RAIRO. RECHERCHE OPÉRATIONNELLE

E. DIDAY

Introduction à l'approche symbolique en analyse des données

RAIRO. Recherche opérationnelle, tome 23, n° 2 (1989), p. 193-236

<http://www.numdam.org/item?id=RO_1989__23_2_193_0>

Recherche opérationnelle/Operations Research
(vol. 23, n° 2, 1989, p. 193 à 236)

INTRODUCTION A L'APPROCHE SYMBOLIQUE EN ANALYSE DES DONNÉES (*)

par E. DIDAY (1)

Résumé. — *L'objectif de ce travail est d'indroduire un point de vue « symbolique » en analyse des données et de montrer que l'on peut ainsi étendre l'analyse des données classiques à des données plus complexes car plus proches de la réalité multidimensionnelle. On définit d'abord divers types d'objets symboliques: les événements élémentaires, les objets assertion, horde, de synthèse, règle. Ils s'expriment sous forme de conjonction logique de propriétés portant sur des variables usuels de l'analyse des données. Des connaissances supplémentaires peuvent venir s'ajouter à la description des objets à l'aide de taxonomies et d'affinités par exemple. On schématise quelques problèmes d'analyse de données où les « individus » sont des objets symboliques. L'analyse des données symboliques est guidée par plusieurs principes: la « fidélité » entre les objets symboliques utilisés et les connaissances à représenter, la « prédominance de la connaissance » pour guider les algorithmes, la « cohérence » entre les objets nécessités en entrée et ceux qui sont obtenus en sortie, « l'explicabilité » des résultats. Plusieurs propriétés des objets symboliques sont ensuite étudiés: on définit la notion d'ordre, d'union et d'intersection entre ces objets et on montre qu'ils sont organisés selon un treillis d'héritage. On étudie ensuite des « qualités » que l'on peut attribuer aux objets symboliques (complétude, simplicité, affinement), à des classes (stabilité, effritement) ou à des classifications d'objets symboliques (degré de recouvrement, qualité de l'héritage). On énonce enfin plusieurs exemples de problème d'analyse des données symboliques.*

Mots clés : Analyse des données, objets symboliques, classification automatique.

Abstract. — *The aim of this paper is to define the symbolic approach in data analysis and to show that it extends data analysis to more complex data which may be closer to the multidimensional reality. We introduce several kinds of symbolic objects ("events", "assertions", and also "hordes" and "synthesis" objects) which are defined by a logical conjunction of properties concerning the variables. They can take for instance several values on a same variable and they are well adapted to the case of missing and nonesens values. Background knowledge may be represented by "pyramidal taxonomies" and "affinities". In clustering the problem remains to find inter-class structures such as partitions, hierarchies and pyramids on symbolic objects instead classical one. Symbolic data analysis is conducted on several principles: accuracy of the representation, coherence between the kind of objects used at input and output, knowledge predominance for driving the algorithms, self explanation of the results. We define the notion of order, union and intersection between symbolic objects and we show that they are organised according to an inheritance lattice. We study several kinds of qualities of symbolic objects, of classes and classification of symbolic objects. Finally we propose several kinds of data analysis relating to the symbolic approach.*

Keywords : Data analysis, symbolic objects, clustering.

(*) Reçu juillet 1988.
(1) Université Paris-IX - Dauphine, place du Maréchal-de-Lattre-de-Tassigny, 75016 Paris et I.N.R.I.A., domaine de Voluceau, B.P. n° 105, 78153 Le-Chesnay Cedex.

Recherche opérationnelle/Operations Research, 0399-0559/89/02 193 44/$ 6.40

INTRODUCTION

Avec l'arrivée des systèmes de saisie automatique dans tous les domaines de l'activité humaine, l'accumulation de données de toutes sortes s'intensifie. Extraire de ces données des informations utiles dans un but explicatif ou décisionnel est l'un des problèmes majeurs auxquels sont confrontés les entreprises. L'analyse des données (qui s'est développée surtout à partir des années 60) grâce aux facilités de calculs et de représentation fournis par les machines a pour but de donner des réponses à ces problèmes en permettant le traitement de grands tableaux numériques où n objets (ou individus) en lignes prennent des valeurs sur p variables en colonne.

Les progrès récents réalisés en Bases de Données et en Languages Orientés Objets permettent de manipuler et représenter des objets exprimant des connaissances d'une complexité grandissante difficiles à exprimer dans le carcan des tableaux de données classiques on assiste aussi à un foisonnement de Bases de connaissances dû en parti au succès des systèmes experts. C'est dans ce contexte que nous introduisons la notion d'objets symboliques.

La distinction entre objets «symboliques» et «numériques» est claire dés lors que l'on considère qu'un objet est «numérique» s'il peut être représenté et utilisé comme un point de l'espace $\mathbf{R}^p$ considéré comme un espace vectoriel muni des opérations habituelles et qu'il est «symbolique» si ce n'est pas le cas (autrement dit, s'il est nécessaire de définir une sémantique propre au domaine d'application car la sémantique des nombres ne convient pas). Il résulte de cette définition que l'analyse des données classique traite depuis longtemps des objets symboliques puisque c'est la cas de tous les objets caractérisés par des variables qualitatives (qu'elles soient nominales ou ordinales). Le but de ce travail est d'étendre l'analyse des données classique à l'étude d'objets symboliques plus complexes qui s'expriment sous forme de conjonction de propriétés portant sur des variables classiques: quantitatives ou qualitatives (nominales ou ordinales). Ils se distinguent des objets classiquement traités en analyse des données d'abord au niveau de leur description:

(*a*) Chaque variable peut prendre des valeurs multiples pour un même objet, par exemple [fièvre = [39°, 41°]] pour exprimer que les patients d'une maladie donnée ont une fièvre variant entre 39° et 41°; ou [couleur = { jaune, marron, noir $\wedge$ marron }] pour exprimer le fait que la couleur d'un cèpe peut être jaune, marron ou noir et marron. Ces valeurs ne sont pas réductibles à une modalité d'une variable qualitative afin de ne pas perdre toute l'information qu'elles contiennent.

(*b*) Comme conséquence de (*a*) on est amené à exprimer différents types de liens entre les variables: quand une variable prend une modalité, une autre peut ne pas avoir de sens (on ne décrit pas les ordinateurs d'une entreprise qui n'en posséde pas) ou l'on doit restreindre son champs de valeurs possibles (si la couleur est jaune, la taille est petite). On obtient ainsi des objets symboliques munis de propriétés.

Ils se distinguent aussi au niveau de leur manipulation:

(*c*) Un objet symbolique est une description en compréhension d'une classe d'objets élémentaires qui en constituent l'extension (l'objet [couleur = { blanc, jaune }] a pour extension tous les objets élémentaires dont la couleur est soit blanche soit jaune).

(*d*) Comme conséquence de (*c*) on peut généraliser ou spécialiser un objet symbolique en modifiant ses propriétés de façon soit à étendre soit à restreindre son extension.

(*e*) Pour généraliser on utilise la sémantique du domaine d'application (qui s'exprime, par exemple, sous forme d'une taxonomie) ce qui évite, par exemple, de généraliser « quelqu'un qui boit du whisky et de l'eau » et « quelqu'un qui boit du vin et de l'eau » par « quelqu'un qui boit de l'eau » mais plutôt par « quelqu'un qui boit de l'alcool et de l'eau ».

Les objets symboliques se rencontrent dans de nombreuses situations. Quand par exemple, un expert (chef d'atelier, banquier, médecin, biologiste...) veut décrire un objet de sa connaissance (comportement d'une chaîne d'usinage, type de clientèle d'une succursale, maladie, plante, ...); quand un utilisateur d'une base de données exprime une question sous forme de requête; quand on désire décrire une classe d'objets classiques obtenue par une classification automatique de façon plus riche, plus explicative que par le centre de gravité et la variance; quand les objets à étudier nécessitent l'utilisation de variables complexes dont la valeur pour chaque objet est par exemple, un arbre, un graphe, ou d'autres objets. Etc.

Le but de l'analyse des données symboliques est d'étendre la problématique, les méthodes et algorithmes de l'analyse des données classiques aux objets symboliques. Les objets symboliques permettent une adéquation plus grande à la réalité multidimensionnelle que les objets habituellement utilisés en analyse des données et apparaissent de façon éparse depuis longtemps dans différentes disciplines. Ils sont par exemples indispensables en biologie pour décrire des espèces d'animaux ou de plantes et les premières classifications de tels objets remontent à l'antiquité au moins à Aristote (Historia animalis). Les flores modernes constituent des exemples parfaits de classifications d'objets symboliques à l'aide de « clés ». Ces classifications ont généralement

été constituées de façon plus ou moins empirique pour rassembler des années si ce n'est des siècles d'expérience. Bien que l'un des premiers algorithmes de classification automatique remonte à Adanson (1757), d'après Sneath et Sokhal (1973) il faut remonter à la fin des années 50 pour trouver une systématisation des algorithmes de taxonomie numérique. Les premiers algorithmes de segmentation remonte à ces dates: Belson (1959), Morgan et Sonquist (1963), avec leur méthode AID (Automatic Interaction Detector). Ces techniques constituent un premier effort pour extraire de données classiques, des objets symboliques. En effet, en segmentation classique on dispose en entrée d'un tableau de données classiques mais en sortie on fournit un arbre dont les branches permettent de décrire des classes d'objets, sous forme de conjonction de propriétés portant sur les variables explicatives, qui sont donc des objets symboliques. L'algorithme de Quinlan (1983) est une version moderne (exprimée dans le contexte de l'intelligence artificielle) de cette ancienne approche. Comment faire de la segmentation dans le cas où des lignes du tableau de données d'entrée représentent des objets symboliques? Il semble que le premier programme de segmentation d'objets symboliques ait été écrit par L. E. Morse (1968) suivi de R. S. Pankurst (1970). M. J. Dallwitz (1984) a proposé un langage standard pour représenter les objets symboliques de types «assertion» qui est couramment utilisé en biologie. J. Lebbe (1984) a réalisé un système de gestion d'objets assertion munis de propriétés appelé XPER qui contient maintenant plusieurs programmes de traitement d'objets symboliques, l'un d'entre eux permet, par exemple, d'obtenir à partir d'un ensemble d'objets symboliques et d'une variable qualitative à expliquer, un graphe d'identification des modalités de cette variable (*voir* R. Vignes, J. Lebbe (1987)).

En intelligence artificielle et plus particulièrement en apprentissage (*voir* Y. Kodratoff (1986)) de nombreux algorithmes d'analyse de données symboliques se développent citons parmi les premiers algorithmes dit de «conceptual clustering» issus de la méthode des nuées dynamiques Diday, Govaert, Lechevallier, Sidi (1980), puis Michalsky, Stepp, Diday (1982). Pour la génération automatique de règles citons Quinqueton, Sallantin (1986), Ho Tu Bao, Diday, Summa (1987), H. Ralambondrainy (1987), Guénoche (1987), Gascuel (1987). Guigues et Duquenne (1986), Ganascia (1987) dans le même but par utilisation de treillis et en étant plus exhaustifs. Wille (1983) pour la représentation graphique de «concepts» par des treillis. Des languages informatiques dits «orientés objets» ou «schémas» adaptés à cette problématique se développent intensivement citons par exemple M. Manago (1988) pour «Gol» et R. Ducournau (1989) pour «Yafool».

Les applications sont nombreuses citons par exemple, Manago (1988) pour le diagnostique des maladies de la tomate, Tronche, Lebbe et Vignes (1987) pour l'aide au diagnostique des causes de surdité, Menessier, Diday (1988) pour la prévision dans les séries pseudo-périodiques, Diday et Roy (1988) par l'identification de sols en géologie, Touati et Diday (1989), pour obtenir une synthèse d'objets « harmonieuse » à partir d'affinités.

Nous montrons d'abord, par de nombreux exemples, comment les différents objets symboliques introduits se distinguent au plan sémantique. Ils se distinguent également par leur champ d'application, les problèmes et algorithmes de classification et d'analyse des données qu'ils soulèvent; on ne pratiquera pas de la même façon pour générer des règles, des assertions, des hordes ou des objets de synthèse. Cependant, au plan syntaxique, ils peuvent tous s'exprimer (par des changements de variables adéquats) sous la forme simple d'un « événement élémentaire ». Cela permet d'énoncer ensuite un ensemble de propriétés qui les concernent tous.

1. LA REPRÉSENTATION DES CONNAISSANCES : LES CHOIX DE BASE

1.1. Choix des variables

Une variable y est par définition une application de $\Omega \rightarrow O$ où Ω est l'ensemble dit des « objets élémentaires » et O l'ensemble d'observation où la variable prend ses valeurs.

Le type de variable dépend de la structure algébrique de O. Si O est un intervalle de l'ensemble des réelles nous dirons que la variable est quantitative. Si O est fini ou dénombrable nous dirons qu'elle est qualitative. Dans ce dernier cas, on dira qu'elle est qualitative ordinale ou nominale suivant que O est ordonné ou non.

	y1	y2	y3
w1	1	2	1,65
w2	3	1	1,80
w3	2	1	1,29
w4	1	3	1,30
w5	4	2	1,58

Figure 1

Exemple: Soit $\Omega = \{w_1, \ldots, w_5\}$ les variables classe d'âge, nationalité et taille définissant le tableau de la figure 1. Ces variables notées y_1, y_2, y_3 sont respectivement qualitatives ordinales, nominales et quantitatives.

1.2. Choix des objets symboliques

1.2.1. *La notion d'objet symbolique*

A partir des objets élémentaires caractérisés par les valeurs prises par plusieurs variables $v_1, \ldots, v_p$ avec v_i: $\Omega \to O_i$ on peut définir beaucoup d'autres types d'objets dont nous présentons maintenant quelques formes utilisées dans l'approche symbolique. On introduit d'abord les «objets assertions» cas particuliers des «objets hordes» qui sont présentés ensuite et on termine par les «objets de synthèse munis de propriétés» qui sont les plus compliqués. Pourquoi utiliser le terme objet et que signifie-t-il? Remarquons d'abord qu'une ligne d'un tableau de données qui caractérise un «objet» (en *AD* on dit souvent aussi «individu») peut s'exprimer sous forme d'une conjonction de propositions logiques que nous appellerons événements. Par exemple si l'on considère la première ligne du tableau de la figure 1 et que l'on restreint aux deux premières variables, l'individu ou objet w_1 est caractérisé par l'expression symbolique définie par la conjonction:

$$[\text{classe d'âge} = 1] \wedge [\text{nationalité} = 2]$$

De façon générale dans ce texte on considérera qu'*un objet symbolique* est *une description qui s'exprime à l'aide d'une conjonction d'événements (on dit aussi «propriétés») portant sur les valeurs prises par les variables.*

Pour faciliter la lecture on présentera d'abord les objets symboliques dans le cas où ces événements sont des fonctions propositionnelles à une variable. Au paragraphe 1.3 on introduit les objets symboliques munis de propriétés où des fonctions propositionnelles à plusieurs variables peuvent apparaître.

