Analogical Proportions: Why They Are Useful in AI

Henri Prade and Gilles Richard
IRIT, CNRS and Université Paul Sabatier, Toulouse, France
{henri.prade, gilles.richard}@irit.fr

Abstract

This paper presents a survey of researches in analogical reasoning whose building block are analogical proportions which are statements of the form “a is to b as c is to d”. They have been developed in the last twenty years within an Artificial Intelligence perspective. After discussing their formal modeling with the associated inference mechanism, the paper reports the main results obtained in various AI domains ranging from computational linguistics to classification, including image processing, I.Q. tests, case based reasoning, preference learning, and formal concepts analysis. The last section discusses some new theoretical concerns, and the potential of analogical proportions in other areas such as argumentation, transfer learning, and XAI.

1 Introduction

Exploiting analogies has been recognized for a long time as a useful approach to problem solving. For instance, Polya’s famous book How to solve it? [Polya, 1945] is a good advocacy in favor of an extensive search for analogies in order to discover solutions of mathematical problems. Even if analogical inference is brittle since it offers no guarantee on the validity of its derived conclusions, it has the merit of laying bare candidate answers or solutions that further investigations may confirm as being ultimately valid. Analogies, as a useful parallel between two situations enabling us to foresee that what is true in the first situation may be true as well in the second situation, has been extensively studied not only by philosophers and psychologists [Hesse, 1966; Gentner et al., 2001; Hofstadter and Mitchell, 1995; Hofstadter and Sander, 2013], but also by AI scientists [Boy de la Tour and Caferra, 1987; Veloso and Carbonell, 1993; Melis and Veloso, 1998]. This survey focuses on analogical proportions (APs for short in this survey), i.e., statements of the form “a is to b as c is to d” such as “cow is to calf as mare is to foal”, usually denoted a : b :: c : d. APs can be related to analogies between two situations if the pairs of items (a, b) and (c, d) refer respectively to these two situations and the items are put in correspondence [Hesse, 1959]. Starting in the late 1990’s there has been a new trend of research in analogical studies dealing with APs: this is what is reviewed here. Although the idea of analogy seems to primarily focus on similarity, APs deal as much with dissimilarity. APs take place as soon as we are comparing items described by multiple attributes. An AP states that the comparisons between a and b, and between c and d, in terms of similarity and dissimilarity, yield results that are very much alike. APs are quite pervasive; in particular, if a and d differ on at least two attributes, one can find (several) pairs (x, y) such that a : x :: y : d holds and the four items are distinct [Couceiro et al., 2018]. Analogical reasoning is often associated with the idea of creative thinking. APs are creative as well, since the equation a : b :: c : x between items described by multiple attributes, when it is solvable, has a solution d different from a, b, c in general. However, feature by feature, (c, x) will be a copy of (a, b), as we shall see. These properties and others presented in the next section, make APs quite useful in many tasks. The paper is organized into three main sections. Section 2 presents how APs can be modeled and used for inference task. Other proportions, strongly related to APs, are also discussed. Section 3 surveys domains where APs have been used with success. Section 4 highlights tracks worth to be investigated.

2 Modeling and Related Issues

The idea of AP between word items dates back to Aristotle [Aristotle, 2011] (at least), and could have been inspired by a parallel with (geometric) numerical proportions, namely \( \frac{a}{b} = \frac{c}{d} \); see [Prade and Richard, 2013] for details. This parallel explains the postulates for APs recalled below. Note that it fits as well with arithmetic proportions that state the equality of two differences (rather than two ratios) between numbers: \( a - b = c - d \) (can be got as a logarithmic transform of \( \frac{a}{b} = \frac{c}{d} \)).

2.1 Postulates

Given a set of items \( X \), APs is a quaternary relation supposed to obey the 3 following postulates (e.g. [Lepage, 2001]):

\( \forall a, b, c, d \in X, \)

1. \( a : b :: a : b \) (reflexivity);
2. \( a : b :: c : d \to c : d :: a : b \) (symmetry);
3. \( a : b :: c : d \to a : c :: b : d \) (central permutation).

