La classification analogique : Une nouvelle manière d'exploiter des exemples *

Myriam Bounhas[1] **Henri Prade**[2] **Gilles Richard**[2]

[1] LARODEC Laboratory, ISG de Tunis, Tunisie
& Emirates College of Technology, Abu Dhabi, United Arab Emirates
[2] IRIT, Université Paul Sabatier, Toulouse, France
myriam_bounhas@yahoo.fr {prade,richard}@irit.fr

Résumé

Introduites il y a quelques années, les méthodes de classification basées sur l'analogie constituent une addition notable à l'ensemble des techniques d'apprentissage à base d'instances. Elles fournissent des résultats surprenants (en terme de précision) sur de nombreuses bases de données classiques. Ces méthodes considèrent tous les triplets d'exemples dans un ensemble d'apprentissage qui sont en proportion analogique avec l'item à classer sur un nombre maximal d'attributs, et pour lesquels l'équation analogique correspondante sur la classe a une solution. Dans cet article, on propose une nouvelle approche où on se focalise sur une petite partie de l'ensemble d'apprentissage pour classer un nouvel item. Pour limiter l'ampleur de la recherche, on considère d'abord les exemples qui sont aussi similaires que possibles au nouvel item à classer. On considère ensuite seulement les paires d'exemples présentant la même dissimilarité qu'entre le nouvel item et l'un de ses plus proches voisins. De cette manière, on construit implicitement des triplets qui sont en proportion analogique sur tous les attributs avec le nouvel item. La classification est alors effectuée sur la base d'un vote majoritaire sur les paires conduisantà une équation sur la classe ayant une solution. Le nouvel algorithme donne des résultats au moins aussi bon que les classifieurs analogiques existants pour une complexité moyenne plus basse.

Abstract

Introduced a few years ago, analogy-based classification methods are a noticeable addition to the set of lazy learning techniques. They provide amazing results (in terms of accuracy) on many classical datasets. They look for all triples of examples in the training set that are in analogical proportion with the item to be classified on a maximal number of attributes and for which the corresponding analogical proportion equation on the class has a solution. In this paper, we demonstrate a new approach where we focus on a small part of the training set when classifying a new item. To limit the scope of the search, we first look for examples that are as similar as possible to the new item to be classified. We then only consider the pairs of examples presenting the same dissimilarity as between the new item and one of its closest neighbors. In this way, we implicitly build triples that are in analogical proportion on all attributes with the new item. Then the classification is made on the basis of a majority vote on the pairs leading to an equation on the class that has a solution. This new algorithm provides results at least as good as previous analogical classifiers with a lower average complexity.

1 Introduction

C'est une idée naturelle que de penser que l'apprentissage humain est affaire d'observation, puis d'imitation et de transposition appropriée de ce qui a été observé [1]. Cela peut être relié à l'idée de raisonnement analogique [9, 11]. Il a été montré qu'une telle idée peut être efficace pour résoudre des tests de QI du type matrices progressives de Raven, où on construit la solution (plutôt que de la choisir dans un ensemble de candidats possibles), en recopiant des observations appropriées [7]. Le processus utilisé est étroitement lié à l'idée de proportion analogique, qui est une notion centrale en raisonnement analogique [22].

Les proportions analogiques sont des énoncés de la forme "A est à B comme C est à D", souvent notées $A : B :: C : D$ qui expriment que "A diffère de B comme C diffère de D", ainsi que "B diffère de A comme D dif-

*Une version anglaise est également disponible [3].

fère de C" [17]. En d'autres termes, la paire (A, B) est analogue à la paire (C, D) [8]. Une vue formalisée des proportions analogiques a été récemment mise au point dans des contextes algébriques ou logiques [13, 24, 17, 19]. Les proportions analogiques s'avèrent être un outil puissant pour l'analyse linguistique morphologique [12], ainsi que dans des tâches de classification où des résultats compétitifs avec ceux des méthodes classiques d'apprentissage artificiel ont été obtenus, d'abord par [16].

En classification, nous supposons ici que des objets ou des situations A, B, C, D sont représentés par des vecteurs de valeurs d'attributs, notés $\vec{a}, \vec{b}, \vec{c}, \vec{d}$. L'approche analogique de la classification fondée sur la proportion analogique repose sur l'idée que la classe inconnue $x = cl(\vec{d})$ d'une nouvelle instance $\vec{d}$ peut être prédite en tant que solution d'une équation exprimant que la proportion analogique $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ tient entre les classes. Cela se fait sur la base de triplets $(\vec{a}, \vec{b}, \vec{c})$ d'exemples de l'ensemble d'apprentissage qui sont tels qu'une proportion analogique $\vec{a} : \vec{b} :: \vec{c} : \vec{d}$ tient composante par composante pour tous les (ou sur un grand nombre d') attributs décrivant les exemples.

La construction de la solution d'un test de QI ou la classification d'un nouvel élément sont des tâches très similaires. En effet, les images d'un test de QI visuel peuvent être décrites en termes d'attributs sous forme de vecteurs de leurs valeurs correspondantes, et on doit alors trouver une autre image représentée de la même manière, sur la base de l'hypothèse implicite que cette dernière image doit être associée avec les images données, cette association étant régie par la disposition spatiale des images en lignes, et peut-être en colonnes. De même, dans une tâche de classification, les vecteurs de description sont associés avec des classes dans l'ensemble des exemples, alors qu'une classe doit être prédite pour un nouveau vecteur. Dans ce papier, ce parallèle nous conduit à proposer une nouvelle approche, plus simple et plus compréhensible à la classification basée sur l'analogie, inspiréd de notre approche pour résoudre les tests de QI des matrices progressives de Raven [7]. Cela contraste avec les approches analogiques précédentes de la classification, où *tous* les triplets d'exemples formant une proportion analogique avec le nouvel élément à classer sont systématiquement pris en compte de manière aveugle.

Le papier est organisé comme suit. Tout d'abord, un rappel des proportions analogiques, à la fois sur les attributs booléens et les attributs discrets est présenté en section 2. Ensuite, dans la section 3, un parallèle est fait entre la résolution de tests de QI et le problème de la classification, où on insiste sur la pertinence des proportions analogiques pour l'analyse d'un ensemble d'exemples. Après avoir rappelé les approches existantes de la classification basées sur les proportions analogiques en section 4, la nouvelle approche est décrite en section 5, et les résultats sur des bases de données classiques sont indiqués en section 6, avec une discussion générale sur l'intérêt du nouveau classifieur, qui s'inscrit dans un nouveau paradigme de l'apprentissage qui peut être appelé *apprentissage artificial créatif*. La conclusion souligne quelques lignes de recherche future.

2 Rappel sur les proportions analogiques

A l'origine de la proportion analogique se trouve la proportion numérique standard, qui est une égalité entre deux ratios $\frac{a}{b} = \frac{c}{d}$, où a, b, c, d sont des nombres. Dans le même esprit une proportion analogique $a : b :: c : d$ qui se lit "a est à b ce que / comme c est à d", ce qui exprime de manière informelle que "a diffère de b comme c diffère de d" et vice versa.