Le terme d'objet étant défini, l'intérêt de l'utiliser est de trois ordres:

(*a*) En terme d'*AD* il permet de bien rappeler que les objets symboliques peuvent être considérés comme des lignes d'un tableau de données et donc sur lesquels on tentera d'adopter la problématique et les méthodes usuelles d'*AD*.

(*b*) Les événements de la connaissance supplémentaire ne seront pas appelés «objet» même s'ils peuvent prendre la même forme que les objets à analyser.

(*c*) Les objets symboliques que nous définissons peuvent être considérés comme des «objets» des languages orientés objet qui prennent une importance croissante.

Dans ces languages chaque objet comprend une partie purement déclarative qui décrit les variables en donnant leur domaine de variation et une partie active formée de méthodes particulières pour calculer certaines variables ou des valeurs par défaut. Les objets sont organisés (selon un graphe dit «d'héritage») à partir d'un objet génétique, père de tous les autres, de façon à ce que chaque objet apparaisse comme une instanciation (ou spécialisation) d'un ou plusieurs objets dont il affine la description. Aussi bien dans leur définition que dans leur organisation (qui sera obtenue par des méthodes de classification), les objets que nous allons maintenant décrire entrent dans le cadre de ceux des langages orientés objets.

Exprimé sous sa forme logique par l'expression symbolique qui le représente un objets symbolique s est dit défini en *intension* (on dit aussi parfois en compréhension). L'ensemble des objets élémentaires de Ω pour lequels il est satisfait (*i. e.* vrai) constitue son *extension* et sera noté $|s|_\Omega$.

Pour simplifier les notations on identifiera souvent le nom s donné à l'expression symbolique de l'objet et la fonction qui lui est associée.

1.2.2. *Les «événements élémentaires»*

Il s'agit du premier type d'objet symbolique que nous allons définir.

Soient p variables $y_1, \ldots, y_p$ définient sur Ω, l'ensemble des objets élémentaires observés et prenant leurs valeurs dans $O_1, \ldots, O_p$. Si l'on identifie chaque élément de Ω avec l'ensemble des valeurs qu'il prend, on peut plonger Ω dans $\Omega' = O_1 \times \ldots \times O_p$ l'ensemble des objets élémentaires possibles. On note V_i une partie de O_i.

DÉFINITION : *Un événement élémentaire est défini par la fonction* $e_{y_i V_i}$: $\Omega \to \{vrai, faux\}$ *telle que* $e_{y_i V_i}(w) = vrai$ *si et seulement si* $y_i(w) \in V_i$.

Un événement élémentaire est noté sous la forme symbolique $e = [y_i = V_i]$ où $[y_i = V_i]$ exprime l'événement: «la variable y_i prend une valeur dans V_i». Il en résulte que $[y_i = V_i]$ est l'union logique des événements $[y_i = v]$ pour tout $v \in V$.

Ainsi pour en revenir aux appelations introduites en (1.2.2) dans le cas d'un objet symbolique défini par un événement élémentaire $[y_i = V_i]$, sa définition en compréhension notée e est $e = [y_i = V_i]$. Son extension est:

$$|e|_\Omega = \{ w \in \Omega / y_i(w) \in V_i \}.$$

Pour simplifier les notations on considérera souvent que e exprime plus qu'une simple notation et constitue une fonction identique à $e_{y_i V_i}$.

Exemple: Considérons la première variable du tableau de la figure 1, l'événement élémentaire noté $e=[y_1=\{1,3\}]$ est défini par la fonction $e_{y_1 V_1}: \Omega \to \{\text{vrai, faux}\}$ telle que $e_{y_i V_i}(w)=\text{vrai}$ si et seulement si $y_1(w) \in \{1,3\}=V_1$. Son extension est $|e|_\Omega=\{w_1, w_2, w_4\}$.

1.2.3. *Les objets assertion*

Soit $y=(y'_1, \ldots, y'_q)$ où $y'_i \in \{y_1, \ldots, y_p\}$ est l'application $\Omega \to O_i \in \{O_1, \ldots, O_p\}$. On note $V=\{V_1, \ldots, V_q\}$ avec $V_i \subset O'_i$. Un objet assertion est une conjonction d'événements élémentaires.

Plus précisément, on a la définition suivante:

DÉFINITION : *Un objet assertion noté* $a=[y'_1=V_1] \wedge \ldots \wedge [y'_q=V_q]$ *où* $V_i \subset O'_i$ *est défini par la fonction* $a_{y\,V}: \Omega \to \{\textit{vrai, faux}\}$ *telle que* $a_{y\,V}(w)=\textit{vrai}$ *si et seulement si pour tout* $i=1, \ldots, q$ *on a* $y_i(w) \in V_i$.

Remarquons que pour bien spécifier que a est une conjonction d'événements élémentaires qui doivent être vrais simultanément pour le même objet élémentaire $w \in \Omega$, on aurait pu aussi noter a sous la forme symbolique suivante:

$$a=[y'_1(w)=V_1] \wedge \ldots \wedge [y'_q(w)=V_q].$$

Nous avons préféré réserver ce type de notation au cas des «objets horde» que nous introduisons en 1.2.4.

Extension d'un objet assertion. — L'ensemble des objets élémentaires qui satisfont l'objet assertion a dans Ω sera noté $|a|_\Omega$ et constitue la définition en extension de a dans Ω. On a donc:

$$|a|_\Omega=\{w \in \Omega / y'_i(w) \in V_i \text{ pour } i=1, \ldots, q\}.$$

Exemples: Supposons que Ω soit un ensemble de champignons décrits par deux variables: y_1 exprime la taille du pied et y_2 la couleur du chapeau. Un objet élémentaire tel que $y_1=1$ et $y_2=\text{blanc}$ peut s'exprimer sous la forme de l'objet assertion suivant: $a=[y_1=1] \wedge [y_2=\text{blanc}]$.

L'ensemble des objets élémentaires de Ω' dont la taille est comprise entre 0 et 10 et la couleur du chapeau est soit marron soit noire peut s'écrire:

$$a=[y_1=[0, 10]] \wedge [y_2=\{\text{blanc, noir}\}].$$

Considérons maintenant un ensemble Ω réduit aux trois objets w_1, w_2 et w_3. Pour w_1 on a $y_1=3$ et $y_2=$noir, pour w_2 on a $y_1=2$ et $y_2=$blanc et pour w_3 on a $y_1=1$ et $y_2=$blanc. A chaque w_i on peut associer une assertion a_i telle que:

$$a_1=[y_1=3] \wedge [y_2=\text{noir}],$$
$$a_2=[y_1=2] \wedge [y_2=\text{blanc}],$$
$$a_3=[y_1=1] \wedge [y_2=\text{blanc}].$$

On peut définir l'objet assertion $a=[y_1=\{2, 3\}] \wedge [y_2=\{\text{noir, blanc}\}]$ dont nous allons chercher l'extension:

On a

$$[y_1=\{2, 3\}]=[y_1=2] \vee [y_1=3]$$

et

$$[y_2=\{\text{noir, blanc}\}]=[y_2=\text{noir}] \vee [y_2=\text{blanc}]$$

donc

$$a=[[y_1=2] \wedge [y_2=\text{noir}]] \vee [[y_1=3] \wedge [y_2=\text{noir}]] \vee [[y_1=2] \wedge [y_2=\text{blanc}]] \vee [[y_1=3] \wedge [y_2=\text{blanc}]]$$

donc $a=a_1 \vee a_2 \vee z_1 \vee z_2$ avec $z_1=[y_1=3] \wedge [y_2=\text{blanc}]$ et $z_2=[y_1=2] \wedge [y_2=\text{noir}]$. Il en résulte que les seuls objets de Ω qui satisfont a sont w_1 et w_2, autrement dit l'extension dans Ω est: $|a|_\Omega=\{w_1, w_2\}$ alors que l'extension dans Ω' est $|a|_\Omega=\{w_1, w_2, z_1, z_2\}$.

Objets assertion simplifiés: Il peut se produire que V_i soit identique à l'ensemble des valeurs possibles O_i, dans ce cas, dans certains contextes on peut supprimer pour simplifier le terme $[y_i=O_i]$ de la conjonction. Autrement dit, par exemple l'objet assertion

$$w_a=[y_1=V_1] \wedge [y_2=V_2] \wedge [y_3=O_3] \ldots \wedge [y_p=O_p]$$

(*i.e.* $V_i=O_i$ pour $i=3, \ldots, p$) peut s'écrire plus simplement $w_a=[y_1=V_1] \wedge [y_2=V_2]$.

Exemple: Supposons que la variable $y_2=$«couleur du chapeau» ne puisse prendre que les deux valeurs : blanc, noir. L'objet assertion $a=[y_1=1] \wedge [y_2=\{\text{blanc, noir}\}]$ se simplifie en $a=[y_1=1]$.

Il pourra se produire dans certains contextes que ce type de simplification ne soit pas possible pour bien distinguer les objets symboliques concernés par une variable y_i de ceux qui ne le sont pas; par exemple la variable «rayon» n'intervient pas pour la description d'un rectangle, contrairement à la variable «surface». D'autre part, si deux variables y_i et y_j prennent des valeurs dans V_i et V_j de façon que l'extension de $e_i=[y_i=V_i]$ et $e_j=[y_j=V_j]$ soit la même sur Ω, on ne s'autorisera pas à simplifier une conjonction de ces deux événements en un seul d'entre eux car les objets symboliques $s_1=e_i$ et $S_2=e_i \wedge e_j$ sont sémantiquement différents bien que leur extension soit identique. Par contre la conjonction de plusieurs événements élémentaires qui concernent la même variable peut être simplifiée en prenant l'intersection des domaines concernés. Par exemple, $s=[y_i=V_i^1] \wedge \ldots [y_i=V_i^l]$ sera simplifiée en $s=[y_i=V_i]$ avec $V_i=[\cap V_i^j/j=1, \ldots, l]$ en conservant l'équivalence logique et sans modifier la sémantique.

Exemple: Reprenons l'exemple de la figure 1. La conjonction des événements $e_1=[y_1=[1,3]]$, $e_2=[y_1=[2,4]]$, $e_3=[y_2=1]$ définit l'objet symbolique $s=e_1 \wedge e_2 \wedge e_3$ qui sera simplifié en $s=[y_1=[2,3]] \wedge [y_2=1]$ qui ne sera pas encore simplifié bien que logiquement équivalent à e_3 puisque $|e_3|_\Omega = |s|_\Omega = \{w_2, w_3\}$.

1.2.4. *Les objets horde*

Reprenons le tableau de la figure 1 et considérons les événements élémentaires $e_1=[y_1=1]$ et $e_2=[y_2=3]$. On a $e_1(w_1)=$vrai et $e_2(w_4)=$vrai mais il n'existe pas d'individu de Ω pour lesquels e_1 et e_2 soient vrais simultanément. Autrement dit, en introduisant une application $h: \Omega \times \Omega \to \{$vrai, faux$\}$ telle que $h(w', w'')=e_1(w') \wedge e_2(w'')$ on a $h(w_1, w_2)=$vrai et $h(w, w)=$faux $\forall w \in \Omega$. Pour bien faire ressortie le fait que h est une conjonction d'événements élémentaires d'instanciation pouvant concerner (contrairement aux assertions) deux objets non nécessairement identiques on notera

$$h=[y_1(u_1)=1] \wedge [y_2(u_2)=3].$$

Il s'agit d'un objet horde que l'on peut définir sous la forme générale suivante (où l'on note y_i, y_i', O_i' et V de la même façon que pour la définition des objets assertion en 1.2.3).

DÉFINITION : *Un objet horde représenté par l'expression symbolique* $h=[y_1'(u_1)=V_1] \wedge \ldots \wedge [y_p'(u_p)=V_p]$ *est défini par la fonction* h_{yV}: $\Omega^q \to \{$*vrai, faux*$\}$ *telle que* $\forall W=(w_1', \ldots, w_q') \in \Omega^q$, $h_{yV}(W)=$*vrai si et seulement si* $\forall_i y_i'(w_i') \in V_i$.

Dans le cas où l'on désire imposer $u_i = u_j$, l'objet horde devient une fonction $\Omega^{q-1} \to \{\text{vrai, faux}\}$. On peut bien sûr généraliser à plus de deux éléments identiques. Quand tous les u_i sont égaux l'objet horde se réduit à un objet assertion. On voit donc que les objets assertion constituent un cas particulier d'objet horde.

Extension d'un objet horde : Soit h un objet horde défini sur Ω^q. L'extension de h est l'ensemble des éléments $W \in \Omega^q$ tels que $h_{y'V}(W) = \text{vrai}$. On note cette extension $|h|_\Omega$ et on a donc

$$h_{y'V}^{-1}(\text{vrai}) = |h|_\Omega = \{ W = w'_1, \ldots, w'_q) \in \Omega^q / y'_i(w'_i) \in V_i \}.$$

Exemple : Considérons dans l'exemple associé à la figure 1, l'objet horde suivant : $h = [y_1(u_1) = 1] \wedge [y_2(u_2) = 2]$. On a $|h|_\Omega = \{(w_1, w_1), (w_1, w_5), (w_4, w_1), (w_4, w_5)\}$. Si l'on considère par contre l'objet assertion $a = [y_1 = 1] \wedge [y_2 = 2]$, son extension $|a|_\Omega$ est réduite à l'objet élémentaire $w_1 \in \Omega$.

Lien entre une horde h et la disjonction des événements élémentaires qui la définissent : Si l'on considère l'union logique $h' = \{ \vee e_i | i = 1, \ldots, q \}$ des événements élémentaires $e_i = [y_i = V_i]$ qui définissent la horde $h = \{ \wedge [y_i(u_i) = V_i] / i = 1, \ldots, q \}$, on obtient un événement défini par $h'_{y'V} : \Omega \to \{\text{vrai, faux}\}$ dont l'extension est l'union des extensions de chacun de ces événements élémentaires. On vérifie facilement que c'est l'union des parties $P(W_i)$ formées des éléments des q-uples qui constituent $|h|_\Omega$. Ainsi dans l'exemple qui précède avec $e_1 = [y_1(u_1) = 1]$ et $e_2 = [y_2(u_2) = 2]$ on a : $|e_1|_\Omega \cup |e_2|_\Omega = \{w_1, w_4, w_5\} = \{ \cup P(W) / W \in |h|_\Omega \}$ où par exemple $P(W_1) = \{w_1\}$ et $P(W_2) = \{w_1, w_5\}$ puisque $W_1 = (w_1, w_1)$ et $W_2 = (w_1, w_5)$.