These postulates have straightforward consequences like:

- \( a : a :: b : b \) (identity);
- \( a : b :: c : d \to b : a :: d : c \) (internal reversal);
• \( a : b : c : d \rightarrow d : b : c : a \) (an extreme permutation);
• \( a : b : c : d \rightarrow d : c : b : a \) (a complete reversal).

Among the 24 permutations of \( a, b, c, d \), the previous postulates induce 3 distinct classes each containing 8 distinct proportions regarded as equivalent due to postulates: \( a : b : c : d \) has in its class \( c : d :: a : b \); \( a : b : c : d \) has in its class \( a : d :: b : c \); and \( a : c : b : d \). But \( b : a : c : d \) and \( a : d : c : b \) do not belong to the class of \( a : b : c : d \) and are elements of the two other classes.

2.2 Models

There are three main approaches to the modeling of APs that we briefly present: logical, algebraic, and complexity-based.

Logical View

There exist two models of analogical reasoning respectively based on first order logic [Russell, 1989], and second order logic [Gust et al., 2006]. We do not discuss them further since they are not based on APs. An AP can be viewed as a quaternary propositional logic connective when \( X \) is the Boolean set \( B = \{0, 1\} \) [Miclet and Henri Prade, 2009]

\[
a : b : c : d = ((a \land \neg b) \equiv (c \land \neg d)) \land ((\neg a \land b) \equiv (c \land d))
\]

It makes explicit that “\( a \) differs from \( b \) as \( c \) differs from \( d \) (and vice-versa)”. It is easy to check that this formula is only valid for the 6 valuations in Table 1. This set of 6 valuations is

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: 6 Valid valuations for strong Boolean AP

the minimal (w.r.t. set inclusion) Boolean model [Prade and Richard, 2018] obeying the 3 postulates of analogy. \( a : b : c : d \) can be equivalently written in a way that reminds a well-known property of numerical proportions (\( a : d = b : c \) and \( a + d = b + c \)):

\[
a : b : c : d = (a \land d \equiv b \land c) \land (a \lor d \equiv b \lor c)
\]

Note that this writing emphasizes similarity, and not dissimilarity as the first one. It can be seen on Table 1 that 1 and 0 play a symmetrical role, which makes the definition code-independent. The Boolean modeling of APs can be related to a set-based view [Lepage, 2003; Miclet and Henri Prade, 2009]. Besides, it has its roots in an early proposal by S. Klein [Klein, 1983] who suggested to also include \((0, 1, 1, 0)\) and \((1, 0, 0, 1)\), which would validate the highly debatable consequence \( a : b : c : d \rightarrow b : a : c : d \). When \( a, b, c, d \) take their values in a finite set \( X \) such that \( |X| > 2 \) (nominal case where the values are discrete or categorical), APs can be taken as true for 3 patterns, namely \((s, s, s, s), (s, t, s, t), (s, s, t, t)\) for \( s, t \in X \), which clearly include as a particular case the 6 patterns of Table 1. This generalization still agrees with the postulates.

When \( X = [0, 1] \subset \mathbb{R} \), there are basically two options [Dubois et al., 2016] for extending the logical definition of \( a : b : c : d \) into multiple-valued logic expressions, where \( a : b : c : d \) becomes a matter of degree. They both generalize the Boolean and the nominal cases, and are code-independent (with respect to the negation \( \nu(a) = 1 - a \)). One is such that \( a : b : c : d = 1 \) only for \( x : y : x : y \) and \( x : x : y : y \) (with possibly \( x = y \)) where \( x, y \in [0, 1] \). The other is such that \( a : b : c : d = 1 \) if and only if \( a \neq b \neq c \neq d \). See [Dubois et al., 2016] for expressions and details. Thus one can define APs for Boolean and nominal features, as well as numerical features (once renormalized within \([0, 1]\)). Let us mention the special case of “continuous AP” of the form \( a : b : c : d \) and are elements of the two other classes.

To deal with items represented by vectors of feature values, AP definitions are extended componentwise from \( X \) to \( \mathbb{R}^n \):

\[
a : b : c : d \iff \forall i \in [1, n], a_i : b_i : c_i : d_i
\]

Using arithmetic proportion \( a - b = c - d \) as a model for \( \mathbb{R} \) [Rumelhart and Abrahamson, 1973] leads in \( \mathbb{R}^n \) to the well-known parallelogram view considering \( a, b, c, d \) as vectors in \( \mathbb{R}^n \). It agrees with Table 1, but then \( a - b \in \{-1, 0, 1\} \).

