Boolean analogical proportions - Axiomatics and algorithmic complexity issues

Henri Prade Gilles Richard

IRIT, CNRS & Université Paul Sabatier,
Toulouse, France

Contents

2 new “justifications” of
the Boolean model for analogical proportions

- Background on analogical proportions
- Postulates for analogical proportions and Boolean models
- An algorithmic complexity view of analogical proportions

Boolean analogical proportion *"a is to b as c is to d"*

"*a* differs from *b* as *c* differs from *d*, and vice-versa"

$$a : b :: c : d = (a \wedge \neg b \equiv c \wedge \neg d) \wedge (\neg a \wedge b \equiv \neg c \wedge d)$$

$$a : b :: c : d = (a \wedge d \equiv b \wedge c) \wedge (\neg a \wedge \neg d \equiv \neg b \wedge \neg c)$$

a	b	c	d	$a : b :: c : d$	a	b	c	d	$a : b :: c : d$
0	0	0	0	**1**	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	**1**
0	0	1	1	**1**	1	0	1	1	0
0	1	0	0	0	1	1	0	0	**1**
0	1	0	1	**1**	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	**1**

Equation and creativity

- x? *equation* $a : b :: c : x$
  $x = c \equiv (a \equiv b)$ iff $(a \equiv b) \vee (a \equiv c)$ holds
  But $0 : 1 :: 1 : x$ or $1 : 0 :: 0 : x$ have **no solution**:
  neither $0 : 1 :: 1 : 0$ nor $0 : 1 :: 1 : 1$,
  neither $1 : 0 :: 0 : 1$ nor $1 : 0 :: 0 : 0$ hold true
- $\overrightarrow{a} = (a_1, \cdots, a_n)$
  $\overrightarrow{a} : \overrightarrow{b} :: \overrightarrow{c} : \overrightarrow{d}$ iff $\forall i \in [1, n],\ a_i : b_i :: c_i : d_i$
- *creativity* $(a_1, a_2) = (1, 1)$
  $(b_1, b_2) = (1, 0)$
  $(c_1, c_2) = (0, 1)$
  — — — — — — — —

  we infer $(d_1, d_2) = (0, 0)$

Analogical proportions postulates

1. $\forall a, b, R(a, b, a, b)$ (reflexivity);
2. $\forall a, b, c, d, R(a, b, c, d) \rightarrow R(c, d, a, b)$ (*symmetry*)
3. $\forall a, b, c, d, R(a, b, c, d) \rightarrow R(a, c, b, d)$ (*central permutation*)

$\forall a, b, c, d, R(a, b, c, d) \rightarrow R(d, b, c, a)$ (*external permutation*)

Boolean model

It is straightforward to get a basic Boolean model
- by reflexivity, $0101, 1010$ should belong to the relation
- and $0000, 1111$ as well since letting $a = b$
- central permutation then leads to add 0011 and 1100

$\Rightarrow$ we get the **minimal** model

$$\Omega_0 = \{0000, 1111, 0101, 1010, 0011, 1100\}$$

which is stable under symmetry

Other models - 1

Due to axioms, we should add to Ω_0 subsets of $\mathbb{B}^4$ stable w.r.t. symmetry and central permutation

1) 1 model with 6 elements: Ω_0 (the smallest one)
2) 1 model with 8 elements: $Kl = \Omega_0 \cup S_2 =$ $\{0000, 1111, 0101, 1010, 0011, 1100, 0110, 1001\}$ first proposed by S. Klein (1982)
3) 2 model with 10 elements:

$M_3 = \Omega_0 \cup S_3 =$
$\{0000, 1111, 0101, 1010, 0011, 1100, 1110, 1101, 1011, 0111\}$

$M_4 = \Omega_0 \cup S_4 =$
$\{0000, 1111, 0101, 1010, 0011, 1100, 0001, 0010, 0100, 1000\}$

Other models - 2

4) 2 models with 12 elements:

$M_5 = M_3 \cup S_2 = \{0000, 1111, 0101, 1010, 0011, 1100, 1110, 1101, 1011, 0111, 0110, 1001\}$,

$M_6 = M_4 \cup S_2 = \{0000, 1111, 0101, 1010, 0011, 1100, 0001, 0010, 0100, 1000, 0110, 1001\}$,

5) 1 model with 14 elements:

$M_7 = M_3 \cup S_4 = M_4 \cup S_3 = \Omega_0 \cup S_3 \cup S_4 = \{0000, 1111, 0101, 1010, 0011, 1100, 1110, 1101, 1011, 0111, 0100, 1000, 0110, 1001\}$

6) 1 model with exactly 16 elements:

