On Autistic Interpretations of Occam's Razor

José Hernández-Orallo
Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació
Camí de Vera s/n, E-46022, València, Spain.
E-mail: jorallo@dsic.upv.es.

Ismael García-Varea
Universitat Politècnica de València
Institut Tecnològic d'Informàtica
Camí de Vera s/n, E-46022, València, Spain.
E-mail: ivarea@iti.upv.es

MODEL-BASED REASONING IN SCIENTIFIC DISCOVERY (MBR'98)
Pavia, Italy, December 17-19, 1998

Induction as Compression

This maxim has become increasingly popular:

- Occam's Razor (Ockham 1290?-1349?)
- Unsupervised Learning as Compression (Solomonoff 1964)
- Universal Distribution based on Description Length (Kolmogorov 60's)
- Minimum Message Length (MML) Principle (Wallace 1968)
- Minimum Description Length (MDL) Principle (Rissanen 1978, 1996)
- "All kinds of computing and formal reasoning may usefully be understood as information compression [...]" (Wolff 1995)

ML Literature is full of assertions like: the shorter the theory the better (the more likely, the more plausible...).

Is this maxim valid for explanatory induction?

Is this maxim valid for scientific discovery?

Formalisation of the MDL principle

Karl Popper objected:

"there is no criterion of simplicity".

Stochastic Complexity and Kolmogorov Complexity are well-established criteria of simplicity:

DEFINITION 1. KOLMOGOROV COMPLEXITY

The *Kolmogorov Complexity* (KC) of a string x on a bias β:

$$K_\beta(x|y) = \min \{ l_\beta(p_x(y)) \}$$

where p_x denotes any "prefix-free" β-program for x using input y and $l_\beta(p_x)$ denotes the length of p_x in β.

$K_\beta(x) = K_\beta(x|\varepsilon)$ where ε denotes the empty string.

Kolmogorov Complexity is an absolute and objective criterion of simplicity. It is independent (upto a constant term) of the descriptional mechanism.

In absence of any other knowledge about the hypotheses distribution (autism), one choice is the prior distribution $P(h) = 2^{-K(h)}$

Success of the MDL principle

MDL Principle (Rissanen 1978)

"The best model to explain a set of data is the one which minimises:
- the sum of the length, in bits, of the description of the theory
- the length, in bits, of data when encoded with the help of the theory.

Then, we enclose the exceptions, if any."

The MDL principle matches with Kuhn's notion of "changing paradigms":
Exceptions are patched until they are long enough to force the revision of the theory.

By using approximations, it has been successful for different descriptive mechanisms and applications.

Provides a compromise between:
- over-generalisation (underfitting)
- under-generalisation (overfitting).

Problems and Paradoxes of the MDLP

- *Not computable*. $K(h)$ is not computable.
- *Relative, in the end*. Many computable approximations, like $Kt(h)$, dynamically change as the learner knows that something can be further compressed.
- *Perfect data*: the MDL under-fits perfect data: new examples are quoted until their compression is worthy.
- *Discontinuous*: The reliability of the theory is not always increasing with the number of examples that have confirmed the theory. E.g. the sequence $(a^n b^n)^*$ is more compressible if $n=10^{10}$ than if $n= 78450607356$.
- *Inconsistent with Deduction*: e.g. given T_a and T_b, intuition (and logic) says that $T = T_a \vee T_b$ should have more probability, but the MDL principle assigns less probability to T because it is larger.
- *Frequently non-explanatory*: For the sake of maximum mean compression, some part of the hypothesis may be not compressed at all.
- *Frequently unmanageable*: For the sake of maximum compression, the theory can be computationally intractable.
- *Frequently Non-Informative or Non-Creative*: If the data is random ($K(x) = l(x)$) $\Rightarrow$ theory = data

Problems with Explanation. Example

EXAMPLE 1. Short data.

Data: 1, 2, 3, 5, 7, 11, 13.

- *Shortest Description:*
  "1,2,3,5,7,11,13"
  → Completely extensional.
- *Shortest 'Predictive' Description:*
  "Odd numbers until n=13 with positive exception 2 and negative exception 9" + Definition of "Odd".
  → Partially extensional.
- *Intensional Description:*
  "Prime numbers until n=13" + Definition of "Prime".
  → Completely intensional.

The last one matches with Popper's criterion of falsifiability.

Extensionalities can never be falsified

⇩

We should avoid exceptions…

What is an intrinsic exception?