Algebraic View

Sprocca and Yvon [Yvon and Sprocca, 2006] have given another more general definition of AP, based on the notion of factorization, when the set of items is a commutative semi-group \( (X, \otimes) \).

\[(x, y, z, t) \in X^4 \text{ is an AP } (x : y : z : t) \text{ if:}
\]

1) either \( (y, z) \in \{(x, t), (t, x)\} \),
\[
2) \text{or } (x_1, x_2, t_1, t_2) \in X^4 \text{ such that } x = x_1 \oplus x_2, y = x_1 \oplus t_2, z = t_1 \oplus x_2 \text{ and } t = t_1 \oplus t_2.
\]

This definition satisfies the postulates of APs. The second part of the definition formally expresses that \( x \) and \( y \) (resp. \( z \) and \( t \)) have in common \( x_1 \) (resp. \( t_1 \)) while \( x \) and \( y \), as \( z \) and \( t \), differ from \( x_2 \) to \( t_2 \), such as in \( mpr::mps::qr:qs \). This definition, which coincides with the logical view on a Boolean lattice, allows us to define APs when \( X \) is equipped with various structures (e.g., [Miclet et al., 2014], or non-distributive lattices in relation with formal concept analysis [Barbot et al., 2019a]).

Complexity-Based View

A machine learning-oriented view where APs are interpreted in terms of Kolmogorov complexity has been presented in [Cornuéjols, 1996]; (see also [Murena et al., 2020]). The main idea is to measure the size of the minimal program \( pr(a, b) \) able to transform \( a \) into \( b \) (a and \( b \) supposed to be formal expressions), in other words, to consider Kolmogorov complexity as the standard for complexity definition (instead of Shannon entropy for instance). If the size of \( pr(a, b) \) and the size of \( pr(c, d) \) are approximately equal, then \( a : b : c : d \) holds, still expressing that \( a \) and \( b \) differ as \( c \) and \( d \) differ [Bayoudh et al., 2012]. Roughly speaking, it amounts to consider the problem of solving an AP equation \( a : b : c : x \) from an inductive learning perspective: knowing \( a, b, c, d \), find \( x \) minimizing the complexity of \( a : b : c : d \) considered as a formula. The truth table of Boolean APs has been also analyzed in terms of Kolmogorov complexity, showing that the complexity of the patterns that make an AP true is smaller than the complexity of any other pattern [Prade and Richard, 2018].
Other Views
From a functional viewpoint, \( x : f(x) :: y : f(y) \) looks like a prototypical AP, meaning that \( x \) is to \( f(x) \) as \( y \) is to \( f(y) \), just applying the same function \( f \) for obtaining \( f(x) \) and \( f(y) \) from \( x \) and \( y \) respectively. This underlies the approach developed in COPYCAT [Hofstadter and Mitchell, 1995] for completing \( a, b, c \) with a plausible \( d \). Note that this view does not fit with the standard arithmetic proportion (e.g., \( a : a^2 :: b : b^2 \), but \( a - a^2 \neq b - b^2 \)). Due to central permutation, one is led to postulate that we have \( x : y :: f(x) : f(y) \) and then it also exists \( g \) such that \( x : g(x) :: f(x) : f(g(x)) \) where \( f \) and \( g \) commute; see [Barbot et al., 2019b] for a detailed discussion.

The relation-based view, generalizing the functional view, is as simple as: \( a : b :: c : d \) holds if there exists a relation \( R \subseteq X \times X \) such that \( R(a, b) = R(c, d) \). It raises a number of questions, including a risk of triviality if \( R \) is too general, and the problem of accommodating (or not) central permutation. See [Lim et al., 2021] for a discussion.

