Predicting preferences by means of analogical proportions

Myriam Bounhas, Marc Pirlot, Henri Prade

Predicting Preferences by Means of Analogical Proportions

Myriam Bounhas[1,2(✉)], Marc Pirlot[3], and Henri Prade[4,5]

[1] LARODEC Laboratory, ISG de Tunis, 41 rue de la Liberté, 2000 Le Bardo, Tunisia
myriam_bounhas@yahoo.fr
[2] Emirates College of Technology, P.O. Box: 41009,
Abu Dhabi, United Arab Emirates
[3] Faculté Polytechnique, Université de Mons, Mons, Belgium
marc.pirlot@umons.ac.be
[4] IRIT – CNRS, 118, route de Narbonne, Toulouse, France
prade@irit.fr
[5] QCIS, University of Technology, Sydney, Australia

Abstract. It is assumed that preferences between two items, described in terms of criteria values belonging to a finite scale, are known for a limited number of pairs of items, which constitutes a case base. The problem is then to predict the preference between the items of a new pair. A new approach based on analogical proportions is presented. Analogical proportions are statements of the form "a is to b as c is to d". If the change between item-1 and item-2 is the same as the change between item-3 and item-4, and a similar statement holds for item'-1, item'-2, item'-3, item'-4, then one may plausibly assume that the preference between item-1 and item'-1 is to the preference between item-2 and item'-2 as the preference between item-3 and item'-3 is to the preference between item-4 and item'-4. This offers a basis for a plausible prediction of the fourth preference if the three others are known. This approach fits well with the postulates underlying weighted averages. Two algorithms are proposed that look for triples of preferences appropriate for a prediction. The first one only exploits the given set of examples. The second one completes this set with new preferences deducible from this set under a monotony assumption. This completion is limited to the generation of preferences that are useful for the requested prediction. The predicted preferences should fit with the assumption that known preferences agree with a unique unknown weighted average. The reported experiments suggest the effectiveness of the proposed approach.

1 Introduction

Analogical reasoning is reputed to be a valuable heuristic means for extrapolating plausible conclusions on the basis of comparisons. A simple form of this idea is implemented by case-based reasoning (CBR) [1], where conclusions known for stored cases are tentatively associated to similar cases. A more sophisticated option relies on the idea of analogical proportions. By analogical proportions,

https://doi.org/10.1007/978-3-030-01081-2_34

we mean statements of the form "a is to b as c is to d". We assume here that, a, b, c, d refer to four items described by their vector of features values for a considered set of features. The analogical proportion is to be understood as "a differs from b as c differs from d" and conversely "b differs from a as d differs from c". Then inference is based on triples of vectors (rather than taking vectors one by one as in case-based reasoning). The underlying idea is that if four items a, b, c, d are making an analogical proportion on describing features, an analogical proportion may hold as well on another mark pertaining to them, and then if this mark is known for a, b and c, one may compute it for d in such a way that the marks make an analogical proportion.

The notion of analogical proportions and their formalization has raised a trend of interest in the last two decades [14,16,18,20,21]. Moreover analogical proportion-based classifiers have been designed and experienced with success [3,4,15], first for Boolean and then for nominal and numerical attributes. In this case, the predicted mark is the label of the class. Although it is not intuitively obvious why analogical proportion-based inference may work well, one may notice that such a proportion enforces a parallel between four situations in such a way that the change between a and b is the same as the change between c and d. So this inference exploits co-variations.

One may wonder if what is working in classification may also be applied to preference prediction. The aim of this paper is to check whether analogical proportions may be a suitable tool for predicting preferences. This idea has been recently advocated in [17], but without providing any experimental evidence that it may work. This paper further investigates this idea and provides an experimentation of this idea. The problem considered is no longer to predict a class for a new item, but a preference relation between two items on the basis of a set of examples made of known comparisons applying to pairs of items. This set of examples plays the role of a case base, where a case is just a pair of vectors describing the two items together with information saying what item is preferred. Cases stating preferences are not usual in CBR literature, although they are clearly cases of interest. Still this should not be confused with "preference-based CBR" [2] where preferences between competing solutions in the context of a problem are handled.[1]

Preference learning has become a popular artificial intelligence topic [7,9,10,12]. Preference learning often relies on the assumption that data sets are massively available. Interestingly enough, analogical proportion-based inference may work with a rather small amount of examples, as we shall see. Preference-learning approaches often rely on the hypothesis that known preferences agree with a

[1] During the time we were finalizing this paper, we become aware of a very recent work [8], also aiming at predicting preferences on an analogical basis. Their approach exploits what is called "the horizontal reading" in [17], while here we investigate "the vertical reading" (also introduced in [17]). Moreover the focus of [8] is on learning to rank evaluated with a loss function, which is slightly different from the one here on predicting preferences and computing the error rate of predictions. A detailed comparison of the relative merits of the two approaches are beyond the scope of this paper, but will be the topic of a forthcoming study.

unique unknown aggregation function or with a conditional preference structure that has to be identified. Analogical proportion-based methods extrapolate predictions from known cases without looking for some underlying explanation model.