Approaches to the Idea of Exception

Necessarily based on the idea of mean compression ratio:

$$CR(T) = l(M(T)) / l(T)$$

→ *First approach*: A part E of a theory T such that:

$$CR(E) << CR(T)$$

→ *Corrected approach*: A part E of a theory T such that can be removed such that the data which is now uncovered $M(T) - M(T–E)$ or erroneous $M(T–E) - M(T)$, follows this equation:

$$l\{M(T) - M(T–E)\} \cup \{M(T–E) - M(T)\} / l(E) << CR(T)$$

With $\Delta(p)= e$ we will denote the length (in bits) of the greater exception of a description p.

- A formal definition of $\Delta(p)$ requires a general definition of *subprogram* or *part*. This must be certainly based on the idea of separation: "*something is separable if the cost of describing the whole is similar to the cost of describing the parts*", which is also closely related to the idea of exception.

$\Delta(p)$ can be easily defined for Model-based languages, like first-order logic, equational languages, …

Intensional Complexity

DEFINITION 2. INTENSIONAL COMPLEXITY

The *Intensional Complexity* (IC) of a string x on a bias β:

$$E_\beta(x|y) = \min \{ l_\beta(p_x(y)) : \Delta(p_x) = 0 \}$$

where p_x denotes any β-program for x using input y and $l_\beta(p_x)$ denotes the length of p_x in β.

i.e. the shortest program for x without intrinsic exceptions.

$E(h)$ integrates:

- avoidance of exceptions, and
- syntactical simplicity.

The prior $P(h) = 2^{-E(h)}$ could be seen as an adaptation for explanation of the MDL principle ($P(h) = 2^{-K(h)}$).

- *Simplicity is important but secondary.*
- *Nothing is noise or casual, all must be explained. All is intensional. All has a meaning, a cause…*

Explanatory Complexity

Intensional Complexity is not enough for explanation.

Something is an explanation *only if* it can be related to others.

In the same way,

DEFINITION 3. LEVIN'S LENGTH-TIME COMPLEXITY

The *Levin Complexity* of a string x on a bias β:

$$Kt_{\beta}(x|y) = \min \{ LT_{\beta}(p_x(y)) \}$$

where $LT_{\beta}(p_x) = l(p_x) + log_2 \text{Cost}(p_x)$

makes Kolmogorov Complexity Computable,

DEFINITION 4. EXPLANATORY COMPLEXITY

The *Explanatory Complexity* (EC) of a string x on a bias β:

$$Et_{\beta}(x|y) = \min \{ LT_{\beta}(p_x(y)) : \Delta(p_x) = 0 \}$$

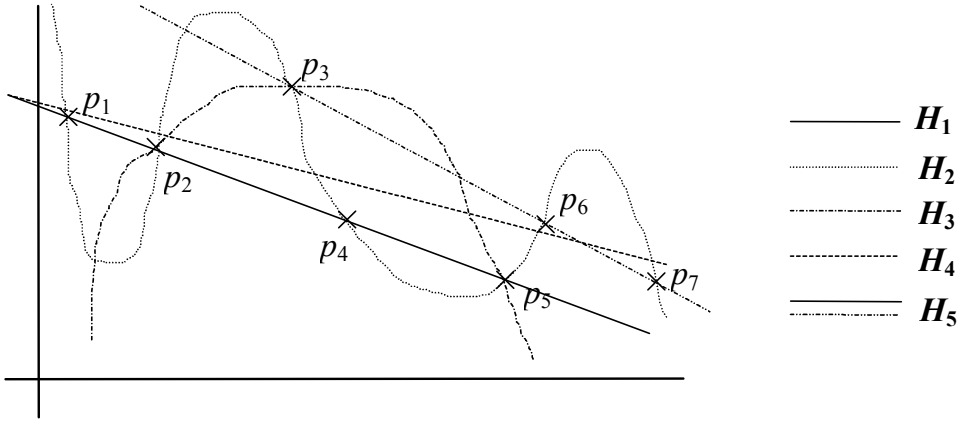
where $LT_{\beta}(p_x) = l(p_x) + log_2 \text{Cost}(p_x)$

avoids intractable descriptions.

Intensional does not mean Creative

Intensional Complexity is not always valuable when E(x) > l(x).

EXAMPLE 2.



$H_0 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7$	*Extensional*
H_1 = 1st order polynomial $+p_3+p_6+p_7$	*Partially Extensional*
H_2 = 6th order polynomial	*Intensional*
H_3 = 2nd order polynomial $+p_1+p_4+p_6+p_7$	*Partially Extensional*
H_4 = 1st order interpolation	*Approximation*
H_5 = Two first order polynomial	*Intensional*

H_6 = *"Spanish petrol prices evolution in this decade"*
H_7 = *"Distance of satellite Europe last year"*

There is always an 'easy' intensional description (H_2).

Computational Information Gain

Which descriptions are really valuable?