Comme c'est le cas pour les proportions numériques, un tel énoncé analogique est censé toujours tenir lorsque les paires (a, b) et (c, d) sont échangées, ou lorsque les termes moyens b et c sont permutés (voir [20] pour une discussion détaillée). Quand on considère des valeurs booléennes (appartenant à $\mathbb{B} = \{0, 1\}$), un moyen simple d'obtenir la contrepartie symbolique des proportions numériques a été donné dans [19] en mettant l'accent sur des indicateurs capturant les idées de "similitude" et de "dissemblance". Pour une paire (a, b) de variables booléennes, ces indicateurs sont définis Ê par :

$a \wedge b$ et $\neg a \wedge \neg b$: *indicateurs de similarité*,
$a \wedge \neg b$ et $\neg a \wedge b$: *indicateurs de dissimilarité*.

Une proportion logique (voir[21] pour une étude approfondie) est une conjonction de deux équivalences entre ces indicateurs et le meilleur "clone" de la proportion numérique est la proportion analogique définie comme suit :

$$(a \wedge \neg b \equiv c \wedge \neg d) \wedge (\neg a \wedge b \equiv \neg c \wedge d).$$

Cette expression de la proportion analogique, utilisant uniquement des différences, peut se lire comme *ce qui est vrai pour a et pas pour b, c'est exactement ce qui est vrai pour c et pas pour d, et vice versa.* En tant que tel, une proportion logique est une formule booléenne impliquant 4 variables et on peut vérifier dans la Table 1 que l'expression logique de $a : b :: c : d$ satisfait la *symétrie* ($a : b :: c : d\ c : d :: a : b$) et la *permutation centrale* ($a : b :: c : d \to a : c :: b : d$), qui sont reconnues depuis longtemps comme des propriétés caractéristiques de la proportion analogique. En outre, cette modélisation booléenne jouit aussi de la propriété de transitivité $(a : b :: c : d) \wedge (c : d :: e : f) \to (a : b :: e : f)$.
En termes de schémas génériques, nous voyons que la proportion analogique tient toujours pour les 3 motifs suivants : $s : s :: s : s$, $s : s :: t : t$ et $s : t :: s : t$, où s et t sont des valeurs booléennes distinctes.

Directement associée aux proportions numériques la "règle de trois" permet de calculer un 4^e terme approprié x afin de compléter une proportion $\frac{a}{b} = \frac{c}{x}$. Cette propriété

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

TABLE 1 – Valuations pour lesquelles $a : b :: c : d$ est vraie

a une contrepartie dans le cas booléen où le problème peut s'énoncer comme suit. Etant donné un triplet (a, b, c) de valeurs booléennes, existe-t-il une valeur booléenne x tel que $a : b :: c : x = 1$, et dans ce cas, cette valeur est-elle unique ? Dans certains cas, l'équation n'a pas de solution puisque le triplet (a, b, c) peut prendre $2^3 = 8$ valeurs, alors que $a : b :: c : d$ n'est vrai que pour 6 4-uplets distincts. En effet, les équations $1 : 0 :: 0 : x = 1$ et $0 : 1 :: 1 : x = 1$ n'ont pas de solution. Il est facile de prouver que l'équation analogique $a : b :: c : x = 1$ est résoluble si et seulement si $(a \equiv b) \vee (a \equiv c)$ est vrai. Dans ce cas, l'*unique* solution est donnée par $x = a \equiv (b \equiv c)$.

Si on revient à nos 3 schémas analogiques $s : s :: s : s$, $s : s :: t : t$ et $s : t :: s : t$, la résolution d'un modèle incomplet (c'est à dire où le 4^e élément est manquant) est juste une question de recopie de la valeur s ou t vue précédemment (une idée déjà soulignée dans [7]).

Les objets sont généralement représentés par des *vecteurs* de valeurs d'attributs (plutôt que par une seule valeur). Ici, chaque composante de vecteur est supposée être une valeur booléenne. Une simple extension de la proportion analogique à $\mathbb{B}^n$ est alors :

$$\vec{a} : \vec{b} :: \vec{c} : \vec{d} \text{ ssi } \forall i \in [1, n],\ a_i : b_i :: c_i : d_i$$

Toutes les propriétés de base (symétrie, permutation centrale) tiennent encore pour cette extension vectorielle. Le processus de résolution de l'équation reste également applicable, mais offre un nouvel aperçu sur la proportion analogique : les proportions analogiques sont *créatives*. En effet, considérons l'exemple suivant où $\vec{a} = (1, 0, 0), \vec{b} = (0, 1, 0)$ et $\vec{c} = (1, 0, 1)$. La résolution de l'équation analogique $\vec{a} : \vec{b} :: \vec{c} : \vec{x}$ donne $\vec{x} = (0, 1, 1)$, qui est un vecteur différent de $\vec{a}, \vec{b}$ et $\vec{c}$. En fait, nous pouvons étendre ce mécanisme booléen à des domaines discrets où le nombre de valeurs d'attributs candidates est fini, mais supérieur à 2. En effet, étant donné un domaine d'attribut $\{v_1, \cdots, v_m\}$, nous pouvons "binariser" l'attribut au moyen des m propriétés "avoir ou non la valeur v_i". Par exemple, étant donné un attribut tri-valué pouvant prendre les valeurs v_1, v_2, v_3, respectivement codées par 100, 010, et 001, la proportion analogique $v_1 : v_2 : v_1 : v_2$ tient toujours *composante par composante* sur le codage sous forme de vecteurs de $\mathbb{B}^3$: 100 : 010 :: 100 : 010, puisque les proportions booléennes $1010, 0101$ et 0000 tiennent. On peut vérifier que le codage binaire permet ainsi de reconnaitre les proportions analogiques qui sous-tendent les deux autres schémas $v_1 : v_1 : v_2 : v_2$ et $v_1 : v_1 : v_1 : v_1$. Inversement, une proportion analogique non valide, par exemple, $v_1 : v_2 : v_2 : v_1$, ne sera pas reconnue comme une proportion analogique dans ce codage binaire. Cette remarque nous permettra de gérer les attributs discrets directement en termes des trois modèles $s : s :: s : s$, $s : s :: t : t$ et $s : t :: s : t$, où s et t sont maintenant des valeurs du domaine d'un attribut discret [1].

3 Des tests de QI aux classifieurs analogiques

Un ensemble d'apprentissage TS d'exemples $\vec{x^k} = (x^k{}_1, ..., x^k{}_i, ..., x^k{}_n)$ avec leur classe $cl(\vec{x^k})$, où $k = 1, \cdots, t$ peut recevoir une lecture informelle en termes de proportions analogiques. A savoir, "$\vec{x^1}$ est à $cl(\vec{x^1})$ comme $\vec{x^2}$ est à $cl(\vec{x^2})$ comme $\cdots$ comme $\vec{x^t}$ est à $cl(\vec{x^t})$" (en supposant ici la transitivité de ces proportions analogiques). Cela peut être (informellement) écrit comme

$$\vec{x^1} : cl(\vec{x^1}) :: \vec{x^2} : cl(\vec{x^2}) :: \cdots :: \vec{x^t} : cl(\vec{x^t})$$

Il est important de noter que ce point de vue correspond exactement à l'idée que dans un problème de classification il existe une *fonction* de classification *inconnue* qui associe une classe unique avec chaque objet, et dont l'ensemble d'apprentissage fournit des valeurs. En effet $\vec{x^k} : cl(\vec{x^k}) :: \vec{x^k} : cl'(\vec{x^k})$ avec $cl(\vec{x^k}) \neq cl'(\vec{x^k})$ est interdit, car ce ne peut être une proportion analogique. Cependant, dans la pratique, nous permettons un TS bruité, c'est-à-dire des exemples $\vec{x^k}$ avec une classe $cl(\vec{x^k})$ erronée. L'ensemble d'apprentissage n'est pas seulement considéré ici comme un simple ensemble de paires $(\vec{x}, cl(\vec{x}))$, mais aussi comme un ensemble de proportions analogiques valides $\vec{x} : cl(\vec{x}) :: \vec{y} : cl(\vec{y})$. C'est la pierre angulaire de l'approche présentée dans ce qui suit.

