LA SIMILARITE :

UNE NOTION SYMBOLIQUE/NUMERIQUE

Gilles Bisson

IMAG-CNRS, Projet SHERPA
Unité de recherche INRIA Rhône-Alpes
ZIRST, 655 avenue de l'Europe
38330 Saint Martin - France
E-mail : gilles.bisson@imag.fr

Résumé

Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur de similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées. Cette notion de similarité a fait l'objet d'importantes recherches dans des domaines extrêmement divers tels que l'Analyse des Données, la Reconnaissances des Formes, les Sciences Cognitives ou encore l'Apprentissage Symbolique. Conséquence de cette diversité, il est parfois difficile d'appréhender l'ensemble des travaux qui ont été effectués autour de ce thème car ils se distinguent par les buts poursuivis, les langages de représentation utilisés ainsi que par l'approche théorique sous-jacente. Ce chapitre vise d'une part, à présenter quelques-uns des liens qui existent entre ces différents domaines et à montrer aux lecteurs toute la richesse de la notion de similarité au travers d'une large étude bibliographique. D'autre part, il s'agit aussi de montrer l'importance de la notion de similarité dans le cadre d'un apprentissage de type «symbolique/numérique».

1 Introduction

1.1 Qu'est qu'une similarité ?

Dans tous les domaines de l'informatique dans lesquels on désire analyser de manière automatique un ensemble de données (au sens le plus large) il est nécessaire de disposer d'un opérateur capable d'évaluer précisément les ressemblances ou les dissemblances qui existent au sein de ces données. Sur cette base, il devient alors possible d'ordonner les éléments de l'ensemble, de les hiérarchiser ou encore d'en extraire des invariants. Pour qualifier cet opérateur nous utiliserons dans ce chapitre le terme de *fonction de similarité* ou plus simplement celui de *similarité*.

Exprimées sous des formes multiples, des fonctions de similarité sont mises en œuvre dans de nombreux domaines. C'est notamment le cas de l'Analyse des Données (AD), de la Reconnaissances des Formes (RF), de l'Apprentissage Symbolique (AS), ou encore de celui des Sciences Cognitives (SC). Même s'il peut sembler difficile au premier abord d'établir un lien entre les différentes approches, nous allons voir qu'il est possible de donner une définition de ce qu'est une similarité ainsi que des principales tâches qu'elle permet d'effectuer.

De manière générale, une fonction de similarité est définie dans un univers *U* qui peut être modélisé à l'aide d'un quadruplet : (L_d, L_s, *T*, *FS*).

- Soit L_d le langage de représentation utilisé pour décrire les données.
- Soit L_s le langage de représentation des similarités.
- Soit *T* un ensemble de connaissances que l'on possède sur l'univers étudié.
- Soit *FS* la fonction binaire de similarité, telle que :

$$FS : L_d \times L_d \varnothing L_s$$

Lorsque, comme c'est souvent le cas, la fonction de similarité a pour objet de quantifier les ressemblances entre les données, le langage L_S correspond à l'ensemble des valeurs dans l'intervalle [0..1] ou bien à l'ensemble $\leftarrow^+$ et l'on parlera alors dans ce chapitre de *mesure de similarité*. Mais avant de détailler davantage la manière dont les éléments (L_d, L_s, *T*, *FS*) du modèle sont instanciés, nous allons décrire le cadre dans lequel les fonctions de similarité sont couramment utilisées. On peut considérer schématiquement qu'elles interviennent dans trois types de traitement de données à savoir (fig. 1) : la *classification*, *l'identification* et la *caractérisation.*

1.2 Utilisations de la similarité

Le processus de classification[1] vise à structurer les données contenues dans *U*, en fonction de leurs ressemblances, sous la forme d'un ensemble de classes à la fois homogènes et contrastées (fig. 1a). Ce processus a été intensivement étudié en AD où il a donné naissance à deux grandes familles de méthodes : d'une part, les méthodes de partitionnement, dont l'algorithme des nuées dynamiques [19] est un bon représentant, et d'autre part, les méthodes de classification ascendante [44] où les données sont rangées au sein d'une hiérarchie de classes. En AS le problème a été étudié dans le cadre de *l'apprentissage non supervisé* et a conduit à l'élaboration des méthodes dites de classification conceptuelle [54], [37] et de formation de concepts[2] [25], [29]. Par ailleurs, le lien entre classification et similarité est étudié en SC (on parle alors de catégorisation) puisqu'il s'agit d'un processus central dans la cognition humaine (voir entre autres : [75], [53], [35]). Dans tous les cas, le critère de formation des classes consiste, de manière plus ou moins directe, à maximiser la mesure de similarité intra-classes et à minimiser la mesure de similarité inter-classes.

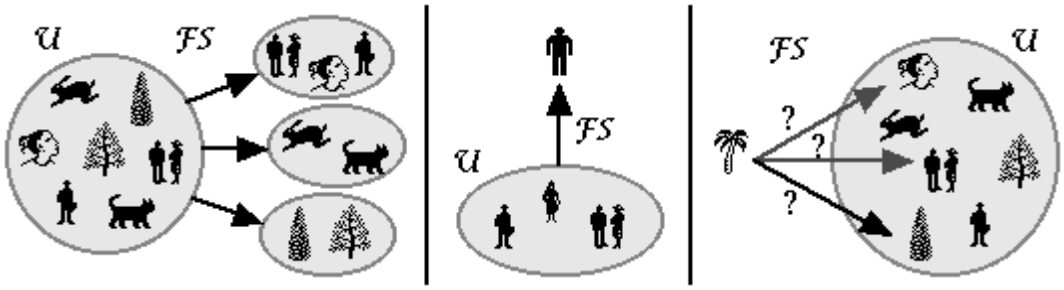


Fig. 1a : Classification *Fig. 1b: Caractérisation* *Fig. 1c: Identification*

Le processus d'identification (fig. 1c) a pour but de connaître la classe à laquelle un objet inconnu est susceptible d'appartenir, ou encore, de trouver à quel(s) objet(s) de *U* il est le plus ressemblant. Dans ce dernier cas, la méthode dite des *plus proches voisins*, initialement introduite en AD, utilise directement la notion de similarité : en effet, l'objet de *U* qui est sélectionné est celui qui maximise la mesure de similarité avec l'instance inconnue. Outre qu'elle soit très simple à implémenter, cette méthode se révèle efficace pour peu que la mesure retenue soit pertinente. Cette approche est également utilisée en AS, sous des formes plus ou moins complexes, dans le cadre de l'apprentissage à partir d'instances (IBL : Instance Based Learning, voir entre autres : [2], [6], [81]) et dans celui du

[1] En pratique, le mot classification est ambigu puisque selon les domaines concernés et la langue utilisée, il s'agit soit de construire des classes d'objets, soit de retrouver la classe à laquelle appartient un objet.

[2] Dans le cas de la formation de concept, on utilise comme critère de classification le «Concept Utility» qui est issus de travaux [33] en SC. Toutefois, ce critère peut être facilement reformulé en terme de similarité, d'ailleurs le système ADECLU [16] est basé sur une mesure de similarité.

raisonnement à partir de cas (CBR : Case Based Reasonning, voir entre autres : [45], [78]). Par ailleurs, le processus d'identification est analysé en SC au travers de l'étude générale de la notion de similarité [75], [36] et de celle d'analogie [31], [39], [27]. Il faut souligner que les mécanismes de classification et d'identification ne sont pas entièrement orthogonaux puisque certains systèmes de classification, notamment en formation de concept, reposent sur un mécanisme d'identification incrémentale. Du point de vue de la similarité, même si les mesures utilisées dans les deux processus sont assez semblables, il existe cependant quelques différences. Ainsi, alors qu'en classification on utilise des mesures «symétriques» (*FS* (x, y) = *FS* (y, x)), cela n'est plus toujours le cas en identification[3] où l'on distingue l'objet à identifier de l'objet de référence.

Enfin, le processus de caractérisation (fig. 1b) permet de construire une représentation explicite des informations qui sont communes à un ensemble de données ; dans ce cas, le langage L_s est souvent un sur-ensemble de L_d ($L_s \in L_d$). Ce processus correspond à l'opération de généralisation[4] qui a fait l'objet de nombreuses recherches [55], [41] en AS et notamment dans le cadre de la programmation logique inductive (voir entre autres : [60], [56], [17]. Ainsi, il devient possible d'associer une *fonction de reconnaissance* symbolique à des classes d'objets produites par classification[5] (apprentissage non-supervisé) ou qui sont fournies directement par l'utilisateur (apprentissage supervisé). L'ensemble des fonctions ainsi apprises constitue une base de connaissances, exprimable par exemple sous la forme de règles d'inférences, qui permet d'identifier la classe d'appartenance de nouveaux objets.

1.3 Une classification des similarités

En examinant schématiquement comment les quatre paramètres (L_d, L_s, *T*, *FS*) de notre modèle de la similatité s'instancient en fonction des différents domaines (voir le tableau 1) on peut définir des critères permettant de classifier les fonctions de similarité en plusieurs familles. Nous allons énoncer rapidement ces critères puis nous les discuterons en détail dans les trois prochaines sections.

Comme nous l'avons déjà évoqué dans la section précédente, les langages de représentation des similarités L_S se scindent en deux familles[6] [63]. D'une part, les

---

[3] Particulièrement dans le cadre du raisonnement analogique ou encore, lorsque l'objet de référence est une description de concept (et non une instance). Par ailleurs, si du point de vue mathématique une similarité est, par définition, symétrique, nous employons ici ce mot dans le sens plus général de «ressemblance».

[4] Notons que peu de travaux présentent la généralisation comme une sorte de fonction de similarité.

[5] l'AD s'intéresse peu au problème de la caractérisation et cette étape est à la charge de l'utilisateur. En classification les classes produites sont définies par la liste des individus qui les composent (extension).

[6] Cette vision est un peu réductrice. Ainsi, dans le cadre d'une étude théorique sur les mesures de similarité utilisée en CBR, Richter s'intéresse à des caractérisations booléennes de la similarité [63]. Par ailleurs, Maher propose une mesure à deux composants [48] : le premier exprimant le degré potentiel de ressemblance entre les données et le second, le degré de certitude que l'on a sur cette ressemblance.

similarités «numériques» qui *quantifient* les ressemblances sous la forme d'une valeur dans ← et qui sont utilisées en AD, RF, AS et SC. D'autre part, les similarités «symboliques» qui permettent de *caractériser* les ressemblances, et qui sont largement utilisées en AS dans le cadre de l'apprentissage de fonctions de reconnaissance de concepts. Bien sûr, la méthode d'évaluation des similarités est différente dans les deux cas : la quantification repose sur des processus de *comptage* alors que la caractérisation repose sur des méthodes de *réécriture*. Dans ce chapitre nous ne nous intéressons qu'aux méthodes de quantification, mais comme nous le verrons, être capable de quantifier une ressemblance peut aider grandement à sa caractérisation.

	AD	**RF**	**AS**	**SC**
L_d	• vecteur	• vecteurs • graphes	• attribut/valeur • ordre 1	• attribut/valeur • ordre 1
L_s	• numérique	• numérique	• numérique • symbolique	• numérique • symbolique
T	• statistique	• statistique	• statistique • symbolique	• symbolique
FS	• comptage	• comptage	• comptage • réécriture	• comptage • réécriture

Tab. 1 : Utilisation de la notion de similarité dans différents domaines

Le second critère de classification des similarités concerne le langage de description des données L_d qui est utilisé. Là également, on peut distinguer deux groupes: les représentations de type propositionnel (vecteurs et attributs) et les représentations de type relationnel (graphes, objets et logique des prédicats). Les premières sont majoritairement utilisées en AD même si certains travaux récents [19] visent à étendre l'expressivité des langages de manière à prendre en compte la notion de relations entre individus. En AS, bien que l'on travaille de plus en plus avec des représentations relationnelles, il existe encore peu de travaux [22], [7], [21] proposant des mesures de similarité numérique capables de traiter ce type de langage et les mesures couramment employées en IBL et CBR s'inspirent largement de celles qui ont été proposées en AS et RF. Enfin, en SC, à la suite des travaux de [30] sur le raisonnement analogique, des mesures de similarités s'appliquant aux représentations relationnelles ont été proposées par [39] et [34].

