed by pressure to the overall compression device in order for a minimization of the representation of the stimulus (corresponding to the bayesian posterior probability update), ending up with having a one-bit (ascii bit, remember) representation. Suppose that the cognitive box sets the 1-bit sequence $\bar{\#}$ (any other non-used character would do) to stand for the program $*\bar{8}H$. Let's measure the surprise now of `HHHHHHHT` now. Let's do the calculation above in this second scenario:

```
l(h) + C(x|h) = 5 + 0
l(alt1) + C(x|alt1)  = 0 + 9
l(alt2) + C(x|alt2) = 2 + 2 = 4   <- favorite hypothesis wins, no surprise
```

The small particularity of this stimulus does not really need to generate a model change. Even if not completely according to the model, the existing hypothesis alt_2 is sufficiently small to encode in the shortest manner the data even with minor corrections. So here is an example of a stimulus that might or not be surprising, function of the internal state (model) of the cognitive device.

Hypothesis priors

When we say "a lottery number is..." we create an expectation, we prime the hypothesis of randomness as we know a lottery is random; when we say "a fair coin gave..." etc. we prime a hypothesis, of p=0.5. It is equivalent to giving temporary shorter codes to hypothesis whose length would make them less available normally. 66666 is not really that interesting without context, it is more interesting when the hypothesis "lottery" is primed.

We may suppose that the cognitive box has a way of priming hypothesis. This is connected to language somehow, difficult to treat it here. But we can model it as an assignment of a shorter code than the hypothesis itself. Making it available, the stimulus is matched against it, as the most available, not against some other hypothesis.

Biased coin scenario: surprise of the more complex than expected

Let us take the scenario "a head-only coin generated the sequence `HHTHTTHT`" and model it. The mention "a head-only coin" makes very available the already-short hypothesis h_{head} = `*8H`. The observation cannot make much sense of the sequence, storing it literally h_{obs} = `HHTHTTHT`. $l(h_{head}) = 3 + 1 = 4$, $l(h_{obs}) = 8 + 1 = 9$. h_{head} starts as favorite shortest hypothesis. $C(x|h_{head}) = 8$ (the hypothesis doesn't help at all so we have to store the sequence in addition to the useless hypothesis) while $C(x|h_{obs}) = 0$.

```
l(h_head) + C(x|h_head) = 4 + 8 = 12 - favorite loses, 3 bits of surprise
l(h_obs) + C(x|h_obs) = 9 + 0 = 9
```

Once again, the favored hypothesis, h_{head} proves a worse encoder than an alternative, less available hypothesis which results in the unexpectedness.

A tentative definition of unexpectedness

Let CB be a finite set of programs for a given universal Turing machine; CB can be seen as a cognitive box of programs. Let x be a piece of data presented to CB. Let $H \subset CB = \{h_i \in CB | C(x|h_i) << b\}$ be a subset of CB which contains reasonable candidates for encoding x (that is, which encode x in a two part code of length C(h) + C(x—h), with less than b extra bits). b represents an effort barrier, and only allows adequate hypothesis to candidate for encoding x, knowing that in principle any program can be modified to encode x even if it does not contain any of x's regularities, by simply adding some code that removes anything produced by it and adding the sequence x at the end.

Let $h_f \in H$ be the "favored" hypothesis of H, that is, the program from H with lowest complexity. Let $h_b \in H$ be the "best" program of H, that is the one that actually best encodes x, the program for which the two-part code whose length $C(h_b) + C(x|h_b)$ is minimal. x is surprising for CB if $h_f \neq h_b$ and the quantity of unexpectedness is given by the difference between the two attempted encodings:

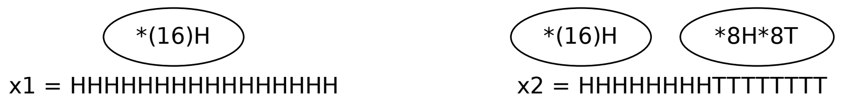
$$U(x, CB) = C(h_f) + C(x|h_f) - C(h_b) - C(x|h_b)$$

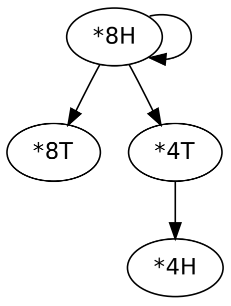
which happens to be also $C(h_f|x) - C(h_b|x)$, the difference of a posteriori complexity of the two hypothesis.

Still, a more adequate definition of interest may be worth exploring : equating unexpectedness with the *quantity of change* that a cognitive box must undergo, following an incoming piece of data. A formal definition of the latter must include a explicit quantization of change in a formal cognitive box. A sketch of it is attempted below. If the cognitive box does not factor out common pieces of code into low-complexity abstract concepts, but keeps programs separately, then the effect of a surprising incoming stimulus is just a complexity rewrite of the Bayesian belief update, like above: the concerned hypothesis change complexity in order to account better for the encoded data; – but if the organization of the cognitive box is graph-like, with programs sharing segments of code, then change should be calculated as the quantity of graph reorganization, of which the above equation (and maybe the CDT equation) is only a particular case (a particular case where nodes and their relations stay the same, only their complexity changes).

