

Comparaison de systèmes de Programmation Logique Inductive et application à la modélisation d'un élève en géométrie

Nicolas VAN LABEKE

*CRIN - CNRS & Université Henri Poincaré, Nancy I
Bâtiment LORIA, Campus Scientifique, BP 239,
F-54506 Vandoeuvre-les-Nancy Cedex, FRANCE
Mailto : Nicolas.Van-Labeke@loria.fr*

Résumé : *Nos travaux se placent dans le cadre du tuteur TALC dont l'objectif est de diagnostiquer la correction d'une figure géométrique construite par un élève vis-à-vis d'une spécification fournie par le professeur. Dans le cas où des constructions sont incorrectes, il paraît très utile pour produire une bonne explication à l'élève de disposer d'un modèle de ses croyances en géométrie. Dans ce contexte des EIAO (Environnements Interactifs d'Apprentissage avec Ordinateur), il est ainsi évidemment intéressant de générer ce modèle automatiquement plutôt que d'en appeler à l'enseignant. La démarche que nous présentons consiste à utiliser comme «boîte noire» des systèmes de Programmation Logique Inductive disponibles et de les expérimenter sur des conceptions erronées déjà analysées par des didacticiens.*

Mots-clefs : *Modèle de l'élève, EIAO, Programmation Logique Inductive, Géométrie, Diagnostic.*

Introduction.

Nos travaux se situent dans le cadre de la conception et de la réalisation du Tuteur d'Aide Logique pour la Construction de figures géométriques TALC [Desmoulins 94]. L'objectif de ce tuteur est de fournir un diagnostic sur la correction d'une figure construite par un élève vis-à-vis d'une spécification fournie par l'enseignant. Ce diagnostic est globalement défini dans la logique du premier ordre par l'expression $TIG \text{ /-- } S \Leftrightarrow F$ où F et S sont les traductions logiques respectivement de la construction de l'élève et de la spécification de l'enseignant, TIG (Théorie Instrumentale de la Géométrie) est une théorie axiomatique représentant les connaissances géométriques utiles à la résolution de l'exercice.

TALC fournit actuellement trois types de diagnostics : la figure est correcte, la figure est insuffisante, la figure est particulière. Dans le cas où la construction est incorrecte, il apparaît préférable de fournir à l'élève des explications lui permettant d'accepter le diagnostic. Une méthode pour générer de telles explications passe par l'utilisation d'un modèle des conceptions géométriques de l'élève.

Appliqué à TALC, la génération d'un modèle des croyances de l'élève consiste à chercher, pour tous les couples (F,S) pour lesquels le contrat n'est pas réalisé une théorie TAG (Théorie de l'Apprenant en Géométrie) telle que $TAG \text{ /-- } S \Leftrightarrow F$, c'est à dire cohérente avec les constructions de l'élève. La représentation des connaissances géométriques sous forme de clauses de Horn dans TALC fait que l'utilisation de systèmes de Programmation Logique Inductive (PLI) est particulièrement appropriée pour cette génération.

La démarche que nous présentons dans cet article a consisté à comparer des systèmes de PLI existant en vue de les utiliser comme une 'boîte noire' de génération de la théorie TAG. Dans une première partie, nous définissons le principe de la PLI et de son utilisation au sein de TALC. Puis nous proposons une comparaison qualitative des différents systèmes retenus pour l'expérimentation que nous présentons dans la dernière partie.

1. PLI et modèle de l'élève.

L'objectif de la PLI est d'inférer une définition d'un ou plusieurs concepts (appelés *concepts cibles*) à partir d'observations, i.e. d'exemples positifs (noté E^+) et d'exemples négatifs (ou *contre-exemples*, noté E^-) de ces concepts. La connaissance relative au domaine où se déroule l'apprentissage automatique (ou *théorie du domaine*, noté B) est constituée d'un ensemble de clauses de Horn. Le principe de la PLI consiste alors à déter-

miner une hypothèse H, elle aussi sous forme de clause de Horn, vérifiant les conditions suivantes [Muggleton et al. 94] :

- la complétude $B, H \text{ /-} E^+$
- la consistance $B, H \text{ /-} E^-$

La condition de complétude exige que H couvre la totalité des exemples positifs. La condition de consistance impose que H ne couvre aucun exemple négatif. A ces deux conditions, il faut ajouter la précondition $B \text{ /-} E^+$ qui assure que la théorie du domaine ne contient pas déjà une définition complète du concept.