L'intérêt des objets horde ressortira surtout quand nous ferons apparaître dans leur expression (*voir* 1.3) des liaisons entre les u_i à l'aide de méthodes ou propriétés exprimées par des prédicats à plusieurs variables.

1.2.5. *Les objets de synthèse*

Soient $\Omega_1, \ldots, \Omega_k$, k ensembles d'objets élémentaires respectivement définis sur l'ensemble de variables $J_1, J_2, \ldots, J_k$ avec card $(J_i) = P_i$. Soit H_i l'ensemble des objets horde que l'on peut définir sur Ω_i.

DÉFINITION : *Un objet de synthèse est la conjonction de k objets horde respectivement définis sur chacun des ensembles $H_1, \ldots, H_k$. Il s'écrira sous la forme générale $s = h_1 \wedge \ldots h_k$ avec $h_i \in H_i$.*

On voit donc que les objets de synthèse constituent une extension des objets vus précédemment puisqu'ils se réduisent à des objets horde ou assertion quand $k=1$. L'ensemble des éléments de $\Omega=\Omega_1 \times \ldots \times \Omega_k$ qui satisfont s constitue l'extension $|s|_\Omega$ de s.

Exemple: A partir de fenêtres Ω_1, portes Ω_2 et toits Ω_3 on peut définir des objets de synthèse appelés «type de maison».

Considérons l'ensemble des fenêtres Ω_1 décrites par l'objet horde: $h_f=$[forme fenêtre $(u)=$ronde] $\wedge$ [type vitre fenêtre $(v)=$épais].

Les couples de portes de Ω_2 satisfaisant à l'objet horde:

$h_p=$[blindage porte $(u)=$acier] $\wedge$ [matière porte $(v)=$chêne] $\wedge$ [numéro porte $(u)=12$] et les toits définis par l'objet assertion:

$a=$[type toit = ardoise] $\wedge$ [surface toit = très grande].

On définit alors l'objet de synthèse «type de maison»: $m=h_f \wedge h_p \wedge a$.

Un objet de l'extension de m dans $\Omega_1^2 \times \Omega_2^2 \times \Omega_3$ est une maison qui comporte une fenêtre ronde, une fenêtre à vitres épaisses, une porte.

Afin de définir l'extension d'un objet de synthèse il faut donner plus de précision concernant la fonction qui lui est associée :

Si l'on considère l'application

$$Y^i=(y_1^i, \ldots, y_{q_i}^i) \in J_i^{q_i} \text{ de } \Omega_i^{q_i} \to O^i=O_1^i \times \ldots \times O_{q_i}^i$$

et $V^i=V_1^i \times \ldots \times V_{q_i}^i$ avec $V_j^i \subset O_j^i$ un objet horde

$$h=[y_1^i(u_1^i)=V_1^i] \wedge \ldots \wedge [y_{q_i}^i(u_{q_i}^i)=V_{q_i}^i]$$

peut s'écrire sous forme d'un événement élémentaire $e_i=[Y^i=V^i]$.

En effet, pour

$$w=(w_1^i, \ldots, w_{q_i}^i) \in \Omega_i^{q_i}\ e_i(w)=\text{vrai} \quad \Leftrightarrow \quad Y^i(w) \in V^i$$

$$\Leftrightarrow \quad \{y_j^i(w_j^i)\} \in V_j^i\ \forall j=1, \ldots, q_i\} \quad \Leftrightarrow \quad h(w)=\text{vrai}.$$

Il en résulte qu'un objet de synthèse peut s'écrire sous forme d'un objet horde noté $s=[Y^1(U_1)=V_1] \wedge \ldots \wedge [Y^k(U_k)=V_k]$ où $U_i=(u_i^1, \ldots, u_i^{q_i}) \in \Omega_i^{q_i}$.

Posons $Y=(Y^1, \ldots, Y^k)$, $V=(V_1, \ldots, V_k)$, $U=(U_1, \ldots, U_k)$ l'objet de synthèse s est le prédicat défini par la fonction $s_{yV}: \Omega=\Omega_1^{q_1} \times \ldots \times \Omega_k^{q_k} \to \{\text{vrai, faux}\}$ telle que $s_{yV}(U)=$vrai signifie que $Y^i(U_i) \in V^i$ et plus précisément que $y_j^i(u_j^i) \in V_j^i$. Il en résulte que l'extension

d'un objet de synthèse s est :

$$|s|_\Omega = s_{yv}^{-1}(\text{vrai}) = \{ w = (w_1^1, \ldots, w_{q_1}^1, \ldots, w_{q_k}^k) \in \Omega / y_j^i(w_j^i) \in V_j^i \}$$

Exemple : En reprenant l'exemple ci-dessus, l'objet de synthèse obtenu est la fonction : $m_{yv} : \Omega_1^2 \times \Omega_2^2 \times \Omega_3 \to \{\text{vrai, faux}\}$ tel que si $W = (w_1^1, w_2^1, w_1^2, w_2^2, w_1^3)$ alors $m_{yv}(W) = \text{vrai} \Leftrightarrow \{$forme fenêtre $(w_1^1) =$ ronde, type vitre $(w_2^1) =$ chêne, numéro porte $(w_1^2) = 12$, type toit $(w_1^3) =$ ardoise, surface toit $(w_1^3) =$ très grande$\}$.

1.2.6. *Les objets règle*

L'objet règle le plus général est défini par un objet de synthèse appelé prémisse et une disjonction d'objets de synthèse appelée conclusion. Une clause de Horn est le cas particulier où la prémisse est un objet assertion et où la conclusion est un objet assertion réduit à un seul événement comme par exemple :

$r =$ [couleur du pied = bleu] $\wedge$ [région = parisienne] $\Rightarrow$ [champignon = comestible].

En reprenant l'objet de synthèse m défini dans l'exemple de 1.2.5 on obtient un objet règle plus compliqué avec par exemple :

$$r = [m \Rightarrow [\text{classe} = \text{luxe}] \wedge [\text{prix} = [200, 300]]].$$

La fonction associée à une règle est définie sur le produit des espaces sur lesquels la prémisse et la conclusion est définie. Par exemple, considérons la règle :

$r =$ [position fenêtre $(v) =$ basse] $\wedge$ [blindage porte $(u) =$ acier]

$\Rightarrow$ [matière (volet) = acier] $\wedge$ [épaisseur (volet) = grand].

Le prédicat associé à r est défini sur l'ensemble $\Omega = \Omega_f \times \Omega_p \times \Omega_v$ qui croise l'ensemble des fenêtres des portes et des volets qui satisfont aux propriétés indiquées dans r.

L'extension d'une règle est le complémentaire de l'ensemble des objets élémentaires de Ω pour lesquels elle est fausse. Elle n'est donc pas identique à l'extension de la conjonction de la prémisse et de la conclusion de la règle puisqu'elle contient tous les objets élémentaires qui ne le contredisent pas et qui ne sont pas dans cette extension.

Exemple : Reprenons l'exemple de la figure 1.1. La règle

$$r = [[y_1 = 1] \wedge [y_2 = 2] \quad \Rightarrow \quad [y_3 = 1{,}65]]$$

et l'assertion

$$a=[y_1=1]\wedge[y_2=2]\wedge[y_3=1{,}65]$$

n'ont pas la même extension puisque : $|r|_\Omega=\Omega$ alors que $|a|_\Omega=w_1$.

Remarquons qu'en utilisant des opérateurs autres que le $\wedge$ et le $\vee$, par exemple le non qui se note $\neg$, un objet assertion peut se mettre sous forme de règle fausse car on a $A\wedge B=\neg(A\Rightarrow\neg B)$ sur Ω si $A\wedge B$ a une extension non vide dans Ω.

1.3. Objets symboliques munis de méthodes et propriétés

On peut généraliser les différents objets que nous avons définis à des objets munis de méthodes et propriétés. Il suffit pour cela d'ajouter par conjonction des événements définissant par exemple des *méthodes* pour calculer une variable ou une fonction (des variables ou des objets élémentaires) ou des propriétés exprimant des liens entre les variables ou entre les objets élémentaires ou entre les variables et les objets élémentaires. Ces liens dépendent du type d'objets et seront de plus en plus complexes en allant des objets assertion jusqu'aux objets règles. Ils peuvent tous s'exprimer sous forme de fonctions de plusieurs éléments qui peuvent être des variables et des objets.

1.3.1. *Objets assertion munis de méthodes et de propriétés*

Dans ce cas les propriétés ne peuvent concerner que les méthodes de calcul de chaque variable ou des propriétés qui les relient. Si l'on note $\mathrm{Meth}(y_l, \ldots, y_k)$ la méthode de calcul de la variable y_i qui dépend des variables $y_1, \ldots, y_k$ (par exemple, si y_i est une surface, la méthode consiste à multiplier la largeur par la longueur) et si $\mathrm{Meth}(y_s, \ldots, y_t)$ est une information donnant par exemple une façon de procéder pour atteindre un objectif (par exemple pour cuisiner, calculer une taxe, etc.) et si enfin $[y_i R_k y_j]$ exprime une relation devant lier y_i et y_j (par exemple $y_i > y_j$), alors la forme la plus générale d'un objet assertion muni de méthodes et propriétés de ce type est :

$$a_p=[y_1=V_1]\wedge\ldots\wedge[y_p=V_p]\wedge[y_i=\mathrm{Meth}(y_l, \ldots, y_k)]\wedge\ldots$$
$$\wedge[\mathrm{Meth}(y_s, \ldots, y_t)]\wedge[y_t R_k y_j]\wedge\ldots\wedge[y_1 R_m y_n].$$

L'extension $|a_p|_\Omega$ de a_p dans Ω est l'ensemble des objets élémentaires qui satisfont $y_i(w)\in V_i$ et pour lesquels les méthodes et relations indiquées sont satisfaites.

Exemple :

$$a_p = [\text{couleur chapeau} = \{\text{blanc, jaune}\}] \wedge [\text{taille pied} = [0,10]]$$
$$\wedge [\text{taille chapeau} = [0,15] \wedge [\text{taille chapeau} = \text{Meth}(c)$$
$$\wedge [\text{méthode Calcong} = f(l, c)] \wedge [\text{taille chapeau} > \text{taille pied}]$$

où méthode (c) signifie que le calcul de la taille du chapeau se fera en prenant la taille de la plus grande lamelle et $f(l, c)$ est une méthode permettant le calcul de la longueur du champignon (qui n'est pas une des variables) à partir de celle du pied et du chapeau.

Une propriété peut aussi s'exprimer sous forme de règle, par exemple si l'on veut exprimer qu'une anémone ne pousse qu'à partir du mois de mars ($[y_1 = \text{mars}]$) au nord ($[y_2 = \text{Nord}]$) et dans le midi ($[y_2 = \text{Sud}]$) qu'à partir d'avril ($[y_1 = \text{avril}]$) on pourra décrire cela sous la forme d'une assertion

$$a = [y_1 = \{\text{mars, avril}\}] \wedge [y_2 = \{\text{nord, midi}\}]$$
$$\wedge [\text{si } [y_2 = \text{Nord}] \text{ alors } [y_1 = \text{mars}]] \wedge [\text{si } [y_2 = \text{sud}] \text{ alors } [y_1 = \text{avril}]].$$

Remarquons que dans ce cas simple une disjonction d'assertion aurait pu exprimer la même idée.

1.3.2. *Objets horde munis de propriétés*

En plus des méthodes et propriétés définies pour les objets assertion, on peut avoir ici des méthodes qui peuvent dépendre de plusieurs variables et objets élémentaires; on peut avoir aussi des propriétés qui relient les objets élémentaires ou qui relient la valeur prise entre les variables sur ces objets. La forme générale prise entre les variables munies de ces propriétés sera donc :

$$h_p = w_{ap} \wedge [y^1(u_1) = V_1] \wedge \ldots \wedge [y_q(u_q) = V_q] \wedge [y_l(u_i) = F(u_j, \ldots, u_k)] \wedge$$
$$\ldots \wedge [\text{meth}(y_l, y_i, \ldots, u_j, u_k)] \wedge [y_1(u) R_k y_n(v)] \wedge \ldots \wedge [y_m(w) R_1 y_k(t)].$$

L'extension de h_p dans Ω est l'ensemble des p-uples $(u_1, \ldots, u_p) \in \Omega^p$ tels que $y_i(u_i) \in V_i$ pour lesquels les méthodes et propriétés indiquées sont satisfaites.

Exemple :

$$h_p = [\text{couleur}(u_1) = \{\text{gris, noir}\}] \wedge [F(u_1, u_2)] \wedge [\text{taille}(u_1) < \text{taille}(U_3)]$$
$$\wedge [\text{vitesse à 2 } h(u_2) > \text{vitesse à 3 } h(u_4)]$$

où $F(u_1, u_2)$ signifie que u_1 se situe à gauche et en avant de u_2 (qui aurait pu s'écrire [gauche $(u_2)=u_1$] $\wedge$ [devant $(u_2)=u_1$]). Une instanciation de w_h est par exemple :

h_p(Auroch 3, Auroch 2, Auroch 6, Mamouth 4)

$$= [\text{couleur (auroch 3} = \{\text{gris, noir}\}] \wedge [[F(\text{auroch 3, auroch 2})]$$
$$\wedge [\text{taille (auroch 3)} < \text{taille (auroch 6)}] \wedge [\text{vitesse à 2 } h(\text{Auroch 2})$$
$$> \text{vitesse à 3 } h(\text{Mamouth 4})].$$

1.3.3. *Objets de synthèse munis de méthodes et propriétés*

Étant donnés k ensembles $\Omega_1, \ldots, \Omega_k$ associés respectivement à k ensembles de variables $J_1, \ldots, J_k$. On obtient un objet de synthèse en faisant la conjonction de k objets hordes h_p^i définis sur (Ω_i, J_i) et munis de méthodes et de propriétés. On peut de plus ajouter des méthodes et propriétés reliant les h_p^i entre eux, donc la forme la plus générale d'objets de synthèse peut s'écrire :

$$h_{sp} = h_p^1 \wedge \ldots \wedge h_p^k \wedge [\text{Meth}(h_p^i, \ldots, h_p^j)] \wedge \ldots \wedge [h_p^i R_k h_p^m].$$

En reprenant l'exemple ci-dessus, une méthode peut être par exemple une façon de situer les fenêtres par rapport aux portes et une propriété peut être par exemple : « la taille d'une fenêtre est inférieure à celle d'une porte ».

1.3.4. *Objets règles munis de méthodes et propriétés*

Dans leur forme la plus générale ce sont les objets règles dont la prémisse et la conclusion sont des objets de synthèse munis de méthodes et propriétés. On peut aussi ajouter des méthodes et propriétés sur des objets et (ou) des variables concernés par la règle. Par exemple une méthode permettant de calculer sa qualité en fonction de la fréquence des événements qui la constituent.