Une idée similaire a récemment été appliquée avec succès à la résolution des tests de QI de matrices progressives de Raven [7, 6]. Dans ce type de tests on doit compléter une matrice 3×3 où le contenu de la 9^e case est manquant. Le point de départ de la démarche a été alors de supposer que

$$(cell(1,1), cell(1,2)) : cell(1,3) :: (cell(2,1), cell(2,2)) : cell(2,3) :: (cell(3,1), cell(3,2)) : cell(3,3)$$

où $cell(i, j)$ est la représentation, par un vecteur de valeurs d'attributs, du contenu de la cellule (i, j), le contenu de $cell(3, 3)$ étant inconnu. L'expression formelle ci-dessus indique que la paire des deux premières cellules de la ligne

1. Il faut cependant noter que certaines proportions éventuellement connues comme tenant entre des valeurs d'attribut (comme par exemple " petit : moyen : : moyen : grand "), ne correspondent à aucun des trois modèles, et en tant que telles ne seront pas reconnues.

	$\mathcal{A}_1 ... \mathcal{A}_{i-1}$	$\mathcal{A}_i .. \mathcal{A}_{j-1}$	$\mathcal{A}_j .. \mathcal{A}_{k-1}$	$\mathcal{A}_k .. \mathcal{A}_{r-1}$	$\mathcal{A}_r .. \mathcal{A}_{s-1}$	$\mathcal{A}_s .. \mathcal{A}_n$	cl
$\vec{x^i}$	1 ... 1	0 ... 0	1 ... 1	0 ... 0	1 ... 1	0 ... 0	$cl(\vec{x^i})$
$\vec{x^j}$	1 ... 1	0 ... 0	1 ... 1	0 ... 0	0 ... 0	1 ... 1	$cl(\vec{x^j})$
$\vec{x^k}$	1 ... 1	0 ... 0	0 ... 0	1 ... 1	1 ... 1	0 ... 0	$cl(\vec{x^k})$
$\vec{x^0}$	1 ... 1	0 ... 0	0 ... 0	1 ... 1	0 ... 0	1 ... 1	$cl(\vec{x^0})$

TABLE 2 – Appariement de deux paires

1 est à la troisième cellule de la ligne 1 ce que la paire des deux premières cellules de la ligne 2 est à la troisième cellule de la ligne 2, ce que la paire des deux premières cellules de la ligne 3 est à la troisième cellule de la ligne 3 (des proportions analogiques similaires sont également supposés pour les colonnes). Ici, la fonction cachée f est telle que $cell(i,3) = f(cell(i,1), cell(i,2))$ et sa sortie n'est pas une classe, mais une description de la cellule. Comme cette approche a permis de résoudre les tests de QI de Raven, l'idée de ce papier est de l'appliquer à la classification.

Par application de la permutation centrale, nous sommes amenés à réécrire la proportion analogique $\vec{x^i} : cl(\vec{x^i}) :: \vec{x^j} : cl(\vec{x^j})$ reliant les exemples $\vec{x^i}$ et $\vec{x^j}$ sous la forme

$$\vec{x^i} : \vec{x^j} :: cl(\vec{x^i}) : cl(\vec{x^j})$$

Cela suggère une nouvelle lecture de l'ensemble d'apprentissage, sur la base de paires. A savoir : la façon dont les vecteurs $\vec{x^i}$ et $\vec{x^j}$ sont semblables / dissemblables devrait être liée à l'identité ou la différence des classes $cl(\vec{x^i})$ et $cl(\vec{x^j})$. Etant donné une paire de vecteurs $\vec{x^i}$ et $\vec{x^j}$, on peut calculer l'ensemble des attributs $A(\vec{x^i}, \vec{x^j})$ où ils sont en accord (c'est-à-dire qu'ils ont des valeurs d'attributs identiques) et l'ensemble des attributs $D(\vec{x^i}, \vec{x^j})$ où ils sont en désaccord (c'est-à-dire qu'ils ont des valeurs d'attribut différentes). Ainsi, dans la table 2, après une réorganisation appropriée de l'ordre des attributs, les vecteurs $\vec{x^i}$ et $\vec{x^j}$ sont en accord sur les attributs de $\mathcal{A}_1$ à $\mathcal{A}_{r-1}$ et sont en désaccord sur les attributs de $\mathcal{A}_r$ à $\mathcal{A}_n$. Considérons maintenant l'exemple $\vec{x^0}$ (voir la table 2) pour lequel nous voulons prédire la classe. Supposons que nous ayons dans l'ensemble d'apprentissage TS, à la fois la paire $(\vec{x^i}, \vec{x^j})$, et l'exemple $\vec{x^k}$ qui, une fois apparié avec $\vec{x^0}$ a exactement le *même ensemble de désaccord* $D(\vec{x^i}, \vec{x^j})$ que $(\vec{x^i}, \vec{x^j})$, et de plus *avec les changements orientés de la même manière*. Puis, comme on peut le voir dans la table 2, nous avons composante par composante une proportion analogique parfaite, entre les quatre vecteurs. Notez que même si $A(\vec{x^i}, \vec{x^j}) = A(\vec{x^k}, \vec{x^0})$, les quatre vecteurs ne sont pas partout égaux sur ce sous-ensemble d'attributs. En outre, le point de vue ci-dessus s'étend directement des attributs à valeurs binaires aux attributs à valeur dans des domaines finis.

Ainsi, en travaillant de cette manière avec des paires, on peut reconstituer implicitement des 4-uplets de vecteurs qui forment une proportion analogique comme dans la méthode de classification basée sur la recherche systématique de triplets. Avant de développer cette approche à base de paires d'un point de vue algorithmique, nous rappelons les approches existantes qui sont basées sur une recherche par force brute de triplets d'exemples.