Enfin, outre les données initiales, on possède souvent des informations supplémentaires *T* sur le contexte dans lequel on travaille ; elles influent à deux niveaux différents. D'une part, en conjonction avec la connaissance des buts poursuivis, ces informations permettent de sélectionner la fonction de similarité la mieux adaptée. D'autre part, ces informations peuvent être exploitées par la fonction de similarité elle-même. Dès lors, on peut distinguées les similarités *non-informées* qui ne travaillent que sur les données, des similarités *informées* qui utilisent explicitement les connaissances présentes dans *T*. Selon les domaines, la

nature de ces connaissances est différente, en AD et RF les informations prises en compte vont être plutôt de nature statistique (modèle de répartition des valeurs) alors qu'en AS et SC on utilisera plutôt des modèles exprimés sous forme symbolique[7] (règles).

1.3.1 Similarités numérique et symbolique

Dans le cadre de l'apprentissage inductif, il est clair que les approches numérique et symbolique de la notion de similarité offrent des caractéristiques qui sont complémentaires. Les mesures de similarités numériques s'avèrent être d'un emploi extrêmement souple. Elles sont capables de travailler sur un large spectre de types de données[8] et il est assez facile d'introduire, lors du calcul, des approximations statistiques si les informations à traiter sont complexes. En outre, la quantification des ressemblances par une valeur continue implique qu'il est toujours possible et facile de comparer des couples d'objets entre eux. Par contre, les résultats fournis par ces mesure restent difficilement explicables[9] même si l'on peut améliorer la compréhension en visualisant les similarités sous la forme de hiérarchies [5] ou de pyramides [13]. Dans certains cas, les résultats peuvent même être trompeurs ; ainsi, lorsque l'on compare des couples de données, des valeurs de similarité identiques entre ces couples peuvent correspondre à des caractérisations (ou généralisation) très différentes. En d'autres termes, le fait que deux couples d'objets (A, B) et (A, C) aient la même similarité n'implique pas que B et C soient identiques. Enfin, notons qu'aujourd'hui il y a relativement peu de mesures de similarité permettant de travailler directement sur des langages relationnels sans passer par une étape de redescription des données. Certes, ce problème est traité en RF, mais comme nous le verrons, les solutions apportées ne sont pas satisfaisantes du point de vue de l'apprentissage.

Les fonctions de similarités symboliques présentent, elles, l'avantage de fournir à l'utilisateur des résultats explicites, puisque les similitudes sont caractérisées dans un langage proche de celui qui est utilisé pour exprimer les données initiales. En outre, il existe des méthodes permettant de travailler directement sur des données relationnelles (voir entre autres : [62], [57]). Cependant, le traitement des valeurs numériques est alors souvent fait de manière peu satisfaisante en redécrivant ces valeurs sous une forme symbolique ; c'est notamment le cas dans les approches s'inspirant de la logique. Enfin, ainsi que nous l'avons dit précédemment, la recherche des ressemblances est fondée sur l'opération de généralisation, qui s'avère être relativement complexe dans le cas des données relationnelles[10].

---

[7] A ce sujet, certains travaux [70] mesurent la proximité de deux exemples en comptant le nombre de règles qu'ils permettent de déclencher simultanément dans une base de connaissances.

[8] Comme le souligne [43], la distinction entre apprentissage «symbolique» et «numérique» n'est pas le simple reflet du type de donnée traitée. On trouvera une discussion complète sur ce sujet dans [11].

[9] Nous ne parlons pas ici de l'explicabilité mathématique, mais de l'intelligibilité du résultat. La phase d'interprétation demande généralement une bonne connaissance de la méthode par l'utilisateur.

[10] En AS, ces données sont souvent représentées à l'aide de sous-ensembles de la logique des prédicats.

En fait, la principale différence entre les deux types de similarité réside dans la manière dont on peut comparer les résultats obtenus. Alors que du point de vue numérique il est toujours possible de répondre à la question «X est-il plus similaire à Y qu'à Z ?», cela n'est plus vrai dans le cas symbolique où les comparaisons reposent sur le test booléen de *subsomption* qui permet seulement de décider si une description est plus générale qu'une autre, mais qui n'apporte aucune information sur leur degré de ressemblance. C'est pourquoi les similarités symboliques ne sont donc pas toujours utilisables pour les problèmes d'identification[11] ou de classification.

De ce point de vue, la similarité numérique peut être vue comme une *extension* de la subsomption. Considérons en effet la fonction booléenne SUB (X, Y) qui vérifie si la description X *subsume strictement* la description Y. Considérons maintenant une mesure de similarité SIM (X, Y) qui permet d'évaluer le degré de ressemblance de la description de X vis à vis de celle d'Y. Précisons tout de suite que la mesure de similarité que nous introduisons ici est bien évidemment *asymétrique* puisque Y constitue la référence. Cette fonction va prendre sa valeur dans l'intervalle continu [0..1], la valeur 1 signifiant la parfaite inclusion de la première description dans le seconde. Les deux propriétés suivantes sont toujours vérifiées[12] :

- **SIM (X, Y) = 1** ⇔ **SUB (X, Y) = vrai**
- **SIM (X, Y) ∈ [0..1[** ⇔ **SUB (X, Y) = faux**

Plusieurs systèmes travaillent directement avec des similarités symboliques, mais dès qu'il s'agit d'évaluer la validité des généralisations produites, ils doivent repasser dans un espace purement numérique. Ainsi, dans CIGOL [56] les généralisations sont comparées entre elles sur la base d'une mesure de l'information. Dans l'algorithme de classification CLUSTER/2 [54], pour évaluer les classes qui ont été construites, on compare le cardinal de l'extension des descriptions, c'est à dire le nombre d'exemples potentiel qu'elles permettent de reconnaître.

En conclusion, on voit donc bien les apports respectifs de ces deux types de similarités : souplesse et calculabilité d'une part et intelligibilité de l'autre. Dans la prochaine section de ce chapitre nous allons montrer une approche capable de combiner ces avantages de façon à obtenir une notion de similarité «symbolique-numérique» capable de traiter, avec un coût algorithmique raisonnable, à la fois des données numériques et relationnelles et susceptible de fournir à l'utilisateur des caractérisations symboliques des ressemblances observées.

---

[11] Bien sur, on peut faire de l'identification à partir d'une base de connaissance symbolique (règles ou objets), mais on connait leur manque de robustesse lorsque l'on doit traiter des données bruitées. C'est pourquoi des extensions diverses ont été proposées à la logique classique, telle la logique «floue».

[12] Il n'y a plus équivalence lorsque la mesure de similarité est de nature heuristique. C'est par exemple le cas dans [10] ou le calcul de similarité n'est qu'une approximation pour des raisons de complexité.

1.3.2 Similarités propositionnelle et relationnelle

La distinction qui existe entre les langages propositionnel et relationnel est fondamentale du point de vue des problèmes mis en jeu. En représentation propositionnelle, telle qu'on l'utilise en AS, les données initiales sont exprimées sous la forme d'un ensemble de conjonctions de littéraux attribut-valeurs. Quant à elle, l'AD utilise un format matriciel équivalent où les lignes correspondent aux valeurs et les colonnes aux différents attributs qui sont appelés des variables. Notons que la terminologie utilisée pour désigner une donnée varie également en fonction du domaine : en AD on parlera d'individus, alors qu'en AS on parlera d'exemples (apprentissage supervisé) ou d'observations (apprentissage non-supervisé).

Dans cette représentation, indépendamment de la manière dont la mesure est calculée, lorsque l'on compare deux exemples il suffit en première approximation de comparer deux à deux la valeur des attributs qui les composent, la méthode de comparaison étant dépendante du *type* de la donnée (nominale, ordonnée, ...). Dans l'exemple de la figure 2a, pour calculer la ressemblance entre les objets X et A on examine leurs valeurs respectives de taille, de couleur et de forme.

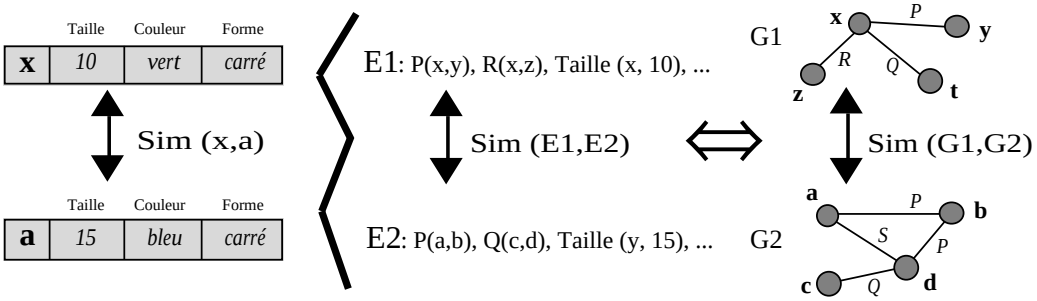


Fig. 2a : Langages propositionnels

Fig. 2b : langages relationnels.

En AS, les données relationnelles sont souvent exprimées sous la forme d'une conjonction de prédicats instanciés. Dans cette représentation un exemple ne décrit plus un objet unique avec la liste de ses attributs, mais plutôt une liste d'objets liés entre eux par des relations et où chaque objet est lui-même décrit par un ensemble d'attributs. Un exemple est donc assimilable à un graphe étiqueté, attribué et éventuellement orienté si les relations entre les objets ne sont pas commutatives. Aussi, pour pouvoir calculer la similarité il faut décider comment les objets vont être comparés entre eux. Par exemple, dans la figure 2b il faut se demander à quel objet de G2 parmi (a, b, c, d) l'objet x de G1 doit-être comparé ? En toute généralité[13], le nombre d'appariement est potentiellement en Θ (N!) s'il y a N objets

[13] Bien sûr, il est inutile de comparer deux objets ne possèdant aucun attribut commun. Par ailleurs, en effectuant un typage des objets on parvient à contraindre efficacement la recherche [58].

dans les deux graphes. Lorsque deux exemples sont décrits à l'aide d'une représentation relationnelle, la mesure de la similarité nécessite donc une nouvelle étape dont le but est de trouver un appariement entre les objets qui *maximise la similarité globale* entre les deux exemples. Une fois le problème d'appariement résolu, on retombe dans un cas de figure identique à celui des représentations propositionnelles et qui est donc aisément traitable. Il faut souligner que ces problèmes d'appariement, loin de n'être qu'un artefact issu des représentations informatiques, sont connus et étudiés en SC sous le nom du problème *d'alignement des structures* [34], [51].

Bien évidemment, on est confronté à la même difficulté lorsque l'on veut généraliser un couple d'exemples ; exprimée dans la terminologie de la théorie des graphes, cette opération se ramène à chercher les plus grands sous-graphes partiels isomorphes entre les deux graphes. Ce problème étant a priori de la classe des problèmes NP complets, toute heuristique permettant de le résoudre est la bienvenue. Or, si l'on peut estimer la similarité entre chaque couple d'objets, il devient possible de traiter le problème d'appariement en un temps polynomial (fig. 3).

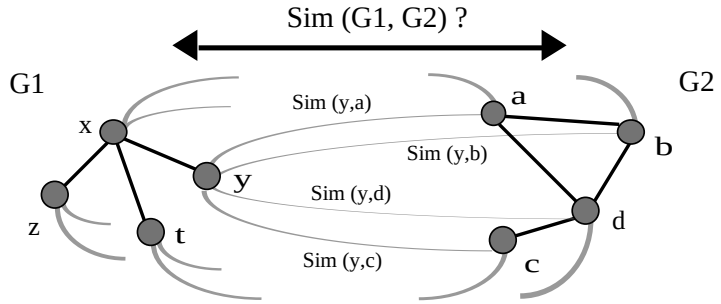


Fig. 3: problème du calcul d'une fonction de similarité entre deux graphes

Le problème est le suivant : on recherche un appariement 1-1 entre les objets de G1 et G2 qui soit «optimal» c'est à dire qui maximise la somme des similarités des objets appariés. D'un point de vue théorique le problème se ramène à un cas connu en théorie des graphes qui est la *recherche du couplage de poids maximum dans un graphe biparti*. Plusieurs algorithmes ont été développés pour résoudre ce problème [4] et on trouvera une bonne synthèse des méthodes existantes dans [12]. Reste évidemment le problème de définir une mesure de similarité capable de mesurer la ressemblance des objets non seulement en terme d'attributs mais également en terme de relations. Plusieurs approches seront présentées dans la section 3.