1.4. La connaissance supplémentaire

1.4.1. *Objets de la connaissance et connaissance sur les objets*

Il faut bien distinguer les objets (que l'on va étudier de façon analogue aux lignes d'un tableau de données usuelles) des connaissances supplémentaires que l'on se donne sur ces objets comme par exemple des mesures de ressemblance, des dénominations regroupant des modalités ou des contraintes exprimées sous forme de règles. Cette remarque qui est évidente en analyse des données l'est beaucoup moins dans l'approche symbolique car les mesures

de ressemblance (ou leur extension sous forme « d'affinités » dont nous allons parler maintenant) et les contraintes peuvent s'exprimer par des objets assertion, hordes ou règles. Pour bien distinguer les objets de la connaissance (*i. e.* le tableau de données) de la connaissance (supplémentaire) sur les objets, on n'utilisera plus dans le second cas le mot « objet » en disant simplement « assertion », « horde », « synthèse », et « règle ». Ainsi, par exemple, une assertion représentant une classe obtenue par classification d'objets élémentaires. Les taxonomies que nous introduisons à présent expriment une connaissance supplémentaire qui servira par la suite pour généraliser et simplifier les différents types ou classes d'objets.

1.4.2. *Les taxonomies*

1.4.2.1. Taxonomie sur les objets élémentaires

Une variable y_i telle qu'elle a été définie en 1.1 est une application de Ω, l'ensemble des objets élémentaires, dans O_i un ensemble d'observation. Comme c'est une application, chaque y_i^{-1} définit une partition. Si l'on désire que y_i^{-1} ne soit pas une partition mais un recouvrement, y_i n'est plus une application, on dira que c'est une variable taxonomique. On peut définir des variables taxonomiques particulières comme les variables hiérarchiques ou pyramidales. Une variable hiérarchique est une application t_h de $\Omega \to O$ telle que $\forall \sigma, \sigma' \in O$ on a :

$$t_h^{-1}(\sigma) \cap t_h^{-1}(\sigma') = \begin{cases} \varnothing \\ \text{ou } t_h^{-1}(\sigma) \subset t_h^{-1}(\sigma') \\ \text{ou } t_h^{-1}(\sigma') \subset t_h^{-1}(\sigma) \end{cases}$$

Une variable t_p est pyramidale si $\forall \sigma, \sigma' \in O$ $t_h^{-1}(\sigma) \cap t_p^{-1}(\sigma')$ est vide ou il existe σ'' tel que $t_p^{-1}(\sigma) \cap t_p^{-1}(\sigma') = t_p^{-1}(\sigma'')$. On peut associer une dénomination (*i. e.* un libellé) à chaque code $\sigma \in O$ d'une variable taxonomique.

Exemple : Reprenons l'exemple de la figure 1 en 1.1. On peut indiquer une connaissance supplémentaire sur la variable classe d'âge à l'aide de la hiérarchie H de la figure 2 et sur la variable taille à l'aide de la pyramide P de la même figure.

Aux paliers 1, 2, 3, de la hiérarchie, on peut associer les dénominations jeunes, adulte, quelconque. De même aux paliers 1, 2, 3, 4, 5, de la pyramide on peut associer respectivement petit, moyen, moyen petit, grand, quelconque. On a donc $t_h^{-1}(\text{jeunes}) = \{ w_1, w_4, w_3 \}$ et $t_p^{-1}(\text{petit}) = \{ w_3, w_4 \}$.

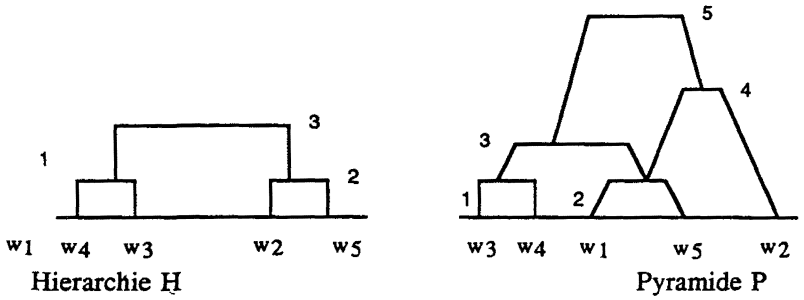

Figure 2

On peut démontrer (*voir* Diday 86 par exemple) qu'il existe toujours un ordre O sur Ω compatible avec toutes les parties $t^{-1}(s)$ définies par une hiérarchie. Autrement dit, tel que toute partie $t^{-1}(s)$ définisse un intervalle de O. Cette condition peut ne pas être satisfaite par les parties induites par une variable pyramidale, elle est alors difficilement représentable graphiquement. Remarquons aussi que l'ensemble des parties de Ω définies par une variable hiérarchique ou pyramidale munie de l'ordre des inclusions forme un treillis si l'on ajoute la partie vide.

Pour adopter deux points de vue différents sur la connaissance supplémentaire associée à une variable on peut être amené à construire deux hiérarchies. Si elles sont compatibles avec le même ordre, elles induisent une variable pyramidale en considérant que le code associé à un groupement d'objet est celui du plus bas palier (des deux hiérarchies) qui le contient. Pour avoir une pyramide il suffit de créer les paliers associés aux intersections non vides des paliers pris dans chacune des deux hiérarchies et de leur associer le code le plus bas de ces deux paliers.

En procédant comme indiqué ci-dessus sur les paliers h_1, h_2, h_3 de la hiérarchie H_1 et h'_1, h'_2, h'_3 de la hiérarchie H'_1 de la figure 3 on construit les paliers $h_1 \cap h'_2 = h_1$, $h_2 \cap h'_3 = h_2$, $h_3 \cap h'_1 = h'_1$, $h_3 \cap h'_2 = h'_2$, $h_3 \cap h'_3 = h'_3$ de la pyramide P (voir *fig.* 3).

1.4.2.2. Taxonomie associée aux variables

Une taxonomie sur des variables permet des regroupements de variables qui peuvent être utiles. Les nouvelles variables ainsi créées ont un espace de définition plus compliqué. Ainsi si l'on regroupe des variables y_1 et y_2 de $\Omega \to O_1$ et $\Omega \to O_2$ en une variable nouvelle Y_1, c'est nécessairement une application de $\Omega \to O_1 \times O_2$. Afin de ne pas démultiplier le nombre de valeurs

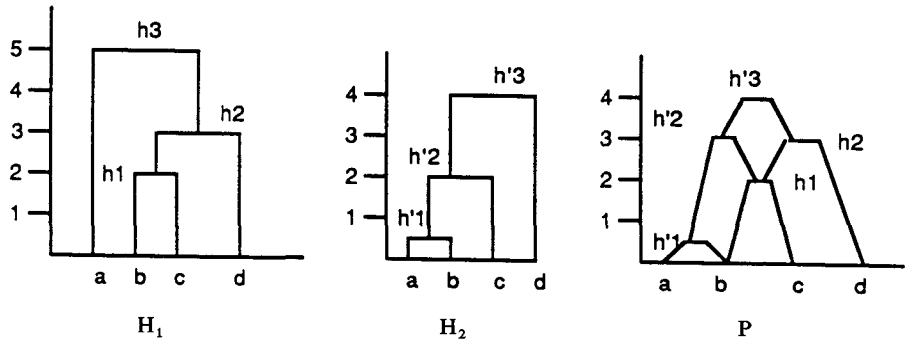

Figure 3

possible on peut être conduit à utiliser une taxonomie définissant des parties de $O_1 \times O_2$.

Exemple : En reprenant l'exemple de la figure 1, on peut regrouper par exemple les variables âge et taille en une nouvelle variable appelée « âgetaille » de $\Omega \to \mathbb{N} \times \mathbb{R}^+$ qui est fonction des valeurs prises par chaque objet élémentaire sur les variables âge et taille. On peut utiliser une combinaison des taxonomies que nous avons défini figure 2 pour simplifier le domaine des valeurs prises par la variable « agetaille » et pour avoir une taxonomie sur ces valeurs. On peut par exemple définir un palier avec les « adultes petits » que l'on appelera « nain » les « grands jeunes » que l'on appellera géants etc.

1.4.3. *Indiçage et affinités*

1.4.3.1. Indiçage d'une taxonomie

Afin de définir une hiérarchie H ou une pyramide il suffit de connaître les parties de Ω qui les forment. Pour les représenter graphiquement et pour avoir une mesure de la cohérence de ces parties (appelées aussi « paliers »), on est conduit à leur associer une valeur numérique. Cette valeur permet d'associer une hauteur à chaque palier de la représentation graphique et s'appelle « indiçage ». Plus précisément, on définit une application f de H ou P dans $\mathbb{R}^+$ telle que $f(w)=0$, $\forall w \in \Omega$ et qui doit satisfaire à la condition suivante : $f(h) \leqq f(h')$ si $h \subset h'$. Cette condition évite d'avoir des inversions, à savoir qu'un palier se trouve à un niveau plus bas qu'un palier qu'il contient. Une hiérarchie munie d'un indiçage est appelée « hiérarchie indicée ».

Exemple : Soit $\Omega = \{w_1, w_2, w_3, w_4\}$ et les hiérarchies indicées H_1 et H_2 de la figure 4. Les deux hiérarchies sont identiques et notées H, elles diffèrent

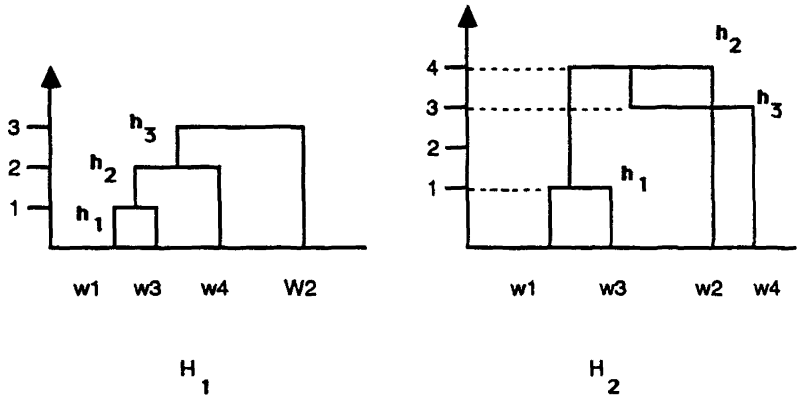

Figure 4

par leur indiçage. L'indiçage de la hiérarchie H_1 est définie par l'application $f_1 : H \to \mathbb{R}^+$ telle que $f_1(w)=0$, $\forall w \in \Omega$ et $f_1(h_1)=1$, $f_1(h_2)=2$, $f_1(h_3)=3$. La hiérarchie H_2 est définie par l'application $f_2 : H \to \mathbb{R}^+$ telle que $f_2(w)=0$, $\forall w \in \Omega$ et $f_2(h_1)=1$, $f_2(h_2)=4$ et $f_2(h_3)=2$. On voit que f_2 donne lieu à une inversion car la condition $f_2(h) \leqq f_2(h')$ si $h \subset h'$ n'est pas satisfaite si $h=h_2$ et $h'=h_3$.

1.4.3.2. Les affinités

L'indiçage d'une taxonomie exprime un degré de cohérence entre les modalités associées à un palier. Les affinités permettent de généraliser cette notion à tout groupe d'objets non nécessairement liés à une taxonomie. Une affinité est une application qui associe à tout ensemble de valeurs prises par des objets élémentaires, sur des variables qui les concernent, une valeur numérique positive négative ou nulle.

Exemple : aff(manteau(Jeanne), chapeau(Julie), pantalon(Josette)) $= -10$, si l'affinité entre le manteau de Jeanne, le chapeau de Julie (qui sont par exemple en vison) et le pantalon de Josette (un jean!) est considéré comme faible.

L'affinité entre deux objets peut se mesurer comme la somme des affinités des valeurs prises par ces objets sur les variables qui les concernent.

Exemple : Reprenons l'exemple des champignons. Si l'on considère que les champignons jeunes ont un pied petit et le chapeau marron et que les champignons plus âgés ont un grand pied et un chapeau de couleur claire, on peut définir les affinités suivantes (supposées symétriques) :

$$\text{aff}(TP \geqq 3,\ CP = \text{marron}) = -1;\ \text{aff}(TP \geqq 3,\ CP = \text{claire}) = +1;$$

$$\text{aff}(TP < 3,\ CP = \text{marron}) = +1;\ \text{aff}(TP < 3,\ CP = \text{claire}) = -1;$$

$$\text{aff}(TP = x,\ TP = y) = |x - y| \text{ et aff}(CP = x,\ CP = y) = 0 \text{ si } x \neq y \text{ et } 1 \text{ si } x = y$$

d'où l'affinité entre les objets $w_1 = [TP = 3] \wedge [CP = \text{claire}]$ et

$w_2 = [TP = 2] \wedge [CP = \text{marron}]$:

$$\text{aff}(w_1, w_2) = \text{aff}(TP = 3,\ TP = 2) + \text{aff}(TP = 3,\ CP = \text{marron}) + \text{aff}(CP = \text{claire},\ TP = 2) + \text{aff}(CP = \text{claire},\ CP = \text{marron})$$

d'où $\text{aff}(w_2, w_4) = 1 + (-1) + (-1) + 0 = -1$.

La connaissance supplémentaire fournie par l'utilisateur concernant les affinités peut s'exprimer sous différentes formes. Il peut donner une liste non exhaustive d'affinités entre variables, entre modalités de variables qualitatives ou entre objets. Dans le cas de variables ou de modalités (transformées en variables) on peut exprimer ces affinités en terme d'assertion contenant une méthode de calcul de l'affinité, par exemple :

$$w_a = [\text{affinité}(y_i, y_j, \ldots, y_1)] \wedge a$$

dont l'extension est l'ensemble des objets élémentaires pour lesquels l'affinité entre variables est définie par affinité $(y_i, \ldots, y_1)$ et qui satisfont l'assertion a.

La méthode « affinité » peut être soit une simple valeur, soit une façon de calculer l'affinité à partir des données. Dans le cas de deux variables on pourra prendre toutes les mesures de ressemblances classiques en analyse des données utilisant par exemple les corrélations dans le cas de variables quantitatives ou les contingences pour les variables qualitatives.

L'affinité entre objets peut s'exprimer sous forme de horde par exemple :

$$w_h = [\text{affinité}(u_1, u_2, \ldots, u_p)] \wedge h(u_1, u_2, \ldots, u_p)$$

où l'extension de w_h est l'ensemble des p-uples d'objets $(u_1, \ldots, u_p)$ dont l'affinité se calcule par affinité $(u_1, u_2, \ldots, u_p)$ et qui satisfont l'objet horde $h(u_1, u_2, \ldots, u_p)$.