4 Classifieurs analogiques existants basés sur la recherche de triplets

Les proportions numériques sont étroitement liées aux notions d'extrapolation et de régression linéaire, c'est-à-dire, à l'idée de prédire une nouvelle valeur sur la base de valeurs existantes. La résolution d'équations analogiques rappelée ci-dessus est à la base d'une méthode de recherche par force brute pour la classification. Elle est basée sur une sorte de principe de continuité proportionnelle : si les attributs à valeur binaire de 4 objets sont en proportion analogique composante par composante, alors cela devrait encore être le cas pour leurs classes. Plus précisément, étant donné un problème de classification binaire, 4 objets booléens $\vec{a}, \vec{b}, \vec{c}, \vec{d}$, 3 dans l'ensemble d'apprentissage avec les classes connues $cl(\vec{a}), cl(\vec{b}), cl(\vec{c})$, le 4^e étant l'objet à classer dans l'une des classes ($cl(\vec{d})$ est inconnu), ce principe peut être énoncé comme suit :

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

Ensuite, si l'équation $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ est résoluble, nous pouvons affecter sa solution à $cl(\vec{d})$. Il est à noter que cela peut s'appliquer à la fois dans le cas où les attributs et les classes sont booléens et dans le cas où les valeurs d'attributs et les classes sont à valeur dans des domaines discrets. Ce principe a conduit à plusieurs implémentations (dans le cas booléen).

Avant de présenter les classifieurs analogiques existants, rappelons d'abord le problème de classification. Nous considérons un univers $\mathcal{U}$ où chaque objet est représenté comme un vecteur de n valeurs d'attributs $\vec{x} = (x_1, ..., x_i, ..., x_n)$ appartenant à un produit cartésien $T = \Pi_{i \in [1,n]} X_i$. Lorsque l'on considère uniquement des attributs binaires, cela signifie simplement que $T = \mathbb{B}^n$. Chaque vecteur $\vec{x}$ est supposé faire partie d'un groupe de vecteurs associés avec une unique classe $cl(\vec{x}) \in C = \{c_1, ..., c_K\}$, où K est fini et il n'y a pas de classe vide dans l'ensemble de données. Si nous supposons que cl est connue sur un sous-ensemble $TS \subset T$ (TS est l'ensemble d'apprentissage pour une configuration donnée), étant donné un nouveau vecteur $\vec{y} = (y_1, ..., y_i, ..., y_n) \notin TS$, le problème de classification consiste à attribuer une valeur plausible à $cl(\vec{y})$ sur la base des exemples stockés dans TS.

L'apprentissage par analogie, décrit dans [2], est une technique "paresseuse" d'apprentissage qui utilise une mesure de *dissemblance analogique* entre 4 objets. Elle estime dans quelle mesure 4 objets sont en proportion analogique. La dissemblance analogique $\mathtt{ad}$ entre 4 valeurs booléennes correspond au nombre minimum de bits qui doivent être changés pour obtenir une proportion analogique satisfaite. Ainsi $\mathtt{ad}(1,0,1,0) = 0, \mathtt{ad}(1,0,1,1) = 1$ et $\mathtt{ad}(1,0,0,1) = 2$. Donc, $a : b :: c : d$ est vraie si et seulement si $\mathtt{ad}(a,b,c,d) = 0$. De plus ad différencie deux cas où l'analogie ne tient pas, à savoir les 8 cas avec un nombre impair de 0 et un nombre impair de 1 parmi les 4 valeurs booléennes, tels que $\mathtt{ad}(0,0,0,1) = 1$ ou $\mathtt{ad}(0,1,1,1) = 1$, et les deux cas $\mathtt{ad}(0,1,1,0) = \mathtt{ad}(1,0,0,1) = 2$. Pour généraliser la dissemblance analogique à des vecteurs booléens de $\mathbb{B}^n$, on ajoute les évaluations ad composante par composante, et on obtient ainsi un entier AD appartenant à l'intervalle $[0, 2n]$. Ce nombre estime dans quelle mesure quatre vecteurs sont, composante par composante, en analogie complète. Il est utilisé par [2] dans la mise en œuvre d'un algorithme de classification où les paramètres d'entrée sont un ensemble TS d'éléments classifiés, un entier k, et un nouvel élément $\vec{d}$ à classer. L'algorithme procède de la manière suivante :

Etape 1 : Calculer la dissemblance analogique AD entre $\vec{d}$ et tous les triplets de TS^3 qui produisent une solution pour la classe de $\vec{d}$.

Etape 2 : Trier ces n triplets selon les valeurs croissantes de AD.

Etape 3 : Soit p la valeur de AD pour le k-ème triplet ; trouver alors k' comme étant le plus grand nombre entier tel que le k'-ième triplet a la valeur p.

Etape 4 : Résoudre les k' équations analogiques sur l'étiquette de la classe. Prendre le vainqueur des k' votes et affecter ce gagnant à $cl(\vec{d})$.

Cette approche fournit des résultats remarquables et, dans plusieurs cas, surpasse les algorithmes les plus connus [16]. D'autres options de classifieurs analogiques ont été testées qui n'utilisent pas la mesure de dissemblance AD, mais qui consistent à appliquer directement le principe de continuité précédent, tout en ajoutant de la flexibilité en permettant d'avoir des composantes où l'analogie ne tient pas. Une première option [19, 23] est de considérer les triplets $(\vec{a}, \vec{b}, \vec{c})$ de telle sorte que l'équation aux classes $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ est résoluble et de telle sorte que $card(\{i \in [1,n] | \ a_i : b_i :: c_i : d_i \text{ est vrai } \})$ est maximal. Ensuite, nous attribuons à $\vec{d}$ la solution de l'équation aux classes correspondante. Une deuxième option [18] est de fixer un nombre entier p qui indique le nombre de composantes pour lesquelles nous tolérons que la proportion analogique ne tienne pas. Dans ce cas, les votants sont les triplets $(\vec{a}, \vec{b}, \vec{c})$ tels que l'équation aux classes $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ est résoluble et tels que $card(\{i \in [1,n] | \ a_i : b_i :: c_i : d_i \text{ holds}\} \geq n-p)$. Dans les deux options [23, 18], un vote majoritaire est appliqué. La principale nouveauté par rapport à l'approche précédente [2] est que nous ne distinguons pas les deux cas où l'analogie ne tient pas. En termes de précision, il n'y a pas de différences significatives entre ces trois algorithmes.

5 Utilisation distincte des similarités et des dissimilarités : Une approche basée sur les paires

Dans cette section, nous étudions une nouvelle approche de la classification analogique, conceptuellement plus simple que les approches précédentes, en ce qu'elle sépare les traitements de la similarité et de la dissimilarité, et calculatoirement moins onéreuse car conduisant à ne considérer qu'un nombre réduit de triplets. L'idée principale est d'abord de rechercher les éléments de TS qui sont les plus proches du nouvel item $\vec{d}$ à classer, puis de rechercher les paires de TS^2 qui présentent la même dissimilarité entre les vecteurs constituants la paire que celle entre $\vec{d}$ et le plus proche voisin considéré. Nous construisons ainsi les triplets utilisés pour la prédiction finale. Détaillons l'idée.

Etape 1 : Utilisation de la similarité. Nous recherchons les exemples les plus similaires au $\vec{d}$. Soit $\vec{c}$ un proche voisin de $\vec{d}$ dans TS. Si $\vec{c} = \vec{d}$, (i.e. $\vec{d}$ appartient à TS), sa classe est connue et il n'y a rien à faire. Dans le cas contraire, $\vec{c} \neq \vec{d}$, et $\vec{c}$ diffère de $\vec{d}$ sur p valeurs (c'est à dire la distance de Hamming entre $\vec{c}$ et $\vec{d}$, $H(\vec{c}, \vec{d})$ est égale à p). Puisque $\vec{c}$ est un plus proche voisin de $\vec{d}$ par rapport à H, il n' y a pas d'exemple $\vec{c'}$ différant de $\vec{d}$ sur moins de p valeurs. Contrairement à un algorithme 1-NN, nous n'attribuons pas la classe de $\vec{c}$ à $\vec{d}$: c'est là où l'apprentissage analogique diffère de la famille des algorithmes k-NN.

