

# Apprentissage supervisé adaptatif de Concepts Formels à partir des données nominales

Nida Meddouri\*, Mondher Maddouri\*

\*Research Unit on Programming, Algorithmics and Heuristics - URPAH  
Faculty of Science of Tunis - FST  
University of Tunis-El Manar  
Campus Universitaire, EL Manar, 1060, Tunis, Tunisia  
nmeddouri@gmail.com, mondher.maddouri@fst.rnu.tn

## 1 Introduction

Nous proposons une amélioration d'une méthode dite *BFC* (*Boosting Formal Concept*) (Meddouri et Maddouri, 2009). Cette méthode est basée sur l'*Analyse de Concepts Formels* et bénéficiant des avantages des algorithmes de *Boosting*. L'amélioration proposée permet de traiter des données non binaires (i.e nominales). Dans la section 3, nous décrivons l'extension proposée après avoir discuter le *BFC* dans la section 2.

## 2 Boosting de Concepts Formels

Le *Boosting* est une approche adaptative, qui permet de classer correctement un objet mal classé par un classifieur ordinaire. L'idée est de construire plusieurs classifieurs (des experts) qui se complètent afin de construire un classifieur plus performant. Ce qui permet de réduire le taux d'erreurs de n'importe quel classifieur. De ce fait, on peut prouver que le *Boosting* est une méthode générale pour convertir un classifieur faible en un classifieur fiable.

*BFC* est une approche adaptative et itérative. À chaque itération, l'algorithme d'apprentissage de *BFC* sélectionne l'attribut qui maximise le *gain informationnel* à partir des données binaires. Les concepts pertinents sont calculés à partir de la fermeture de cet attribut binaire. Une règle de classification est obtenue en cherchant la classe majoritaire associée à l'extension du concept pertinent.

## 3 Boosting des Concepts à partir des données nominales

La méthode proposée est une extension aux données nominales de la méthode *BFC* conçue pour les données binaires.

Notre approche, basée essentiellement sur l'*AdaBoost.M2*. Nous exécutons  $T$  fois l'algorithme d'apprentissage sur des distributions variées d'instances d'apprentissage. Pour chaque

itération, nous n’obtenons pas de nouvelles instances mais nous nous satisfaisons de perturber la distribution en modifiant les poids des instances de la base d’apprentissage.

Initialement, l’algorithme affecte des poids égaux à ces instances d’apprentissage :  $\mathcal{O}$ . Ensuite, notre algorithme d’apprentissage proposé, débute à ce niveau afin de générer le concept pertinent comme étant un classifieur faible. Il sélectionne un autre ensemble  $O_t$  (à partir de l’ensemble des données nominales sélectionnées précédemment) par un tirage probabiliste à partir de l’ensemble d’apprentissage  $\mathcal{O}$ . Nous rappelons que l’ensemble d’apprentissage  $\mathcal{O}$  est décrit par un ensemble d’attributs nominaux  $AN$  (qui ne sont pas forcément binaires).

$$AN = \{AN_l | l = \{1, \dots, L\}, L \leq m \text{ et } \exists o_i \in \mathcal{O}, \exists p \in P, AN_l(o_i) = p\}. \quad (1)$$

Notre algorithme d’apprentissage proposé se concentre sur  $O_t$ . Il extrait le concept pertinent à partir de cet ensemble  $O_t$  en sélectionnant l’attribut qui maximise le *gain informationel*. Une fois l’attribut nominal  $AN^*$  est retenu, nous cherchons les instances associées à chaque valeur  $v_j$  de cet attribut,  $\delta(AN^* = v_j)$ .

**Proposition 1 :** Dans un contexte nominal (multi-valué), l’opérateur  $\delta$  peut être calculé par :

$$\delta(AN^* = v_j) = \{o \in O_t | AN^*(o) = v_j\}. \quad (2)$$

Ensuite, nous cherchons les attributs vérifiés par les ensembles des instances retenues précédemment (en utilisant l’opérateur de fermeture  $\delta \circ \varphi(AN^* = v_j)$ ) sachant que l’opérateur  $\varphi$  est défini comme suit :

**Proposition 2 :** Dans un contexte nominal (multi-valué), l’opérateur  $\varphi$  peut être calculé par :

$$\varphi(B) = \{v_j | \forall o, o \in B \text{ et } \exists AN_l \in AN | AN_l(o) = v_j\}. \quad (3)$$

À ce niveau là, on peut avoir un concept pertinent correspondant à chaque valeur  $v_j$  ( $\delta(AN^* = v_j)$ ,  $\delta \circ \Phi(AN^* = v_j)$ ). Une règle de classification est obtenue en cherchant la classe majoritaire associée à l’extension du concept pertinent ( $\delta(AN^* = v_j)$ ). La partie conclusion de la règle est formée par la classe majoritaire. La partie condition de la règle est formée par les attributs de l’intention du concept ( $\delta \circ \varphi(AN^* = v_j)$ ). Dans la suite, notre algorithme utilise les règles générées pour classer l’ensemble des données d’apprentissage  $\mathcal{O}$ . Donc pour chaque itération, notre algorithme génère un classifieur  $h_t$  représenté sous forme d’un ensemble de règles de classification.

## 4 Conclusion

Nous avons proposé une amélioration d’une méthode adaptative de classification : le *do-page* (*Boosting*) de classifieurs basés sur les concepts formels (*Formal Concepts*) : *BFC*. L’extension proposée vise à élargir les champs d’application de l’*ACF* sur un type de données autre que le binaire (i.e nominal).

## Références

Meddouri, N. et M. Maddouri (January 2009). Générer des règles de classification par *do-page* de concepts formels. In *Proceedings of the 9èmes Journées Francophones en Extraction et Gestion des Connaissances (EGC’09), Strasbourg, FRANCE*.