Ici aussi cette affinité peut être une simple valeur fournie par l'utilisateur ou une méthode de calcul. Dans le cas de deux objets on pourra aussi utiliser

dans la mesure du possible, les indices de ressemblance classiques en analyse des données, comme la distance euclidienne. L'utilisation des affinités et surtout le calcul des distances au sens usuel est toujours possible (voir le paragraphe suivant) avec les nouveaux objets que nous avons introduit mais pour utiliser toute la puissance du symbolique on sera amené à utiliser aussi d'autres notions pour effectuer les comparaisons comme par exemple celle d'héritage (*voir* 4.3).

1.4.3.3. *Utilisation des taxonomies pour construire une dissimilarité entre objets assertion*

On suppose disposer d'un tableau de données à n lignes et p colonnes (voir figure 5) où chaque case contient $V_i^j \subset O_j$. Suivant le choix des O_j, V_i^j peut être aussi bien un intervalle ou une union d'intervalle (cas d'une variable

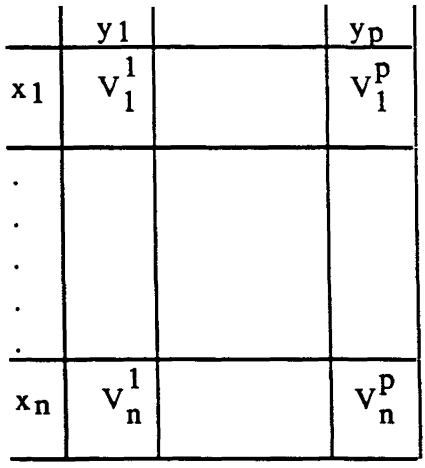

Figure 5

quantitative) qu'un ensemble de valeurs discrètes (cas d'une variable qualitative).

A chaque ligne x_i on peut associer l'objet assertion $a_i = [y_1 = V_i^1] \wedge \ldots \wedge [y_p = V_i^p]$. Nous supposons disposer d'une taxonomie sur les valeurs prises par les variables. Si ce n'est pas le cas, on peut la calculer à partir des exemples $x_1, \ldots, x_n$. Par exemple, dans le cas de variables quantitatives, la taxonomie se déduit d'une hiérarchie H_j dont les racines sont des singletons associés aux extrémités des intervalles V_i^j (ordonnés par ordre de valeurs croissantes) et les paliers sont obtenus en utilisant un algorithme de classification hiérarchique ou pyramidale muni d'un indice

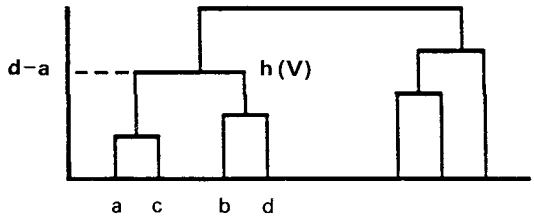

Figure 6. V=[a, b]

classique. On associe ensuite à chaque V_i^j le plus bas palier qui le contient. Par exemple (voir *fig.* 6), si $V_i^j = V = [a, b]$, la hauteur de $h(V)$ est d-a si l'indice d'agrégation utilisé est celui de la distance maximum entre les singletons pris dans chaque palier. Dans le cas de variables qualitatives, si la taxonomie sur les modalités n'est pas fournie, on peut la calculer en considérant, par exemple, que deux modalités sont d'autant plus proches que les exemples qui les concernent prennent des valeurs proches sur les autres variables. Si les deux modalités que l'on veut comparer appartiennent à des variables différentes il faut tenir compte de plus de leur nombre d'occurences communes sur chaque ensemble de l'ensemble d'apprentissage. A l'aide de cet indice de proximité on peut comme dans le cas des variables quantitatives construire une taxonomie sur les valeurs prises (ici, les modalités). La distance entre objets assertion peut alors se calculer, par exemple, de la façon suivante où $h_j(V)$ est la hauteur du plus bas palier de la taxonomie H_j qui contient V :

$$D(a_l, a_k) = \sum_{j=1}^{p} \alpha^j(V_l^j, V_k^j) \text{ où } \alpha^j(V_l^j, V_k^j) = (h_j(V_l^j \cup V_k^j) - h_j(V_l^j)) + (h_j(V_l^j \cup V_k^j) - h_j(V_k^j)) = 2h_j(V_l^j \cup V_k^j) - (h_j(V_l^j) + h_j(V_k^j)).$$

Remarquons que les taxonomies peuvent être aussi utilisées en segmentation pour trouver les endroits où couper chaque variable explicative en utilisant les paliers stables vis-à-vis des exemples et contre-exemples.

1.4.4. *Les règles*

Les règles de la connaissance supplémentaire, n'interviennent pas au niveau de la construction des classes puisqu'elles ne font pas partie des objets à classifier, car dans ce cas ce serait des « objets règles ». Elles interviennent pour tenir compte de certaines contraintes provenant (de façon fréquente dans la pratique) de la multiplicité des valeurs dans chaque case du tableau

ou comme conséquence de classes d'objets proposés par l'utilisateur ou obtenus par un algorithme de classification.

Par exemple, l'objet assertion $a = [\text{couleur chapeau} = \{ \text{marron, clair} \}] \wedge [\text{longueur pied} = [0,15]]$ peut nécessiter la règle : si [couleur chapeau = clair] alors [longueur pied < 7]. Autre exemple, considérons les trois règles suivantes

$A_1 \wedge A_2 \Rightarrow A$, $B_1 \wedge B_2 \Rightarrow B$, $A \wedge B \Rightarrow C$. Si les objets symboliques $a = A_1 \wedge A_2$ d'une part et $b = B_1 \wedge B_2$ représentent deux classes différentes, la conjonction de ces représentations permettra d'appliquer la règle $A \wedge B \Rightarrow C$ où C peut être par exemple une méthode de fabrication, un positionnement, une alerte, etc.

Dans l'exemple des maisons, une règle pourra par exemple s'exprimer sous la forme : « si porte blindé et fenêtres sans volets alors mettre les fenêtres hautes ». Si une classe de maisons est définie par la propriété d'avoir une porte blindée et des fenêtres sans volets on lui imposera d'avoir les fenêtres hautes.

2. DES DONNÉES NUMÉRIQUES DE L'ANALYSE DES DONNÉES AUX DONNÉES LOGIQUES DE L'APPROCHE SYMBOLIQUE

2.1. Les données numériques

Les données classiques de l'analyse des données se présentent généralement sous forme d'un ensemble d'objets élémentaires Ω caractérisé par un nombre fini p de variables qualitatives ou quantitatives. Si l'on considère ces objets comme des points de l'espace $\mathbb{R}^p$ en transformant les variables qualitatives en variables binaires, nous dirons que de telles données sont numériques.

Il peut se produire que l'on privilégie une ou plusieurs variables (qualitatives ou quantitatives) particulières dans le tableau de données. Dans ce cas, chaque objet peut être considéré comme un « objet règle élémentaire » dont la conclusion est définie par la valeur des variables privilégiées et la prémisse est la valeur des autres variables.

En analyse des données la connaissance supplémentaire est définie par le choix d'une mesure de dissimilarité entre les objets. Cette mesure est généralement utilisée pour réaliser une classification des objets ou une représentation plane par analyse factorielle.

2.2. Les données logiques de l'approche symbolique

Comme nous venons de le voir en 1.2, 1.3, 1.4 les objets symboliques se présentent sous forme de conjonction d'événements. Ils sont de différentes sortes et peuvent se présenter sous forme d'objets élémentaires, d'objets assertion, d'objets hordes, d'objets de synthèse, d'objets règle (avec ou sans méthodes et propriétés). La connaissance supplémentaire est définie (*voir* 1.4) par les taxonomies, les affinités et les règles.

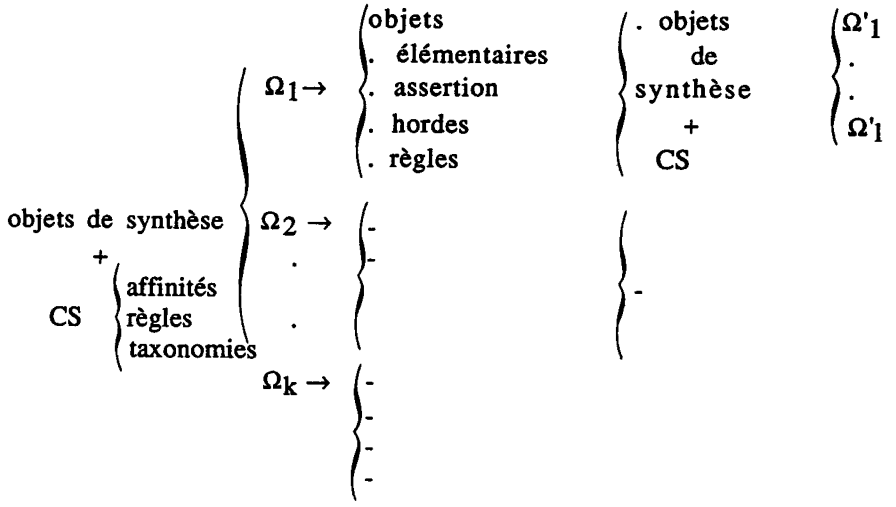

Figure 7

La représentation de la connaissance peut nécessiter uniquement des objets d'un seul type, par exemple, que des objets assertion (voir *fig.* 5) ou que des objets hordes (dans ce cas le tableau des objets élémentaires est « fictif »). Elle peut aussi nécessiter des combinaisons beaucoup plus complexes. Par exemple (voir *fig.* 7) des objets de synthèses (munis d'affinités et de règles et taxonomies) obtenus à partir d'espaces $\Omega_1, \ldots, \Omega_k$ eux-mêmes formés d'objets élémentaires, assertions, hordes, eux-mêmes définis comme objets de synthèse munis de connaissance supplémentaires, etc. Par exemple, des types de maisons peuvent être considérés comme des objets de synthèse de portes, fenêtres, toits, murs, eux-mêmes considérés comme objets de synthèse, etc.

2.3. Comparaison des données de l'analyse des données et des données de l'approche symbolique

Il y a une relation biunivoque entre les objets élémentaires de l'analyse des données et ceux de l'approche symbolique. En effet, l'application qui associe un élément de $\mathbb{R}^p$ à la conjonction des événements associés aux valeurs prises par ses coordonnées est une bijection. Les objets règles de l'analyse des données peuvent donc s'exprimer sous forme d'une implication de deux objets élémentaires et constituent ainsi un cas particulier d'objets règles symboliques (*voir* 1.2.6).

Les objets de l'Approche numérique	Les objets de l'approche symbolique
Objets élémentaires (éléments de $\mathbb{R}^p$)	. Objets élémentaires (conjonction d'évènements)
	. Objets assertion (horde, de synthèse)
Objets règles (implication entre éléments de $\mathbb{R}^p$)	. Objets règles (implication entre objets symboliques)
La connaissance supplémentaire	
Sémantique des nombres	Sémantique du domaine
Indice de dissimilarités	Affinités
	Taxonomies
	Règles

Figure 8

En ce qui concerne la connaissance supplémentaire les indices des dissimilarité constituent des cas particuliers d'affinité, les taxonomies sont très rarement utilisées en analyse des données (pour exprimer une connaissance supplémentaire). La notion de règle telle qu'elle a été introduite en 1.2.6 n'apparaît pas en analyse des données. Par exemple, les conséquences déduites des classes d'une classification ne sont pas prévues dans les programmes classiques d'analyse des données. Toute cette comparaison est résumée en figure 8.

3. LE TRAITEMENT DES DONNÉES SYMBOLIQUES

3.1. Le problème général et les quatre principes

Comme pour l'analyse des données numériques il s'agit de résumer et synthétiser l'information contenue dans les données symboliques pour servir de base à un processus de décision, de reconnaissance ou plus généralement pour appréhender la nature des phénomènes sous-jacents aux données.

Afin de répondre à ce problème, on peut adopter deux points de vue. Le premier plus « numérique » consiste dans la mesure du possible ! à adapter les méthodes classiques de l'analyse des données aux nouveaux objets qui ont été définis en 1.2 et 1.3. On adaptera par exemple l'analyse factorielle au traitement des objets assertion : comment, par exemple en entrée, faire une analyse factorielle ou une classification automatique d'un tableau où dans chaque case apparaît un intervalle différent ?

Le second point de vue plus « symbolique » *qui sera adopté dans toute la suite de ce texte* consiste à appliquer les quatre principes suivants :

(*a*) *Principe de fidélité :* les données doivent être fidèles à la réalité multidimensionnelle en évitant le plus possible l'artéfact des codages réducteurs.

On acceptera ainsi les valeurs multiples dans les cases d'un tableau de données, en ne les réduisant surtout pas à leur valeur moyenne. On traitera les variables hiérarchiques (par exemple la variable « existence du chapeau » précède la variable « couleur du chapeau »). On associera aux objets des méthodes et des connaissances supplémentaires correspondant à la sémantique la plus adéquate à leur domaine.

(*b*) *Principe de la prédominance de la connaissance :* la connaissance dirige les algorithmes par la sémantique du domaine (les taxonomies, affinités, règles), nouvelles données, questions posées, etc.

Autrement dit, la connaissance supplémentaire qu'elle soit fournie par interactivité homme-machine (question posées) ou machine-machine (mesure de la qualité des résultats obtenus), guide les algorithmes.

Il existe de nombreuses méthodes d'analyse des données classique où les algorithmes sont dirigés par les questions posées citons par exemple : la regression par boule, où l'on fait une régression associée à la boule correspondant à la question posée, il en est de même pour la discrimination par les k plus proches voisins, citons enfin la segmentation dirigée par les données où à chaque réponse le système propose une série de questions par ordre de pouvoir discriminant (*voir* J. Lebbe, 1984, par exemple).

L'approche symbolique est particulièrement propice au dialogue homme-machine car les résultats sont d'interprétation aisée. Ainsi dans un processus d'apprentissage automatique une machine peut fournir des exemples et contre-exemples par simulation suivant la qualité des règles obtenues par l'algorithme et conduire ainsi à une modification de son comportement.

(*c*) *Principe de cohérence :* les résultats des analyses devront s'exprimer selon les mêmes termes que les objets fournis au départ.

Ce principe est utilisé de façon naturelle en analyse des données puisque les objets de départ sont des vecteurs numériques comme les résultats (combinaisons linéaires des variables en régression ou en analyse factorielle, centre de gravité des classes en classification, par exemple).

L'application de ce principe aux données symboliques signifie que les résultats ne seront pas seulement numériques mais devront s'exprimer en terme d'objets assertion, hordes, de synthèse ou règles. Réciproquement si l'on cherche des résultats s'exprimant sous forme d'objets symboliques (dans un problème de regression, discrimination ou classification automatique, par exemple) il faudra également considérer les données d'entrée comme des objets symboliques.