1.3.3 Similarités informée et non-informée

A l'instar de Marcotorchino [50], on peut établir une distinction entre les similarités non-informées et les similarités informées[14]. Dans le premier cas, le calcul se fait sur une base purement *locale* en ne prenant en compte que les informations qui sont explicitement présentes dans les exemples alors que dans le second cas on utilise en outre des informations d'ordre statistique ou encore symbolique portant sur *l'ensemble* des exemples de l'univers dans lequel on travaille.

Nous allons illustrer notre propos à l'aide de l'exemple présenté dans le tableau 2 et qui est emprunté à Marcotorchino. L'une des manières les plus classique, voire intuitive, de mesurer la similarité entre deux exemples consiste diviser la somme des modalités (valeurs) communes aux deux exemples par le nombre d'attributs qui apparaissent dans ces exemples. Ainsi, la similarité entre A et B est de 2/3 car ils ont deux modalités communes parmi les trois attributs qui les caractérisent.

	Forme	**Percé**	**Couleur**
A : 	carré	non	noir
B : 	carré	non	gris
C : 	rond	non	gris
D : 	rond	oui	gris

Table 2 : Similarités informées et non-informées

Supposons maintenant que l'on connaisse la fréquence d'apparition des différentes modalités dans l'ensemble des exemples. On peut alors définir la similarité de la manière suivante : la ressemblance entre deux modalités du même attribut va être, comme précédemment, égale à 0 si elles sont différentes et va être égale à l'inverse du nombre total d'apparitions de la modalité sur l'ensemble des exemples, si elles sont identiques. Par exemple, la similarité entre A et B est égale à 5/6 car d'une part, les deux objets sont de même forme et la modalité *carré* est présente chez deux individus dans l'ensemble des exemples (1/2), d'autre part, ils sont tous deux non-percés et cette modalité est présente chez trois individus (1/3). Dans ce contexte, la ressemblance entre deux exemples est inversement proportionnelle à la fréquence d'apparition des modalités dans l'univers considéré. On peut justifier ce calcul par le fait qu'une ressemblance est d'autant plus significative qu'elle concerne une caractéristique plutôt rare dans la population. Ces deux modes de calcul conduisent aux matrices de similarités présentée dans les tableaux 3a&b :

A B C D A B C D

[14] Dans son article qui ne concerne que les travaux en AD, Marcotorchino utilise une terminologie un peu différente puisqu'il parle de *similarité logique* (ici informée) et de *similarité statistique* (non informée).

A	1	2/3	1/3	0
B	2/3	1	2/3	1/3
C	1/3	2/3	1	2/3
D	0	1/3	2/3	1

Tab. 3a : Similarités non-informées

A	1	5/6	2/6	0
B	5/6	1	4/6	2/6
C	2/6	4/6	1	5/6
D	0	2/6	5/6	1

Tab 3b : Similarité informées

Ici, la similarité informée (tab. 3b) semble refléter de façon plus fine les caractéristiques du domaine dans lequel on travaille : alors que, sans l'information statistique, les couples (A, B), (B, C) et (C, D) sont considérés comme identiques, ce n'est plus le cas avec la seconde méthode de calcul et les couples (A, B) et (C, D) deviennent les plus similaires. Par ailleurs, quand on examine le comportement de la seconde mesure on constate qu'en pratique elle va avoir implicitement tendance à augmenter l'importance des attributs dont les modalités sont les plus dispersées.

Cette dernière remarque peut être généralisée de la manière suivante. En fait, calculer la similarité totale entre deux exemples se ramène toujours plus ou moins à faire la somme des similarités partielles sur les attributs communs. Dès lors, le moyen le plus immédiat de paramétrer le comportement d'une mesure va être d'associer, de manière explicite ou implicite, un coefficient numérique à chaque attribut, ce qui va permettre de pondérer son influence dans la somme finale. Il faut noter que ce problème de la pondération est tout à fait crucial : il est évident que, dans un problème d'identification ou de classification, si l'échelle des pondérations entre les attributs est mal établie, le résultat final risque fort d'être non pertinent[15]. La pondération des attributs peut être effectuée de manière manuelle lorsque l'utilisateur connait les informations importantes pour le problème considéré. Toutefois de nombreux travaux en AD [18], [28] et en AS [66], [80] visent également à établir ces poids de manière automatique, soit par une étude statistique des exemples (comme c'est le cas ci-dessus), soit en utilisant des connaissances symboliques, soit enfin, en mettant en œuvre un mécanisme d'apprentissage des poids (au sens de l'AS).

Le problème des pondérations est également très étudié en SC car il est relié à la notion de *contexte* dans lequel s'effectue une tâche. Par exemple, Tversky [75] montre que dans les problèmes de catégorisation, les critères de comparaison qui sont utilisés par les êtres humains varient en fonction des catégories d'objets préexistantes. Par ailleurs, au travers de plusieurs expériences, Goldstone [36] illustre l'influence des divers facteurs contextuels (culturels, personnels, ...) sur la perception de la similarité. Enfin, les humains semblent être capables d'apprendre des règles de pondération qui s'appliquent dynamiquement en fonction du contexte [3].

[15] En effet, cela peut revenir à priviligier des attributs non significatifs ou au contraire, à faire disparaitre des informations importantes. On trouvera dans [27] une illustration de ce problème en classification.

1.4 Quelques critères d'analyse

Au terme de cette longue introduction qui nous a permis d'introduire les principaux critères définissant notre domaine d'étude, nous allons décrire au cours des deux prochaines sections quelques unes des mesures de similarité qui sont couramment utilisées ainsi que celles qui présentent un caractère original. Le découpage des sections a été effectué par rapport au langage de représentation des données qu'elles utilisent. La section 2 introduit les similarités dans les langages propositionnels et la section 3 présente les similarités dans les langages relationnels.

Les mesures de similarité sous examinées sous plusieurs aspects complémentaires : mathématique, comportemental et cognitif. L'aspect mathématique, qui est principalement étudié dans le cadre de l'AD, vise à caractériser les propriétés formelles des similarités (symétrie, transitivité, ...). Cette étude permet d'analyser la convergence des algorithmes ou l'applicabilité d'une mesure à un problème donné. Toutefois, la majorité des utilisateurs veulent connaitre, plus prosaïquement, le comportement «effectif» de la mesure sur les données étudiées. En d'autres termes, ils désirent obtenir une réponse concrète aux questions : «Quelle mesure dois-je utiliser et comment dois-je interpréter la valeur numérique obtenue ?». Bien que la réponse à ces questions soit une conséquence logique des propriétés mathématiques de la mesure, il est souvent extrêmement délicat pour un non-spécialiste d'établir le lien entre les deux. Enfin, l'analyse cognitive des mesures des similarités est importante afin de pouvoir juger de la pertinence du calcul : en effet, le résultat d'une mesure sera d'autant mieux perçu et accepté par l'utilisateur que le comportement «apparent» de la mesure de similarité sera isomorphe[16] à celui d'un être humain.

2 Similarités dans le cadre propositionnel

2.1 La notion mathématique de distance

La plupart des travaux concernant les mesures de similarité dans le cadre des représentations propositionnelles ont comme base la notion mathématique de *distance* (notion inverse de la similarité) qui a été intensivement étudiée en AD. Elle se définit de la façon suivante : soit Ω l'ensemble des individus du domaine étudié et soit une métrique d qui est une fonction de $\Omega \times \Omega$ dans $\leftarrow^{+}$.

$\forall$ **a, b, c** $\in \Omega$ **1)** d **(a, a) = 0** (propriété de minimalité)

[16] Ce qui ne veut pas dire qu'une mesure de similarité doit constituer un modèle du raisonnement humain.

2) d **(a, b)**	**= d (b, a)** (propriété de symétrie)

Lorsque la fonction d vérifie les propriétés 1 et 2, on l'appelle un *indice de dissimilarité* (ou plus simplement une *dissimilarité*). D'autres propriétés sont également intéressantes :

$\forall$ **a, b, c** $\in \Omega$		
3) d **(a, b)**	**= 0** $\Rightarrow$ **a = b**	(propriété d'identité)
4) d **(a, c)**	$\leq$ d **(a, b) +** d **(b, c)**	(inégalité triangulaire)
5) d **(a, c)**	$\leq$ **Max [**d **(a, b),** d **(b, c)]**	

Si la fonction d vérifie les propriétés (1, 2 et 3) on parle d'un *indice de distance.* Si cet indice vérifie également la propriété (4) on l'appelle une *distance* et s'il vérifie aussi la propriété (5) on l'appelle une *distance ultramétrique*. Par ailleurs, lorsque la fonction d vérifie les propriétés (1, 2 et 4) on parle d'un *écart*, et lorsqu'il vérifie les propriétés (1, 2 et 5) on parle d'un *écart ultramétrique.* Soulignons enfin que le passage d'un indice de dissimilarité d à la notion duale *d'indice de similarité*, que l'on va noter s, est aisée. Si l'on appelle S_{max} la similarité d'un élément avec lui même, il suffit de poser : d (a, b) = | S_{max} - s (a, b) |

2.2 Quelques mesures courantes

Nous ne parlons dans ce chapitre que des mesures de ressemblances entre exemples. En effet, on peut mesurer également des ressemblances sur des attributs ou encore sur des groupes d'exemples en particulier dans le cadre de la classification. En AD de nombreuses distances ont été proposées telles que le χ^2 (chi 2) ou Mahalanobis (voir entre autres [20], [68]), les plus connues et les plus utilisées[17] étant sans contestation la distance euclidienne ou celle dite «de Manhattan», qui ne sont en fait que des cas particuliers de la mesure de Minkowski :

$$d_p(x,y) = \left(\sum_{i=1}^{K} W_i \times \left| x_i - y_i \right|^p \right)^{\frac{1}{p}}$$

En fonction du paramètre p :
- p = 1 distance de Manhattan
- p = 2 distance euclidienne
- p = ∞ distance de Chebychev (Max $|x_i - y_i|$)

Dans cette formule, les variables x_i et y_i représentent respectivement les valeurs du i^{eme} attribut décrivant les exemples x et y et le terme W_i représente le poids associé à cet attribut. Cette fonction s'applique lorsque les exemples sont décrits par des variables numériques. Il est à noter que ce que nous appellons une *distance euclidienne* correspond à la notion classique de distance entre deux points dans un espace à deux dimensions ; elle est calculée à l'aide du théorème de Pythagore. Ce terme de distance euclidienne est aussi utilisé pour caractériser une autre famille de distances dites *distances quadratiques* dont l'expression est la suivante :

[17] Non seulement en AD mais aussi en AS et plus particulièrement pour le raisonnement par cas (CBR).

$$d(x,y) = \sqrt{{}^t(x_i - y_i) M (x_i - y_i)}$$

En fonction de la matrice M :

- M = I : distance euclidienne classique
- M = V^{-1} : distance de Mahanalobis
- M = D^{-1} : distance du χ^2

Dans cette formule, le terme (x_i-y_i) représente le vecteur colonne des différences entre les attributs des deux exemples x et y. Pour Mahanalobis, la matrive V correspond à la matrice de variance-covariance entre l'ensemble des exemples. Pour le χ^2 qui est une distance plutôt adaptée à la comparaison de profils de distribution de modalités, la matrice D est une matrice diagonale exprimant la fréquence relative d'apparition de chacunes des modalités dans les exemples.