2.3 Inference
This inference principle, first proposed in [Pirrelli and Yvon, 1999] for nominal values, can be stated as follows:

\[
\forall i \in \{1, \ldots, n\}, \quad a_i : b_i :: c_i : d_i \text{ holds if } \exists j \in \{n + 1, \ldots, m\}, \quad a_j : b_j :: c_j : d_j \text{ holds}
\]

As can be seen, we transfer knowledge from some components of source vectors to their remaining components, implicitly assuming that the values of the \( n \) first components determine the values of the others. When \( n + 1 = m \), this inference pattern can be used to predict, e.g., the class of \( d \), say \( cl(d) \), from \( cl(a), cl(b), cl(c) \), after checking that \( a : b :: c : d \) holds componentwise. This amounts to assume that the equation \( cl(a) :: cl(b) :: cl(c) :: x \) has a solution. But it is not always the case (e.g., if \( 0 : 1 :: 1 : 1 \)). It is easy to see that the AP equation \( a : b :: c : x \) is solvable if and only if \( a = b \) or \( a = c \) in the nominal case. Then, the unique solution is given by \( x = c \) if \( a = b \) and \( x = b \) if \( a = c \). The inference pattern can be generalized when \( a, b, c, d \) involve numerical features [Bouhas et al., 2017a]. It is the basis of the AP solving process: finding \( x \) such that \( a : b :: c : x \). In [Bouhas et al., 2017a], it is shown that the above inference pattern can be derived from a basic analogical pattern that derives \( Q(y) \) from \( P(x), P(y) \) and \( Q(x) \) (e.g., [Russell, 1989]).

2.4 Related Proportions
A logical proportion\(^1\) \( T(a, b, c, d) \) [Prade and Richard, 2013] is the conjunction of two logical equivalences, each one between indicators for \( (a, b) \) and indicators for \( (c, d) \). By indicators, we mean operators such that \( \land \neg b, \neg a \land \neg b \) expressing dissimilarity, or such that \( a \land b, \neg a \land \neg b \) expressing positive and negative similarity respectively, and the like with \( c \) and \( d \). It has been established that there are 120 syntactically and semantically distinct logical proportions. Each of them are only true for 6 patterns (and false for the 10 = \( 2^4 - 6 \) others). Among them, 8 are particularly noticeable as the only ones to be code-independent (i.e., \( T(a, b, c, d) = T(\neg a, \neg b, \neg c, \neg d) \)).

These 8 proportions split into 4 homogeneous proportions that are symmetrical (one can exchange \( (a, b) \) with \( (c, d) \)) and 4 heterogeneous ones that are not symmetrical. Homogeneity here refers to the fact that in the expression of the proportions, both equivalences link indicators of the same kind (similarity or dissimilarity), while in the case of heterogeneous proportions they link indicators of opposite kinds. Moreover, the homogeneous proportions are true only for 6 patterns involving even numbers of 0 and 1, while the 4 heterogeneous proportions are true for 6 patterns of the form \((s, t, t, t), (t, s, t, t), (t, t, s, t), (t, t, t, s) \) with \( s, t \in \{0, 1\} \). The four heterogeneous proportions express that there is an intruder among \( (a, b, c, d) \), which is not \( a \), which is not \( b \), which is not \( c \), and which is not \( d \), respectively [Prade and Richard, 2014].

Among the 4 code independent homogeneous proportions, 3 of them are directly related to the AP \( a : b :: c : d \). Apart AP itself, they are named “reverse analogy” (Rev) and “paralogy” (Par) and are directly related to AP through a permutation, namely \( Rev(a, b, c, d) = b : a :: c : d \) and \( Par(a, b, c, d) = c : b :: a : d \). Interestingly enough, paralogy can be written \( Par(a, b, c, d) = ((a \land b) \equiv (c \land d)) \land ((\neg a \land \neg b) \equiv (\neg c \land \neg d)) \), which expresses that \( a \) and \( b \) have in common (positively or negatively), \( c \) and \( d \) have it also, and conversely. The existence of these three related proportions mirrors the fact that given 3 non aligned points in \( \mathbb{R}^2 \), there are 3 ways to build a parallelogram from them [Prade and Richard, 2013]. The fourth homogeneous proportion is called “inverse paralogy” (Inv) and defined by \( Inv(a, b, c, d) = ((a \land b) \equiv (\neg c \land \neg d)) \land ((\neg a \land \neg b) \equiv (c \land d)) \), which expresses a kind of “orthogonality” between the pairs \( (a, b) \) and \( (c, d) \): “what \( a \) and \( b \) have in common, \( c \) and \( d \) do not have it and conversely”. Remarkably enough \( Inv \) is the unique proportion among the 120 that is stable under any permutation of the 4 terms [Prade and Richard, 2013].

