Aspects de la classification dans un système de représentation des connaissances par objets

Arnaud Simon, Amedeo Napoli, Jean Lieber, Alain Ketterlin

Aspects de la classification dans un système de représentation des connaissances par objets

Arnaud Simon, Amedeo Napoli, Jean Lieber, Alain Ketterlin*
LORIA – UMR 7503, BP 239, 54506 Vandœuvre-lès-Nancy Cedex
* LSIIT – ULP, 7, rue Descartes, 67084 Strasbourg Cedex
({asimon,napoli,lieber}@loria.fr, alain@dpt-info.u-strasbg.fr)

1 Introduction

Dans cet article, nous étudions certains aspects que peut prendre le processus de classification dans le cadre des systèmes de représentation de connaissances par objets (RCO). Nous nous intéressons essentiellement aux opérations de classification de classes et d'instances : la *classification de classes* recouvre les processus de construction — automatique et manuelle — de classes, l'organisation des classes en hiérarchie(s) et l'insertion d'une classe dans une hiérarchie ; la classification d'instances recouvre le processus de reconnaissance de la classe d'un individu (ou objet individuel). Les classifications de classes et d'instances sont à la base du *raisonnement par classification*, qui s'applique à une variété de problèmes traités en intelligence artificielle, comme la configuration, le diagnostic, la planification, etc. [Stefik, 1995].

Nous évoquons ensuite trois applications particulièrement importantes dans le cadre de la conception de systèmes intelligents qui reposent toutes trois sur le processus de classification : la formation incrémentale de classes et de hiérarchies de classes, l'extraction de connaissances à partir de bases de données et le raisonnement à partir de cas.

2 Éléments sur les systèmes de RCO

2.1 *Préliminaires*

Un système de RCO s'appuie sur une *hiérarchie* $\mathcal{H} = (\mathcal{X}, \sqsubseteq, \omega)$, qui est un graphe orienté sans circuit, où $\mathcal{X}$ est un ensemble de *classes*, $\sqsubseteq$ est une relation d'ordre partiel — ou relation de *subsomption* — et ω est l'élément maximal de $\mathcal{X}$ suivant $\sqsubseteq$; ω est appelée la *racine* de la hiérarchie et elle est supposée toujours exister.

Une classe `C` représente un concept du monde réel et possède une identité et un ensemble de propriétés qui sont caractéristiques de l'état et du comportement du concept représenté (l'ensemble des propriétés de la classe `C` est l'*intension* de `C`). Dans $\mathcal{H}$, le fait qu'une classe `D` *spécialise* une classe `C` — ou encore que `D` est *subsumée* par `C` — se note $\texttt{D} \sqsubseteq \texttt{C}$ et la donnée de cette relation constitue un arc de la hiérarchie $\mathcal{H}$. Une classe regroupe un ensemble d'individus, ou *instances* de la classe ; elle peut se voir associer de nouvelles instances ou encore être *instanciée* pour produire — selon le modèle qu'elle définit — de nouveaux individus (l'ensemble des instances de `C` est l'*extension* de `C`).

2.2 *Le raisonnement dans un système de RCO*

En toute généralité, le raisonnement dans un système de RCO consiste à exploiter les propriétés d'une hiérarchie représentant des connaissances relatives à un certain domaine de référence. Les opérations principales qui sont à la base du raisonnement sont les suivantes :

- Le test de *subsomption* consiste à vérifier qu'une classe `C` subsume qu'une classe `D` ; `C` est alors le *subsumant* et `D` le *subsumé*. Intuitivement, une classe `C` subsume une classe `D` si et seulement si l'extension de `C` contient nécessairement l'extension de `D`.
- La *classification de classes* consiste à placer une nouvelle classe `X` dans l'ordre associé à une hiérarchie $\mathcal{H}$. Parallèlement, la *classification d'instances* — ou *test d'instanciation* — consiste à déterminer les classes dont un objet `x` donné peut être une instance. En particulier, une classe `C` n'a un sens que si elle peut avoir effectivement des instances (*satisfiabilité* d'une classe).
- La *recherche de propriétés* consiste à trouver les propriétés détenues par une classe ou une instance, les restrictions associées à ces propriétés et/ou leurs valeurs.