Dans les mesures précédentes, pour un attribut donné, la distance est calculée à partir de la différence | x_i-y_i | entre les modalités. Or, cette manière de procéder risque de fausser le résultat lorsque les attributs décrivant les exemples ont des domaines de valeurs de tailles différentes. Par exemple, dans le cas de Manhattan, les attributs ayant une grande dispersion de valeurs sont implicitement favorisés ce qui conduit à augmenter exagérément la distance finale. Il est donc nécessaire de normaliser la valeur de similarité. En AD, cela s'effectue par l'intermédiaire d'une opération dite de *recodage des données*. Par exemple, dans le cas d'attribut numérique continu le problème est résolu en faisant pour tous les attributs un *centrage* (on retire à toutes les valeurs la moyenne des valeurs de l'attribut pour tous les exemples) et une *réduction* (on divise toutes les valeurs par l'écart-type de l'attribut). Ainsi, tous les attributs sont de moyenne nulle et d'écart-type égal à 1. Toutefois, ce processus ne peut être effectué que lorsque l'on connait tous les exemples, ce qui n'est pas toujours le cas notamment dans les traitements incrémentaux. En AS, la solution consiste plutôt à exprimer de manière explicite le domaine de définition des attributs et à intégrer cette information dans le calcul de la similarité. Par exemple, si l'on introduit une fonction Dom_i qui calcule la différence entre la borne haute et la borne basse du domaine de définition de l'attribut, la distance de Manhattan s'énonce de la manière suivante :

$$d(x,y) = \sum_{i=1}^{K} W_i \times Dist(x_i, y_i) \qquad \text{avec} \quad Dist(x_i, y_i) = \frac{(x_i - y_i)}{Dom_i} \in [0..1]$$

Par ailleurs, dès lors que les attributs sont typés, il est facile de traiter des exemples composés de données hétérogènes à l'aide d'une seule mesure générale pour peu que l'on utilise lors du calcul la fonction *Dist* correspondant au type de la donnée en cours de traitement. Les méthodes en AD utilisent trois types de données (nominaux, ordonnés finis et infinis) qui permettent de représenter une large variété d'informations. L'AS s'intéresse, en outre, au type hiérarchique qui permet de représenter des objets en relation taxonomique. Nous ne détaillerons pas ici les mesures associées à ces types qui sont généralement assez intuitives, on en trouvera plusieurs exemples dans [20], [78], [10] ainsi qu'une intéressante généralisation au cas des types multivalués dans [76]. Toutefois, lorsque l'on veut manipuler des données d'un type spécial, par exemple pour comparer les acides aminés d'une

protéine, il n'est pas toujours aisé de se ramener à l'un de ces types «génériques» et il devient pratique, voire nécessaire, de construire une mesure de similarité ad-hoc. Cette mesure peut consister tout simplement à construire une matrice de ressemblance (ou dissimilarité) entre tous les couples de valeurs lorsque le domaine est petit. Ainsi, dans le cas des protéines, on utilisera généralement la matrice de Dayhoff. Bien évidemment, si l'on souhaite garder les propriétés mathématiques des distances, il est nécessaire que les différentes fonctions *Dist* utilisées les vérifient également.

2.3 Pondération des attributs

Comme nous l'avons déjà évoqué dans l'introduction, la manière dont les attributs sont pondérés est tout fait cruciale pour que la mesure effectuée soit pertinente[18]. Cependant, cette pondération n'est pas toujours facile à réaliser, notamment si l'on n'a pas «d'expert du domaine» capable de la fournir ; aussi, l'automatisation de ce problème à fait l'objet de nombreuses recherches dont nous allons donner quelques aperçus. Concernant le domaine de la classification automatique, on peut citer [18], [28] en AD ou [16] en AS. C'est toutefois dans les problèmes d'identification que l'on trouve le plus de travaux car il est plus facile de déterminer automatiquement des poids lorsque l'on a des classes d'objets qui sont déjà bien définies. En AS, dans la synthèse effectuée par [81], les méthodes sont divisées en deux familles : la pondération par apprentissage et la pondération par analyse statistique.

Le système EACH [65] est un bon représentant de la première famille. Le mécanisme d'apprentissage proposé est incrémental : lorsqu'un exemple est bien classé par la procédure d'identification (c'est à dire lorsque la classe trouvée par le système est la classe réelle) les poids associés aux attributs qui ont eu tendance à prédire cette classe sont légèrement augmentés tandis que ceux des attributs en désaccord sont légèrement diminués. Le même principe est repris dans les systèmes RELIEF [40] et IB4 [1]. Inversement, dans la mesure VDM (Value Difference Metric) [72], la pondération est évaluée à partir des probabilités d'apparitions des attributs dans les différentes classes (probabilités approximées aux fréquences relatives). Dans cette approche, la distance entre l'objet «x» que l'on cherche à identifier et un objet «y» quelconque est de nature non-symétrique et s'énonce de la manière suivante :

$$\mathrm{d}(x,y) = \sum_{i=1}^{K} \left(\sqrt{\sum_{c \in C} \left(\frac{P(c|x_i)}{P(x_i)} \right)^2} \times \sum_{c \in C} \left(\frac{P(c|x_i)}{P(x_i)} - \frac{P(c|y_i)}{P(y_i)} \right)^2 \right)$$

[18] D'une certaine manière, la pondération modifie implicitement le langage de représentation : à la limite, mettre à 0 le poids d'un attribut revient à le supprimer. Nous n'évoquerons pas ici le problème dual de l'ajout de nouveaux descripteurs qui concerne plutôt le domaine de l'acquisition des connaissances.

Cette formule est décomposable en deux parties. La partie gauche calcule la pondération du $i^{ème}$ attribut (parmi K) en favorisant les attributs dont la valeur x_i est spécifique à une classe. La partie droite calcule la distance entre les valeurs x_i et y_i en considèrant que leur ressemblance est fonction de la ressemblance de leur distribution et de leur «typicité» dans toutes les classes. Ainsi, dans VDM deux valeurs différentes x_i et y_i d'un même attribut peuvent être perçues comme similaires si elles ont des probabilités d'apparition identiques. Cette mesure est reprise dans [15].

Sur un plan plus symbolique, il est également envisageable d'utiliser des connaissances exprimées sous la forme de règles ou de réseaux sémantiques pour contrôler la pondération des attributs. C'est le cas dans CLUSTER/S [73] où les attributs pertinents sont sélectionnés (il n'y a pas de pondération) en fonction du type de problème qui est à résoudre. Mais ici on se ramène en fait à une pondération manuelle puisqu'il faut introduire initialement l'ensemble des connaissances qui vont permettre de juger de la pertinence des descripteurs.

Notons que ces travaux sur la pondération automatique sont relativement spécifiques aux représentations propositionnelles. Ainsi, dans les approches basées sur des critères statistiques on suppose que les domaines de variation des attributs sont a priori connus. Malheureusement, dès lors que l'on travaille avec des représentations relationnelles il n'est généralement plus possible de connaître la répartition exacte des valeurs pour un descripteur donné car celle-ci varie en fonction des appariements entre les objets. Par ailleurs, apprendre les poids de manière globale suppose que l'on fait une hypothèse d'indépendance des caractéristiques entre elles. Or, cette hypothèse n'est pas toujours exacte [80] particulièrement dans le cas relationnel où les différents descripteurs sont explicitement connectés entre eux par le biais des variables.

2.4 La similarité en Sciences Cognitives

Dans le domaine des SC ou plus exactement celui de la psychométrie, on constate expérimentalement que les propriétés vérifiées par les distances mathématiques (minimalité, symétrie et inégalité triangulaire) ne sont pas toujours respectées dans la manière dont les sujets humains perçoivent la notion de ressemblance (voir entre autres : [75], [46], [36], [52]). De manière synthétique, ces différences s'expliquent par le fait que les critères sur lesquels les êtres humains jugent la similarité ne sont pas stables et qu'ils varient dynamiquement en fonction du contexte dans lequel on se trouve, des connaissances «a priori» que l'on possède sur le domaine, du degré de typicité des exemples, etc. Par ailleurs, les instances que l'on est amené à comparer dans les tâches de classification ou d'identification sont assez souvent décrites par des ensembles différents d'attributs, chose qui n'est pas prise en compte dans la formule de Minkowski. Tversky [66] suggère de calculer la similarité S (x, y) entre deux exemples x et y décrits respectivement par les ensembles d'attributs A et B, à partir des quatre termes $A \leftrightarrow B$, $A \approx B$, A - B et B - A combinés dans le modèle suivant :

$$S(x,y) = \frac{f(A \cap B)}{f(A \cup B) + \alpha . f(A - B) + \beta . f(B - A)} \qquad \text{avec } \alpha, \beta \geq 0$$

Selon la manière dont les paramètres f, α et β du modèle sont instanciés, on peut exprimer différents modèles cognitifs et mathématiques de la similarité. Par exemple, on peut imaginer «combiner» les définitions proposées par Tversky et Minkowski afin de dériver deux mesures de similarité, l'une symétrique *Sym* et l'autre asymétrique *Asy*. Pour ce faire on va comparer les deux ensembles attributs[19] à l'aide du modèle de Tversky et utiliser comme fonction d'évaluation f la mesure de Manhattan[20] ; en effet, Tversky fait l'hypothèse que la fonction f du modèle s'exprime sous la forme d'une combinaison linéaire des différentes caractéristiques.

● Soit la mesure de similarité entre deux valeurs: $Sim(x_i, y_i) = \frac{Dom_i - |x_i - y_i|}{Dom_i}$

$Sym(x,y) = \frac{\sum_{i}^{A \cap B} W_i \times Sim(x_i, y_i)}{\sum_{i}^{A \cup B} W_i} \in [0..1]$	$Asy(x,y) = \frac{\sum_{i}^{A \cap B} W_i \times Sim(x_i, y_i)}{\sum_{i}^{A} W_i} \in [0..1]$

La fonction *Sym* est une mesure symétrique qui est couramment utilisée en classification ou en identification [78]. Elle est produite pour les valeurs α=β=0 dans le modèle de Tversky. La seconde fonction *Asy* est une mesure asymétrique que l'on pourra utiliser en identification lorsqu'il est nécessaire de distinguer la cible de la référence. Elle s'obtient avec les valeurs de paramètres α=0 et β=-1.

Lorsque les exemples A et B sont décrits par le même ensemble d'attributs les résultats obtenus par *Sym* sont parfaitement semblables à ceux que l'on obtiendrait avec Manhattan à une normalisation et une inversion d'échelle près. Par ailleurs, la mesure *Asy* est également utilisable moyennant une modification de la mesure *Sim* (x_i, y_i) dans les problèmes d'identification où la référence est non pas une instance mais la description intentionnelle d'une classe comme c'est le cas dans le domaine de la représentation des connaissances «centrée objets»[21] (voir entre autres : [49], [64], [59]). Dès lors, les valeurs y_i correspondent au domaine de définition de l'attribut et la mesure *Sim* (x_i, y_i) doit exprimer le degré "d'inclusion" du premier argument (cible) dans le second (référence). On retrouve ici une utilisation des mesures de similarité que nous avons déjà évoqué dans l'introduction (paragraphe 1.3.1) et qui vise à quantifier le degré de subsomption entre A et B. Pour conclure, voici (tab. 4) un exemple d'utilisation de *Sym* et *Asy* sur deux exemples E_1 et E_2 :

---

[19] Précisons que dans son modèle, Tversky ne fait aucune supposition sur le type des attributs utilisés.

[20] En fait, on triche un peu puisqu'on utilise la mesure de Manhattan que pour le numérateur.

[21] Notons que, dans ce domaine, le mécanisme que nous appelons *identification* est appelé *classification*.

Attributs	W_i	E_1	E_2	Dom_i	$Sim\ (x_i, y_i)$
hauteur	1	10	20	[0..99]	90%
longueur	2	2	2	[0..9]	100%
poids	1	8	-	[0..59]	-
résistance	2	-	10	[-5..5]	-

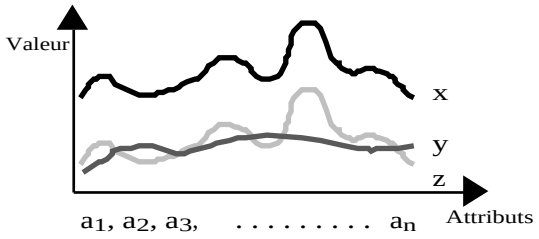
$\Rightarrow$ $Sym\ (E_1, E_2) = 48\%$
$Asy\ (E_1, E_2) = 73\%$
$Asy\ (E_2, E_1) = 58\%$

Tab. 4 : Exemples de similarités symétrique et asymétrique

2.5 Choix d'une mesure de similarité

Le choix d'une mesure de similarité est tout à fait crucial [37][22] pour la bonne exécution d'une tâche. Il s'agit en effet de trouver la meilleure adéquation entre le but qui est poursuivi et le comportement effectif de la mesure. Or, ce comportement est extrêmement variable comme l'illustre la figure 4 qui est empruntée à [20].