3 Some Achievements
APs (as well as some other related proportions) have been used with success in diverse fields of AI that we survey now.

3.1 Computational Linguistics
Natural language processing is a field of choice to investigate the use of APs such as the famous “man:king::woman:queen” example. The pioneering work of [Rumelhart and Abrahamson, 1973] leads the way on this track: it is all about solving APs on concepts represented as vectors. Solving analogies on words, i.e., given 3 words \( a, b, c \), looking for a word \( d \) such that the proportion \( a : b :: c : d \) holds can be tackled from diverse angles. For instance, if \( a \) and \( b \) belong to a language \( L_1 \), \( c \) belonging to another language \( L_2 \), it allows to automatically translate the word \( b \) into its counterpart in \( L_2 \). This is done for instance in [Lepage, 1998; Lepage and Denoual, 2005]. More recently, this has also been developed in [Bayoudh et al., 2012; Murena et al., 2020] with a constraint of minimal complexity (in the sense of Kolmogorov complexity).

\(^1\)Piaget [Piaget, 1953] uses this name for logical expressions of the same form as the second Boolean formula we gave for \( a : b :: c : d \).
From a different perspective, in the past 10 years, the assumption that words that appear in “similar” contexts tend to have “similar” meanings has been the foundation of all the successful embedding techniques such as word2vec [Mikolov et al., 2013], GloVe ([Pennington et al., 2014], etc. Turning words \( a, b, c, d \) into vectors \( a', b', c', d' \) and considering the parallelogram perspective telling that, for the word analogy \( a : b :: c : d \) to hold, \( (a', b', c', d') \) should constitute a parallelogram in the embedding space\(^2\), has led to powerful mechanisms to achieve diverse tasks such as:

- Solving the analogical equation \( a : b :: c : x \) where \( x \) is an unknown word. It is amazing to note that solving word analogies has become one of the most popular benchmarks for quality of word embeddings [Linzen, 2016; Allen and Hospedales, 2019]. The arithmetic view \( (a - b = c - d) \) applied to words represented as vectors, is often not sufficient to find an \( x \) referring to (or close to) the existing word. More sophisticated formulas have been proposed to better match an existing word ([Levy and Goldberg, 2014; Drozd et al., 2016]). The problem can also be considered as a regression task in a vector space as in [Lim et al., 2021].

- Focusing on what is known as relation induction [Bourauil et al., 2018; Lu et al., 2019], i.e., seeking to classify pairs of words \((a, b)\). With the implicit assumption that if 2 pairs of words \((a, b)\) and \((c, d)\) are in the same class, then \( a : b :: c : d \) should constitute an AP.

- Recently, in [Diallo et al., 2019; Zhu and de Melo, 2020], AP between words have been extended to sentences. Combining classical word embeddings to get “sentence embeddings”, the authors are able to solve APs with practical application to chatbots for instance. The underlying model still is the parallelogram.

### 3.2 Image Processing

Analogy between images has long been applicable to a wide variety of image related problems, including image filtering, texture synthesis, texture transfer, artistic filters, etc. [Hertzmann et al., 2001]. Recently, image processing has largely benefited from the emergence of deep neural networks, involving subtle operators like convolution, pooling, dropout, etc. One of the most important tools is the concept of distance, leading to a lot of works in what is known as distance metric learning (see [Law et al., 2017] for a more extensive description). When considering images, the concept of AP still makes sense as can be seen in the work of [Hwang et al., 2013; Law et al., 2017]. For instance, when talking about pictures, \( \text{pic}(\text{collie}) : \text{pic}(\text{dalmation}) :: \text{pic}(\text{lion}) : \text{pic}(\text{leopard}) \) (with informal notations) is visually acceptable. [Hwang et al., 2013] use the concept of Analogy-preserving Semantic Embedding (ASE) to classify pictures. The analogical view is still the parallelogram: the authors learn an embedding \( u \) of pictures \( a, b, c, d \) into a vector space where analogy is preserved thanks to a specific regularizer \( R \) enforcing the proportion. \( R \) (to be minimized) is just a weighted sum of \( ||u(a) - u(b)|| + ||u(c) - u(d)|| \) and \( ||u(a) - u(c)|| + ||u(b) - u(d)|| \) (involving \( L^2 \) norms), taking into account the central permutation postulate. With a distance learning objective, [Law et al., 2017] exploit analogical constraints over quadruplets of images such as “image \( a \) is closer to image \( b \) than image \( c \) is to image \( d \)”, formalized as \( \text{dist}(a, b) \leq \text{dist}(c, d) \) where \( \text{dist} \) has to be learned, which is the basis of their \( Q \)wise method. This has suggested the development in [Prade and Richard, 2018] of a logical framework for analogical inequalities for having an oriented modeling of such constraints (not captured by a distance since \( \text{dist}(a, b) = \text{dist}(b, a) \)).