Etape 2 : Utilisation de la dissimilarité. Maintenant, nous cherchons dans TS^2 une paire $(\vec{a}, \vec{b})$ d'exemples qui diffèrent de la même manière que $\vec{c}$ et $\vec{d}$ sont dissemblables. De plus $\vec{a}$ et $\vec{b}$ doivent être identiques sur les mêmes attributs où $\vec{c}$ et $\vec{d}$ sont identiques. Par exemple, si $\vec{c} = (1,0,1,1,1,0), \vec{d} = (0,1,1,0,1,0)$, $\vec{a}$ et $\vec{b}$ doivent être similaires (c'est à dire identiques) sur les attributs 3, 5 et 6 et différer sur les attributs 1, 2 et 4, exactement comme $\vec{c}$ et $\vec{d}$ sont dissemblables. Dans ce cas particulier, la paire $\vec{a} = (1,0,0,1,0,0), \vec{b} = (0,1,0,0,0,0)$ satisfait l'exigence précédente. De toute évidence, $\vec{a} : \vec{b} :: \vec{c} : \vec{d}$ est alors une proportion analogique valide.

Si l'équation aux classes $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ a une solution, nous allouons à $\vec{d}$ la solution de cette équation. Sinon, nous cherchons une autre paire $(\vec{a}, \vec{b})$. Evidemment, dans le cas où nous avons plus d'un plus proche voisin $\vec{c}$, ou plusieurs paires $(\vec{a}, \vec{b})$, ce qui peut conduire à des étiquettes différentes pour $\vec{d}$, nous mettons en œuvre un vote à la majorité pour l'étiquette finale de $\vec{d}$. Dans le cas où il existe des ex-aequos, c'est-à-dire que plusieurs classes ont le même nombre maximal de voix, nous considérons l'ensemble des voisins $\vec{c}$ à la distance $p+1$, et nous répétons la même procédure restreinte au sous-ensemble des classes gagnantes. Nous réitérons le processus jusqu'à avoir un gagnant unique.

Formalisons maintenant ces idées dans le cas booléen (l'extension au cas discret est facile). Etant donné deux vecteurs distincts $\vec{x}$ et $\vec{y}$ de $\mathbb{B}^n$, ils définissent la partition suivante de $[1,n]$

– $A(\vec{x},\vec{y}) = \{i \in [1,n] | \ x_i = y_i\}$ qui prend en compte les similarités,
– $D(\vec{x},\vec{y}) = \{i \in [1,n] | \ x_i \neq y_i\} = [1,n] \setminus A(\vec{x},\vec{y})$ qui prend en compte les différences. Le cardinal de $D(\vec{x},\vec{y})$ est exactement la distance de Hamming $H(\vec{x},\vec{y})$ entre les deux vecteurs.

Soit $J \subseteq [1,n]$, notons $\vec{x}|_J$ le sous-vecteur de $\vec{x}$ constitué des $x_j, j \in J$. De toute évidence, $\vec{x}|_{A(\vec{x},\vec{y})} = \vec{y}|_{A(\vec{x},\vec{y})}$ et, dans le cas binaire, si nous connaissons $\vec{x}|_{D(\vec{x},\vec{y})}$, nous pouvons calculer $\vec{y}|_{D(\vec{x},\vec{y})}$. La paire $(\vec{x},\vec{y})$ nous permet de construire un *schéma de désaccord* $Dis(\vec{x},\vec{y})$ comme une liste de paires $(valeur, indice)$ $(valeur \in \mathbb{B}^n, indice \in D(\vec{x},\vec{y}))$ où les deux vecteurs diffèrent. Dans le cas binaire, avec $n = 6, \vec{x} = (1,0,1,1,0,0), \vec{y} = (1,1,1,0,1,0)$, $Dis(\vec{x},\vec{y}) = (0_2, 1_4, 0_5)$. Il est clair que d'avoir un schéma de désaccord $Dis(\vec{x},\vec{y})$ et un vecteur $\vec{x}$ (resp. $\vec{y}$), on peut obtenir $\vec{y}$ (resp. $\vec{x}$). De la même façon, le schéma de désaccord $Dis(\vec{y},\vec{x})$ est déductible de $Dis(\vec{x},\vec{y})$. Pour l'exemple précédent, $Dis(\vec{y},\vec{x}) = (1_2, 0_4, 1_5)$.

Algorithm 1 Classifieur analogique

Input : $\vec{d} \notin TS$ une nouvelle instance à classer
$r = 1; classified = false$;
while $not(classified)$ AND $(r \leq n)$ **do**
  build up the set $\mathcal{B}_H(\vec{d},r) = \{\vec{c} \in TS | H(\vec{c},\vec{d}) = r\}$
  if $\mathcal{B}_H(\vec{d},r) \neq \emptyset$ **then**
    for chaque étiquette l **do** $CandidateVote(l) = 0$
    end for
    for chaque $\vec{c}$ dans $\mathcal{B}_H(\vec{d},r)$ **do**
      for chaque paire $(\vec{a},\vec{b})$ telle que : $Dis(\vec{a},\vec{b}) = Dis(\vec{c},\vec{d})$ AND $cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x$ a pour solution l **do**
        $CandidateVote(l)++$
      end for
    end for
    $maxi = max\{CandidateVote(l)\}$
    if $maxi \neq 0$ AND $unique(maxi, CandidateVote(l))$ **then**
      $cl(\vec{d}) = argmax_l\{CandidateVote(l)\}$
      $classified = true$
    end if
  end if
  $r++$
end while
if $classified$ **then**
  return $(cl(\vec{d}))$
else
  return $(not\ classified)$)
end if

L'explication précédente peut maintenant être décrite avec le pseudo-code de l'algorithme 1, où la fonction $unique(value, Array)$ renvoie une valeur booléenne $true$ si et seulement si la valeur $value$ apparait exactement une fois dans le tableau $Array$, $false$ sinon. $\mathcal{B}_H(\vec{d},r) = \{\vec{c} \in TS \mid H(\vec{d},\vec{c}) = r\}$ dénote la boule dans TS de centre $\vec{d}$ et de rayon r par rapport à la distance de Hamming H. L'algorithme s'arrête dès que nous avons une étiquette pour $\vec{d}$, et nous n'explorons pas d'autres boules avec un plus grand rayon où nous pourrions trouver d'autres options. Notez que ces boules ne se chevauchent pas par construction. Si nous ne trouvons pas d'étiquette pour $\vec{d}$, l'algorithme s'arrête lorsque $r = n+1$ et retourne $not\ classified$. Nous ne pouvons pas classer dans le cas où il s'agit d'un vote 50/50 : en pratique, cette situation est extrêmement peu probable et n'a pas eu lieu pour les bases que nous avons traitées.