The paper is structured as follows. Section 2 recalls the necessary background on analogical proportions and their use in plausible inference mechanisms, while Sect. 3 restates important facts in multiple criteria decision that are relevant for applying analogical proportion-based inference to preference prediction. Section 4 presents two algorithms, one exploiting a set of given examples, one using an extension of this set relying on a monotony assumption of the preferences. Section 5 shows that promising results can be obtained, while Sect. 6 provides some lines for further research.

2 Background on Analogical Proportions

As already said, an analogical proportion is a statement of the form "a is to b as c is to d". It is denoted $a : b :: c : d$ usually. The name comes from a parallel, already made in Ancient Greek time, with geometric proportions that equate the ratio of two numbers a and b with the ratio of c and d, which may appear as a particular case of an analogical proportion. This is maybe why analogical proportions are supposed to obey the three following postulates (which correspond to properties of geometric proportions):

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

Let us assume that a, b, c and d take their values in a finite set U with at least two elements. Let $u, v \in U$. It follows immediately that (u, u, u, u), (u, u, v, v), and (u, v, u, v) are valid schemas for the analogical proportion. Other possible schemas with two distinct values are on the one hand (u, u, u, v), (u, u, v, u), (u, v, u, u), and (v, u, u, u), and on the other hand (u, v, v, u). They all disagree with the idea of proportion: indeed, e.g., it would be weird to claim that "u is to u as u is to v" if $v \neq u$, while "u is to v as v is to u" looks debatable as well, since the change from v to u is opposite to the one from u to v.

So the simplest model for analogical proportions, is to consider that the quaternary relation in U^4 is valid if and only if $a : b :: c : d$ is of the form (u, u, u, u), (u, u, v, v), or (u, v, u, v). In the case where U would be replaced by the set of real numbers, a more liberal model may be considered where $a : b :: c : d$ holds as soon as $a - b = c - d$ [19], and where more than two distinct values may appear in the 4-tuples that make an analogical proportion, e.g., $0 : 0.5 :: 0.5 : 1$, or $0.2 : 0.4 :: 0.6 : 0.8$. However, we shall not use this modeling in the following since we use a finite set U, encoded by numbers, but where the difference operation '$-$' is not a closed operation.

The above definition of an analogical proportion extends to vectors straightforwardly. Let $a = (a_1, \cdots, a_n)$, $b = (b_1, \cdots, b_n)$, $c = (c_1, \cdots, c_n)$, $d = (d_1, \cdots, d_n)$, then

$$a : b :: c : d \text{ if and only if } \forall i, a_i : b_i :: c_i : d_i, i = 1, n.$$

Then analogical proportion-based inference [4,20] is usually defined by the following pattern of plausible inference

$$\frac{\forall i \in [[1, n]], a_i : b_i :: c_i : d_i}{m(a) : m(b) :: m(c) : m(d)}$$

where $m(x)$ denotes a mark associated with vector x, e.g., the label of the class of x, in a classification problem. More generally, $m(x)$ may be also a vector. Here m represents the sign of the preference relation (the value of m is in $\{\preceq, \succeq\}$). Then if $m(a)$, $m(b)$, $m(c)$ are already known, while $m(d)$ is unknown, we can infer a plausible value for $m(d)$ by solving the equation $m(a) : m(b) :: m(c) : ?$ whose solution is always unique, when it exists. However, note that $u : v : v : ?$ has no solution, since neither $u : v : v : v$ nor $u : v : v : u$ are valid analogical proportions.

In the following, the items we consider are made of preferences between two vectors of criteria values, of the form $a^1 \preceq a^2$, $b^1 \preceq b^2$, $c^1 \preceq c^2$ and $d^1 \preceq d^2$. Then an instance of the analogical proportion-based preference inference is now

$$\begin{array}{l} \forall j \in [[1, n]], a_j^1 : b_j^1 :: c_j^1 : d_j^1 \text{ and } a_j^2 : b_j^2 :: c_j^2 : d_j^2 \\ \qquad\qquad a^1 \preceq a^2, b^1 \preceq b^2, c^1 \preceq c^2 \\ - \\ \qquad\qquad d^1 \preceq d^2. \end{array}$$

Two other instances, which involve the reversed preference relation $\succeq$, and which are in agreement with the valid patterns of analogical proportions, would correspond to following the analogical entailments *with the same first premise as above* that we do not repeat:

$$\begin{array}{l} a^1 \preceq a^2, b^1 \preceq b^2, c^1 \succeq c^2 \\ - - - - - - - - - - - - \\ d^1 \succeq d^2. \end{array}$$

and

$$\begin{array}{l} a^1 \preceq a^2, b^1 \succeq b^2, c^1 \preceq c^2 \\ - - - - - - - - - - - - \\ d^1 \succeq d^2. \end{array}$$

As we are going to see now, other concerns should be taken into account for a proper preference prediction mechanism. For that we need to first recall some results in multiple criteria analysis, and to make some observations.

3 Multiple Criteria-Based Preference and Analogical Proportions

In practice, the most largely used multiple criteria aggregation operators are weighted sums. Let $x = (x_1, \ldots, x_n)$ be a vector of evaluations representing a choice according to n criteria. The same finite scale $S = \{1, 2, ..., k\}$ is used for all criteria (the greater the value, the better it is). As pointed out in [17], an important property satisfied by many aggregation operators, in particular weighted sums, is that *contradictory tradeoffs are forbidden* (in fact, not showing contradictory tradeoffs is a property shared by many preference models, not only by weighted sums). This is expressed by the following postulate.