On constate que pour la distance de Manhattan, deux exemples sont d'autant plus similaires que les valeurs de leurs attributs sont proches deux à deux. Par contre, dans le cas du χ^2, le résultat est très différent puisqu'on compare le profil global des deux descriptions, c'est à dire les variations relatives des attributs les uns par rapports aux autres. Ainsi, lorsque l'on veut comparer la forme d'un signal numérique, il est clair que le χ^2 va donner un résultat plus significatif. Avec une distance comme le VDM [72] le résultat serait encore différent puisque dans ce cas on compare non pas le profil des valeurs, mais le profil de leur distribution dans les classes.



•Distance du χ^2 :
d(x,z) < d(y,z) (mêmes profils)

•Distance de Manhattan :
d(y,z) < d(x,z) (valeurs proches)

Fig. 4 : Illustration du comportement de différentes mesures de similarité.

[22] Cet exemple se situe dans le cadre d'une discussion générale sur les mérites comparés du symbolique et du numérique. Répondant à Michalski qui avait donné un exemple de concept (le cercle) qui, à son avis, ne pouvait être appris qu'avec la présence d'informations exprimées sous la forme de connaissances symboliques, Hanson montre qu'en changeant la mesure de similarité utilisée lors de la classification, ce concept devient alors directement apprenable avec une méthode numérique.

Lorsque les exemples sont décrits à l'aide de descripteurs différents ou lorsque l'on a besoin d'une comparaison asymétrique, on utilisera les mesures *Sym* et *Asy*. Dans le cas contraire et si l'on a aucun présupposé sur le type de similarité que l'on souhaite mettre en évidence, la distance de Minkowski constitue un choix raisonnable car les résultats qu'elle produit constituent une bonne modélisation du jugement humain [61]. De façon plus détaillée, on observe que la distance de Manhattan est le meilleur modèle pour comparer des attributs psychologiquement et/ou physiologiquement séparables (par exemple : le volume et la fréquence d'un son) et que, inversement, c'est la distance euclidienne qui modélise le mieux la comparaison de deux attributs liés (par exemple : la saturation et la luminosité d'une couleur). Dans [65], Salzberg teste ces deux mesures sur différentes bases pour faire de l'identification par la méthode des plus proches voisins. Or, il montre que les résultats obtenus à l'aide de ces deux mesures ne sont pas très éloignés et que, chose étonnante, il obtient également des résultats assez corrects (mais cependant inférieurs) avec la mesure de Chebychev qui ne prend pourtant en compte qu'un seul attribut.

Enfin, lorsqu'on utilise une méthode qui pondère les attributs sur la base de critères statistiques, il faut être attentif à la pondération *implicite* qu'elle effectue. Ainsi, dans le cas de Mahanalobis, on favorise les attributs ayant une faible dispersion. Dans le χ^2 ce sont les attributs ayant des modalités peu courantes qui sont privilégiés. La mesure VDM favorise, quant à elle, les attributs possédant les valeurs les plus discriminantes (typiques d'une classe) ce qui est logique puisque cette mesure est utilisée dans un contexte d'identification de classes.

Après ce rapide tour d'horizon des mesures de similarité qui sont utilisées dans le cadre des langages propositionnels à la fois en AD, en AS et en SC, nous allons passer à la dernière partie de ce chapitre qui porte sur les langages relationnels.

3 Similarités dans le cadre relationnel

3.1 Spécificité du problème

Lorsque l'on passe dans une représentation relationnelle, les mesures de similarité décrites dans le paragraphe précédent deviennent insuffisantes. Car si elles permettent de comparer les objets présents dans les exemples, elles ne prennent pas explicitement en compte l'étape d'appariement que nous avons décrit dans l'introduction (paragraphe 1.3.3). On pourrait certes imaginer de calculer la ressemblance entre tous les couples d'objets et d'apparier les objets qui ont les N meilleures valeurs de similarité. Cette façon de faire n'est pas du tout satisfaisante car l'appariement que l'on obtiendrait ne serait basé que sur les attributs et pas du

tout sur les relations entre les objets. Pour traiter ce problème, nous allons présenter dans cette section trois familles d'approches qui sont issues de recherche en RF, en SC et en AS.

3.2 Appariement de graphes

En reconnaissance des formes on s'est attaqué très tôt au problème de l'appariement de descriptions relationnelles et à l'évaluation de distances entre graphes dans le but de classifier ou d'identifier des structures produites par des algorithmes de segmentation d'images.

Dans l'approche de Sanfeliu et Fu [67] la distance entre deux *graphes étiquetés et attribués* [23] est calculée à partir d'une somme pondérée de plusieurs facteurs qui expriment le coût des transformations qu'il faut réaliser pour passer du graphe à identifier au graphe de référence. Ce type de mesure est classiquement appelé une *distance d'édition.* Les facteurs pris en compte sont ici au nombre de cinq : le premier quantifie les ressemblances entre les attributs des nœuds appariés, il est donc équivalent à une mesure de similarité propositionnelle et les quatre suivants expriment le nombre de nœuds et de relations qu'il faut respectivement ajouter ou supprimer pour obtenir un appariement complet entre les deux graphes. Chacun de ces critères peut être pondéré par l'utilisateur : par exemple, il peut exprimer que le coût d'un ajout de relation est double de celui d'un retrait. Cette approche est reprise dans [71] ou elle est généralisée de manière à obtenir une vraie distance au sens mathématique du terme (minimalité, symétrie et inégalité triangulaire). L'approche présentée dans [82] utilise également sur des graphes attribués en associant de plus des probabilités d'apparitions sur les relations et les nœuds : ainsi, il devient possible de travailler avec des informations bruitées ou incertaines. Dans ce cas, la distance calculée repose sur la minimisation du calcul d'entropie de Shannon.

Toutefois, comme le fait remarquer [79] ces différentes approches nécessitent souvent la détermination de poids et de seuils qui n'ont pas de sémantique précise et qui sont difficiles à déterminer. La méthode qu'il préconise repose sur le principe de *description de longueur minimale* ("minimum description length") qui est issue de travaux en théorie de l'information. Enfin, dans toutes ces méthodes où le but est de rechercher l'appariement des objets (nœuds) qui *minimise la distance* entre les graphes, cette recherche s'effectue soit à l'aide d'heuristiques et de seuils, soit par l'intermédiaire de méthodes de recherche arborescente "informées" (fig. 5) tel que le *Branch and bound* (séparation et évaluation).

[23] C'est à dire, dans lequel les propriétés des sommets des graphes sont décrites par un ensemble d'attributs.

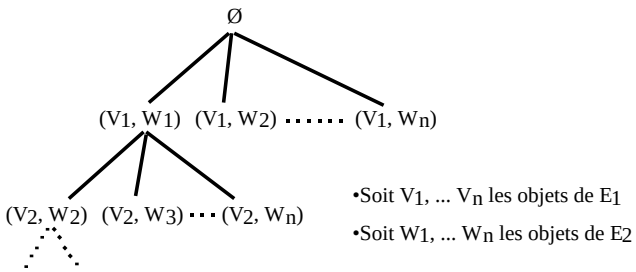


Fig. 5 : Arbre d'exploration des appariements dans une méthode «Branch and Bound»

Avec divers aménagements, les distances d'édition sont également assez utilisés dans le domaine de l'AS. Ainsi, dans les travaux de [38] qui concernent la découverte de motifs récurrents dans un graphe, l'appariement entre graphes est fondé sur des critères de coût transformationnel identiques à ceux utilisés dans la méthode de Sanfeliu and Fu. De même, dans le cadre du CBR, [14] propose également d'utiliser une distance d'édition pour comparer les cas entre eux. Pour [42], la similarité entre deux formules exprimées en logique des prédicats est fonction du nombre minimum de modifications qu'il est nécessaire d'effectuer pour passer de l'une à l'autre, mais ici les opérateurs utilisés reposent davantage sur l'utilisation de connaissances symboliques. Ils distinguent ainsi quatre groupes d'opérateurs : l'identité, l'utilisation de règles de la théorie du domaine, l'utilisation de l'idempotence de la conjonction, l'abandon de littéraux. Enfin, d'autres travaux [46], [47] s'appuient sur une approche semblable mais en employant des opérateurs différents.

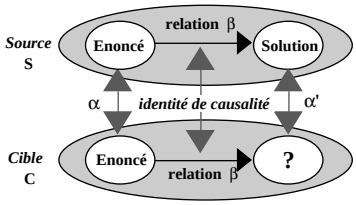
La RF n'est pas le seul domaine dans lequel la recherche des meilleurs appariements entre les objets s'effectue à l'aide de méthodes de recherche arborescente, cette approche est également utilisée en AS [22], [23]. Dans ce travail, les auteurs s'intéressent au problème de l'unification de deux formules bruitées exprimées à l'aide d'un sous-ensemble du langage APC [54]. Pour le résoudre, ils utilisent une méthode de *filtrage flou* qui repose sur la définition d'une *distance probabiliste* entre les valeurs, les littéraux et les conjonctions de littéraux qui composent les deux formules. La recherche de l'appariement *optimum* entre les variables (entités) s'effectue à l'aide d'un algorithme de Branch and Bound dont la fonction d'évaluation est la suivante. Elle combine d'une part, la distance entre les formules qui est une fonction monotone décroissante au fur à mesure que les appariements progressent et d'autre part, le nombre de littéraux appariés qui est une fonction monotone croissante.

3.3 Analogie et similarité

3.3.1 Le raisonnement analogique

Le raisonnement par analogie [69], [77] vise à caractériser une situation inconnue C, appelée cible, en la mettant en correspondance C avec une autre situation S, appelée source, que l'on a déjà observée. L'idée sous-jacente est que si deux situations se ressemblent, les conclusions que l'on peut en tirer sont analogues. En toute généralité, le raisonnement par analogie permet d'établir des relations entre des informations de types différents : c'est typiquement le cas de l'analogie classique entre un atome et le système solaire [30] où l'on met en correspondance le noyau de l'atome avec le soleil et les planètes avec les électrons. Sous une forme plus pragmatique, le processus d'analogie est utilisé en AS dans le cadre du raisonnement par cas (CBR) et du raisonnement à partir d'instances (IBL) dans le but de résoudre un nouveau problème à partir d'un problème similaire déjà résolu. Toutefois, alors que dans le cas général une analogie peut-être effectuée entre des domaines très différents (l'astronomie et la physique par exemple) dans le cas du CBR et de l'IBL, les analogies sont réalisées sur le même type de domaine et de problème. Le raisonnement analogique s'effectue en quatre étapes :

❶ Recherche de la source S potentielle
❷ Elaboration d'un appariement α entre S et C
❸ Evaluation des mises en correspondance
❹ Transfert de la solution de S vers C avec α'
  • Soit par modification directe (adaptation)
  • Soit par raisonnement (explication)



En SC, divers travaux se sont intéressés à ce problème [31], [35], [39], [77] et plus particulièrement aux deux premières étapes du processus (recherche et appariement). Dans le cadre de ce chapitre sur la similarité, plusieurs points sont intéressants : tout d'abord, les langages utilisés dans ces travaux sont dérivés de la logique des prédicats, ensuite les modèles qui ont été proposés ont fait l'objet d'implémentation. De plus, les méthodes et les concepts développés sont bien fondés, originaux et constituent une remarquable source d'inspiration.

3.3.2 Le principe de conservation des structures

Dans Gentner [31], [32] les concepts sont décrits à l'aide d'un langage structuré dans lequel l'importance sémantique des informations (leur *ordre*) est indiqué par la syntaxe. Sur cette base, Gentner propose une classification des méthodes d'appariement (tab. 5) en fonction du type d'information mise en œuvre[24]. Le

[24] Dans ce langage, les attributs correspondent à des prédicats d'arité 1 et les relations à des prédicats d'arité supérieure (2 ou plus). *L'ordre* d'une expression est défini à partir de l'ordre maximum de ses arguments plus 1 ; l'ordre des objets élémentaires (constantes) est égal à 0. Ainsi, Rouge (a) est une expression d'ordre 1, Plus-grand (Taille (a), Taille (b)) est d'ordre 2 et ainsi de suite. Pour exprimer que l'on observe un écoulement d'eau entre un bécher et un verre consécutif à une différence de pression , on écrira l'expression : CAUSE [SUPERIEUR (PRESSION (bécher), PRESSION (verre)), FLUX (bécher, verre)]

premier niveau est celui des *similarités apparentes* dans lequel on ne prend en compte que les attributs directement vérifiés par les individus ; ce niveau pourrait correspondre aux similarités dans les langages propositionnels. Ensuite, on trouve le niveau des *similarités littérales* dans lequel on tient compte de l'ensemble des attributs et des relations, mais sans privilégier les unes ou les autres, c'est à ce niveau que nous nous situons dans cette section (voir aussi [26]). Au niveau des *analogies* on favorise l'appariement des relations qui sont d'ordre le plus élevé ; de ce fait, les appariements entre les objets de la source et de la cible sont déterminées par le rôle que ces objets jouent dans la structure relationnelle plus que par leurs similarités intrinsèques. Finalement, le dernier niveau recensé par Gentner est celui des *abstractions* qui est également basé sur l'appariement des relations d'ordre supérieur et dans lequel les propriétés des objets ne sont quasiment plus prises en compte.