Comme tout système à base de connaissances, un système de RCO peut se voir comme un *système logique*, qui fournit un processus de construction de classes — aspect *syntaxique* — et une procédure d'inférences qui repose sur la subsomption. Au processus de construction des classes peut être couplée une *fonction d'interprétation* qui donne la *sémantique* attachée à une classe (qui peut être l'ensemble des objets faisant partie de l'extension de la classe, comme dans les logiques de descriptions [Napoli, 1997]). De plus, un ensemble d'*assertions* décrivant des faits dans lesquels interviennent des individus peuvent être pris en compte. Ces aspects logiques des RCO sont aussi discutés dans [Euzenat, 1993] et [Ducournau, 1996] (voir aussi [Ducournau et al., 1998]).

2.3 *La construction des classes et l'organisation en hiérarchie(s)*

La *classification de classes* est une opération qui consiste à découvrir des régularités pour regrouper des entités individuelles en classes [Napoli, 1994]. Dans le cadre des systèmes de RCO, deux approches principales sont à distinguer. La première est relative à la conception de hiérarchies d'héritage. La construction d'une telle hiérarchie est « manuelle» et descendante : les nouvelles classes spécialisent les classes existantes par adjonction de nouvelles propriétés et par substitution de valeurs dans des propriétés déjà définies (nous supposerons ici que ces modifications se font de façon *monotone*). Le comportement associé aux différentes concepts modélisés est distribué parmi l'ensemble des classes de la hiérarchie.

La deuxième approche est incrémentale et ascendante. Elle est similaire aux techniques de classification automatique en analyse de données et peut être qualifiée de formation incrémentale de classes et de hiérarchies de classes (cette méthode d'apprentissage est couramment appelée « formation incrémentale de concepts et de hiérarchies de concepts » [Gennari et al., 1989]). Les objets de base sont traités globalement ou bien l'un après l'autre ; ils sont regroupés en des classes et l'ensemble des classes produit est organisé en une hiérarchie. Les caractéristiques des concepts du monde réel étudiés fournissent les propriétés des classes qui sont formées et qui servent à représenter les concepts étudiés. Les hiérarchies obtenues peuvent être des arbres, des graphes ou encore des treillis [Godin et al., 1995] [Ketterlin, 1995].

2.4 *L'exploitation d'une hiérarchie de classes*

Les processus de classification de classes et d'instances opèrent sur une hiérarchie $\mathcal{H} = (\mathcal{X}, \sqsubseteq, \omega)$ et cherchent à mettre en évidence les dépendances implicites classes–classes et classes–instances qui existent entre les objets dans $\mathcal{H}$. En particulier, les processus de classification permettent de placer un objet `x`, classe ou instance, dans la hiérarchie $\mathcal{H}$.

Le *raisonnement par classification* s'appuie sur les processus de classification et s'appréhende comme une procédure de déduction opérant sur une hiérarchie. La mise en œuvre du raisonnement par classification repose sur un cycle comprenant trois étapes [Napoli and Laurenço, 1993] :

- Initialisation : création d'un nouvel objet `x`, qui peut être une classe ou une instance.
- Classification : parcours de la hiérarchie $\mathcal{H}$; recherche des subsumants les plus spécifiques de `x` (SPS), recherche des subsumés les plus généraux de `x` (SPG) et mise en place de `x` dans $\mathcal{H}$.
- Exploitation : la mise en place de `x` dans $\mathcal{H}$ déclenche des opérations de mise à jour d'objets interdépendants et/ou la production de nouveaux objets, ce qui ramène le cycle à sa première étape.

Le processus de classification repose sur le caractère nécessaire et/ou suffisant des propriétés attachées à une classe. Si $\texttt{x} \in \texttt{Extension}(\texttt{C})$, alors `x` vérifie *toutes* les propriétés de `C` (conditions *nécessaires* d'appartenance de `x` à $\texttt{Extension}(\texttt{C})$). Réciproquement, s'il s'avère que l'objet `y` vérifie *toutes* les propriétés de `C`, alors il est (éventuellement) possible de déduire que $\texttt{y} \in \texttt{Extension}(\texttt{C})$ (conditions *collectivement suffisantes* d'appartenance de `y` à l'extension de `C`).