Nous devons noter qu'un pré-traitement hors ligne peut être fait sur l'ensemble d'apprentissage TS : pour chaque $r \in [1,n]$, nous pouvons construire le sous-ensemble de $TS \times TS$ constitué par les paires $(\vec{a},\vec{b})$ tel que $H(\vec{a},\vec{b}) = r$: cela accélérera le processus en ligne dans la boucle interne FOR. On peut même envisager de limiter ce pré-traitement à un sous-ensemble de TS.

A cause des deux boucles FOR imbriquées, il est clair que la complexité dans le pire cas est cubique (comme pour les autres algorithmes basés sur l'analogie) puisque l'espace de recherche est juste $TS \times TS \times TS$. Néanmoins, en moyenne, nous allons trouver une solution dans la boule $\mathcal{B}_H(\vec{d},r)$ avec $r < n$, ce qui permet ainsi un élagage d'une grande partie de l'espace de recherche.

Il est important de remarquer que la cardinalité de la boule de Hamming dépend de la représentation des données. Illustrons le problème avec seulement 2 attributs a_1 et a_2 où a_1 est binaire et a_2 a 3 valeurs distinctes possibles. Etant donné deux exemples distincts $\vec{d} = (1, v1)$ et $\vec{d'} = (1, v2)$ respectivement représenté par $\vec{d} = (1,1,0,0)$ et $\vec{d'} = (1,0,1,0)$, après binarisation. Avec le codage discret, $H(\vec{d},\vec{d'}) = 1$, soit $\vec{d'} \in \mathcal{B}_H(\vec{d},1)$, mais avec le codage binaire, $H(\vec{d},\vec{d'}) = H((1,1,0,0),(1,0,1,0)) = 2$, soit $\vec{d'} \notin \mathcal{B}_H(\vec{d},1)$. Le nombre de votants ne sera donc pas le même, et nous pouvons anticiper des résultats différents selon le procédé de codage.

6 Expérimentations

Cette section fournit les résultats expérimentaux obtenus avec notre nouvel algorithme. L'étude expérimentale est basée sur six ensembles de données provenant du site de UCI [15].

- `Balance` et `Car` sont des bases de données avec des classes multiples.

- `TicTacToe`, `Monk1`, `Monk2` et `Monk3` sont des bases de données associées à des problèmes de classification binaire. `Monk3` a du bruit ajouté (dans l'ensemble

TABLE 3 – Description des benchmarks

Bases de données	Nb instances	Att. Nominaux	Att. Binaires	Classes
Balance	625	4	20	3
Car	743	7	21	4
TicTacToe	958	9	27	2
Monk1	432	6	15	2
Monk2	432	6	15	2
Monk3	432	6	15	2

TABLE 4 – Résultats (moyenne et écart type)

	non binarisé			binarisé		
	r=1	r=2	r=3	r=1	r=2	r=3
Balance	86 $\pm$ 4	**88** $\pm$ 2	72 $\pm$ 5	84 $\pm$ 4	87 $\pm$ 3	74 $\pm$ 5
Car	**95** $\pm$ 3	89 $\pm$ 3	72 $\pm$ 6	95 $\pm$ 3	94 $\pm$ 5	77 $\pm$ 6
TicTacToe	**98** $\pm$ 5	96 $\pm$ 5	98 $\pm$ 5	98 $\pm$ 5	97 $\pm$ 5	98 $\pm$ 5
Monk1	**99** $\pm$ 1	99 $\pm$ 1	90 $\pm$ 4	99 $\pm$ 1	99 $\pm$ 1	99 $\pm$ 1
Monk2	**99** $\pm$ 1	97 $\pm$ 3	91 $\pm$ 5	60 $\pm$ 7	99 $\pm$ 1	94 $\pm$ 5
Monk3	**99** $\pm$ 1	97 $\pm$ 2	91 $\pm$ 5	99 $\pm$ 1	99 $\pm$ 1	98 $\pm$ 2

d'apprentissage uniquement).

Une brève description de ces ensembles de données est fournie dans la Table 3.

La Table 4 fournit les résultats de la précision de l'algorithme proposé obtenue par validation croisée 10 fois pour ces six ensembles de données. Les meilleurs résultats sont en gras. Plusieurs commentaires s'imposent :

– Dans la Table 4, il est clair que les résultats ne sont pas sensibles au codage (à l'exception de Monk2) : quelle que soit la façon dont on code les données, en gardant les valeurs discrètes ou en passant à un code binaire, nous obtenons la même précision (la différence n'est pas statistiquement significative). En ce qui concerne le cas de Monk2, il est bien connu que la fonction sous-jacente ("avoir exactement deux attributs valant 1") est plus compliquée que les fonctions sous-jacentes à Monk1 et Monk3, et implique tous les attributs (alors que dans les deux autres cas les fonctions utilisent seulement 3 attributs parmi 6). Nous pouvons donc conjecturer que plus d'exemples sont nécessaires pour prédire la fonction relative à Monk2, et que dans le cas *binarisé*, la boule de Hamming de rayon 1 ne contient pas suffisamment d'information (il a été vérifié que dans ce cas, la boule ne contient que 64 éléments).
– Généralement, les meilleurs résultats sont obtenus pour les petites valeurs de r (souvent $r = 1$), au moins dans le cas discret, puisque dans le cas binaire, on observe sur certains jeux de données qu'un très bon résultat peut être obtenu avec une valeur plus élevée de r dans le cas d'un codage binaire, ce qui est cohérent avec l'analyse ci-dessus.
– D'un point de vue pratique, ces résultats suggèrent que les paires qui ne diffèrent que sur un attribut sont suffisantes pour classer de nouveaux vecteurs $\vec{d}$: nous avons alors simplement à explorer $\mathcal{B}_H(\vec{d}, 1)$ pour associer une étiquette à $\vec{d}$. Cette règle empirique permet de réduire considérablement la taille de l'espace de recherche et la complexité moyenne de l'algorithme, mais une expérimentation plus large est nécessaire pour valider cette observation. Cette valeur de r est également susceptible d'être liée à la taille de l'ensemble d'apprentissage (un trop petit ensemble d'apprentissage pourrait conduire à une boule vide pour $r = 1$).
– Notons aussi que, pour les 6 ensembles de données considérés, nous n'avons pas eu d'items non classés. Cela signifie qu'en pratique, il existe toujours une boule $\mathcal{B}_H(\vec{d}, r)$ où le vote majoritaire s'applique correctement.

TABLE 5 –Résultats obtenus par des algorithmes classiques

Datasets	**SVM**	**IBk(k=1, k=10)**	**JRip**	**C4.5**	**WAPC**
Balance	90	84, 84	72	64	86
Car	92	92, 92	88	90	n/a
Tic tac toe	98	99, 99	98	85	n/a
Monk1	75	99,96	94	96	98
Monk2	67	60, 63	66	67	100
Monk3	100	99, 98	99	100	96

Il est intéressant d'estimer le nombre de paires $(\vec{a}, \vec{b})$ qui sont construites à partir de l'ensemble d'apprentissage (à l'étape de pré-traitement) qui diffèrent sur r attributs pour $1 \leq r \leq n$. Prenons le cas de `Monk2` (avec un codage discret et avec TS égal à 90% de l'ensemble des instances). Le nombre de paires $(\vec{a}, \vec{b})$ est alors :

$r = 1 : 3946$; $r = 2 : 17346$; $r = 3 : 39630$; $r = 4 : 49580$; $r = 5 : 32068$; $r = 6 : 8362$.