### 3.3 I.Q. Tests

Consider a completion test such as the example of Figure 2 where pictures \( a, b, c, d \) are given, and \( d \) has to be found (in the usual tests, among a set of potential solutions). The problem can be encoded using the 5 Boolean predicates \( \text{hasSquare}(hS), \text{hasBlackDot}(hBD), \text{hasTriangle}(hT), \text{hasCircle}(hC), \text{hasEllipse}(hE) \) (taken in that order).

This leads to the code of Figure 3. Applying componentwise the AP solving process (described in subsection 2.3), we get \( \vec{x} = (0, 1, 1, 1, 0) \), i.e., the code of the expected solution. The approach is constructive since the missing picture \( \vec{x} \) is obtained by computation from \( \vec{a}, \vec{b}, \vec{c} \). This contrasts with classical AI approaches to this problem, pioneered by Thomas Evans [Evans, 1968], where \( d \) has to be chosen among a set of candidate pictures containing the right answer: the change between \( \vec{a} \) and \( \vec{b} \) is compared with the change between \( \vec{c} \) and \( \vec{x} \) for each potential \( \vec{x} \), leading to choose \( \vec{x} \) as the one maximizing similarity between the changes. In the above example, the solution could also be obtained by applying the AP solving method.
Still related to machine learning in the large, [Miclet et al., 2008; Couceiro et al., 2017] have shown that APs could also be used for data extension, which is a well known process for task such as image classification or segmentation. The power and theoretical limitations of analogy-based methods have been precisely described in [Couceiro et al., 2017; Couceiro et al., 2018], showing that APs-based inference never fails in case the Boolean function underlying the classification is linear; it has been extended to the case of nominal attributes [Couceiro et al., 2020].

As shown in [Bounhas et al., 2017b; Bounhas et al., 2018], heterogeneous logical proportions can also be used for evaluating how much an item is at odd with a class, and are thus a tool of interest in classification.

A main issue for analogy-based classification is that it relies on a cubic algorithm (due to the search of appropriate triplets \((a, b, c)\) for classifying a \(d\), which is not satisfactory in a world where a standard training set counts more than 100000 items. Nevertheless, such a cubic procedure can be much improved by i) restricting \(c\) to be a neighbor of \(d\), and ii) looking for a limited number of pairs \((a, b)\) that are particularly “competent” as shown in [Lieber et al., 2019] (the search for these pairs can be done in a pre-processing step; thus there is no need to walk through the whole domain to get accurate classification). Let us also mention the use of APs for completing missing values [Correa Beltran et al., 2014].

### 3.5 Case-Based Reasoning

Case based reasoning is similarity-based. Given a repertory of experienced cases describing problems \(p_i\)’s with solutions \(s_i\)’s, and facing a new problem \(p^*\), it amounts to looking for (a) similar problem(s) in the repertory and to adapting its/their solution(s) to this new problem. It already involves APs since one can say that the new solution is to the new problem \(p^*\) as the solution \(s_i\) is to problem \(p_i\) (when \(p^*\) is similar to \(p_i\)). But one cannot apply the AP-based inference pattern here because problems and solutions are not described with the same features. But, one can say that “\((p_a, s_a)\) is to \((p_b, s_b)\) as \((p_c, s_c)\) is to \((p_d, s_d)\)”. Thus from 3 cases, one can extrapolate an unknown \(s_d\), using the AP-based inference, provided that \(p_a : p_b :: p_c : p_d\) holds and that \(s_a : s_b :: s_c : x\) is solvable. Note that this extrapolation incorporates the adaptation step of case-based reasoning for producing \(s_d\) [Lieber et al., 2018].