3 Illustrations des processus de classification dans le cadre des RCO

3.1 *Cobweb et la formation incrémentale de classes et de hiérarchies de classes*

L'algorithme COBWEB [Fisher, 1987] fournit un premier exemple de méthode de construction de hiérarchies de classes. COBWEB procède à partir des instances — appelées *observations* dans ce contexte — qu'il traite une à une. La première instance donne simplement lieu à une classe, unique. La deuxième instance, à supposer qu'elle soit distincte de la première, donne elle aussi lieu à une nouvelle classe, et une nouvelle classe « racine », généralisant les deux premières classes, est créée. La hiérarchie est ainsi « amorcée ». Toute nouvelle instance est ensuite intégrée dans cette hiérarchie. L'algorithme procède à partir de la racine et recherche de quelle sous-classe immédiate l'instance est la plus proche. En alternant d'une part cette phase de recherche de proximité et d'autre part d'éventuelles opérations de réorganisation locale de la structure hiérarchique (par exemple la fusion de deux classes ou encore la suppression d'un niveau de généralité), COBWEB adapte la hiérarchie de classes — qui est un arbre — à la nouvelle instance. Ce parcours descendant se poursuit éventuellement jusqu'aux feuilles de l'arbre. Au passage, les descriptions des classes explorées sont mises à jour afin de refléter la nouvelle instance.

Le résultat du traitement d'une instance par COBWEB (que ce soit à des fins de construction ou de reconnaissance) est donc toujours une classe, laquelle, associée à la liste de ses super-classes, identifie parfaitement l'instance dans l'« espace conceptuel » ainsi construit.

3.2 *La fouille de données en milieu hiérarchique*

La fouille de données (FDD) — ou extraction de connaissances à partir de données — consiste à analyser des données brutes d'un certain domaine de façon à en extraire un ensemble d'unités de

connaissances pouvant être exploitées dans un système à base de connaissances [Frawley et al., 1992] [Brachman et al., 1993]. L'utilisation d'un système de RCO, du raisonnement par classification, et des techniques de construction d'arbres de décision [Simon and Napoli, 1997] et de treillis de Galois [Simon and Napoli, 1998], permettent de répondre à certains besoins de la FDD. Les concepts du domaine étudié sont représentés par des classes organisées en une hiérarchie, et les données à traiter sont alors vues comme des instance de ces classes.

En particulier, la construction d'un treillis de Galois sur la base des données étudiées — et en adéquation avec les connaissances du domaine — permet d'élaborer des *points de vue hiérarchiques* différents sur le domaine et les données. Une classe représente un concept formel défini par un couple intension – extension, muni d'attributs (non nécessairement booléens) et de relations mono et multi-valués. L'utilisation des treillis de Galois comme outils pour la FDD a été mise en évidence et discutée dans [Godin et al., 1995], mais aussi, sans que ce soit exprimé ainsi, dans [Duquenne, 1996]. En particulier, des règles peuvent être extraites du treillis et peuvent constituer des éléments de connaissances recherchés dans le cadre du processus de FDD.

3.3 *Éléments sur le RàPC en milieu hiérarchique*

Les systèmes de RCO et le raisonnement par classification s'associent de façon très naturelle au raisonnement à partir de cas (RÀPC) dès lors que les cas sont représentés et/ou indexés par des objets organisés en hiérarchie [Lieber and Napoli, 1998]. Le RÀPC se propose de faire correspondre à l'énoncé d'un nouveau problème `P` une solution $\texttt{Sol}(\texttt{P})$ en tirant parti d'un ensemble de *cas*, qui sont des problèmes déjà résolus accompagnés de leur solution. Le processus du RÀPC se décompose en trois opérations principales : la *remémoration*, l'*adaptation* et la *mémorisation*.

Lorsque la base de cas possède une organisation hiérarchique, notée $\mathcal{H}_{\texttt{idx}}$, l'opération de remémoration peut se diviser en deux étapes comparables aux deux premières étapes du raisonnement par classification : (1) construction d'une représentation du problème `cible` et (2) classification de `cible` dans $\mathcal{H}_{\texttt{idx}}$; tous les énoncés de problèmes sources dont l'index est un subsumant de `cible` fournissent alors une solution qui peut être potentiellement réutilisée pour produire une solution du problème `cible`. La remémoration et l'adaptation, qui constituent l'essentiel de la procédure d'inférence du RÀPC, peuvent se combiner par l'intermédiaire de la notion de *chemin de similarité*, pour retrouver dans une base de cas le meilleur cas adaptable et capable de résoudre un nouveau problème [Lieber and Napoli, 1998].