	Attributs	**Relations**	**Exemple**
Sim. apparentes	Beaucoup	Peu	Une luciole ressemble à une lampe
Sim. littérales	Beaucoup	Beaucoup	Le lait ressemble à de l'eau
Analogie	Peu	Beaucoup	L'atome et le système solaire
Abstraction	Très peu	Beaucoup	La chaleur s'écoule par diffusion

Table 5 : Les différents types d'appariements

Pour effectuer une analogie, Gentner propose quatre principes d'importance croissante, qui doivent guider la mise en correspondance des objets et des relations.

❶ Mise en correspondance des objets de la source et de la cible
❷ Pas de prise en compte des objets isolés décrits par quelques attributs
❸ On essaye d'apparier les objets liés par des relations communes
❹ Préservation des systèmes de relations connexes (systematicity principle)

En pratique le dernier point est le plus intéressant puisque d'une certaine manière il résume tous les autres. On peut le paraphraser de la manière suivante : pour assurer l'analogie la plus pertinente il faut privilégier les appariements qui permettent de conserver dans le système final les objets qui sont les plus *inter-connectées* possible. Or, ce principe de «conservation des structures» [69] reste tout à fait applicable au cas des similarités littérales[25]. Si réexprime ce principe en terme de graphes et de généralisation, cela signifie simplement que les appariements que l'on fait entre les objets du graphe doivent permettre, autant que possible, d'obtenir une généralisation qui soit exprimable sous la forme d'un graphe connexe. Toutefois, il faut noter que l'analogie se différencie de la similarité dans le fait qu'elle accepte de mettre en appariement des attributs qui portent des noms différents.

[25] D'ailleurs, dans les expériences effectuées par Gentner, c'est très souvent la similarité littérale qui a le comportement le plus proche de celui des sujets humains, même sur des problèmes typiques d'analogie.

Plus récemment, les travaux de Gentner sont relatifs à la modélisation du processus cognitif de recherche de la source [32], [27] et ils ont donné lieu à une théorie appellée, non sans humour, MAC/FAC (Many Are Called / Few Are Chosen). L'idée est que la recherche en mémoire d'une situation analogue est basée sur un processus à deux étapes. Dans un premier temps on recherche toutes les sources qui présentent des similarités apparentes (attributs) avec la cible. Dans un deuxième temps, on ne conserve que les sources qui ont une bonne similarité littérale (relations).

Ces travaux ont été modélisés [24] à l'aide du système SME (Structure Mapping Engine). Il a permis de vérifier le bien fondé de ces théories en confrontant les résultats produits par SME avec les résultats obtenus lors d'expérimentations avec des sujets humains. L'algorithme utilisé ne nous intéresse pas directement dans ce chapitre puisque d'une part, il est assez «coûteux» et d'autre part, il repose sur un appariement explicite des graphes sans passer par une mesure de similarité.

3.3.3 Utilisation d'un réseau de contraintes

Les travaux de Thagard et Holyoak se situent aussi dans le cadre du raisonnement par analogie. Ils se différencient de ceux de Gentner par la prise en compte de connaissances supplémentaires lors des phases de recherche et d'appariement et surtout par le processus de calcul mis en place dans les systèmes ACME [39] et ARCS[26] [74] qui repose sur la résolution numérique d'un réseau de contraintes.

Chez ces auteurs, la mise en appariement d'une source et d'une cible repose sur des contraintes syntaxiques, sémantiques et pragmatiques. Les contraintes syntaxiques sont les mêmes que celles proposées par Gentner avec le principe de préservation des systèmes de relations : il s'agit donc de rechercher un isomorphisme, aussi strict que possible, entre les structures de la source et celles de la cible. Cependant, dans le cadre d'une analogie Thagard pense qu'il est nécessaire (contrairement à Gentner) d'autoriser la mise en appariement de relations ayant des noms différents. Les contraintes sémantiques utilisées dans ACME permettent de contrôler ces appariements et se scindent en quatre familles : identité, synonymie, hyperonymie (taxonomie de relations) et méronymie (taxonomie de composants). Enfin, les contraintes pragmatiques servent à exprimer les buts de l'analogie et correspondent en pratique à une pondération des descripteurs. Thagard insiste sur le fait que l'appariement de la source et de la cible doit être *simultanément* contrôlé à l'aide des trois types de contraintes, aucune n'étant prépondérante. Pour atteindre ce but, le système ACME est basé sur un algorithme distribué reposant sur un réseau qui permet de réaliser, en parallèle, la satisfaction des contraintes.

A partir d'une expression cible C et d'une expression source S, toutes les deux exprimées dans un langage proche de la logique des prédicats, la construction du réseau s'effectue en trois étapes (figure 6). Dans un premier temps, on construit

[26] Les deux systèmes utilisent un processus de mise en appariement identique. ACME permet de faire du raisonnement analogique alors que ARCS vise surtout modéliser un mécanisme de recherche de la source.

dans le réseau autant de nœuds qu'il existe d'appariements possibles entre les éléments de S et de C, c'est à dire entre les littéraux (C_i, S_j), les prédicats et les objets (arguments). Dans un deuxième temps, des *liens d'activation* entre les nœuds sont placés. Ils expriment le fait que deux appariements sont effectivement «compatibles». Par exemple, dans la figure 6, il y a un lien d'activation (en noir) entre les nœuds (A=M) et (B=N) car le fait d'apparier A avec M n'empêche pas l'appariement entre B et N. En outre, on ajoute si nécessaire des liens d'activation entre les unités pragmatique et sémantique et les nœuds qui sont concernés par les informations contenues dans ces unités. Ainsi, dans notre exemple, il y a un lien qui part de l'unité sémantique vers (A=M) et qui renforce le fait que cet appariement semble pertinent ; de même des liens partent de l'unité pragmatique vers l'ensemble des nœuds contenant N. Dans un troisième temps, des *liens d'inhibition* entre les nœuds sont établis. Ils expriment le fait que deux appariements sont «incompatibles». Par exemple, il y a un lien d'inhibition (en gris) entre les nœuds (A=M) et (A=N) car A ne peut pas être apparié de deux manières différentes. Il pourrait également avoir des liens d'inhibition partant des unités sémantiques et pragmatique.

Cible : Source :

C1: A(a) S1: M(m)
C2: B(b) S2: N(n)
C3: C(a,b) S3: O(m,n)
C4: D(b,a) S4: P(n,m)

Sémantique : A ressemble à M
Pragmatique: N est important

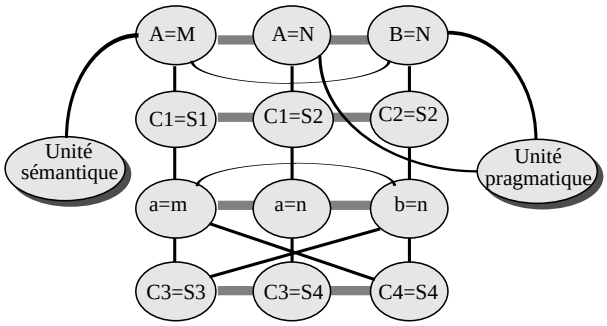


Figure 6 : Un exemple (très simplifié) du réseau construit par ACME pour apparier Cible et Source.

Une fois que ce réseau est construit, le calcul de l'appariement optimal est des plus simples. A l'étape T_0 les différents nœuds du réseau sont initialisés avec une valeur constante. Puis, à chaque étape T_{+1}, on calcule de manière synchrone le nouvel état des nœuds N_j réseau à l'aide la formule suivante :

$$N_j(t+1) = \alpha N_j(t) + \beta \sum (\text{nœuds\_activateurs}) - \gamma \sum (\text{nœuds\_inhibiteurs})$$

Après quelques centaines d'itérations, on constate expérimentalement une stabilisation du réseau. Les nœuds N_j ayant les plus fortes valeurs correspondent aux appariements les plus pertinents du points de vue de l'expression de S et C et des contraintes. Cette méthode est intéressante car elle permet de calculer une mesure de similarité entre tous les éléments de deux expressions quelconques[27] ; par

[27] Il est à noter que [33] utilise une approche tout à fait identique dans le système SIAM.

contre, le temps de calcul nécessaire (plusieurs secondes) interdit toute utilisation intensive. Cependant, comme nous allons le voir maintenant, cette idée de faire propager les ressemblances par l'intermédiaire des liens peut être exploitée différemment.

3.4 Le calcul de la similarité dans le système KBG

3.4.1 Bases de l'approche

Nous l'avons souligné au paragraphe 3.1, lorsqu'on travaille dans le cadre d'une représentation relationnelle, les mesures de similarité doivent caractériser à la fois les similitudes qui existent entre les objets pris séparément, et celles qui apparaissent au sein de leur «environnement» relationnel. Par exemple, dans la figure 7a, pour comparer les individus JOHN et PAUL, il faut évaluer la similitude entre les valeurs des attributs qu'ils partagent (l'âge et la nationalité), mais il faut aussi prendre en compte la ressemblance entre MIKE et PIERRE qui sont leur fils. Or, intuitivement, plus les situations de MIKE et de PIERRE seront voisines, plus les situations de leurs pères respectifs le seront également. Le problème est évidemment *symétrique* et lorsqu'on compare les enfants, il faut aussi tenir compte de la situation des parents.

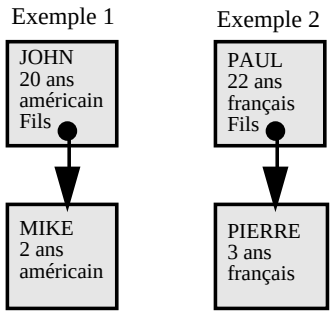


Fig. 7a : Exemple de relation.

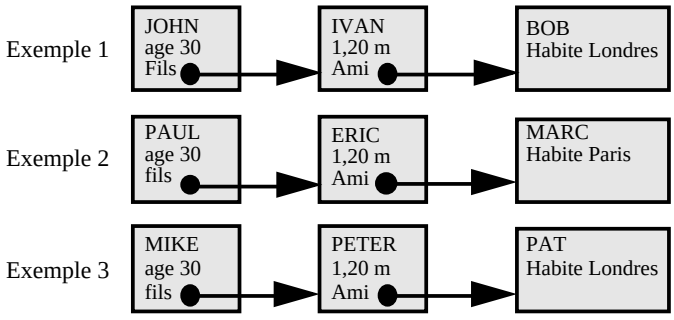


Fig. 7b : Combinaison de relations.

L'exemple suivant (figure 7b) illustre quelques unes des propriétés que cette mesure devrait aussi posséder. Dans cet exemple, lorsque l'on compare la similarité de JOHN et PAUL avec celle de JOHN et MIKE, il serait raisonnable que la première soit un peu plus faible que la seconde : SIM (JOHN , PAUL) < SIM (JOHN , MIKE). En effet, bien que les différents pères, ainsi que leurs enfants, soient indiscernables du point de vue de leurs propriétés, il existe au niveau des amis des enfants une différence portant sur le lieu d'habitation : alors que les amis des fils de JOHN et MIKE habitent Londres, celui du fils de PAUL habite Paris.

Idéalement, la mesure de similarité doit vérifier deux propriétés : d'une part, il faut que la similarité entre les objets se transmette de manière *transitive* et *infinie* d'un

objet à ses voisins et d'autre part, que *l'influence* des ressemblances décroisse en fonction inverse du nombre de prédicats qui apparaissent dans le *chemin de connexion* entre les objets[28]. Par exemple, si l'objet A est connecté à B et que B est lui-même connecté à C, les similarités dans lesquelles A est impliqué auront une influence sur celles de C mais cette influence devra être inférieure à celle que B peut avoir sur C. Dans notre exemple, la différence de similarité entre (JOHN, PAUL) et (JOHN, MIKE) doit donc être faible dans la mesure où l'information discriminante est accessible par l'intermédiaire d'une double indirection (Ami et Habite).