### 3.6 Preference Learning

The first implemented approach to preference learning based on AP inference to preference learning is due to Fürnkranz and Hüllermeier [Fahandar and Hüllermeier, 2018], and focuses on learning to rank user preferences based on the evaluation of a loss function. It is based on an inference pattern stating that since \(a\) differs from \(b\) as \(c\) differs from \(d\) (and vice-versa), and \(b\) is preferred to \(a\), \(d\) should be preferred to \(c\) as well. In [Bounhas et al., 2019], the focus is on predicting preferences (rather than getting a ranking) using the previous inference pattern, or a more sophisticated one that states that if \(a : b :: c : d\) and \(a' : b' :: c' : d'\) hold and \(a\) is preferred to \(a'\), \(b\) is preferred to \(b'\), \(c\) is preferred to \(c'\), then \(d\)
should be preferred to \( d' \). A comparison of the these different approaches can be found in [Bounhas et al., 2019].

3.7 Formal Concept Analysis

With analogical reasoning, categorization is a noticeable activity of the mind. This suggests a possible bridge between APs and formal concept analysis (FCA). FCA relies on a so-called formal context, which is a relation \( R \) linking items with (Boolean) properties, from which one can define a formal concept as a pair made of a subset of items \( E \) and a subset of properties \( f \) that characterize them (where \( E \times f \) is maximal in \( R \)). Formal concepts can be organized into a lattice.

As shown in [Barbot et al., 2019a], APs can be defined between formal concepts (in the sense of formal concept analysis) and algorithms to compute them are proposed, exploiting special subcontexts, called analogical complexes. Moreover, APs between concepts can be related to another form of analogy, called “relational proportion”, which involves two universes of discourse, such that “Carlsen is to chess as Mozart is to music”, or more compactly “Carlsen is the Mozart of chess”, which is not anymore a relation between four items of the same kind, but can be extracted from a formal context. See [Barbot et al., 2019b] for a link between dichotomous trees, FCA and analogical complexes.

4 Toward Future Developments

The usages of APs as a first class concept or as a side tool for AI tasks have to be better understood in order to improve their effectiveness. In the next subsections, we provide a list of tracks that are worth exploring.

4.1 New Theoretical Concerns

Still many theoretical issues are open. Here are two examples. The central permutation postulate is sometimes debatable (e.g., [Barbot et al., 2019a]). It is legitimate to abandon it and to replace it by the weaker internal reversal property \((a : b :: c : d \rightarrow b : a :: d : c)\), which still entails complete reversal assuming symmetry. The minimal Boolean model of an AP \( a : b :: w : c : d \) obeying only reflexivity, symmetry and internal reversal postulates is given by \( a : b :: w : c : d = (a \equiv c) \land (b \equiv d) \) (which gets rid of the two patterns \((0, 0, 1, 1)\) and \((1, 1, 0, 0)\) in Table 1). Another theoretical issue has its root in an old treatise by French mathematician Gaspard Monge [Monge, 1793]. In this book, relations between strengths and lengths are described in terms of APs and generalized APs such as \( a : b :: c : d :: e : f \). Such double APs deserves attention in further research (see [Prade and Richard, 2021]).

4.2 Argumentation

According to [Bartha, 2009], an analogical argument is an explicit representation of analogical reasoning that cites accepted similarities between two systems in support of the conclusion that some further similarity exists. There are still few works on analogical argumentation [Amgoud, 2020]. Besides, a sequence of analogical arguments may lead to consider APs. Indeed in a debate, a discussant \( d \) may state that situation \( S_2 \) is like situation \( S_1 \) and that what took place in \( S_1 \) will happen in \( S_2 \) as well. The opponent discussant \( d' \) may argue that there is an (important) feature where they differ, and that what took place in \( S_1 \) may not happen in \( S_2 \). Then \( d \) may produce another pair of situations \((S_1', S_1')\), where the same difference can be observed without affecting the conclusion advocated by \( d \) for \( S_2 \). Then \( d' \) may counter-argue if he knows another pair of situations \((S_2', S_2')\) where the same difference does lead to a different conclusion. Thus this kind of exchange can be analyzed in terms of APs.