La somme de ces nombres est égale au nombre de paires que l'on peut construire à partir de l'ensemble TS.
Comme on peut s'y attendre, ce nombre augmente d'abord avec r et diminue ensuite. Quand on prend un item $\vec{d}$, et un voisin $\vec{c} \in \mathcal{B}_H(\vec{d}, r)$ le nombre de votants $(\vec{a}, \vec{b})$ est un petit sous-ensemble de l'ensemble des couples qui diffèrent sur r attributs, en raison du fait que deux contraintes doivent être remplies : les paires $(\vec{a}, \vec{b})$ et $(\vec{c}, \vec{d})$ doivent différer sur le(s) même(s) attribut(s) et l'équation aux classes associée doit être résoluble. Par exemple, avec `Monk2` (codage discret), le nombre moyen d'électeurs pour $r = 1$ est de 220, pour $r = 2$ de 350, pour $r = 3$ de 270.

Afin de comparer les classifieurs analogiques avec les classifieurs existants, la Table 5 reprend les résultats sur les mêmes ensembles de données que certains algorithmes d'apprentissage bien connus : SVM, k-plus proches voisins IBk (pour k = 1, k = 10), JRip un classifieur construisant des règles propositionnelles de manière optimisée, C4.5 (arbre de décision), et enfin WAPC, le classificateur analogique pondéré (qui utilise la dissemblance analogique présentée dans [16]). Les résultats pour SVM, IBk, JRip et C4.5 sont obtenus en utilisant Weka.

Nous pouvons tirer les conclusions suivantes de cette comparaison :

– Comme attendu, nos résultats sont très proches de ceux obtenus par l'autre approche analogique WAPC.
– Cependant, en contraste avec WAPC, nous n'utilisons pas de poids pour optimiser l'algorithme de base.
– Notons que, avec notre nouvelle approche, si l'on compare avec l'algorithme WAPC utilisant la dissemblance analogique AD, nous n'utilisons que les votants avec $\mathtt{ad} = 0$ (analogie parfaite sur tous les attributs). En effet, il a été observé dans [16] que le nombre de triplets $(\vec{a}, \vec{b}, \vec{c})$ tels que $\mathtt{AD}(\vec{a}, \vec{b}, \vec{c}, \vec{d}) = 0$ est souvent grand. Nos résultats montrent qu'il suffit de considérer ces triplets "parfaits" (obtenus comme la combinaison d'un proche voisin $\vec{c}$ et d'une paire $(\vec{a}, \vec{b})$ dans la nouvelle version) pour faire des prédictions précises.
– Comme WAPC, le nouvel algorithme fournit des résultats qui peuvent être favorablement comparés à (et parfois même surpasser) ceux des méthodes classiques (au moins sur ces ensembles de données).

Lorsque la distance de Hamming r est faible, ce qui est généralement le cas, $\vec{a}$ est assez proche de $\vec{b}$, puisque $\vec{c}$ est proche de $\vec{d}$. Néanmoins, il a été vérifié que, en général, ce n'est pas le cas que $\vec{a}$ ou $\vec{b}$ sont proches de $\vec{d}$. Ceci met en évidence le fait que ces classificateurs ne fonctionnent pas dans le voisinage de l'élément à classer, mais plutôt cherchent des éléments d'information loin de l'item cible $\vec{d}$.

Cependant, pour rapprocher les classifieurs analogiques des classifieurs k-NN, nous avons également testé une autre version de l'algorithme proposé qui considère comme candidats votants uniquement les paires $(\vec{a}, \vec{b})$ qui coïncident avec la paire $(\vec{c}, \vec{d})$ sur le plus grand nombre possible d'attributs. Les résultats obtenus pour les ensembles de données (de codage discret) sont les suivants :

r=1 Monk1 : 87±8 ; Monk2 : 64±5 ; Monk3 : 91±6 ; Balance : 78±5 ; Car : 88±2.

r=2 Monk1 : 88±9 ; Monk2 : 58±7 : Monk3 : 95±3 ; Balance : 86±5 ; Car : 80±6.

A partir de ces résultats par rapport à ceux donnés dans le Tableau 4, il est clair que cette nouvelle version est moins précise que celle de départ. Nous pouvons conclure que tenir compte seulement des paires $(\vec{a}, \vec{b})$ égales à $(\vec{c}, \vec{d})$ sur un nombre maximum d'attributs n'est pas très efficace pour la classification. Cela suggère que, ce faisant, nous restons trop proche de l'esprit des classifieurs k-NN pour obtenir le plein bénéfice de l'approche par proportion analogique.

Enfin, nous avons testé une troisième version de la démarche proposée qui est différente de la deuxième version. Au lieu de classer l'exemple $\vec{d}$ en utilisant les paires $(\vec{a}, \vec{b})$ qui sont égales à la paire $(\vec{c}, \vec{d})$ sur un nombre maximum d'attributs, nous classons l'exemple à l'aide de paires $(\vec{a}, \vec{b})$ qui sont égales à $(\vec{c}, \vec{d})$ sur un nombre r *choisi* d'attributs. Ce nombre r est déterminé comme celui pour lequel le nombre de paires $(\vec{a}, \vec{b})$ est maximum, c'est-à-dire

$$r = argmax_{r'}\{card(E_{(\vec{c},\vec{d})} \cap E'_{(\vec{c},\vec{d})}) \mid H(\vec{d}, \vec{c}) = r'\}$$

avec $E_{(\vec{c},\vec{d})} = \{(\vec{a}, \vec{b}) \in TS^2 | Dis(\vec{a}, \vec{b}) = Dis(\vec{c}, \vec{d})\}$ et $E'_{(\vec{c},\vec{d})} = \{(\vec{a},\vec{b}) \in TS^2 | cl(\vec{a}) : cl(\vec{b}) :: cl(\vec{c}) : x \text{ a une solution}\}$

L'expérience montre que cette version donne des résultats très proches de ceux du Tableau 4 calculés avec le premier algorithme.

Ainsi, dans les classifieurs analogiques, contrairement à k-NN, nous traitons des paires d'exemples. En outre, les deux paires qui sont impliquées dans une proportion analogique ne sont pas nécessairement très semblables comme paires, au-delà du fait qu'elles doivent présenter la même différence (généralement sur un petit nombre d'attributs). Du point de vue expérimental, cette façon de procéder semble être assez efficace pour obtenir de bonnes performances.

Notre approche peut apparaître quelque peu similaire aux travaux provenant de la communauté CBR. Néanmoins, notre méthode repose entièrement sur des proportions analogiques impliquant un triplet d'exemples et un élément nouveau à classer. Au lieu de considérer tous les triplets de candidats, nous nous concentrons sur un sous-ensemble de triplets en choisissant l'un des trois éléments du triplet comme voisin le plus proche du nouvel élément. Il n'y a aucune contrainte sur les deux autres éléments du triplet qui peuvent alors se trouver loin du voisinage immédiat de l'objet à classer. Nous ne rangeons pas l'élément entre deux exemples comme dans [14], ni n'utilisons des techniques Bayésiennes comme dans [5]. D'un autre point de vue, l'adaptation est cruciale pour les problèmes CBR et les connaissances d'adaptation peuvent aussi être apprises (voir [10] par exemple), pour être appliquées aux paires (nouvel item, plus proche voisin). Dans notre approche, les proportions analogiques gèrent similarité et dissimilarité simultanément, et réalisent une forme d'adaptation à leur manière, sans nécessiter une étape d'induction. Enfin, il est intéressant de noter qu'une approche similaire pour le traitement des attributs numériques a été étudiée dans [4], conduisant également à des résultats prometteurs.