Example growth of a cognitive box

Suppose the cognitive box is subject to minimum overall description length. Algorithms that factor out duplication in the cognitive box should ensure minimality, typically, by removing duplication of code. Here is how it might grow over some data:





x3 = HHHHHHHHTTTTHHHH with refactoring

The `*8H` program, as the most connected one, should be the less complex one; a smaller code than its actual "executable" length could be assigned to it, for example; its being central and very interrelated with other nodes makes incoming data that does not conform to it very surprising. For instance if the string `*7H*9T` comes, it will be difficult to encode, given the fact that the model is now quite established around the fact that strings are supposed to start with 8 H's. If the model has to encode `*7H*9T` and, for overall minimality considerations, has to give up the `*8H` node, the change should be painful because some other structure should be proposed to replace *8H in all the connection that it has created.

This network ressembles a bayesian network; the differences are that nodes encode programs; also network structure is not predefined, attributes that typically correspond to nodes in a bayesian machine learning framework are not known a priori in this case; they are gradually induced from common regularities in data in an unsupervised manner. The structure of the network is unknown a priori, it grows with incoming data and describes its regularities.

On such a network unexpectedness could be measured as the quantity of change that the network needs to be subjected to in order to account for the incoming data.

One important aspect of this model is that a small (simple) program is automatically highly connected. Any data that affects this small program is interesting (to relate to landmark-like interest).

References

[Baldi, 2002] Pierre Baldi. *A computational theory of surprise.* 2002.

[Bennett *et al.*, 1998] C. H. Bennett, P. Gacs, M. Li, P. Vitanyi, and W. H. Zurek. Information distance. *IEEETIT: IEEE Transactions on Information Theory*, 44, 1998.

[Chaitin, 2005] Gregory Chaitin. *Hasard et complexité en mathématiques.* Flammarion, Paris, traduction français edition, 2005.

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

[Cilibrasi and Vitányi, 2007] R. Cilibrasi and P. M. B. Vitányi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[Dessalles, 2006] J. L. Dessalles. A structural model of intuitive probability. In *International Conference on Cognitive Modeling*, pages 86–91, April 2006.

[Dessalles, 2008a] J. L. Dessalles. Coincidences and the encounter problem: A formal account. In *30th Annual Conference of the Cognitive Science Society*, pages 2134–2139, July 2008a.

[Dessalles, 2008b] J. L. Dessalles. *La pertinence et ses origines cognitives.* Hermes-Science publications, 2008b.

[Dimulescu and Dessalles, 2009] Adrian Dimulescu and Jean-Louis Dessalles. Understanding narrative interest: Some evidence on the role of unexpectedness. In N. A. Taatgen and H. van Rijn, editors, *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 1734–1739, 2009.

[Falk and Konold, 1997] Ruma Falk and Clifford Konold. Making sense of randomness: Implicit encoding as a basis for judgment. *Psychological review*, 104:301–318, 1997.

[Farkas, 2007] András Farkas. The analysis of the principal eigenvector of pairwise comparison matrices. *Acta Polytechnica Hungarica*, 4(2), 2007.

[Hutter, 2004] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions Based On Algorithmic Probability.* Springer, 1 edition, November 2004.

[Itti and Baldi, 2009] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306, June 2009.

[Kahneman *et al.*, 1982] Daniel Kahneman, Paul Slovic, and Amos Tversky. *Judgment under Uncertainty : Heuristics and Biases.* Cambridge University Press, April 1982.

[Kullback and Leibler, 1951] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[Li *et al.*, 2003] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi. The similarity metric, 2003.

[Maguire and Maguire, 2009] Phil Maguire and Rebecca Maguire. Investigating the difference between surprise and probability judgements. In *Cogsci 2009*, 2009.

[Saaty, 1994] T. L. Saaty. How to make a decision: The analytic hierarchy process. *Interfaces*, 24(6):19–43, 1994.

[Schmidhuber, 2002] Juergen Schmidhuber. Optimal ordered problem solver, Dec 2002.

[Schmidhuber, 2008] Juergen Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. Dec 2008.

[Schmidhuber, 2010] J. Schmidhuber. Artificial scientists & artists on the formal theory of creativity. Technical report, 2010.

[Solomonoff, 1964] R. Solomonoff. A formal theory of inductive inference, part i. *Information and Control*, 7(1):1–22, 1964.

[Solomonoff, 1997] Ray J. Solomonoff. The discovery of algorithmic probability. *Journal of Computer and System Sciences*, 55(1):73–88, 1997.

[Teigen and Keren, 2003] K. Teigen and G. Keren. Surprises: low probabilities or high contrasts? *Cognition*, 87(2):55–71, March 2003.

[Tenenbaum and Griffiths, 2001] Joshua B. Tenenbaum and Thomas L. Griffiths. The rational basis of representativeness. In *23rd Annual Conference of the Cognitive Science Society*, pages 84–98, 2001.

[Tversky and Kahneman, 1973] A. Tversky and D. Kahneman. Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, 5(2):207–232, 1973.

[Vitanyi and Li, 2000] Paul Vitanyi and Ming Li. Minimum description length induction, bayesianism and kolmogorov complexity. 2000.