

Un logiciel permettant d'apprendre des règles et leurs exceptions : Area

Sylvain Lagrue*, Jérémie Lussiez*, Julien Rossit*

* CRIL - Université d'Artois
Faculté des Sciences Jean Perrin
Rue Jean Souvraz - SP 18
62307 Lens Cedex
lagrue@cril.univ-artois.fr
jeremie_lussiez@ens.univ-artois.fr
julien_rossit@ens.univ-artois.fr

1 Problématique

Le raisonnement à base de règles générales pouvant comporter différentes exceptions et le raisonnement non-monotone sont des domaines qui ont été bien étudiés et formalisés en Intelligence Artificielle. Ainsi, le Système *P* (Kraus et al., 1990) fournit un ensemble de postulats de rationalité permettant de définir les conclusions plausibles pouvant être obtenues à partir d'un ensemble de règles pouvant contenir des exceptions. De plus, différentes méthodes de raisonnement, en accord avec le Système *P*, ont été proposées. Une question cependant subsistait : comment obtenir de telles règles à partir d'informations fréquentielles, en d'autres termes, comment apprendre de telles règles ?

De récents travaux ont montré comment se baser sur des distributions de probabilités particulières, les distributions de probabilités à *grandes marches* (Snow, 1999), afin d'obtenir des règles et leurs exceptions. Dans une distributions de probabilités à *grandes marches*, chaque élément à une probabilité supérieure à la somme des probabilités des évènements qui lui sont moins probables. Contrairement aux approches classiques basées sur les règles associatives, les règles ainsi extraites peuvent être utilisées dans le cadre du raisonnement non-monotone, en accord avec le Système *P* et avec la base initiale (Benferhat et al., 2003).

Cependant, ces distributions de probabilités à *grandes marches* ne peuvent être obtenues qu'en regroupant les différents individus de la base (simple) d'apprentissage, chaque regroupement pouvant aboutir à des ensembles de règles différents et incompatibles. À ce jour, aucun algorithme de regroupement réellement satisfaisant, tant d'un point de vue de temps de calcul que des règles générées, n'avait été proposé.

Les contributions de ce travail se déclinent en deux points principaux :

- la proposition de différents algorithmes de regroupement d'où des règles peuvent être générées ;
- l'implémentation de ces différents algorithmes afin de les valider (tant sur les temps de calcul que sur la qualité des règles extraites).

Un logiciel permettant d'apprendre des règles et leurs exceptions : Area

2 Présentation de la méthode et du logiciel

L'extraction des règles depuis la base d'observations se décompose en 3 phases bien distinctes : une phase de dénombrement de chaque observation ; une phase de regroupement de ces observations afin d'obtenir des distributions à *grandes marches* ; une phase d'extraction des règles à partir du regroupement précédent.

La phase de regroupement est une phase déterminante pour la génération de règles. En effet, l'unique moyen d'influer sur la qualité et la pertinence des règles apprises se situe donc dans la construction des différentes classes de la base initiale. Or, il est impossible de calculer tous les regroupements possibles et de choisir le meilleur, le nombre de ces regroupements étant exponentiel.

Le logiciel. Le logiciel en démonstration et développé par les auteurs, Area, comporte différents algorithmes reposant sur différents facteurs, le but de chacun de ces algorithmes étant de tendre vers un regroupement optimal permettant de générer le meilleur ensemble de règles possible. Le logiciel offre également la possibilité de modifier les regroupements générés par les algorithmes, au moyen d'outils de manipulation des regroupements permettant à un utilisateur de déplacer les observations. Ces fonctions permettent d'affiner les règles apprises par le système.

Il peut être noté que la robustesse d'une règle dépend du regroupement dont elle est issue : plus la règle provient d'un regroupement avec une population élevée (ie. plus elle est générique) moins elle sera sensible à l'ajout de nouveaux éléments dans la base.

Limites et perspectives. De par le formalisme dont il est issu, le moteur d'Area ne permet actuellement l'apprentissage de règles qu'à partir des bases d'observations décrites suivant des attributs binaires. Or, la plupart des attributs contenus dans des bases réelles ne sont pas binaires. Des attributs multivalués devraient pouvoir être discrétisés et binarisés. De plus, une validation beaucoup plus poussée devra être effectuée afin de tester nos algorithmes. Un vaste champ d'investigation pour ces algorithmes pourrait être la sécurité et la détection d'intrusion.

Références

- Benferhat, S., D. Dubois, S. Lagrue, et H. Prade (2003). A big-stepped probability approach for discovering default rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS) 11*, 1–14.
- Kraus, S., D. Lehmann, et M. Magidor (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence 44*, 167–207.
- Snow, P. (1999). Diverse confidence levels in a probabilistic semantics for conditional logics. *Artificial Intelligence 113*, 269–279.

Summary

Area is a Java software which purpose is to extract default rules from simple databases. The main property of these rules is to be compatible with the System *P* of Kraus, Lehmann and Magidor. Hence, these rules can be used with non-monotonic reasoning systems.