(*d*) *Principe d'explicabilité :* il faut fournir des résultats explicables et d'utilisation aisée pour définir une base de connaissance dans le domaine d'où sont issues les données même si c'est au détriment de l'efficacité.

Ce principe est dans l'ordre de préoccupations des recherches en intelligence artificelle [*voir* Kodratoff (1986), p. 21] puisqu'il s'agit d'obtenir des résultats qui s'expriment dans les termes que manipulent l'intelligence humaine. Il va de soi qu'un expert exprimera plus facilement des faits observés sous forme d'objets assertion que de combinaison linéaire même si cette combinaison donne des résultats sensiblement meilleurs. De façon générale, on voit que les principes de cohérence et d'explicabilité se rejoignent pour les données de l'approche symbolique puisque les deux conduisent à utiliser en sorties les objets symboliques. En effet, d'une part ce sont ceux qui sont utilisés en départ et d'autre part ils fournissent des résultats d'interprétation immédiate et d'utilisation aisée pour construire une base de connaissance d'un système expert par exemple.

Remarquons que cette convergence des deux principes distingue le second point de vue du premier où le second principe vient contredire le premier. En effet, si l'on sort des données symboliques (pour être « explicatif ») alors que l'on s'impose de n'entrer que des données considérées comme numériques on contredit le principe de cohérence.

3.2. Quelques problèmes d'analyse de données symboliques

Il s'agit de partir d'un ensemble d'objets A (qui peut aller en s'agrandissant) et d'obtenir un ensemble d'objets B plus simple (qui peut aller en s'améliorant). Dans le cas où A et B sont formés chacun d'un même type d'objets (tous les objets de A sont des assertions par exemple) on peut schématiser un ensemble de problèmes d'analyse des données symboliques par les cheminements indiqués figure 9 où chaque flèche indique l'espace A d'où l'on part et l'espace d'arrivée B des connaissances apprises. L'un des premiers algorithmes de "conceptual clustering" [*voir* Michalsky, Diday, Stepp (1982)] entre dans le cadre de la flèche 1. Le travail de Ganashia (1987) et Ralambondrainy (1987) correspond à la troisième flèche; Quinqueton et Sallantin (1986) et Ho Tu Bao, Diday, Summa (1987) ont mis au point des algorithmes d'apprentissage correspondant à la flèche 4.

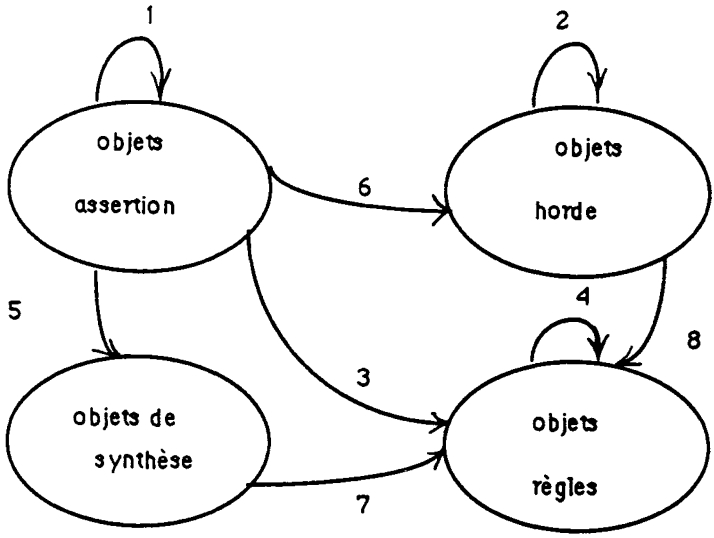

Figure 9

4. QUELQUES PROPRIÉTÉS DES OBJETS SYMBOLIQUES

4.1. Une expression générale des objets symboliques

Bien que sémantiquement différents, les objets élémentaires, assertion, horde et de synthèse peuvent se mettre sous une forme équivalente d'un point de vue algébrique. Cela résulte de la proposition suivante.

PROPOSITION 1 : 1. Si E, A, H, S sont respectivement l'ensemble des objets élémentaires, assertion, horde et de synthèse alors on a $E \subset A \subset H \subset S$.

2. Il est toujours possible de trouver un ensemble d'objets élémentaires Ω^*, un ensemble d'observation O^* et une variable $Y : \Omega^* \to O^*$ telle que tout objet symbolique de E, A, H et S soit considéré comme un événement élémentaire $e = [Y = V]$.

Démonstration. — La première propriété a été démontrée au paragraphe 1, elle résulte facilement des définitions des différents types d'objets symboliques.

La seconde propriété peut se démontrer en considérant successivement chaque type d'objet symbolique : un événement élémentaire quelconque $e_i = [y_i = V_i]$ peut bien sûr se mettre sous la forme générale $e = [Y = V]$ en choisissant $Y = y_i$ et $V = V_i$. Considérons un objet assertion $a_i = [y_1^i = V_1^i] \wedge \ldots \wedge [y_{q_i}^i = V_{q_i}^i]$. On peut choisir $Y = (y_1^i, \ldots, y_{q_i}^i)$ application de Ω dans $O = O_1^i \times \ldots \times O_{q_i}^i$ telle que $Y(w) = (y_1^i(w), \ldots, y_{q_i}^i(w))$. En posant $V = (V_1^i, \ldots, V_{q_i}^i)$, on voit que $e = [Y = V]$ et a_i sont équivalents car

$$e(w) = \text{vrai} \Leftrightarrow \{ Y(w) \in V \} \Leftrightarrow \{ \forall j, y_j^i(w) \in V_j^i \} \Leftrightarrow a_i(w) = \text{vrai}.$$

On peut faire une démonstration analogue dans le cas des objets hordes en considérant que $Y = (y_1^i, \ldots, y_{q_i}^i)$ est une variable de Ω^{q_i} dans $O^i = O_1^i \times \ldots \times O_{q_i}^i$ (voir § 1.2.5). De même pour les objets de synthèse en posant $y^{q_i} = (y_1^i, \ldots, y_{q_i}^i)$ et en considérant que $Y = (y^{q_1}, \ldots, y^{q_k})$ est une variable définie sur $\Omega_1^{q_1} \times \ldots \times \Omega_k^{q_k}$ et prenant ses valeurs dans $O_1^{q_1} \times \ldots \times O_k^{q_k}$ et que $V = (V^1, \ldots, V^k)$ avec $V^i = (V_1^i, \ldots, V_{q_i}^i) \subset O_i^{q_i}$. ■

Conséquences : (*a*) D'après cette proposition un objet de synthèse, par exemple, peut être considéré comme un événement élémentaire et il est alors possible de définir de nouveaux objets assertion, horde, de synthèse avec ce type d'événements élémentaires. Ces objets symboliques peuvent être considérés à leur tour comme des événements élémentaires définis sur de nouveaux espaces Ω^* et O^* et ainsi de suite. Ainsi la représentation symbolique peut être compliquée à l'infini pour représenter la réalité multidimensionnelle.

(*b*) Il résulte aussi de cette proposition que toute définition ou propriété algébrique satisfaite par un type d'objet symbolique et non spécifique à son type sera satisfaite par les autres. En effet, si l'on donne par exemple une définition ou une propriété qui concerne les objets assertion, elle est vrai pour les objets élémentaires (cas particulier d'objets assertions) et elle est aussi vraie pour les objets horde ou de synthèse (même ceux qui ne sont pas des objets assertion) car ils peuvent également se mettre sous forme d'événements élémentaires.

4.2. Ensemble des objets symboliques et « extension symbolique »

Afin de simplifier les notations nous supposerons dans toute la suite que l'on dispose d'un ensemble d'objets symboliques noté S défini sur un ensemble Ω caractérisé par des variables $y_i : \Omega \to O_i$. D'après la proposition 1 en choisissant bien y_i, Ω, et O_i, S peut aussi bien être un ensemble d'événements élémentaires, d'objets assertion, horde ou de synthèse. Dans la suite nous supposerons que S est l'ensemble des objets assertion car ces objets sont plus explicites que les événements élémentaires tout en étant d'expression plus simple que les objets horde ou de synthèse.

La bijection φ entre Ω et S :

Soit φ (voir *fig.* 10) l'application $\Omega \to S$ qui associe à tout élément de Ω l'assertion $\varphi(w) = [y_1 = y_1(w)] \wedge \ldots \wedge [y_p = y_p(w)]$.

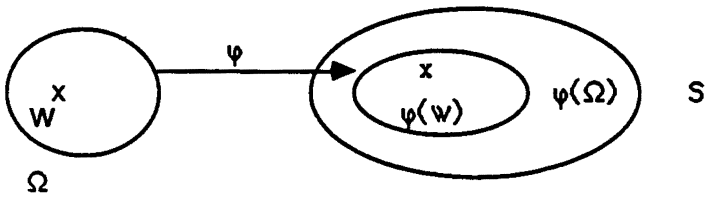

Figure 10

On suppose que deux éléments différents de Ω ne peuvent prendre de valeurs identiques sur toutes les variables. Il résulte de cette condition que φ est une bijection de Ω dans $\varphi(\Omega)$. On notera pour simplifier w^s l'objet symbolique de S associé à l'objet élémentaire $w \in \Omega$ par l'application φ [*i. e.* $w^s = \varphi(w)$].

Extension symbolique : Par la suite l'extension « symbolique » d'un objet symbolique $s \in S$ sera notée s' et sera formée de l'ensemble des objets symboliques élémentaires $\varphi(w) \in S$ tels que $w \in \Omega$ appartienne à l'extension de s dans Ω. Autrement dit : $s' = \{\varphi(w) \in S / w \in |s|_\Omega\}$. On peut maintenant définir un ordre sur les objets symboliques.

4.3. Ordre, héritage et treillis des objets symboliques

Définition de l'ordre symbolique :

$$\forall s_1, s_2 \in S \text{ on dit que } s_1 \leqq s_2 \text{ si et seulement si } s'_1 \subseteq s'_2.$$

L'inclusion étant en ordre sur les parties de Ω, l'inégalité Ω que nous venons d'introduire est donc un ordre sur les objets symboliques donc le terme « ordre symbolique » est justifié.

Définition de l'héritage et de la généralisation :

$\forall s_1, s_2 \in S$ *on dit que* s_1 *hérite de* s_2 *et que* s_2 *est plus général que* s_1 *si et seulement si* $s'_1 \subset s'_2$.

L'ensemble des objets symboliques s tels que $s \geqq s_1$ (resp $\leqq s_1$) sont les ascendants (resp. les descendants) de s.

Définition de l'union et de l'intersection :

$s_1 \cup s_2$ *(resp.* $s_1 \cap s_2$*) est la conjonction de tous les objets symboliques de S dont l'extension symbolique contient l'ensemble des objets symboliques de* s'_1 *et* s'_2 *(resp. les objets symboliques communs à* s'_1 *et* s'_2*).*

Nous allons introduire des conventions de notation qui vont nous permettre de faire deux remarques concernant cette définition.

Dans toute la suite, on adoptera les conventions de simplification suivantes :

(i) $s \wedge [y_i = O_i] = s$ *et* $[y_1 = O_1] \wedge \ldots \wedge [y_p = O_p] = \Omega^s$ *où* Ω^s *est l'objet symbolique dit « plein »;*

(ii) $[y_i = v_i] \wedge [y_i = V_i] = [y_i = v_i]$ *si* $v_i \subset V_i$, $= [y_i = \varnothing]$ *si* $v_i \cap V_i = \varnothing$ *et* $[y_1 = \varnothing] \wedge \ldots \wedge [y_p = \varnothing] = \Phi^s$ *où* Φ^s *est l'objet symbolique dit « vide ».*

Avec la convention (ii), on voit que la définition de l'union et de l'intersection peut s'exprimer de façon équivalente en remplaçant dans la première ligne « objets symboliques » par « événements élémentaires de plus petite extension ».

D'autre part, on voit que $s'_1 \cap s'_2 = \varnothing \Leftrightarrow |s_1|_\Omega \cap |s_2|_\Omega = \varnothing \Leftrightarrow s_1 \cap s_2$ est la conjonction de tous les événements élémentaires dont l'extension est vide dans Ω. Parmi ces événements se trouvent en particulier les $[y_i = \varnothing]$ pour $i = 1, \ldots, p$. Il en résulte que $s_1 \cap s_2 = [y_1 = \varnothing] \wedge \ldots [y_p = \varnothing]$ et que $s_1 \cap s_2$ est donc l'objet vide Φ^s. De même $s'_1 \cup s'_2 = (\Omega) \Leftrightarrow |s_1|_\Omega \cup |s_2|_\Omega = \Omega \Leftrightarrow \{ s_1 \cup s_2$ est la conjonction de tous les événements élémentaires dont l'extension est $\Omega \} \Leftrightarrow s_1 \cup s_2 = [y_1 = O_1] \wedge \ldots \wedge [y_p = O_p]$ si l'on suppose que les $y_i : \Omega \to O_i$ sont surjectives, ce que l'on supposera dans toute la suite, en attribuant une valeur particulière (par commodité, pour les raisonnements) aux données manquantes et à celles qui n'existent pas (par exemple, couleur du chapeau d'un champignon qui n'a pas de chapeau) à ne pas confondre avec la valeur vide. Il en résulte que $s_1 \cup s_2$ est l'objet symbolique plein. On peut résumer

toutes ces remarques dans la proposition suivante qui sera utile par la suite :

PROPOSITION 2 : 1. *L'union (resp. l'intersection) de deux objets symboliques* s_1 *et* s_2 *est la conjonction de tous les événements élémentaires de plus petite extension dont l'extension symbolique contient* $s'_1 \cup s'_2$ *(resp* $s'_1 \cap s'_2$*).*

2. *L'union (resp. l'intersection) de deux objets symboliques* s_1 *et* s_2 *est l'objet symbolique plein :* Ω^s *(resp. vide :* Φ^s*) si et seulement si* $s'_1 \cup s'_2 = \varphi(\Omega)$ (resp. $s'_1 \cap s'_2 = \varnothing$).