Dans le cadre de l'AS, nous avons développé dans le système KBG[29] [8], [9] une mesure de similarité [7], [10] qui vérifie toutes ces propriétés et qui possède deux avantages. D'une part, la complexité du calcul est dans tous les cas polynomiale (quadratique en fonction du nombre d'objets considérés) et d'autre part, son comportement est cohérent avec les théories cognitives que nous venons d'évoquer[30], notamment avec le principe de préservation des structures de Gentner.

3.4.2 Utilisation d'un système d'équations

Notre méthode peut être vue comme une extension des mesures de similarité présentées dans le cadre propositionnel. Pour mesurer la similarité entre un couple d'objets, il y a deux types d'informations à considérer : les attributs et les relations.

Nous avons proposé au paragraphe 2.2 diverses mesures permettant de comparer des attributs (numériques). Concernant les relations, le calcul est basé sur l'idée que la similarité entre deux relations est directement fonction de la similarité entre les objets qui interviennent dans ces relations. Dans l'exemple de la figure 6a cela signifie que la similarité entre les relations FILS de JOHN et PAUL dépend de la similarité entre MIKE et PIERRE. Plus formellement, la mesure se définit de la façon suivante :

● Soit un couple d'individus X et Y décrits respectivement par deux ensembles d'attributs A et B et par deux ensembles de relations C et D.

1) Similarité entre les attributs communs de X et Y

---

[28] Si l'on représente les exemples sous la forme de graphe, un chemin de connexion entre deux objets X et Y correspond à la liste des prédicats (arcs) qu'il faut parcourir pour aller du sommet correspondant à X à celui correspondant Y. Eventuellement, il peut y avoir plusieurs chemins possibles.

[29] Il s'agit d'un système permettant de catégoriser des observations et de construire des bases de connaissances d'identification. Le langage de représentation est dérivé de la logique des prédicats.

[30] On peut argumenter sur ce dernier point. Sans détailler, il nous semble important d'appliquer sur les connaissances, des traitements (ici la notion de ressemblance) ayant un comportement apparent qui soit isomorphe à des mécanismes cognitifs humains. Ce faisant, on peut espérer que l'utilisateur comprenne de façon plus complète le fonctionnement du système et qu'il en fasse ainsi une meilleur utilisation.

Dans le cas d'attribut numérique, cette similarité peut être calculée à l'aide d'une fonction $Sim_A(x_i, y_i)$ parfaitement identique à celle qui a été décrite au paragraphe 2.2.

$$Sim_A(x_i, y_i) = \frac{Dom_i - |x_i - y_i|}{Dom_i}$$

2) Similarité entre les relations communes de X et Y

Elle est calculée à l'aide de la fonction $Sim_R(x_i, y_i)$ dans laquelle les arguments x_i et y_i correspondent en fait à d'autres objets[31]. Pour simplifier, nous considérerons ici que les relations sont toutes mono-valuées (ainsi, PAUL n'a qu'un seul fils ...).

$$Sim_R(x_i, y_i) = \frac{1}{2}\left(1 + RelSym(x_i, y_i)\right)$$

L'idée de ce calcul est la suivante : le premier terme "1" correspond à la similarité entre X et Y vis à vis de cette relation. Elle est forcément maximale puisque la relation est commune aux deux objets. Le second terme *RelSym* correspond à la similarité effective entre les objets x_i et y_i qui sont liés à X et Y par le biais de la $i^{ème}$ relation.

● Soit la fonction *RelSym* qui calcule la similarité pour *l'ensemble des relations communes* à X et Y. Elle calcule la somme des contributions apportées par les attributs et les relations qui sont communes à X et Y et divise le tout par le poids total des descripteurs mis en jeu afin à normaliser le résultat. Il s'agit donc d'une similarité symétrique qui peut être vue comme extension de la mesure *Sym* proposée au 2.4.

$$RelSym(x, y) = \frac{\sum_{i}^{A \cap B} W_i \times Sim_A(x_i, y_i) + \sum_{i}^{C \cap D} W_i \times Sim_R(x_i, y_i)}{\sum_{i}^{A \cup B} W_i + \sum_{i}^{C \cup D} W_i} \in [0..1]$$

En lisant cette définition, on voit apparaître le phénomène suivant : pour calculer la similarité entre chacune des relations de X et Y il est nécessaire de connaître la similarité finale *RelSym* entre les objets qui sont reliés à X et Y par l'intermédiaire de ces relations. Bien évidemment, la situation est *symétrique* et les objets qui sont connectés à X et Y ont eux-mêmes besoin dans leur calcul de connaître la similarité finale entre X et Y. Dans cette approche, mesurer la similarité entre deux graphes

[31] Par exemple, si l'objet sur lequel on travaille est PAUL et que l'indice i correspond à la relation fils, la valeur x_i représente l'objet PIERRE. En fait, la représentation est la même que pour les attributs.

d'objest se ramène donc à *poser et résoudre un système d'équations linéaires* dont les inconnues représentent les différents couples d'individus qui composent les graphes et dont on cherche à évaluer la ressemblance.

3.4.3 Propriétés de la méthode

Cette façon de poser le problème permet d'obtenir une mesure qui prend simultanément compte dans ses résultats des *ressemblances locales et contextuelles* entre les individus. Ainsi, deux objets auront une similarité d'autant plus forte que leurs propriétés seront voisines et que les objets auxquels ils sont reliés seront analogues. Ainsi, on met en pratique le principe de Gentner de préservation des systèmes de relations. En effet, comme les similarités se propagent de manière transitive au sein du graphe des relations, on va observer des effets de *renforcement mutuel* et les fortes similarités correspondront à des zones dans lesquelles il est possible de faire des appariements entre des objets connexes[32]. Si l'on applique cette mesure à l'exemple de la figure 6b, on obtient bien l'effet souhaité (tab. 6)

• Tous les poids W_i étant égaux à 1		• Avec le poids de *Habite* égal à 2	
RelSym (JOHN, PAUL)	= 92%	*RelSym* (JOHN, PAUL)	= 90%
RelSym (JOHN, MIKE)	= 100%	*RelSym* (JOHN, MIKE)	= 100%

Tab. 6 : Evolution de la similarité entre les objets en fonction des relations et des poids.

Enfin, on peut démontrer que le système d'équations est résolu en temps polynomial et qu'il admet toujours une solution même si les graphes de relations comprennent des cycles [9]. Enfin, grâce à cette méthode *aucune recherche arborescente* n'est nécessaire pour trouver des appariements quasi-optimaux entre les objets.

Cependant il n'est pas toujours possible de se ramener directement à un système d'équations linéaires. En effet, lorsqu'un objet possède plusieurs occurrences similaires dans l'un des exemples, par exemple si PAUL à plusieurs enfants, il y a dès lors plusieurs façons de comparer, et donc d'apparier, les objets. Afin de résoudre ce problème, il est nécessaire de modifier la fonction $Sim_R(x_i, y_i)$ en introduisant une fonction *Meilleur-App*[33] qui va retourner la valeur moyenne des meilleurs appariements possibles entre les éléments des deux listes d'objet exprimées par x_i et y_i. L'introduction de cette fonction *Meilleur-App* a pour conséquence que le calcul de similarité n'est plus exprimable sous la forme d'un système d'équations *linéaires*. Il faut alors utiliser une méthode de résolution

[32]Le résultat de la mesure est donc tout à fait utilisable pour faire une recherche de l'appariement optimal entre les objets à l'aide, par exemple, d'algorithmes de recherche de couplage de poids maximum dans un graphe bipartie que nous avons évoquées à la fin de l'introduction (paragraphe 1.3.2).

[33] Le lecteur intéressé par les détails de la méthode pourra consulter [7], [9].

itérative (Jacobi par exemple) pour résoudre le système d'équations. Cependant le nombre d'itérations reste toujours très faible (3 ou 4) car le système d'équations converge très rapidement.

Si l'on compare cette approche à celle de Thagart dans ACME, on constate une certaine similitude. En pratique, le réseau d'ACME pourrait facilement être redéfini sous la forme d'un système d'équations qui seraient toutefois différentes de celles utilisées dans KBG. Par contre, il est clair que la recherche dans ACME est beaucoup moins contrainte puisqu'elle autorise l'appariement de prédicats de noms différents ; il serait donc intéressant de faire des expérimentations pour analyser le comportement respectif des deux approches sur des problèmes de complexité identique. Notons pour conclure que la mesure de similarité de KBG à été récemment reprise dans [21] pour être appliquée dans le cadre de l'IBL.

4 Conclusion

Par l'intermédiaire des différents travaux que nous avons examiné dans ce chapitre, nous avons voulu montrer le caractère clairement pluridisciplinaire de la notion de similarité. Il est en effet assez rare que des travaux issus de domaines aussi différents dans leur approches que, par exemple, l'analyse des données et les sciences cognitives puissent comporter des contributions croisées.

A notre sens, la notion de similarité construit incontestablement un lien entre les approches symbolique, cognitive et numérique de l'apprentissage. En effet, au travers de son utilisation, il devient tout à fait possible de réaliser des systèmes informatiques capables de communiquer de manière bidirectionnelle avec l'utilisateur à l'aide des langages de représentations symboliques puissants, tout en basant les processus internes de calcul du système sur des notions purement numériques. Ainsi, cette symbiose permet d'obtenir des mécanismes d'apprentissage qui sont à la fois robustes et rapides dans leur traitement et intelligibles dans leur résultats.

BIBLIOGRAPHIE

[1] AHA D.W. 1990. A study of instance-based learning algorithms for supervised learning tasks: Mathematical, empirical, and psychological evaluations (Technical Report 90-42). Irvine. University of California. Department of Information and Computer Science.

[2] AHA D.W., KIBLER D., ALBERT M.K. 1991. Instance-Based Learning Algorithms. *Machine Learning journal.* Volume 6. 37-66.

[3] AHA D., GOLDSTONE R. 1992. Concept learning and flexible weighting. In Proceedings of the 14th Annual Conference of the Cognitive Science Society. Bloomington, IN. 534-539.

[4] AHUJA R.K, MAGNANTI T.L., ORLIN J.B. 1993. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall.

[5] BARTHELEMY J-P. 1991. Similitude, Arbres et Typicalité. *Sémantique et Cognition*. D. Dubois (Ed). Presses du CNRS. Chapitre 11. 205-224.

[6] BIBERMAN Y. 1994. A Context Similarity Measure. In proceedings of European Conference on Machine Learning (ECML). Catania (Italy). 49-63.

[7] BISSON G. 1992a. Learning in FOL with a similarity measure. In Proceedings of 10th AAAI Conference. San-Jose. 82-87.

[8] BISSON G. 1992b. Conceptual Clustering in a First Order Logic Representation. Proceedings of 10th ECAI. Vienna. 458-462.

[9] BISSON G. 1993. KBG : induction de bases de connaissances en logique des prédicats. Thèse de l'université Paris-sud soutenue le 30 avril 1993.

[10] BISSON G. 1995. Why and how to define a similarity measure for object-based representation systems. Proceedings of 2nd international conference on building and sharing very large-scale knowledge bases (KBKS). IOS press. Enschede (NL). 10-13 avril 1995. pp 236-246

[11] BOUCHON-MEUNIER B., DESPRES S., DUBOIS D., GASCUEL O. GENOCHE A., PRADE H. 1990. Interface entre symbolique et numérique. Actes des 3ème journée nationales du PRC-IA. Hermès. 89-138. Paris.

[12] BRISSAC O., LIQUIERE M. 1994. Un algorithme optimal pour l'association d'entités à partir de leurs similarités. Actes des 9eme JFA. Strasbourg. B1-B14.

[13] BRITO P. 1991. Analyse de données symboliques, pyramides d'héritage. Thèse de l'université Paris XI Dauphine soutenue le 26 février 1991.

[14] BUNCKE H., MESSMER B.T. 1994. Similarity Measure for Structured Representations. *Topics in Case-Based Reasoning*. S. Wess, K-D. Althoff, M. Richter (Eds.). Springer-Verlag (LNCS 837). 106-118.

[15] COST S., SALZBERG S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine learning Journal* 10. 57-78

[16] DECAESTECKER C. 1993. Apprentissage et outils statistiques en classification conceptuelle incrémentale. *Revue d'Intelligence Artificielle.* Volume 7, numéro 1. 33-71.