We *cannot have together* the four following preference statements: $\forall i, j$

$$\begin{array}{ll} & A : x_{-i}\alpha \preceq y_{-i}\beta \\ \text{and} & B : x_{-i}\gamma \succeq y_{-i}\delta \\ \text{and} & C : v_{-j}\alpha \succeq w_{-j}\beta \\ \text{and} & D : v_{-j}\gamma \prec w_{-j}\delta \end{array}$$

So the first three statements should entail

$$D' : v_{-j}\gamma \succeq w_{-j}\delta$$

where $\succeq$ denotes a preference relation ($x \succeq y$ is the same as $y \preceq x$) and x_{-i} denotes the n-1-dimensional vector made of the evaluations of x on all criteria except the i^{th} one for which the Greek letter denotes the substituted value. This postulate ensures that the difference between γ and δ is at least as large as that between α and β, independently of the criterion on which this difference shows up. In other words, in context x_{-i} the values of α, β are not enough for reversing the preference, while γ, δ are sufficient; in context v_{-j} the values of α, β are now sufficient for getting the preference reversed, then it should be a fortiori the case with γ, δ in this new context.

This postulate is verified by preferences that can be represented by a weighted sum of utilities, i.e., in case there exist a real-valued function u defined on S and a set of n weights p_i summing up to 1, such that, $\forall x, y$, $x \succeq y$ if and only if

$$U(x) = \sum_{i=1}^{n} p_i u(x_i) \geq U(y) = \sum_{i=1}^{n} p_i u(y_i).$$

where $U(x)$ is the global utility of the choice associated with vector x.

The above pattern of inference is compatible with the analogical proportion-based patterns of the previous section. In fact, a violation of this pattern would lead to observe a *reversed* analogical proportion (of the form a is to b as b is to a) on the preference symbols, which is opposite to what analogical proportion expresses [18].

Besides, the problem considered in this paper is the following. Given a set E of preferences of the form $x^k \succeq y^k$ $(k = 1, \cdots, m)$, representing what we know about the preferences of an agent about some pairs of choices, can we predict its preference between two other choices x and y? First, an idea is to make the assumption that the preferences of the agent obey a weighted sum aggregation scheme, whose weights are unknown. Then, we might think of finding a sampling of systems of weights summing to 1 that are compatible with the constraints induced by E. But, enumerating the vertices of the polytope defined by the system of inequations corresponding to the preferences in E is a NP hard problem that is not easy at all to handle in practice [13]. Indeed given a feasible system of linear inequalities, generating all vertices of the corresponding polyhedron is hard. Yet, in the case of bounded polyhedra (i.e., polytope) the complexity remains open. It is why we have chosen to explore another route in this study, based on the exploitation of a pattern avoiding contradictory trade-offs, and patterns expressing that preferences should go well with analogical proportions. This idea which may sound fancy at first glance is based on the empirical evidence that analogical proportion-based classifiers work well and the theoretical result that such classifiers make no error in the Boolean case when the labeling function is affine [6]. A result of the same nature might be conjectured when attributes are nominal rather the Boolean. In our case since the scale S is finite, criteria may be regarded as nominal attributes.

4 Analogy-Based Preference Prediction

As just said, we investigate how analogical proportions can help for predicting preference relations from a given set of such relations, while avoiding the generation of contradictory trade-offs. We call APP such a method, which is short for "Analogy-based Preference Prediction".

Let us consider a preference relation $\succeq$ over the universe set S^n (n is the number of criteria) and a set of preference examples $E = \{e_i : x^i \succeq y^i\}$ telling us that choice x^i is preferred to choice y^i. We may apply monotony on examples in the set E, in order to produce other new valid examples. Namely, if $(x^i \succeq y^i) \in E$ and if x', y' are such that $x' \succeq x^i$ and $y^i \succeq y'$ due to dominance, then $x' \succeq y'$ should hold as well. We denote $comp(E)$ this completion of set E by repeated application of monotony. Moreover the scale $S = \{1, 2, ..., k\}$. In the experiments we use $S = \{1, 2, 3, 4, 5\}$. Such a scale is usual in practice.

4.1 Methodology

Given a new item $D : X, Y$ whose preference is to be predicted, the basic principle of APP is to find the *good* triples (A, B, C) of examples in E (or if possible in $comp(E)$) that form with D either the non-contradictory trade-offs pattern (considered in first), or one of the three analogical proportion-based inference patterns. Such triples, when applicable, will help to guess the preference of D by applying a majority vote on the solutions provided by each of these triples.

Let us consider one of the basic patterns:

$$A : x_{-i}\alpha \succeq y_{-i}\beta$$

$$B : x_{-i}\gamma \preceq y_{-i}\delta$$

$$C : v_{-j}\alpha \succeq w_{-j}\beta$$

$$D : v_{-j}\gamma \ ? \ w_{-j}\delta$$

where preference of D is unknown.