Références

[Brachman et al., 1993] Brachman, R., Selfridge, P., Terveen, L., Altman, B., Borgida, A., Halper, F., Kirk, T., Lazar, A., McGuinness, D., and Resnick, L. (1993). Integrated support for data archaeology. *International Journal of Intelligent and Cooperative Information Systems*, 2(2):159–185.

[Ducournau, 1996] Ducournau, R. (1996). Les incertitudes de la classification incertaine. In Dennebouy, Y., editor, *Actes du Colloque Langages et Modèles à Objets (LMO'96), Leysin, Suisse*, pages 183–200. École Polytechnique Fédérale de Lausanne.

[Ducournau et al., 1998] Ducournau, R., Euzenat, J., Masini, G., and Napoli, A., editors (1998). *Langages et modèles à objets — État des recherches et perspectives*. Collection Didactique D-019. INRIA, Le Chesnay.

[Duquenne, 1996] Duquenne, V. (1996). On lattices approximations: Syntactic aspects. *Social Networks*, 18:189–199.

[Euzenat, 1993] Euzenat, J. (1993). Définition abstraite de la classification et son application aux taxonomies d'objets. In Habib, M. and Oussalah, M., editors, *Actes de la Conférence Représentations Par Objets (RPO'93), La Grande Motte, France*, pages 235–246. EC2, Nanterre.

[Fisher, 1987] Fisher, D. (1987). Knowledge Acquisition Via Conceptual Clustering. *Machine Learning*, 2:139–172.

[Frawley et al., 1992] Frawley, W., Piatetsky-Shapiro, G., and Matheus, C. (1992). Knowledge Discovery in Databases: An Overview. *The AI Magazine*, 14(3):57–70.

[Gennari et al., 1989] Gennari, J., Langley, P., and Fisher, D. (1989). Models of Incremental Concept Formation. *Artificial Intelligence*, 40:11–61.

[Godin et al., 1995] Godin, R., Mineau, G., Missaoui, R., and Mili, H. (1995). Méthodes de classification conceptuelle basées sur les treillis de galois et applications. *Revue d'intelligence artificielle*, 9(2):105–137.

[Ketterlin, 1995] Ketterlin, A. (1995). *Découverte de concepts structurés dans les bases de données*. Thèse d'Informatique, Université Louis Pasteur, Strasbourg.

[Lieber and Napoli, 1998] Lieber, J. and Napoli, A. (1998). Représentation par objets et classification pour le raisonnement à partir de cas. In Chein, M. and Schmitt, F., editors, *Actes du 11e congrès AFCET Reconnaissance des formes et intelligence artificielle, Clermont-Ferrand*, pages 345–354 (Tome III). AFCET – AFIA – Université de Clermont-Ferrand.

[Napoli, 1994] Napoli, A. (1994). Catégorisation, raisonnement par classification et raisonnement à partir de cas. In *Actes des Journées Acquisition, Validation, Apprentissage (JAVA'94), Strasbourg*, pages E1–E14.

[Napoli, 1997] Napoli, A. (1997). Une introduction aux logiques de descriptions. Rapport de Recherche RR-3314, INRIA.

[Napoli and Laurenço, 1993] Napoli, A. and Laurenço, C. (1993). Représentations à objets et classification. Conception d'un système d'aide à la planification de synthèses organiques. *Revue d'intelligence artificielle*, 7(2):175–221.

[Simon and Napoli, 1997] Simon, A. and Napoli, A. (1997). Un algorithme de fouille dans une représentation des données par objets : une application au domaine médical. In Charlet, J., editor, *Actes des Journées Ingénierie des Connaissances et Apprentissage Automatique, Roscoff*, pages 593–604. INRIA Rennes.

[Simon and Napoli, 1998] Simon, A. and Napoli, A. (1998). Treillis de Galois et représentation par objets pour la fouille de données. In Bourigault, D., editor, *Ingénierie des connaissances (IC'98), Pont à Mousson, France*. INRIA.

[Stefik, 1995] Stefik, M. (1995). *Introduction to Knowledge Systems*. Morgan Kaufmann Publishers, Inc., San Francisco, California.