[17] DE RAEDT L., BRUYNOOGHE M. 1992. An Overview of the Interactive Concept-Learner and Theory Revisor CLINT. *Inductive Logic Programming*, Academic Press, 163-191.

[18] DE SOETE G., DESARBO W.S., CARROLL J.D. 1985. Optimal Variable Weighting for Hierarchical Clustering: An Alternating Least-Squares Algorithm. *Journal of classification*. Number 2. 173-192.

[19] DIDAY E. 1991. Des objets de l'analyse des données à ceux de l'analyse des connaissances. *Induction Symbolique et Numérique à partir de données*. CEPADUES

[20] DIDAY E., LEMAIRE, J., POUGET J., TESTU F., 1982. *Eléments d'analyse des données.* Edition Dunod.

[21] EMDE W., WETTSCHERECK D. 1996. Relational Instances-Based Learning. Proccedings of 13th International Conference on Machine Learning (ICML96). Saitta L. (ed). Morgan Kaufmann. Bari, Italy.

[22] ESPOSITO F., MALERBA D., SEMERARO G. 1991a. Flexible Matching for Noisy Structural Descriptions. In Proceedings of 12th IJCAI, 658-664. Sydney.

[23] ESPOSITO F., MALERBA D., SEMERARO G. 1991b. A Syntactic Distance for Partially Matching Learned Concepts Against Noisy Structural Objects Descriptions. *International Journal of Expert Systems*. Volume 4 number 4. 409-451.

[24] FALKENHAINER B., FORBUS K., GENTNER D., 1989. The Structure Mapping Engine: Algorithm and Examples. Artificial Intelligence Journal. Number 41. 1-63.

[25] FISHER D.H 1987. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning Journal* 2, 139-172.

[26] FISHER D., YOO J. 1991. Combining Evidence of Deep and Surface Similarity. Proceedings of 8th International Workshop on Machine Learning. Illinois. 46-50

[27] FORBUS K. D., GENTNER D., LAW K. 1995. MAC/FAC : A Model of Similarity Based Retreival. Cognitive Science . Number 19. 141-205.

[28] FOWLKES E., GNANADESIKAN R., KETTENRING J. 1988. Variable Selection in Clustering. *Journal of Classification* 5, 205-228.

[29] GENNARI J., LANGLEY P., FISHER D. 1989. Model of Incremental Concept Formation. *Artificial Intelligence Journal*, Volume 40, 11-61.

[30] GENTNER D. 1983. Structure-Mapping: A Theoritical Framework for Analogy. In *Readings in Cognitive Science.* Morgan Kaufmann. 303-310. (de*Cognitive Science* 7, 155-170. 1983).

[31] GENTNER D. 1989. The mechanisms of analogical learning. *Similarity and Analogical Reasoning.* S. Vosniadou, A. Ortony (eds). CU Press. (reprinted in

Readings in Machine Learning. J. Shavlik, T. Dietterich (eds). Morgan Kaufmann 1990. 601-622).

[32] GENTNER D., RATTERMANN M., FORBUS K. 1993. The Roles of similarity in Transfer: Separating Retrieviability from Inferential Soundness. *Cognitive Psychology* 25. 431-467.

[33] GLUCK M., CORTER J., 1985. Information, uncertainty and the utility of categories. Proceedings of the 7th Annual Conference of the Cognitive Science Society. 283-287.

[34] GOLDSTONE R. 1991. Similarity As Structural Alignment. Indiana University. Cognitive Science Program. Research Report 55.

[35] GOLDSTONE R. 1994. The Role of Similarity in Categorization: Providing a Groundwork. *Cognition Journal.* Number 52 (2). 125-157

[36] GOLDSTONE R. 1995. Mainstream and Avant-garde Similarity. Indiana University. Cognitive Science Program. Research Report 132.

[37] HANSON S. 1990. Conceptual Clustering and Categorization: Bridging the Gap between Induction and Causal Model. In *Machine Learning 3*. Morgan Kaufman. 235-268.

[38] HOLDER L., COOK D., BUNKE H. 1992. Fuzzy Substructure Discovery. In Proceeding of the 9th International Workshop on Machine Learning. 218-223.

[39] HOLYOAK K., THAGARD P. 1989. Analogical Mapping by Constraint Satisfaction. In *Cognitive Science Journal*. number 13. 295-355.

[40] KIRA K., RENDELL L.A. 1992. A Practical Approach to Feature Selection. Proceedings of International Conference on Machine Learning (ICML). Aberdeen. 249-256.

[41] KODRATOFF Y., GANASCIA J.G. 1986. Improving the generalization step in Learning. *Machine Learning 2 an Artificial Intelligence Approach.* Morgan Kaufmann. 215-244.

[42] KODRATOFF Y., TECUCI G. 1988. Learning Based on Conceptual Distance. *IEEE Transactions on Patern Analysis and Machine Intelligence*. PAMI 10. 897-909.

[43] KODRATOFF Y. 1991. Faut-il choisir entre science des explications et science des nombres. *Induction Symbolique et Numérique à partir de données*. CEPADUES.

[44] LANCE G.N., WILLIAMS W.T. 1967. A General Theory of Classification Sorting Strategies: 1=Hierarchical Systems, 2=Clustering systems. *Computer Journal* 9-10. 373-380.

[45] LEAKE D. 1996. CBR in context : The Present and the Future. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park: AAAI press.

[46] LEBBE J., NICOLAS J., VIGNES R. 1991. From Knowledge to Similarity. In Proceedings of the International Conference Symbolic-Numeric, Data Analysis and Learning. Nova Science. 585-597. Paris (France).

[47] LERMAN I.C., NICOLAS J., OUALI M., PETER P. 1991. Classification conceptuelle: une approche centrée sur la similarité. *Induction Symbolique et Numérique*, Cepaduès Edition. 153-177.

[48] MAHER P. 1993. A Similarity Measure for Conceptual Graphs. *International Journal of Intelligent Systems*. Volume 8. 819-83.

[49] MARIÑO O., RECHENMANN F., UVIETTA P. 1990. Multiple Perspectives and Classification Mechanism in Object-Oriented Representation. Proceedings of 9th European Conference on Artificial Intelligence (ECAI). Stockholm. 425-430.

[50] MARCOTORCHINO F. 1991. La classification automatique aujourd'hui : bref aperçu historique applicatif et calculatoire. Publications Scientifiques et Techniques d'IBM. Numéro 2. 35-94.

[51] MARKMAN A., GENTNER D. 1993. Structural Alignment during Similarity Comparisons. In *Cognitive Psychology* 25. 525-575.

[52] MARKMAN A., GENTNER D. 1996. Commonalities and differences in similarity comparisons. In *Memory & Cognition.* 24(2). 235-249.

[53] MEDIN D.L., WATTENMAKER W.D., HAMPSON S.E. 1987. Family resemblance, conceptual cohesiveness and category construction. *Cognitive Psychology.* volume 19. 242-279

[54] MICHALSKI R.S., STEPP E. 1983. Learning from Observation : Conceptual Clustering. In *Machine Learning 1 an Artificial Intelligence Approach*, Tioga, 331-363.

[55] MITCHELL T. 1982. Generalization as Search. *Artificial Intelligence Journal* 18. 203-226.

[56] MUGGLETON S., BUNTINE R. 1992a. Machine Invention of First Order Predicates by Inverting Resolution. In *Inductive Logic Programming,* Academic Press, 261-280.

[57] MUGGLETON S., FENG C. 1992b. Efficient Induction of Logic Program. In *Inductive Logic Programming*, Academic Press, 281-298.

[58] MYAENG S., LOPEZ-LOPEZ O. 1992. Conceptual graph matching: a flexible algorithm and experiments. Journal of Experimental and Theoritical Artificial Intelligence 4. 107-126.

[59] NAPOLI A. 1992. Représentations à objets et raisonnement par classification en intelligence artificielle. Thèse d'état de l'université Nancy I, soutenue le 31 janvier 1992.

[60] NIBLETT J. 1988. A Study of Generalization in Logic Programming. Proceeding of 3th European Session on Learning (EWSL). Glasgow.

[61] NOSOFSKY J. 1984. Choice, Similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. Volume 10. 104-114.

[62] QUINLAN J.R 1993. FOIL: A Midterm Report. Proceedings of 6th European Conference on Machine Learning (ECML). Vienna. 3-20.

[63] RICHTER M.M. 1992. Classification and Learning of Similarity Measures. *Studies in Classification, Data Analysis and Knowledge Organization* (Opitz, Lausen, Klar eds). Springer Verlag. Also exists as SEKI report SR-92-18 SFB.

[64] SALOTTI S. 1992. Filtrage flou et représentation centrée objet pour raisonner par analogie : le système FLORAN. Thèse de l'université Paris-sud (n°2339) soutenue le 4 décembre 1992.

[65] SALZBERG S. 1991a. Distance Metrics for Instance-Based Learning. Proceedings of International Symposium on Methodologies for Intelligent System (ISMIS). Charlotte NC. Lecture Notes in Artificial Intelligence 542. 399-408.

[66] SALZBERG S. 1991b. A nearest hyperrectangle learning method. Machine Learning Journal. Nomber 6. 251-276

[67] SANFELIU A., FU K. 1983. A Distance Measure between Attributed Relational Graphs for Pattern Recognition.*IEEE Transactions on System, Man & Cybernetics.* SMC-13. 353-362.

[68] SAPORTA G. 1990. Probabilités, Analyse des Données et Statistique. Edition Technip.

[69] SAVELLI J. 1993. Facettes statique et dynamique de la notion d'analogie : relation d'analogie et processus analogique. Thèse de la faculté des sciences et techniques de St-Jérome (Aix-Marseille III) soutenue le 25 janvier 1993.

[70] SEBAG M. SCHOENAUER M. 1993. A Rule-Based Similarity Measure. *Topics in Case-Based Reasoning*. S. Wess, K-D. Althoff, M. Richter (Eds.). Springer-Verlag (LNCS 837).

[71] SHAPIRO L., HARALICK R. 1985. A Metric for Comparing Relational Structures. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 7. 90-94.

[72] STANFILL C., WALTZ D., 1986. Toward memory-based reasoning. *Communications of the ACM*. Number 29. 1213-1228.

[73] STEPP E., MICHALSKI R.S. 1986. Conceptual Clustering of Structured Objects: A Goal-Oriented Approach. *Artificial Intelligence 28*, 43-69 (une version modifiée de cet article se trouve également dans *Machine Learning 2*, 471-498).

[74] THAGARD P., HOLYOAK K., NELSON G., GOCHFELD D. 1990. Analog Retreival by Constraint Satisfaction. In *Artificial Intelligence*. number 46. 259-310.

[75] TVERSKY A. 1977. Features of similarity. In *Readings in Cognitive Science*. Morgan Kaufmann 1988. (from *Psychological Review* 84, 327-352, 1977).

[76] VALTCHEV P., EUZENAT J. 1997. Dissimilarity measure for collections of objects and values. Lecture notes incomputer science 1280 (Xiaohui Liu, Paul Cohen, Michael Berthold (éds.). Actes 2nd international symposium on intelligent data analysis. London (UK), pp259-272, (21-25 août) 1997), 1997

[77] VOSNIADOU S., ORTONY A. (EDS) 1989. *Similarity and Analogical Reasoning*. CU Press.

[78] VOß (ED) 1994. *Similarity Concepts and Retreival Methods*. FABEL project Technical Report number 13. Gesselschaft für Mathematik und Datenverarbeitung (GMB). Sankt Augustin.

[79] VOSSELMAN G. 1992. Relational Matching. *Lectures Notes in Computer Science n°628.* Springer Verlag.

[80] WETTSCHERECK D., AHA D. 1995. Weighting Features. Proceedings of 1st International Conference on Case-Based (ICCBR-95). Springer-Verlag. Lisbon, Portugal. 347-358. (Also available as technical reports : NCARAI TR: AIC-95-026).

[81] WETTSCHERECK D. 1994. A study of Distance-Based Machine Learning Algorithms. PhD thesis in Computer Science presented on June 7, 1994. Oregon State University.

[82] WONG A., YOU M. 1985. Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition. In *Transactions on Pattern Analysis and Machine Intelligence.* PAMI-7. 599-609.