The APP can be described by this basic process:

- For a given D, search for good triples in E.
- In case no good triples could be found in E, search for such triples in $comp(E)$.
- Apply a majority vote on the candidate solutions of these good triples to predict the preference of D.

The process of searching for good triples can be summarized by the following 3 steps:

1. **Find good** C: In the basic pattern, we can see that the item C may be any example in the set E which is identical to D except on one criterion that is denoted by its index j. The intuitive idea of APP is to start by searching for the *best* examples $C \in E$ that fit the basic pattern considered. As j may be any index in the set of criteria, a loop on all possible criteria $j \in \{1, ..., n\}$ should be executed in order to find j. Once a candidate C is found, this helps to also fix parameters α, β, γ and δ for the *current* candidate triple. We save such parameters as $param = \{\alpha, \beta, \gamma, \delta, j\}$.
2. **Find good** A: Once parameters α and β are fixed for each example C, it is easy to find a good example $A \in E$ in which α and β appears on the same criterion, indexed by i. As in the case of C, a similar process is to be applied to find such examples A. This helps to fix a new parameter i and update the set of parameters to be $param = \{\alpha, \beta, \gamma, \delta, j, i\}$.
3. **Find good** B: As a result of the previous step, to each candidate pair (A, C) along with D corresponds a set of candidate parameters $param = \{\alpha, \beta, \gamma, \delta, j, i\}$. The last step is to find *all* good examples $B \in E$ to enclose the triple (A, B, C), i.e., those that fit exactly the pattern: $p : x_{-i}\gamma, y_{-i}\delta$ regardless of the sign of the preference relation.

The next step of the APP is to predict preference based on the selected good triples. Each candidate triple helps to predict an atomic preference solution for D by inference based on any of the previous patterns described in Sects. 2 and 3. A global preference solution is computed through a majority vote applied on *all* atomic solutions and finally assigned to D.

As expected, the proposed APP may fail in case no examples C (or A, or B) could be found in the set E especially when E has a limited size (only few examples of preferences between choices are available). To overcome this problem, we propose to expand the set E and search for examples e in $comp(E)$.

For any example $e \in E$ s.t.: $e : x_{-i}\alpha \succeq y_{-i}\beta$, one may produce a new valid preference example by *dominance* (monotony) defined as:

$$newe \in comp(E) \text{ iff } newe : newx_{-i}\alpha \succeq newy_{-i}\beta \\ \text{and } newx_{-i} \geq x_{-i} \text{ and } y_{-i} \geq newy_{-i} \quad (1)$$

For any relation e with opposite preference sign corresponds a $newe$ by reversing the operators.

4.2 Algorithms

Based on the previous ideas, we propose two different algorithms for predicting preferences in the following.

Let E be a training set of examples whose preference is known. Given a new preference relation $D \notin E$ whose preference is to be predicted, the first alternative is to look at all good triples $(A, B, C) \in E$ that provide a solution for the item D. It is important to note that in a pre-processing step, one may search for the appropriate pairs $(A, B) \in E$, s.t: $A : x_{-i}\alpha, y_{-i}\beta$ and $B : x_{-i}\alpha^{'}, y_{-i}\beta^{'}$, i.e., A is identical as B except in one attribute. This step aims to filtering the high number of pairs and keeping only those that fit the previous patterns. This first option is described by Algorithm 1.

Algorithm 1. APP with restricted set

Input: a training set E of examples with known preferences
a new item $D \notin E$ whose preference is unknown.
$PredictedPref = false$
Preprocess: $S_{(A,B)} = FindPairs(E)$.
CandidateVote(p)=0, for each $p \in \{\preceq, \succeq\}$
for each $C \in E$ **do**
  if IsGood(C) **then**
    for each $(A, B) \in S_{(A,B)}$ **do**
      if IsGood(A) AND IsGood(B) **then**
        $p = Sol(A, B, C, D)$
        $CandidateVote(p)$++
      end if
    end for
  end if
end for
$maxi = max\{CandidateVote(p)\}$
if $maxi \neq 0$ AND $unique(maxi, CandidateVote(p))$ **then**
  $Preference(D) = argmax_p\{CandidateVote(p)\}$
  $PredictedPref = true$
end if
if $PredictedPref$ **then**
  return$Preference(D)$
else
  return (*not predicted*)
end if

In case Algorithm 1 fails to find good triples, the second alternative (described by Algorithm 2) aims at expanding the set of preference examples E by searching for good triples (A, B, C) in $comp(E)$. In this set, examples are produced by applying dominance (monotony) on elements found in E.

Note that the algorithms that we proposed in this paper are quite different from the one in [17], where only a brute force procedure for preference prediction is presented without giving any clue for implementation. Neither an evaluation process nor comparisons are provided. Moreover, the algorithm in [17] assumes that a completed set Comp(E) is first computed and used as input for prediction. Generating the whole set Comp(E) is computationally very expensive, while it may be useless. In this paper we search for useful elements in Comp(E) only in case no appropriate triples can be found in E. This clearly reduces the computational burden. We then describe a precise process for searching for appropriate triples and also present a way to evaluate the algorithm's performance (which is not done in [17]). In terms of complexity, due to the use of triples of items, our algorithms have a cubic complexity while the approach to find the set of weights compatible with the set E has at least a complexity $O(|E| * |E'|^n)$, where n is the number of weighted averages to be generated and E' is the new set generated from one of these weights.