1.2. Caractérisation qualitative des systèmes de PLI

Pour pouvoir sélectionner des systèmes de PLI adaptés à notre problème, nous avons dégagé les éléments pouvant caractériser tout système de PLI du point de vue d'un utilisateur. Cette caractérisation, basée sur la comparaison proposée par Aha dans [Aha 92], concerne uniquement le processus d'apprentissage automatique mis en oeuvre et les restrictions apportées au langage de description. Les caractéristiques de comparaison sont les suivantes :

- apprentissage incrémental ou empirique : les données d'induction sont respectivement entrées les unes après les autres ou données en une seule fois;
- apprentissage interactif : l'apprentissage nécessite l'intervention de l'utilisateur pour confirmer ou infirmer les inférences du système;
- apprentissage mono ou multi-conceptuel : le système permet l'induction soit d'un unique concept, soit de plusieurs concepts à la fois;
- révision de théories : il y a révision de théorie quand l'apprentissage s'appuie sur une définition préalable incomplète et/ou incorrecte des concepts à induire et en fournit une nouvelle définition;
- invention de prédicats : le système est capable d'inventer de nouveaux prédicats utiles pour la définition du concept;
- théorie du domaine extensionnelle ou intentionnelle : la théorie est extensionnelle si elle est donnée sous forme de faits exclusivement ou intentionnelle si elle est donnée sous forme de règles;

1.3. Utilisation de la PLI pour TALC.

Nous présentons ici le principe de l'utilisation de la PLI en «boite noire» en identifiant les données de la PLI fournies par TALC et ses besoins par rapport aux caractéristiques ci-dessus.

1.3.1. Correspondance des éléments de TALC et des observations d'un système de PLI.

Définir les observations de l'apprentissage automatique à partir des éléments de TALC n'est pas immédiat.

Dans la suite de cet exposé, nous considérons uniquement le cas où la construction de l'élève est insuffisante par rapport à l'énoncé, i.e. $TIG \text{ /-} F \Rightarrow S$. Notre objectif est donc de trouver une théorie TAG telle que $TAG \text{ /-} F \Rightarrow S$, pour tout couple <figure, spécification> (F,S). Pour les formules $F \Rightarrow S$ correctes dans TIG, il est inutile de chercher un modèle puisque TIG en est un.

Sachant qu'une formule S est une conjonction de propriétés ($S = s_1 \wedge s_2 \wedge \dots \wedge s_n$), le diagnostic de TALC nous fournit l'ensemble des propriétés s_i non vérifiées dans F, i.e. telles que $TIG \text{ /-} F \Rightarrow s_i$. Les exemples positifs au sens de l'apprentissage automatique sont donc l'ensemble de ces formules $F \Rightarrow s_i$. Par exemple, si la propriété $symétrique(a, b, l, a', b')$ n'est pas vérifiée dans F, i.e. $TIG \text{ /-} F \Rightarrow symétrique(a, b, l, a', b')$, la formule $F \Rightarrow symétrique(a, b, l, a', b')$ constitue un exemple positif de l'apprentissage automatique. Malheureusement, les systèmes de PLI n'acceptent pas des formules comme observations mais des littéraux. Pour résoudre ce problème, nous incluons la formule F dans la théorie du domaine au même titre que la théorie initiale TIG. De manière à conserver le lien existant entre F et S, nous appliquons alors la solution classique consistant à renommer toutes les objets de la formule $F \Rightarrow s_i$ avec des noms différents pour chaque couple (F,S). Par exemple, si la propriété $symétrique(a, b, l, a', b')$ de la spécification S n'est pas vérifiée à la fois dans F_1 et dans F_2 , les exemples positifs peuvent être $symétrique(a_1, b_1, l_1, a_1', b_1')$ et $symétrique(a_2, b_2, l_2, a_2', b_2')$ en introduisant les F_1 et F_2 renommées dans la théorie du domaine.

Notons qu'il n'y a pas a priori d'exemples négatifs car l'élève ne fournit pas au diagnostic de TALC une construction qu'il considère lui-même incorrecte.