DEFINITION 4. COMPUTATIONAL INFORMATION GAIN
The *Computational Information Gain* of a string x wrt. a string y on a bias β:

$$G_\beta(x \mid y) = Kt(\mathrm{x} \mid \mathrm{y}) \;/\; Kt(\mathrm{x})$$

THEOREM 1. LIMITS OF $G_\beta(x \mid y)$
For every x and y, $log\ l(x)/(l(x) + log\ l(x)) \stackrel{+}{<} G(x \mid y) \leq 1$.

THEOREM 2. ROBUSTNESS TO POLYNOMIALITY
Consider a *learning* or *discoverer* algorithm A^* in $\odot$ (i.e. polynomial), namely $\exists p \in \odot : O(n^{p-1}) \leq O(A^*) \leq O(n^p)$, being A^* of constant size, i.e., $l(A^*) = c$, such that this algorithm deterministically transforms y into x, where x is a program for y, being $n = l(y)$.

If $Kt(x) > k \cdot p \cdot log\ n$, then $G(x \mid y) \leq 1 / k$.

Proof of Theorems 1 and 2

PROOF OF THEOREM 1. The second inequality $G(x \mid y) \leq 1$ is obvious by choosing $y = \varepsilon$ and the definition of $Kt(x)$ as $Kt(x \mid \varepsilon)$. The first inequality is justified by the fact that the numerator

$$\mathrm{Kt}(x / y) \geq \log l(x)$$

because x must be printed and this takes at least $l(x)$ units of time. In fact this limit can be come close if $x = y$ because the program "print y" has cost approximately $2 \cdot l(x)$. The denominatior must follow

$$\mathrm{Kt}(x) \stackrel{+}{<} l(x) + \log l(x)$$

because in the worst case, when x is random, we need $l(x)$ + c bits of information for the program "print x" and at least $l(x)$ units of time to be printed. By (1) and (2) we have that $log\ l(x)/(l(x) + log\ l(x)) \stackrel{+}{<} G(x \mid y)$. □

PROOF OF THEOREM 2. For every string of data y, let us construct x in the following way: $x =$ *"apply A* to y"*. Since we can construct x from <A*,y> in an easy way p= "apply 1st argument to 2nd argument" Kt(x / <A*,y>) <= $LT(p)$ = $l(p)$ + $log\ cost\ (p)$<= c + log n^p). It is obvious that Kt(x / y) <+ Kt(x / <A*,y>). So we have that ≤ log n^p = p log n. If, as supposed, $Kt(x) > k \cdot p \cdot log\ n$, then the quotient $G(x \mid y)$ = K(x/y) / K(x) ≤ 1 / k. □

Valuable Descriptions

There are infinite descriptions and theories to some data.

Which are useful to remember?

Is it valuable to store the computational effort which has been invested?

$G_\beta(x \mid y)$ *provides a uniform measure to evaluate theories:*

If x is the theory and y is the data, we have

Minimum: $G_\beta(x \mid y) = \log l(x) \,/\, (l(x) + \log(l(x)) \approx 0$

The theory is evident from the data. It is very easy to describe the theory from the data. $Kt(x \mid y)$ ↓↓

Examples: → the polynomial obtained using the data.
→ Exceptions ($Kt(x \mid y)$ ↑↑)
→ Extensionalities (part of x is in y)

Maximum: $G_\beta(x \mid y) = 1$

We have that $Kt(\mathrm{x} \mid \mathrm{y}) = Kt(\mathrm{x})$. The data is useless (in time-space terms) to describe the theory. It is necessary a great computational work on the data y to obtain the theory or there is a need for external information.

What is to Discover?

A concept x is *surprising* wrt. y in a context β iff:

$$G_\beta(x \mid y) \uparrow\uparrow$$

A concept or theory x is a *discovering* wrt. y in a context β iff:

$$G_\beta(x \mid y) \uparrow\uparrow \text{ and } G_\beta(y \mid x) \approx 0$$

i.e, x is surprising for y and x is an efficient theory for y.

In a proper way, discovering must be accompanied by a confirmation, however x is valuable *per se*.

Induction must be non-autistic in this way. The value of the inductive theory must be evaluated regarding:

- Its intrinsic value.
- The context.
- The purpose.

Conclusions

Motivation:

We have commented on the problems of the maxim "learning as compression". In any case, the maxim "discovering as compression" is not sustainable.

Two main problems of MDL's 'autism':

- explanation
- creativity or informativeness.

Many times the MDL principle does not explain all the data and/or gives naive theories (few informative).

Partial Solutions:

- We have presented "Intensional Complexity" to address the problems of "Kolmogorov Complexity" for explanation.
- We have introduced the idea of "Computational Information Gain" to clarify what is to discover and what is not.

Current and Future Work:

- Establish the relation between *E* and *G*.
- Give a unified and operative alternatives to *E* and *G*.
- Relate to other complexity notions like logical depth.