Algorithm 2. APP with a completion set

Input:a training set E of examples with known preferences
a new item $D \notin E$ whose preference is unknown.
Preprocess: $S_{(A,B)} = FindPairs(E)$.
if Algo1(D,E)=*not predicted* **then**
  CandidateVote(p)=0, for each $p \in \{\preceq, \succeq\}$
  for each $C \in E$ **do**
    newC=comp(C)
    for each $(A, B) \in E \times E$ **do**
      if IsGood(A) AND IsGood(B) **then**
        $p = Sol(A, B, newC, D)$
        $CandidateVote(p)$++
      end if
    end for
    if CandidateVote(p)=0, for each $p \in \{\preceq, \succeq\}$ **then**
      for each $A \in E$ **do**
        newB=comp(B)
        if IsGood(A) AND IsGood(NewB) **then**
          $p = Sol(A, newB, newC, D)$
          $CandidateVote(p)$++
        end if
      end for
    end if
  end for
end if
$Preference(D) = argmax_p\{CandidateVote(p)\}$
return $Preference(D)$

5 Experiments

In order to evaluate the proposed APP algorithms, we have developed a set of experiments that we describe in the following. We finally compare these algorithms to a nearest neighbors method.

5.1 Datasets and Validation Protocol

The experimental study is based on three datasets, the two first ones are synthetic data generated from a chosen weighted average function. For each of these datasets, all possible combinations of the feature values over the scale S are considered. For each pair of vectors $(x, y) \in E^2$, the preference is determined, through computing weighted averages as follows: $x \succeq y$ if and only if $U(x) = \sum_{i=1}^{n} w_i x_i \geq U(y) = \sum_{i=1}^{n} w_i y_i$, where w_i is the weight associated to criterion i.

- **Dataset 1**: we consider only 3 criteria in each preference relation i.e., $n = 3$ and we test with 3 different options of this dataset. In each of them, examples are generated using a different weighted average function:
  - Weights1(noted w_1) with $0.6, 0.3, 0.1$ weights respectively for criteria 1, 2 and 3.
  - Weights2(w_2) with $0.5, 0.3, 0.2$ weights respectively for criteria 1, 2 and 3.
  - Weights3(w_3) with $0.7, 0.2, 0.1$ weights respectively for criteria 1, 2 and 3.
- **Dataset 2**: we expand each preference relation to support 5 criteria, i.e.: $n = 5$ and similar to dataset1, we tried different options of weights:
  - Weights1(w_1) with $0.4, 0.3, 0.1, 0.1, 0.1$ weights respectively for criteria 1, 2, 3, 4 and 5.
  - Weights2(w_2) with $0.3, 0.3, 0.2, 0.1, 0.1$ weights respectively for criteria 1, 2, 3, 4 and 5.
  - Weights3(w_3) with $0.6, 0.2, 0.1, 0.05, 0.05$ weights respectively for criteria 1, 2, 3, 4 and 5.

We may consider that 5 criteria is already a rather high number of criteria for the cognitive appraisal of an item by a human user in practice. For both datasets, each criterion is evaluated on a scale with 5 different values, i.e., $S = \{1, ..., 5\}$.

To check the applicability of APP algorithms, it is important to measure their efficiency on real data. For such data, two choices/options are provided to a human judge and ask him/her to pick one of them. To the best of our knowledge, there is no such a dataset that is available in this format [5]. For this purpose, we select the Food dataset from context aware recommender systems available in[2].

[2] https://github.com/trungngv/gpfm.

– The **Food dataset** used by [5] contains 4036 user preferences among 20 food menus picked by 212 users. In each of them a user is supposed to provide a numerical rating. Each item in this dataset is represented by three features that correspond to three different level of user hunger. Each of them could be in 3 possible situations. To test this dataset, we first pre-process it to generate the preferences in the format recommended by our model: We group all the input data by user and by foods. For any two inputs with different ratings, we generate a preference relation. Since we are only dealing with nominal values in this paper, we limit our study to 5 different foods.

Regarding the validation protocol, for each dataset we have investigated different sizes of E between 20 to 1000 examples. For each subset of data, we repeat the experiment 100 times to get stable results. In each experiment, a standard 10 fold cross-validation technique is applied. The prediction accuracies shown in next subsection for both algorithms are for the testing set and are the average over the 100 rounds.

5.2 Results

Figures 1 and 2 show prediction accuracies of APP algorithms respectively for Datasets 1 and 2 for different sizes of each dataset and different weights (see curves "Algo1_$w_i(E)$" and "Algo2_$w_i(E)$"; the other curves using $InterE$ data are explained at the end of Sect. 5.2).