Exemple : Considérons le tableau de données de la figure 11. Les variables y_i sont des applications de $\Omega = \{w_1, w_2, w_3, w_4, w_5\}$ dans $O_i = \{0,1\}$. On a

$$\varphi(w_1) = w_1^s = [y_1 = 1] \wedge [y_2 = 0] \wedge [y_3 = 0] \wedge [y_4 = 0].$$

$$\varphi(w_2) = w_2^s = [y_1 = 1] \wedge [y_2 = 0] \wedge [y_3 = 1] \wedge [y_4 = 0].$$

$$(w_1^s)' \cup (w_2^s)' = \{w_1^s, w_2^s\} \quad \text{donc} \quad s_1 \cup s_2 = [y_1 = 1] \wedge [y_2 = 0] \wedge [y_4 = 0].$$

$$[y_1 = 1]' \cup [y_2 = 1]' = \varphi(\Omega) \quad \text{donc} \quad [y_1 = 1] \cup [y_2 = 1] = \Omega^s$$

$$[y_2 = 0]' \cap [y_4 = 1]' = \varnothing \quad \text{donc} \quad [y_2 = 0] \wedge [y_4 = 1] = \Phi^s.$$

	y1	y2	y3	y4
w1	1	0	0	0
w2	1	0	1	0
w3	1	1	0	1
w4	1	1	1	1
w5	0	1	0	0

Figure 11

On a aussi les propriétés suivantes qui caractérisent l'union et l'intersection d'objets symboliques.

PROPOSITION 3 : 1. *L'union et l'intersection d'objets symboliques est commutative, associative et existe toujours.*

2. *Si* $s = s_1 \cup s_2$ *(resp.* $s_1 \cap s_2$*) on a* $s' \supseteq s'_1 \cup s'_2$ *(resp.* $s' \equiv s'_1 \cap s'_2$*).*

Démonstration :

1. La commutativité est évidente. L'associativité vient du fait que $(s_1 \cup s_2) \cup s_3 = \{$la conjonction des événements élémentaires dont la conjonction contient s'_1 et $s'_2 \cup s'_3\} = s_1 \cup (s_2 \cup s_3)$. On a une démonstration analogue pour montrer l'associativité de l'intersection. L'union et l'intersection de deux

objets symboliques s_1 et s_2 existent toujours puisque au moins l'extension de $[y_i = O_i]$ contient $s'_1 \cup s'_2$ et $s'_1 \cap s'_2$.

2. Si $s = s_1 \cup s_2$, on a forcément $s' \supseteq s'_1 \cup s'_2$ puisque par définition s est la conjonction de tous les objets symboliques dont l'extension contient $s'_1 \cup s'_2$. Il peut se produire que l'objet $s = s_1 \cup s_2$ soit tel que $s' \supset s'_1 \cup s'_2$ strictement. En effet, en reprenant l'exemple ci-dessus de la figure 11 et en posant $s_i = [y_i = 1]$ on a : $s_3 \cup s_4 = s_1$ avec $s'_1 = \{ w_i^s / i = 1,4 \} \supset s'_3 \cup s'_4 = \{ w_2^s, w_3^s, w_4^s \}$. L'extension de l'intersection de $s = s_1 \cap s_2$ contient par définition $s'_1 \cap s'_2$, elle est aussi contenue dans $s'_1 \cap s'_2$ car tous les événements élémentaires qui définissent s_1 et s_2 ont une extension qui contient $s'_1 \cap s'_2$. Donc $s' = s'_1 \cap s'_2$.

PROPOSITION 4 : *Muni de l'ordre symbolique l'ensemble des objets symboliques S est un treillis et la borne supérieure de tout couple est leur union, la borne inférieure est leur intersection).*

Démonstration : Rappelons qu'un treillis est un ensemble muni d'une relation d'ordre pour laquelle tout couple d'éléments admet une borne supérieure et une borne inférieure.

Montrons que $s = s_1 \cup s_2$ est la borne supérieure de s_1 et $s_2 \forall s_1, s_2 \in S$. Pour cela il faut prouver que si $s_3 \geqq s_1$ et $s_3 \geqq s_2$ alors $s_3 \geqq s_1 \cup s_2$. Or on sait que $s_3 \geqq s_1$ et $s_3 \geqq s_2$ signifie que $s'_3 \supseteq s'_1$ et $s'_3 \supseteq s'_2$ donc que $s'_3 \supseteq s'_1 \cup s'_2$. Donc s_3 fait partie de l'ensemble des objets symboliques dont l'extension contient $s'_1 \cup s'_2$ et dont la conjonction définit $s_1 \cup s_2$. Ainsi $s_1 \cup s_2 = s_3 \wedge \ldots$ d'où il résulte que $s'_3 \supseteq (s_1 \cup s_2)'$ et donc que $s_3 \geqq s_1 \cup s_2$. On peut faire une démonstration analogue pour montrer que la borne inférieure de deux objets symboliques est leur intersection. ■

5. QUALITÉ DES OBJETS SYMBOLIQUES

5.1. Complétude d'un objet symbolique

Si je vois dans ma rue des chiens qui sont tous blancs et que j'énonce : « je vois des chiens » mon assertion ne décrit pas de façon « complète » l'observation des chiens de ma rue puisque j'omets de dire qu'ils sont blancs. C'est cette idée qu'il s'agit de traduire par la notion de complétude d'un objet symbolique. Plus formellement cela revient à mesurer l'écart entre l'ensemble des événements élémentaires dont la conjonction définit un objet symbolique s et l'ensemble de tous les événements élémentaires dont l'extension symbolique contient s'.

5.1.1. *Notations $d(s)$, $c(s)$ et s^c*

Dans la suite nous noterons $d(s)$ (d comme « définition ») l'ensemble des événements élémentaires dont la conjonction définit s, $c(s)$ (c comme « complet ») l'ensemble de tous les événements élémentaires de plus petite extension symbolique qui contiennent s' et enfin s^c, l'objet symbolique défini par la conjonction de tous les éléments de $c(s)$.

Exemple : Reprenons les données de la figure 11, et considérons l'objet symbolique

$$s=[y_1=1] \wedge [y_4=1]$$

on a :

$$d(s)=\{[y_1=1],[y_4=1]\}, \qquad c(s)=\{[y_1=1],[y_2=1],[y_4=1]\}$$

et

$$s^c=[y_1=1] \wedge [y_2=1] \wedge [y_4=1].$$

5.1.2. *Calcul de la complétude et objet « complet »*

La complétude d'un objet symbolique peut se calculer de différentes manières. Donnons deux exemples de critères de complétude :

$$c_1(s)=\text{card}\,(c(s)-d(s))$$
$$c_2(s)=\text{card}\,(\{(\wedge e_i)'/e_i \in (c(s)-d(s))\})$$

Ainsi en reprenant l'exemple précédent où $s : [y_2=1] \wedge [y_4=1]$ on a : $c(s)-d(s)=\{[y_4=1]\}$ et donc $e_i=[y_1=1]$. Il en résulte que $c_1(s)=1$ et $c_2(s)=2$.

Par la suite nous retiendrons le premier critère et nous dirons d'un objet symbolique qu'il est *complet* si et seulement si $c(s)=d(s)$ autrement dit si $c_1(s)=0$.

5.1.3. *Propriétés des objets complets*

Étant donné un objet symbolique quelconque $s \in S$ nous avons d'abord la proposition suivante :

PROPOSITION 5 : 1. $d(s^c)=c(s)$ et $(s^c)'=s'$.

2. s^c *est complet* (i. e. $d(s^c)=c(s^c)$).

3. $s^c=\{\bigcup \varphi(w_i)/w_i \in |s|_\Omega\}$.

4. $s^c=\{\bigcap e_i/e_i \in c(s)\}$.

Démonstration : 1. $d(s^c) = c(s)$ puisque $d(s^c)$ contient par définition tous les événements élémentaires qui décrivent s^c qui est lui-même décrit par tous les éléments de $c(s)$. Pour montrer que $(s^c)' = s'$ il suffit de montrer que $|s^c|_\Omega = |s|_\Omega$. Or on a d'une part, $|s|_\Omega \subseteq |s^c|_\Omega$ puisque s^c n'est défini que par des événements dont l'extension dans Ω contient $|s|_\Omega$. D'autre part $|s^c|_\Omega \subseteq |s|_\Omega$ car s'il existait un élément de $|s^c|_\Omega$ qui n'était pas dans $|s|_\Omega$ cela signifierait que tous les événements élémentaires de s^c auraient une extension qui le contiendrait, ils seraient tous d'extension plus grande qu'au moins un des événements élémentaire de $d(s)$ qui eux ne le contiennent pas tous tout en contenant $|s|_\Omega$, ce qui est contradictoire avec la définition de s^c puisque cet événement devrait être aussi dans $c(s)$.

2. s^c est complet car $d(s^c) = c(s^c)$ puisque $d(s^c)$ est l'ensemble de tous les événements élémentaires qui décrivent s^c et $c(s^c)$ est l'ensemble de tous les événements élémentaires de plus petite extension dont l'extension contient $(s^c)' = s'$.

3. $s^c = \{ \bigcup \varphi(w_i) / w_i \in |s|_\Omega \}$. Cette propriété résulte immédiatement de la définition des objets symboliques complets et de la propriété 1 de la proposition 2.

4. $s^c = \{ \bigcap e_i / e_i \in c(s) \}$. En effet, par définition de l'intersection symbolique $\bigcap e_i$ est la conjonction des événements élémentaires dont l'extension symbolique est commune à celle des e_i qui par définition de $c(s)$ est celle de s^c et dont celle de s (d'après 1). Cette conjonction est donc par définition celle qui donne s^c. ■

De cette propriété on déduit facilement la proposition suivante qui fournit la possibilité de construire de proche en proche un treillis représentant graphiquement l'ensemble des objets symboliques complets.

PROPOSITION 6 : 1. Si un objet symbolique s est l'union ou l'intersection d'autres objets symboliques, alors s est complet.

2. L'ensemble des objets symboliques complets muni de l'ordre symbolique est un treillis.

Démonstration : 1. Supposons que $s = s_1 \bigcup s_2$ avec $s_1, s_2 \in S$. Par définition s est la conjonction de tous les événements élémentaires de plus petite extension à contenir $s'_1 \bigcup s'_2$ et *a fortiori* $s' \supseteq s'_1 \bigcup s'_2$ donc $s = s^c$. On peut faire une démonstration analogue pour l'intersection.

2. L'ensemble des objets symboliques complets muni de l'ordre symbolique est un treillis puisque tout couple est borné supérieurement (resp. inférieurement) par l'union (resp. l'intersection) symbolique d'après la proposition 4 et que de plus ces bornes sont des objets symboliques complets d'après 1.

5.2. Affinement d'un objet symbolique

On dira qu'un objet symbolique est d'autant plus affiné que les événements élémentaires qui le définissent ont une extension proche de celle de s.

Pour exprimer le degré d'affinement d'un objet symbolique, on peut par exemple utiliser un critère du type :

$$A(s) = \frac{1}{\operatorname{card} d(s)} \operatorname{card}(\cup e_i'(s) - \cap e_i'(s))$$

où les $e_i(s)$ sont les événements élémentaires de s. On divise par $\operatorname{card} d(s)$ car un affinement est d'autant plus « méritoire » que $\operatorname{card} d(s)$ est grand.

Exemple : $s = [y_1 = 1] \wedge [y_2 = 1]$ en reprenant les données de la figure 11 donne : $A(s) = 1/2 \operatorname{card}(\Omega - \{w_3, w_4\}) = 3/2$.

Un objet sera dit « affiné » si $A(s) = 0$.

5.3. Simplicité d'un objet symbolique

On dira qu'un objet symbolique s est d'autant plus simple que le nombre d'événements élémentaires qui le décrit est proche d'un ensemble d'événements élémentaires de cardinal minimum dont la conjonction notée s_p a la même extension.

Il n'y a bien sûr, par unicité de s_p et la simplicité peut se mesurer par exemple à l'aide d'un critère de type suivant :

$$S(s) = \operatorname{card} d(s) - \operatorname{card}(d(s_p)).$$

En reprenant l'exemple précédent avec $s = [y_1 = 1] \wedge [y_2 = 1]$ on a $s_p = [y_4 = 1]$ d'où $S(s) = d(s) - d(s_p) = 2 - 1 = 1$.

On peut combiner la simplicité et la complétude d'un objet symbolique en définissant sa « redondance » par l'écart entre le cardinal de s_p et celui de $c(s)$. Un objet symbolique peut être à la fois simple et complet (comme par exemple un objet élémentaire w_i^s) sa redondance est alors nulle.

On pourrait bien sûr donner beaucoup d'autres propriétés et qualités aux objets symboliques mais faute de place ici le lecteur pourra se reporter, par exemple, à Kodratoff [1986].

6. QUALITÉ DES CLASSES D'OBJETS SYMBOLIQUES

Comme nous l'avons fait pour les objets symboliques, pour étudier les qualités d'une classe, il faudrait d'abord définir ce que l'on entend par extension, ordre, union et intersection de classes. Une façon simple de procéder consiste à associer à chaque classe un objet symbolique (en prenant par exemple l'union ou l'intersection des objets de la classe) puis à utiliser sur ces objets les différentes notions, propriétés et qualités des objets symboliques. On pourra dire qu'une classe est complète affinée ou simple si l'objet symbolique que l'on a décidé de lui associer satisfait ces qualités.

On peut aussi plus directement considérer que l'extension d'une classe est l'union ou l'intersection de l'extension des objets de la classe. Pour l'ordre on peut dire qu'une classe est inférieure à une autre si son extension est contenue dans celle de l'autre ou si son plus grand élément (au sens de l'ordre symbolique) est inférieure au plus petit élément de l'autre, etc.

On peut attribuer aux classes les propriétés des objets qui leur sont associées mais on peut aussi leur attribuer des propriétés caractéristiques de la notion de « classe ». Décrivons en deux : la stabilité et l'effritement.

Stabilité d'une classe : c'est la capacité d'une classe à être représentée par l'objet symbolique de plus petite extension qui contient l'union des extensions des éléments de la classe. On peut donc exprimer la stabilité d'une classe C dont les éléments sont notés c_i, par exemple, à l'aide du critère suivant :

$$\mathrm{st}(c) = \mathrm{card}\left(\left|\bigcup c_i\right|_\Omega - \bigcup \left|c_i\right|_\Omega\right).$$

Ce critère a toujours un sens puisque l'on sait par définition que l'union symbolique de deux objets symboliques doit contenir l'union de leur extension. En utilisant la propriété d'associativité de l'union symbolique on peut généraliser cette propriété à plusieurs objets et on a donc toujours $\bigcup \left|c_i\right|_\Omega \subseteq \left|\bigcup c_i\right|_\Omega$.

On dira qu'une classe C est stable si $\mathrm{st}(C)=0$. Cela est équivalent à dire que $\varphi\left(\left|\bigcup c_i\right|_\Omega\right) = \varphi\left(\bigcup \left|c_i\right|_\Omega\right)$ puisque φ est une bijection et donc que $\left(\bigcup c_i\right)' = \bigcup \varphi\left(\left|c_i\right|_\Omega\right) = \bigcup c_i'$ (remarquer que dans $\left(\bigcup c_i\right)'$ l'union est symbolique alors que dans $\bigcup c_i'$ elle est ensembliste).