$\Omega = \Omega_0 \cup S_2 \cup S_3 \cup S_4 = \mathbb{B}$

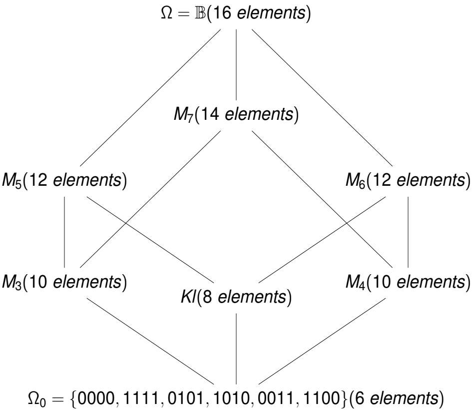

Figura: The lattice of Boolean models of analogy

A (very) short course on algorithmic complexity - 1

U a universal Turing machine; p a program running on U

- $U(p, x) = y$: p stops for input x and outputs a finite string y
- $K_U(y/x)$ is the size of the shortest program able to reconstruct y with the help of x

Kolmogorov complexity of y obtained with empty string ϵ

$$K_U(y) = \min\{|p|, U(p, \epsilon) = y\}$$

Given a string s, $K_U(s)$ is an integer which is a measure of the information content of s

A (very) short course on algorithmic complexity - 2

- complexity is independent of the underlying U
- $\forall x, y, K(xy) = K(x) + K(y/x) + \mathcal{O}(1)$
- K is a non computable function
- an upper bound of K can be estimated
  $|-log_2(m(s)) - K(s)| < c$ with
  $m(s) = \Sigma_{p:U(p)=s} 2^{-|p|}$
  OACC (http://www.complexitycalculator.com/)
  allows to estimate the complexity of *short* strings

Back to analogy

- Are 0000, 1111, 0101, 1010, 0011, 1100
  the 6 less complex strings with 4 elements ?
- **NO !**
  True only for the first 4
  but 4 other strings ranked before 0011, 1100 !
- but we did not try to take into account
  the very meaning of analogy ...

Let $k(x_1x_2x_3x_4)$
the following formula measuring, in some sense, the quality of an analogy:

$$|K(x_2/x_1) - K(x_4/x_3)| + |K(x_1x_2/x_3x_4) - K(x_3x/x_1x_2)| + K(x_1x_2x_3x_4)$$

This leads us to postulate that
the "best" x_4 we are looking for
to build a valid analogy $x_1 : x_2 :: x_3 : x_4$
is the one minimizing this expression:
So, we have: $x_4 = argmin_u k(x_1x_2x_3u)$

Using OACC http://www.complexitycalculator.com/

k(abcd) = |K(b/a) – K(d/c)| + |K(cd/ab) – K(ab/cd)| + K(abcd)

	A	B		C	D	
abcd	K(b/a)	K(d/c)	K(abcd)	k(cd/ab)	K(ab/cd)	|A-B|+|C-D| +K(abcd)
0000	1.8667131652	1.8667131652	11.2174683967	5.8033671621	5.8033671621	11.2174683967
1111	1.8667131652	1.8667131652	11.2174683967	5.8033671621	5.8033671621	11.2174683967
0101	1.8667159505	1.8667159505	11.7002793293	6.2861753096	6.2861753096	11.7002793293
1010	1.8667159505	1.8667159505	11.7002793293	6.2861753096	6.2861753096	11.7002793293
0001	1.8667131652	1.8667159505	11.5731249872	6.1590237527	6.3435937193	11.757697739
1110	1.8667131652	1.8667159505	11.5731249872	6.1590237527	6.3435937193	11.757697739
0111	1.8667159505	1.8667131652	11.5731249872	6.1590209675	6.3435965045	11.7577033094
1000	1.8667159505	1.8667131652	11.5731249872	6.1590209675	6.3435965045	11.7577033094
0011	1.8667131652	1.8667131652	11.8099819092	6.3958806747	6.3958806747	11.8099819092
1100	1.8667131652	1.8667131652	11.8099819092	6.3958806747	6.3958806747	11.8099819092
0100	1.8667159505	1.8667131652	11.757697739	6.3435937193	6.1590237527	11.9422704908
1011	1.8667159505	1.8667131652	11.757697739	6.3435937193	6.1590237527	11.9422704908
0010	1.8667131652	1.8667159505	11.757697739	6.3435965045	6.1590209675	11.9422760613
1101	1.8667131652	1.8667159505	11.757697739	6.3435965045	6.1590209675	11.9422760613
1001	1.8667159505	1.8667159505	12.0548412692	6.6407372495	6.6407372495	12.0548412692
0110	1.8667159505	1.8667159505	12.0548412692	6.6407372495	6.6407372495	12.0548412692

Concluding remark

- maybe in line with
  a Kolmogorov complexity view
  of analogical proportion-based inference
- analogical proportion-based inference
  makes **no error** on affine Boolean functions
  (projection, *x-or* functions) (IJCAI-2017)

$$\frac{\vec{a} : \vec{b} :: \vec{c} : \vec{d}}{cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : cl(\vec{d})}$$