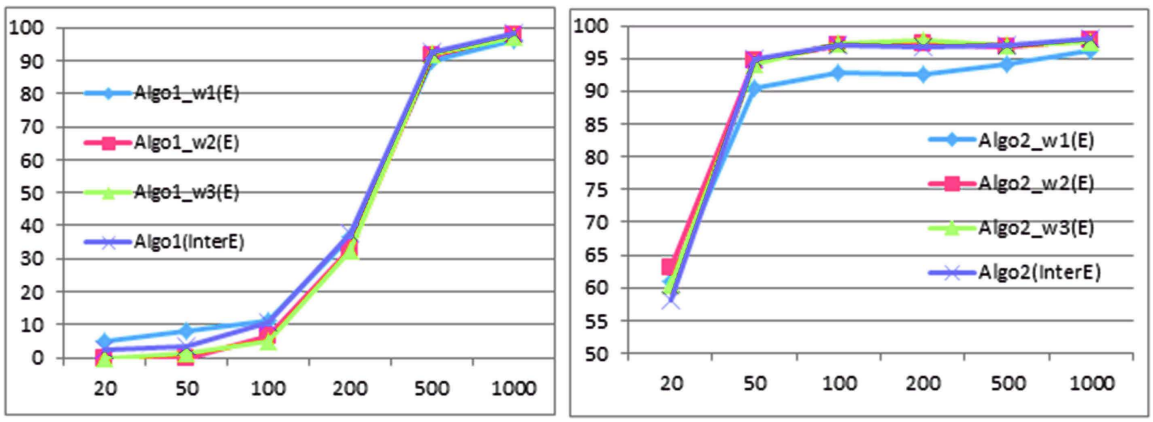

Fig. 1. Prediction accuracies for Dataset 1 with 3 criteria and different sizes of subsets of data.

If we compare results of Algorithm 1 and 2 in Figs. 1 and 2, we can draw the following conclusions:

– In case of Dataset 1, Algorithm 2 is largely better than Algorithm 1 for small sizes of the dataset (size < 500). Algorithm 1 and 2 provide close results for datasets with more than 500 examples even if Algorithm 2 is always better.

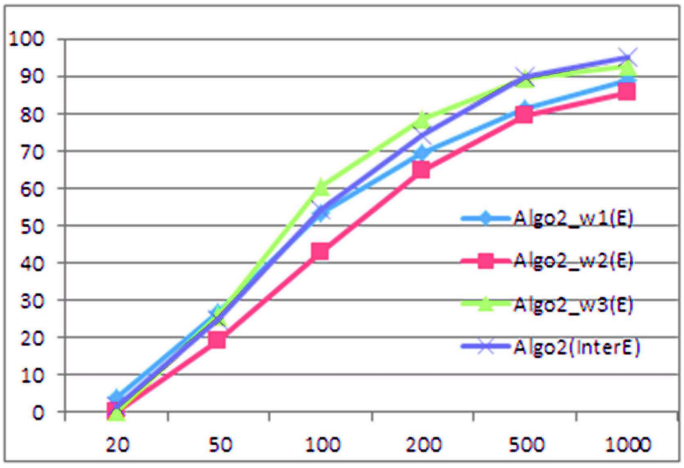

Fig. 2. Prediction accuracies for Dataset 2 with 5 criteria and different dataset sizes

- Algorithm 1 seems inefficient for prediction when only a small sample E of examples is given. Since we are only dealing with grades in a nominal way in this paper, many triples are rejected due to the 'yes-no' acceptance condition for appropriate triples. This may be relaxed and extended to deal with truly numerical values (which is a process under study).
- Algorithm 2 shows a very good prediction ability even with rather small samples (i.e., 50 preference examples). Due to monotony exploited by this algorithm, it is always possible to find appropriate triples in Comp(E).
- Since Algorithm 2 is clearly better than Algorithm 1, we limit our experiments on Algorithm 2 when applied to Dataset 2.
- In case of Dataset 2, prediction accuracy of Algorithm 2 is clearly improved when dataset size increases as in case of Dataset1.
- If we compare results for the two datasets, it is obvious that Algorithm 2 is significantly better when applied to Dataset 1 (with 3 criteria) than to Dataset 2 (with 5 criteria). This means that it is easier for Algorithm 2 to predict preferences with a limited number of criteria, as we may expect.
- The three weighted averages for Datasets 1 and 2 yield close results even though the weights w_1 for Dataset 1 and w_2 for Dataset 2 give slightly worse results. In the two latter, the weights are less contrasted (i.e., closer to each other), which may indicate that it is somewhat easier for the algorithm to predict preferences generated from a weighted average with more contrasted weights. Still the three weights sets yield very similar results for large dataset sizes.

The previous results show the effectiveness of Algorithm 2 as preference predictor which fits with a given weighted average used to produce the preference examples especially for small number of criteria.

It is worth pointing out that the predicted preferences have been evaluated as being valid, or not, on the basis of 3 weighted averages, the ones used for generating the dataset with its three versions. It is clear that a given set E of preference examples is compatible with a more large collections of weights.

Strictly speaking a prediction is valid if it is correct with respect to at least one of the collections of weights compatible with E. As already said, determining all the extreme points of the polytope of the weights compatible with E is quite tricky. So in the above reported experiments, we have compared the prediction to ones obtained by using the weighted averages used in the generation of the training set, and thus the reported accuracies are *lower bounds* of the true accuracies.