7 Conclusion

Dans cet article, nous avons présenté une nouvelle façon de concevoir des classifieurs en utilisant des proportions analogiques. Au lieu de parcourir tous les triplets $\vec{a}, \vec{b}, \vec{c}$ pour constituer une proportion valide avec le nouvel élément $\vec{d}$ à classer, nous forçons $\vec{c}$ à être un voisin de $\vec{d}$. Puis, sur la base des différences entre $\vec{c}$ et $\vec{d}$, on cherche les paires $(\vec{a}, \vec{b})$ ayant exactement les mêmes différences. Ces paires, associées à $\vec{c}$ constituent un triplet candidat votant à condition que l'équation aux classes correspondante soit résoluble. Nos premières expériences donnent de très bons résultats sur 6 bases de données de UCI et les classifieurs obtenus bénéficient d'une complexité moyenne inférieure à celle des classifieurs analogiques classiques. Ces résultats restent à confirmer sur des bases plus grandes en termes de nombre d'exemples et de nombre d'attributs.

Comme expliqué plus haut, un relativement petit nombre de votants est utilisé pour classer. Il reste à vérifier si ceci est une propriété générale et si il est possible d'obtenir des résultats précis en se concentrant uniquement sur un nombre encore plus restreint de votants.

Les proportions analogiques ne sont pas seulement un outil de classification, mais, grâce au procédé de résolution d'équation, permettent aussi le calcul d'un quatrième élément à partir de trois autres. Comme nous l'avons vu, ce 4ème élément peut être tout à fait nouveau. Ainsi, alors que les classifieurs comme k-NN focalisent sur le voisinage de l'élément cible, les classifieurs analogiques vont au-delà de ce voisinage, et plutôt que de "copier" ce qui émerge des proches voisins, "s'inspirent" d'informations pertinentes recueillies éventuellement loin d'un voisinage immédiat de l'item à classer. Cette façon de procéder avec des proportions analogiques ouvre la voie à ce qui pourrait être appelé "apprentissage artificiel créatif".

Références

[1] A. Bandura. *Social Learning Theory*. Prentice Hall, 1977.

[2] S. Bayoudh, L. Miclet, and A. Delhay. Learning by analogy : A classification rule for binary and nominal data. *Proc. Inter. Joint Conf. on Artificial Intelligence IJCAI07*, pages 678–683, 2007.

[3] M. Bounhas, H. Prade, and G. Richard. Analogical classification : A new way to deal with examples. In *Proc. 21th Eur. Conf. on Artificial Intelligence, Praha, Aug. 18-22*, 2014.

[4] M. Bounhas, H. Prade, and G. Richard. Analogical classification : Handling numerical data. Technical Report RR–2014-06–FR, Institut de Recherche en Informatique de Toulouse (IRIT), May 2014.

[5] W.w. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn.*, 76(2-3) :211–225, Sep 2009.

[6] W. Correa, H. Prade, and G. Richard. Quand l'intelligence est juste une question de recopie. In *Actes 6ièmes Journées de l'Intelligence Artificielle Fondamentale (JIAF'12), Toulouse, 22-24 Mai*, pages 77–86, 2012.

[7] W. Correa, H. Prade, and G. Richard. When intelligence is just a matter of copying. In *Proc. 20th Eur. Conf. on Artificial Intelligence, Montpellier, Aug. 27-31*, pages 276–281. IOS Press, 2012.

[8] M. Hesse. On defining analogy. *Proceedings of the Aristotelian Society*, 60 :79–100, 1959.

[9] K. J. Holyoak and P. Thagard. *Mental Leaps : Analogy in Creative Thought*. MIT Press, 1995.

[10] J. Jarmulak, S. Craw, and R. Rowe. Using case-base data to learn adaptation knowledge for design. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 1011–1016, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[11] S. E. Kuehne, D. Gentner, and K. D. Forbus. Modeling infant learning via symbolic structural alignment. In *Proc. 22nd Annual Meeting of the Cognitive Science Society*, pages 286–291, 2000.

[12] J. F. Lavallée and P. Langlais. Moranapho : un système multilingue d'analyse morphologique basé sur l'analogie formelle. *TAL*, 52(2) :17–44, 2011.

[13] Y. Lepage. Analogy and formal languages. *Electr. Notes Theor. Comput. Sci.*, 53, 2001.

[14] D. McSherry. Case-based reasoning techniques for estimation. In *IEE Colloquium on Case-Based Reasoning*, pages 6/1–6/4, Feb 1993.

[15] J. Mertz and P.M. Murphy. Uci repository of machine learning databases. *Available at : ftp ://ftp.ics.uci.edu/pub/machine-learning-databases*, 2000.

[16] L. Miclet, S. Bayoudh, and A. Delhay. Analogical dissimilarity : definition, algorithms and two experiments in machine learning. *JAIR, 32*, pages 793–824, 2008.

[17] L. Miclet and H. Prade. Handling analogical proportions in classical logic and fuzzy logics settings. In *Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09),Verona*, pages 638–650. Springer, LNCS 5590, 2009.

[18] R. M. Moraes, L. S. Machado, H. Prade, and G. Richard. Classification based on homogeneous logical proportions. In M. Bramer and M. Petridis, editors, *Proc. of AI-2013, The Thirty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, England, UK,*, pages 53–60. Springer, 2013.

[19] H. Prade and G. Richard. Reasoning with logical proportions. In *Proc. 12th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2010, Toronto, May 9-13, 2010 (F. Z. Lin, U. Sattler, M. Truszczynski, eds.)*, pages 545–555. AAAI Press, 2010.

[20] H. Prade and G. Richard. Homogeneous logical proportions : Their uniqueness and their role in similarity-based prediction. In G. Brewka, T. Eiter, and S. A. McIlraith, editors, *Proc. 13th Int. Conf.*

on Principles of Knowledge Representation and Reasoning (KR'12), Roma, June 10-14, pages 402–412. AAAI Press, 2012.

[21] H. Prade and G. Richard. From analogical proportion to logical proportions. *Logica Universalis*, 7(4) :441–505, 2013.

[22] H. Prade and G. Richard, editors. *Computational Approaches to Analogical Reasoning : Current Trends*, volume 548 of *Studies in Computational Intelligence*. Springer, 2014.

[23] H. Prade, G. Richard, and B. Yao. Enforcing regularity by means of analogy-related proportions-a new approach to classification. *International Journal of Computer Information Systems and Industrial Management Applications*, 4 :648–658, 2012.

[24] N. Stroppa and F. Yvon. Du quatrième de proportion comme principe inductif : une proposition et son application à l'apprentissage de la morphologie. *Traitement Automatique des Langues*, 47(2) :1–27, 2006.