1.3.2. Caractéristiques requises par TALC.

La PLI utilise souvent des contre-exemples pour diriger et contraindre la recherche des hypothèses. Les données issues de TALC n'en fournissant pas, nous envisageons la possibilité que l'utilisateur valide (exemples) ou invalide (contre-exemples) des propriétés proposées par le système et reliant des objets d'une construction donnée. Une interaction entre le système et l'utilisateur est alors requise. Par contre, de la façon dont nous

l'avons défini, le processus d'apprentissage automatique de TAG peut être indifféremment incrémental (après chaque diagnostic de TALC) ou empirique (après par exemple chaque série de construction).

La théorie du domaine incluant TIG, elle doit être impérativement donnée de façon intentionnelle. La capacité d'inventer des prédicats peut aussi se montrer utile mais pas forcément indispensable, de même que l'apprentissage multi-conceptuel et la révision de théorie. Seules des expérimentations pourront le déterminer.

2. Choix de systèmes de PLI pour la modélisation de l'élève.

Le choix de systèmes de PLI pour l'expérimentation en «boîte noire» s'est fait en deux filtrages successifs : une première sélection de systèmes à partir de leur disponibilité et de leur documentation théorique et une évaluation sur plusieurs exemples issus de la littérature [Van labeke 95].

A l'issue de ces sélections, les quatre systèmes suivants ont été retenus :

- FOIL [Quinlan 90] est un système mono-conceptuel et empirique qui ne nécessite pas obligatoirement de contre-exemples, ceux-ci étant automatiquement générés lors d'une phase de prétraitement des exemples.
- FOCL [Pazzani et al. 92] est un système mono-conceptuel et empirique qui combine des extensions de l'apprentissage inductif appliqué dans FOIL et l'*Explanation-Based Learning* [Ellman 89]. FOCL étend les capacités de FOIL notamment par l'utilisation d'une théorie du domaine intentionnelle et par la révision d'une définition initiale approximative du concept à apprendre.
- GOLEM [Muggleton et al. 92] est un système empirique et mono-conceptuel basé sur le principe d'inversion de la résolution introduit par Muggleton. Les contre-exemples du concept cible doivent être impérativement donnés.
- CLINT [De Raedt 91] est le seul système étudié qui soit incrémental et multi-conceptuel. Il se distingue aussi des autres systèmes par l'utilisation de contraintes d'intégrité (propriétés sur les connaissances de la base). La vérification de ces contraintes peut amener CLINT à supprimer de sa base des éléments contradictoires.

	FOCL	FOIL	CLINT	GOLEM	TALC
Caractéristiques comportementales					
Apprentissage interactif			X		X
Apprentissage incrémental			X		P
Apprentissage multi-conceptuel			X		P
Révision de théories	X		X		X
Invention de prédicats	X		X		P
Restrictions du langage de représentation					
Pas de littéraux négatifs	P	P		X	X
Pas de symboles de fonctions	X	X	X		X
Théorie du domaine intentionnelle	X		X		X

X : le système possède cette propriété, *P* : le système peut utiliser cette propriété

Le tableau ci-dessus présente une synthèse des caractéristiques principales de ces systèmes et de TALC. Cette comparaison des besoins de TALC et des systèmes sélectionnés nous amène à exclure les systèmes FOIL et GOLEM. Leur manque le plus criant vis-à-vis de TALC réside dans la manipulation d'une théorie du domaine exclusivement extensionnelle. Le fait que le système soit incrémental et interactif est a priori un avantage et désigne le système CLINT comme meilleur choix. Malheureusement, l'évaluation de ce système a montré de gros problèmes liés à ses capacités machine et à la définition de méta-règles utilisées dans la restriction de l'espace de recherche. De ce fait, nous avons aussi retenu FOCL, qui est empirique, en construisant l'ensemble des exemples et contre-exemples avant son utilisation.

3. Expérimentation.

Les premières expérimentations se sont basées sur des conceptions erronées de l'élève liées à la notion de symétrie orthogonale d'un segment. Ces conceptions ont été analysées en détail par des didacticiens [Tahri 93] et constituent un corpus intéressant pour notre approche. Particulièrement, nous nous intéressons à la conception dite de «*parallélisme*» où l'élève confond les notions de symétrie orthogonale et de parallélisme (voir figure ci-dessous).