In the following, we aim at estimating the extent to which APP algorithms succeed to fit with a larger variety of weighted averages. For this purpose, we also experiment our algorithms on datasets obtained by applying the following procedure:

1. Select m different weighted averages $P_1, P_2, ...P_m$, all of them satisfying the same ranking for the importance of each criterion. We denote $E_1, E_2, ..., E_m$ respectively the corresponding sets of preference examples.
2. Find the *intersection* set of these preference sets denoted $InterE = E_1 \cap E_2 \cap ... \cap E_m$ containing only examples that *any* of all weighted averages $\in \{P_1, P_2, ...P_m\}$ can generate.
3. Apply APP Algorithms to predict preferences on subsets of $InterE$.

To test the previous idea, we use 5 different weighted averages ($m = 5$) keeping the same importance ranking for the criteria. Results of Algorithm 1 and 2 are given in Figs. 1 and 2 (See Algo1(InterE) and Algo2(InterE)). These results show that:

- Accuracy of Algorithm 2 is clearly improved when the set $InterE$ is used instead of the set E which is produced from *one* of the 5 weighted averages. This can be noticed for most dataset sizes especially large sizes of the data.
- For Algorithm1, a slight improvement is noticed when using the set $InterE$ especially for large dataset sizes exceeding 100 examples.

This confirms our intuition and shows that if preference examples agree with a variety of weighted averages, more predicted preferences can be considered as fitting with these examples.

5.3 Comparison with a Nearest Neighbor Method

In the following, we aim at comparing APP algorithms to a basic nearest-neighbor (NN) preference learning approach. That is why we implemented and tested a NN preference learning algorithm that we call $NNPL$. The basic principle of this algorithm is to predict any new preference example d in the same way as its $nearest - neighbor$ preference example(s) belonging to the training set. For this purpose, we compute the distance of d to *all* training examples and we select those being sufficiently close. In case of ties, a majority vote is applied. Let us consider an example $d : u \ ? \ v$ to be predicted and $c : x^i \succeq y^i \in E$, we compute two distances to c defined as:

$$Dis1(c, d) = (\mid u - x^i \mid, \mid v - y^i \mid)$$

$$Dis2(c,d) = (\mid u - y^i \mid, \mid v - x^i \mid)$$

where $\mid a - b \mid$ is simply the Manhattan distance of vectors components.

We define:

$$NN(c,d) = \{c \in E \text{ s.t.:} Dis(c,d) \leq \theta\}$$

where $Dis(c,d) = Min(Dis1, Dis2)$.

We want to check if the preference d is predicted by APP in the same way as by NNPL. For this purpose, we computed the frequency of the cases where both APP and NNPL predict the correct preference for d (this case is denoted SS), the frequency of the cases where both algorithms predict an incorrect label (denoted FF), the frequency where APP prediction is correct and NNPL prediction is wrong (SF) and the frequency where APP prediction is wrong and NNPL prediction is correct (FS). For this experiment we exploit APP algorithms applied to Dataset 1 for which we only consider the cases where both APP and NNPL are able to provide a prediction (we only include examples that can be predicted by the two compared algorithms). Regarding the threshold θ, we tried 3 different values in $\{1, 2, 3\}$ and we report the results for the best one (θ is fixed to 2 in this experiment). Results are saved in Table 1.

Table 1. Frequency of success/fail of APP predictions that are predicted same/not same as the $NNPL$ approach

Dataset size	Algo1				Algo2			
	SS	FF	SF	FS	SS	FF	SF	FS
50	0.901	0	**0.099**	0	0.697	0.026	**0.2**	0.076
100	0.807	0.04	**0.113**	0.04	0.811	0.035	**0.134**	0.02
200	0.841	0.02	**0.081**	0.058	0.826	0.023	**0.106**	0.045
500	0.887	0.015	**0.056**	0.042	0.863	0.012	**0.069**	0.056
1000	0.92	0.011	**0.042**	0.027	0.911	0.013	**0.045**	0.031

In this table, we note that:

- For most cases, APP and NNPL agree and predict the same preference (the highest frequency can be seen in column SS).
- If we compare results in column SF and FS, we can see that the frequency of cases where APP provides the correct prediction, while $NNPL$ does not (column SF) is significantly better than the opposite case (column FS). This can be seen for all dataset sizes (except the smallest one). Especially for a size of 100 examples, more than 10% of the total correctly predicted examples are predicted differently from the NNPL.

Lastly, we also compare the prediction accuracy of APP algorithms to NNPL when applied to the Food dataset as representative of real data. Results of Food dataset, in Table 2, shows that:

Table 2. Classification accuracy of Algo 1 and 2 applied to Food dataset

Dataset	Algo1	Algo2	NNPL	[5]
Food	**73.31 ± 2.81**	**73.35 ± 2.63**	61.57 ± 3.58	61

- APP algorithms performs well when dealing with real dataset as in case of synthetic data.
- Algorithms 1 and 2 significantly outperform the probabilistic approach proposed by [5] applied to this dataset.
- Algorithms 1 and 2 also do better than the NNPL.