Exemple : Reprenons les données de la figure 11, et considérons la classe $C = \{ w_2^s, [y_1 = 1] \wedge [y_3 = 1], [y_4 = 1] \}$ on a :

$$\bigcup c_i = [y_1 = 1] \qquad \text{puisque} \quad \bigcup \left|c_i\right|_\Omega = \{ w_2, w_3, w_4 \}$$

d'où $\mathrm{st}(C)=\mathrm{card}(\{w_1, w_2, w_3, w_4\}-\{w_2, w_3, w_4\})=1$. Donc c n'est pas stable. Par contre $C_1=\{w_3^s, w_4^s\}$ ou la classe $C_2=\{w_2^s, w_4^s\}$ est stable.

Soit C_t l'ensemble des classes stables formées d'éléments de $\varphi(\Omega)$ et telles que pour tout $w^s \in C \in C_t$ on ait $(w^s)' \subseteqq C$. On a alors la proposition suivante :

PROPOSITION 7 : *Il existe une bijection entre l'ensemble des classes stables C_t et l'ensemble des objets complets.*

Démonstration : Notons O_c l'ensemble des objets complets, f l'application $O_c \to C_t$ qui associe à tout objet complet, la classe formée des éléments de son extension et g l'application $C_t \to O_c$ qui associe à toute classe de C_t l'union symbolique de ses éléments. Montrons que $C=f(s)=s'$ est une classe stable. Puisque s est complet on a $s=s^e$ et donc d'après la troisième propriété de la proposition 5 l'union symbolique des éléments de $C=s'$ est s et son extension symbolique est $s'=C$ donc C est une classe stable. $g(c)$ est complet $\forall C \in C_t$ puisque par définition c'est une union d'objets symboliques.

Il reste à montrer que f et g sont des applications inverses l'une de l'autre. Pour cela, montrons que $f \circ g(C)=C$ et que $g \circ f(s)=s$. On a $f \circ g(C)=f(\{\cup w^s / w^s \in C\})=(\cup w^s)'$ par définition de f et g donc $f \circ g(C)=\cup (w^s)'=C$ puisque C d'une part C est une classe stable et d'autre part $(w^s)' \subseteqq C$ par hypothèse. On a aussi $g \circ f(s)=g(s')=\{\cup w^s / w^s \in s'\}=s$ puisque s est complet. ■

Effritement d'une classe : c'est le plus petit nombre d'objets symboliques $a \in A \subset S$ dont la réunion des extensions est contenu dans l'extension des éléments d'une classe C tout en s'en écartant le moins possible.

Étant données une classe $C \subset S$ d'élément générique c_i et une classe $A \subset S$ d'élément générique a_i, on peut, par exemple, utiliser le critère suivant pour mesurer l'effritement :

$$E_1(C)=\mathrm{Min}\{(1+\mathrm{card}(\cup |c_i|_\Omega - \cup |a_i|_\Omega))\,\mathrm{card}\,A \mid A \subset S : |a_i|_\Omega \subseteqq |\cup c_i|_\Omega\}.$$

Si l'on impose à s d'être tel que $\cup |c_i|_\Omega = \cup |a_i|_\Omega$ on obtient le critère :

$$E_2(C)=\mathrm{Min}\{\mathrm{card}\,A / \cup |a_i|_\Omega = \cup |c_i|_\Omega\}.$$

E_1 et E_2 sont minimaux s'il existe un objet symbolique A tel que $A=\cup c_i$. A est alors complet et $E_1=E_2=1$. E_2 est maximal quand $A \equiv C$ et vaut $\mathrm{card}\,C$.

Exemple : On prend les données de la figure 11. Soit $C=\{w_3^s, w_3^s, [y_2=1] \wedge [y_3=1]\}$. L'effritement minimal est donné par

$A = \{[y_4 = 1],\ w_2^s\}$ on a donc card $A = 2$,

$$\cup\,|a_i|_\Omega = \{w_2, w_3, w_4\} \cup |c_i|_\Omega = \{w_2, w_3, w_4\} \text{ d'où } E_1(C) = E_2(C) = 2.$$

En choisissant $A = \{w_2^s\}$ on a

$$(1 + \text{card}\,(\cup\,|c_i|_\Omega - \cup\,|a_i|_\Omega))\,\text{card}\,A = (1+2).1 = 3.$$

La meilleure solution n'est pas unique car on aurait pu prendre aussi $A = \{[y_3 = 1],\ w_3^s\}$ qui donnerait aussi $E_1(C) = E_2(C) = 2$.

Remarquons qu'il serait possible aussi de définir un effritement « supérieur » qui serait le plus petit nombre d'objets symboliques dont la réunion des extensions contiendrait (au lieu d'être contenue dans) l'extension des éléments de la classe.

7. QUALITÉ DES CLASSIFICATIONS

Une classification d'une classe d'objets symboliques $C \subset S$ est un ensemble de classes qui recouvrent C. Une partition, une hiérarchie, une pyramide constituent des exemples de recouvrement possibles. Une classification peut être complète, affinée, simple, avoir une bonne stabilité et un bon effritement suivant que ses classes ou leurs représentants ont ces qualités. Comme nous l'avons fait pour les classes, on peut ajouter à ces qualités des qualités caractéristiques des classifications :

Le *« degré de recouvrement »* des extensions des classes de la classification.

La *« qualité de l'héritage »* des classes entre elles. Elle peut par exemple être mesurée en tenant compte du nombre de classes qui héritent d'une autre classe, de la quantité de propriétés qui sont héritées et de leur valeur.

8. LES QUATRE TYPES D'ANALYSE DES DONNÉES

On peut grossièrement définir quatre types d'analyse de données dont les frontières ne sont pas nécessairement clairement séparées.

(*a*) « L'analyse des données » classique : on traite des données quantitatives ou qualitatives avec des méthodes numériques utilisant l'algèbre linéaire et les outils de la statistique.

(*b*) « L'analyse numérique des données symboliques » : on introduit une mesure sur les valeurs prises par les variables et on utilise la théorie de la mesure et des probabilités, mais comment tenir compte alors des connaissances supplémentaires ?

(*c*) « L'analyse symbolique des données classiques ». Il s'agit de traiter des tableaux de données classiques (objets caractérisés par des variables quantitatives et (ou) qualitatives) par l'approche symbolique (en utilisant : extensions, ordre symbolique, généralisation, héritage, qualité des objets, etc.) soit dès le départ sur les données soit après avoir utilisé une méthode de l'analyse des données classique (afin d'automatiser l'interprétation, par exemple).

Exemple d'analyse symbolique de données classiques :

Extraire les variables les plus « explicatives » d'un axe factoriel et les 2 classes d'individus les plus contributifs de chaque extrêmité de l'axe. Considérer l'ensemble des objets symboliques associés à ces variables. Trouver dans cet ensemble des objets symboliques complets et d'effritement minimum caractéristiques de chacune des classes. Trouver les objets de meilleurs stabilité qui minimisent le recouvrement de la partition associée à ces classes.

(*d*) « L'analyse symbolique des données » on utilise l'approche symbolique pour traiter des données qui sont aussi symboliques.

Exemple de problème d'analyse symbolique de données :

Étant donné S un ensemble d'objets symboliques et $S_1 \subset S$. Trouver une partition de S_1 qui maximise la stabilité et minimise le recouvrement. Trouver une hiérarchie de classes stables et de meilleure héritance. Trouver des assertions d'effritement minimal pour chaque classe.

Le tableau suivant (*fig.* 12) schématise les quatre approches.

Données / Analyse	Classiques	symboliques
Numérique	(a)	(b)
Symbolique	(c)	(d)

Figure 12

CONCLUSION

Nous avons introduit des objets qui étendent le champ des données habituellement traitées par l'analyse des données classiques. Ces objets dits « symboliques » s'avèrent particulièrement aptes à extraire ou à exprimer une connaissance évoluée, sous une forme explicative. La connaissance mise sous

forme d'objets symboliques fait apparaître généralement la nécessité de définir une connaissance supplémentaire issue de la "sémantique du domaine" qui permet d'approcher d'encore plus près la réalité multidimensionnelle. Pour traiter de telles données, l'utilité d'introduire des outils souvent issus de la logique classique et de l'intelligence artificielle apparaît naturelle et commode. Ces outils et des principes de traitement que nous avons énoncés constituent ensemble un fond commun d'idées de base que l'on peut appeler « l'analyse symbolique des données ». Cette approche s'est avérée à son tour utile pour traiter des données classiques afin d'en extraire des objets symboliques facilitant, par exemple, une interprétation automatisée ou la construction d'une base de connaissance d'un système expert.

Dans le cadre de l'analyse symbolique des données sous quelle forme nouvelle s'exprime le problématique, les méthodes et les algorithmes de l'analyse des données classiques? Quels algorithmes pour passer d'un type d'objets symboliques à un autre? Comment s'affranchir du point de vue volontairement déterministe (car déjà suffisamment riche et compliqué) que nous avons gardé jusque-là, afin de traiter des données munies de connaissances supplémentaires imprécises par exemple? Autant de question parmi beaucoup d'autres qui restent très ouvertes.

BIBLIOGRAPHIE

1. W. A. Belson, *Matching and Prediction on the Principle of Biological Classification*, Applied statistics, vol. III, 1959.
2. M. J. Dallwitz, *Automatic Type Setting of Computer Generated Keys and Descriptions*, In Data Basis in Systematics, R. A. Allkin et F. A. Bisby éd., 1984, p. 279-290, Acad. Press London and Orlando.
3. E. Diday, *Selection of Variables and Clustering*, Int. Conf on Pattern Recognition. Kyoto. Japan, 1978.
4. E. Diday, G. Govaert, Y. Lechevallier et J. Sidi, *Clustering in Pattern Recognition*, Proc. 5th Conf. Pattern Recognition Miami Beach FL. Plus complet dans NATO, Bonas. J. C. Simon éd., 1980.
5. E. Diday et J. V. Moreau, *Learning Hierarchical Clustering from Examples - Application to the Adaptative Construction of Dissimilarity Indices*, Pattern Recognition Letter 2, 1984, p. 365-378.
6. E. Diday, *Order and Overlapping Clusters by Pyramids* in De Leeuw *et al.* éd., Leiden: DSWO Press, 1987.
7. E. Diday et L. Roy, *Generating Rules by Symbolic Data Analysis and Application to Soil Feature Recognition*, Actes des 8es Journées Internationales « Les systèmes experts et leurs applications », Avignon, 1988.
8. R. Ducournau et J. Quinqueton, *Yafool : Encore un langage objet à base de frames*, Rapport technique n° 72 INRIA, « Y 3 : Yafool Le langage à Objets », Rapport Sema-Group février 1989.

9. J. G. GANASCIA, *Charade : apprentissage de bases de connaissances*, Actes des journées « Symboliques-Numériques » pour l'apprentissage de connaissances à partir d'observations, Université Paris-IX - Dauphine, CEREMADE, E. DIDAY et Y. KODRATOFF éd., 1987.

10. J. G. GANASCIA, *Apprentissage de connaissance par les cubes de Hilbert*, Thèse d'Etat, Université d'Orsay, 1987.

11. A. GUENOCHE, *Propriétés caractéristiques d'une classe relativement à un contexte*, Actes des journées « Symboliques-Numériques » pour l'apprentissage de connaissances à partir d'observations. Université Paris-IX - Dauphine, CEREMADE, E. DIDAY et Y. KODRATOFF éd., 1987.

12. J. L. GUIGUE et V. DUQUENNE, *Familles minimales d'implications informatives résultant d'un tableau binaire*, Mathématiques et sciences humaines, 24[e] années, 95, 1986, p. 5-18.

13. Bao Ho TU, E. DIDAY et M. SUMMA, *Generating Rules for Expert System from Observations* in « Les systèmes experts et leurs applications », 7[es] journées internationales les systèmes experts et leurs applications, EC2, 269, rue de la Garenne, 92000 Nanterre (France), 1987.

14. Y. KODRATOFF, *Leçons d'apprentissage symbolique*, Cepadues-éditions, 111, rue Nicolas Vauquelin, 31000 Toulouse, 1986.

15. J. LEBBE, *Manuel d'utilisation du logiciel XPER*, Micro application, Paris, 1984.

16. R. MICHALSKI, R. E. STEPP, *Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy*, IEEE Trans. pattern analysis and Machine intelligence, vol. PAMI-5, n° 4, 1983.

17. R. MICHALSKI, E. DIDAY et R. E. STEPP, *A Recent Advances in Data Analysis: Clustering Objects into Classes Characterized by Conjonctive Concepts*, Progress in Pattern Recognition, vol. 1, L. KANAL et A. ROSENFELD éd., 1981.

18. M. MANAGO, *Intégration de techniques numériques et symboliques en apprentissage automatique*, Thèse d'état, Université d'Orsay, LRI.

19. M. O. MENESSIER et E. DIDAY, *Approches symbolique pour la prévision de séries chronologiques pseudo-périodiques*, Actes des journées « Symboliques-Numériques » pour l'apprentissage de connaissances à partir d'observations, LRI, Université d'Orsay, Y. KODRATOFF et E. DIDAY éd., 1988.

20. L. E. MORSE, J. A. PETER et P. B. HAMEL, *A General Data Format for Summarizing Taxonomic Information*, Bio Science, 21, (4), 1971, p. 174-181.

21. J. N. MORGAN et J. A. SONQUIST, *Problems in the Analysis of Survey Data a Proposal*, J.A.S.A., 58, 1963, p. 415-434.

22. R. J. QUINLAN, *Learning Efficient Classification Procedure and their Application in Chess and Game. Machine Learning: an Artificial Intelligence Approach*, Michalski, Carbonell, Mitchell éd., Pub. TIOGA, Palo, Alto, California, 1983, p. 463-482.

23. J. QUINQUETON et J. SALLANTIN *CALM: Contestation for Argumentative Learning Machine*, in Machine Learning, a Guide to Current Research, Michalski Carbonnel Mitchell ed., Kluwer and sons, 1986.

24. R. J. PANKHURST, *A Computer Program for Generating Diagnostic Keys*, Computer Journal, 13, 1970, p. 145-151.

25. H. RALAMBONDRAINY, *GENREG : un générateur de règles à partir de données*, Actes des journées « Symboliques-Numériques » pour l'apprentissage de connaissances à partir d'observations, Université Paris-IX - Dauphine, CEREMADE, 1987.

26. M. TOUATI et E. DIDAY, *Synthèse d'objets*, Cahiers de CEREMADE, Université Paris-IX - Dauphine, 1989.

27. R. TRONCHER, J. LEBBE et R. VIGNES, *Présentation d'un système expert d'aide au diagnostique des causes de surdité*, Congrès O.P.A., Paris, 1987.
28. R. WILLE, *Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts*, Proceedings of the Symposium on Ordered Sets, Ivan RIVAL éd., 1981.