4.3 Case-Based Decision

Case-based decision [Gilboa and Schmeidler, 1995] is based on similarity only. AP-based inference may be of interest for taking advantage of both similarity and dissimilarity between situations. Let us just illustrate the basic idea using a simple scenario; see [Billingsley et al., 2017] for details and various scenarios. Suppose that a decision \( \delta \) was experienced in two different situations \( sit_1 \) and \( sit_2 \) in the presence or not of some “special circumstances”, leading to “good” or “bad” results respectively depending on the absence or on the presence of these special circumstances. Suppose we have in our repository two cases pertaining to \( sit_1 \) with and without the presence of special circumstances, and a case pertaining to \( sit_2 \) with special circumstances, then using AP-based inference, one can conclude if decision \( \delta \) in \( sit_2 \) without the special circumstances may be good or not.

4.4 XAI

With the emergence of sophisticated AI/ML techniques, the need for human understandable explanations of the output has become a major research theme known as eXplainable AI (XAI). In [H"ullermeier, 2020], the idea of analogy-based explanations is developed. Because most machine learning methods do obey the regularity assumption underlying similarity-based inference, similarity-based explanations are very natural. Not only may a meaningful measure of similarity require complex individual aggregations, but it can also happen that, for a given item, we cannot find “similar” items among the available data. This is where an analogy-based approach becomes relevant because the notion of similarity at item level is not involved. With an AP \( a : b :: c : d \), we do not have to compare \( a \) to \( b \) or \( c \) and \( d \); we just have to estimate the relation between \( a \) and \( b \) which has to be the same as the relation between \( c \) and \( d \). As such, APs can be used in a model-agnostic way, without taking into account the very details of the underlying ML model. This is work in progress. In [Hug et al., 2019], APs are still used to suggest explanations in recommender systems (see also [Sakaguchi et al., 2011]). Performances of recommender systems, largely based on matrix factorization, are generally made via diverse metrics such as precision, recall, coverage, surprise, etc. Whatever their performances w.r.t. these metrics, they are still considered as “black-boxes”, suggesting a movie without explaining why (Netflix for instance). Mining APs in incomplete database-known dataset for recommender systems) allows to find analogies like “Star Wars (1977) is to Raiders of the Lost Ark (1981) as Return of the Jedi (1983) is to Indiana Jones and the Last Crusade (1989)”, then to bring an explanation to “Last Crusade (1989)” recommendation.
4.5 Transfer Learning

The idea of transfer learning, which may be viewed as a kind of analogical reasoning performed at the meta level, is to take advantage of what has been learnt on a source domain in order to improve the learning process in a target domain. Re-using knowledge across different learning tasks (involving different domains), also known as multi-tasks learning, has long been addressed in the machine learning literature [Caruana, 1998; Daumé III and Marcu, 2006; Dai et al., 2009], generally assuming that the tasks are related at a low level (sharing the same feature space, or the same parametric family of models for instance). From a practical viewpoint, in the context of deep learning, transfer learning is usually implemented by retraining an existing neural network on new data to only update the weights corresponding to the last layers. In [Wang and Yang, 2011] the problem of transfer learning is addressed using structural analogy [Gentner, 1989] between two domains with completely different low-level representations. By making use of sophisticated statistical tools, the authors are able to estimate high level dependencies across domains, then to successfully validate the approach on a large number of transfer learning scenarios.

5 Conclusion

Drawing on past experiences to effectively solve current problems is a fundamental insight of human intelligence. Using APs as a formal model linking past pairs (problem, solution) to new observations is a natural way to describe this learning paradigm. As we have seen, APs can be used in a large variety of contexts, from computational linguistics to image processing, via multi-tasks learning. APs are not only a matter of similarity, but also involved dissimilarity. In a prediction perspective, APs bear at least a superficial link with conformal predictors where non-conformity is treated as a first class citizen for ML purposes [Bounhas et al., 2017b]. Ultimately, APs could help to implement a model of human high-level perception leading to the flexible build-up of representations appropriate to a given context, paving the way for creativity in AI.

References