In fact, APP benefits from two basic differences if compared to the classic NNPL (i) using a large amount of *triple* voters for prediction while NNPL uses a simpler voting-based strategy that directly applies a vote on the nearest neighbor examples and (ii) using more complex calculation by comparing pairs of items instead of comparing simply two items. We note that the four items involved in each comparison are not necessarily closes as we shall see from the basic pattern described in Sect. 4.1. In this pattern, it is clear that D is neighbor to C (only one criteria is different) and B is neighbor to A but D is not necessarily neighbor to A or B. This increase in terms of complexity (which is cubic in case of APP) may explain the good results of APP if compared to NNPL having linear complexity.

6 Conclusion

The approach presented in the paper does not amount to inducing a general representation of the set of examples under the form of a particular class of aggregation functions or of a graphical preference representation, from which we could predict a preference relation between any pair of choices. We simply apply an analogical inference principle (or in the comparative study, a nearest neighbor principle) for making directly the prediction. This type of approach is successful in classification. The present study shows that it is applicable as well in preference learning. Still in classification, the classes have just to be mutually exclusive. Preferences are more structured in the sense that they are expected not to exhibit contradictory trade-offs and to be monotone, which has to be taken into account in the learning process. In our approach, our experiments have been on training sets generated by means of weighted sums, which is a quite standard aggregation function, and we have obtained good results for rather small subsets of examples. Still it is known that the representation of multiple-criteria preferences may require more general settings such as Choquet integrals [11] where the condition for avoiding contradictory trade-offs is weaker. Adapting the proposed approach to guess preferences generated by such more general settings is a topic for further research.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning; foundational issues, methodological variations, and system approaches. AICom **7**(1), 39–59 (1994)
2. Abdel-Aziz, A., Hüllermeier, E.: Case base maintenance in preference-based CBR. In: Hüllermeier, E., Minor, M. (eds.) ICCBR 2015. LNCS (LNAI), vol. 9343, pp. 1–14. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24586-7_1
3. Bayoudh, S., Miclet, L., Delhay, A.: Learning by analogy: A classification rule for binary and nominal data. In: Proceedings International Joint Conference on Artificial Intelligence IJCAI07, pp. 678–683 (2007)
4. Bounhas, M., Prade, H., Richard, G.: Analogy-based classifiers for nominal or numerical data. Int. J. Approx. Reason. **91**, 36–55 (2017)
5. Chen, S., Joachims, T.: Predicting matchups and preferences in context. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 775–784. ACM, New York (2016)
6. Couceiro, M., Hug, N., Prade, H., Richard, G.: Analogy-preserving functions: a way to extend boolean samples. In: Proceedings 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017, pp. 1575–1581 (2017)
7. Domshlak, C., Hüllermeier, E., Kaci, S., Prade, H.: Preferences in AI: an overview. Artif. Intell. **175**(7–8), 1037–1052 (2011)
8. Fahandar, M.A., Hüllermeier, E.: Learning to rank based on analogical reasoning. In: Proceedings 32th National Conference on Artificial Intelligence (AAAI 2018), New Orleans, 2–7 February 2018
9. Fürnkranz, J., Hüllermeier, E. (eds.): Preference Learning. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14125-6
10. Fürnkranz, J., Hüllermeier, E., Rudin, C., Slowinski, R., Sanner, S.: Preference learning (dagstuhl seminar 14101). Dagstuhl Rep. **4**(3), 1–27 (2014)
11. Grabisch, M., Labreuche, C.: A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. Annals Oper. Res. **175**, 247–286 (2010)
12. Hüllermeier, E., Fürnkranz, J.: Editorial: preference learning and ranking. Mach. Learn. **93**(2–3), 185–189 (2013)
13. Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V.: Generating all vertices of a polyhedron is hard. Discret. Comput. Geom. **39**(1), 174–190 (2008)
14. Lepage, Y.: Analogy and formal languages. In: Proceedings FG/MOL 2001, pp. 373–378 (2001). http://www.slt.atr.co.jp/lepage/pdf/dhdryl.pdf.gz
15. Miclet, L., Bayoudh, S., Delhay, A.: Analogical dissimilarity: definition, algorithms and two experiments in machine learning. JAIR **32**, 793–824 (2008)
16. Miclet, L., Prade, H.: Handling analogical proportions in classical logic and fuzzy logics settings. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS (LNAI), vol. 5590, pp. 638–650. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02906-6_55
17. Pirlot, M., Prade, H., Richard, G.: Completing preferences by means of analogical proportions. In: Torra, V., Narukawa, Y., Navarro-Arribas, G., Yañez, C. (eds.) MDAI 2016. LNCS (LNAI), vol. 9880, pp. 135–147. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45656-0_12
18. Prade, H., Richard, G.: From analogical proportion to logical proportions. Logica Universalis **7**(4), 441–505 (2013)

19. Rumelhart, D.E., Abrahamson, A.A.: A model for analogical reasoning. Cognitive Psychol. **5**, 1–28 (1973)
20. Stroppa, N., Yvon, F.: Analogical learning and formal proportions: Definitions and methodological issues. Technical report, June 2005
21. Yvon, F., Stroppa, N., Delhay, A., Miclet, L.: Solving analogical equations on words. Technical report, Ecole Nationale Supérieure des Télécommunications (2004)