L'expérimentation sous CLINT a confirmé ses limites déjà mises en évidence lors de l'évaluation de ce système [Van Labeke 95] : quelles que soient les modifications apportées à la théorie du domaine ou à la définition du concept cible, le processus entraîne un temps d'induction trop important et ne franchit pas le cap du premier

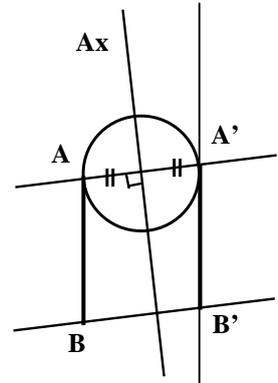
exemple introduit.

Nous avons alors utilisé le système FOCL qui présente l'inconvénient d'exiger la présence de contre-exemples du concept cible. Cette limitation a été contournée en ne se basant que sur un seul exemple de la conception «*parallélisme*» et en autorisant le système à générer automatiquement les contre-exemples à partir de cet ensemble d'exemples positifs supposé complet.

Les résultats varient selon la paramétrisation du système, notamment selon l'utilisation de la théorie du domaine, mais la définition obtenue la plus conforme aux résultats attendus est la suivante :

symétrique($s1, ax, s2$) :- *segment*($s1, p1, p2, l1$), *segment*($s2, p3, p4, l2$),
parallèle($l1, l2$), *perpendiculaire*($l3, ax$).

Nous obtenons bien le fait que $s1$ et $s2$ sont des segments de droites supports respectives $l1$ et $l2$ et que ces deux droites sont parallèles. Par contre, il n'y a aucune information concernant l'orthogonalité de l'axe et des droites (AA') et (BB'), ni l'égalité des longueurs $|AA'|$ et $|BB'|$.



Conclusion.

Notre objectif était d'étudier la réutilisation en «*boîte noire*» de systèmes de Programmation Logique Inductive. Les résultats obtenus sur des exemples de conceptions erronées analysées par des didacticiens sont globalement positifs mais mettent en évidence une inadéquation partielle des systèmes utilisés.

Les problèmes rencontrés avec CLINT et FOCL proviennent essentiellement de la particularité de notre approche vis-à-vis de la Programmation Logique Inductive en générale : un nombre réduit d'exemples positifs, l'absence d'exemples négatifs et une théorie du domaine importante et intrinsèquement récursive.

Pour prendre en compte cette spécificité, d'autres systèmes sont actuellement en cours d'expérimentation. A plus long terme, il nous semble indispensable d'envisager la construction d'un processus d'induction adapté à notre contexte. Cette démarche peut être facilitée par l'existence de nombreuses bibliothèques implémentant la plupart des mécanismes de PLI abordés dans la littérature.

Bibliographie.

- [Aha 92] Aha D.W., *Relating Relational Learning Algorithms*, in Muggleton S. (Ed.), *Inductive Logic Programming*, Academic Press, London, pp 233-259 (1992)
- [De Raedt 91] De Raedt L., *Interactive Theory Revision: an Inductive Logic Programming Approach*, Academic Press, London (1991)
- [Desmoulin 94] Desmoulin C., *Etude et réalisation d'un système tuteur pour la construction de figures géométriques*, Thèse, Université Joseph Fourier - Grenoble I (1994)
- [Ellman 89] Ellman T., *Explanation-Based Learning: A Survey of Programs and Perspectives*, ACM Computing Surveys, Vol. 21, 2, pp 163-221 (1989)
- [Muggleton et al. 92] Muggleton S., Feng C., *Efficient Induction of Logic Programs*, in S. Muggleton (Ed.), *Inductive Logic Programming*, pp 281-298 (1992)
- [Muggleton et al. 94] Muggleton S., De Raedt L., *Inductive Logic Programming: Theory and Methods*, Journal of Logic Programming 19-20, pp 629-679 (1994)
- [Pazzani et al. 92] Pazzani M., Kibbler D., *The Utility of Knowledge in Inductive Learning*, Machine Learning, 9, pp 57-94 (1992)
- [Quinlan 90] Quinlan J.R., *Learning Logical Definitions from Relations*, Machine Learning, 5, pp 239-266 (1990)
- [Tahri 93] Tahri S., *Modélisation de l'interaction didactique : un tuteur hybride sur Cabri-Géomètre pour l'analyse de décisions didactiques*, Thèse, Université Joseph Fourier - Grenoble I (1993)
- [Van labeke 95] Van labeke N., *Programmation Logique Inductive et génération automatique d'un modèle des croyances de l'apprenant*, Mémoire de DEA, Université Henri Poincaré - Nancy I (1995)