

Générer des règles de classification par Dopage de Concepts Formels

Nida Meddouri*, Mondher Maddouri**

*Unité de Recherche en Programmation, Algorithmique et Heuristiques - URPAH
Faculté des Sciences de Tunis - FST
Tunis - Université d'El Manar
Campus universitaire El Manar, 1060, Tunis, Tunisie
nmeddouri@gmail.com

**Département des sciences mathématiques et informatiques
Institut National des Sciences Appliquées et de Technologie de Tunis - INSAT
Université 7 Novembre à Carthage.
Zone industrielle nord, B.P. 676, 1080 TUNIS CEDEX, TUINISIE
mondher.maddouri@fst.mu.tn

Résumé. La classification supervisée est une tâche de fouille de données (Data Mining), qui consiste à construire un classifieur à partir d'un ensemble d'exemples étiquetés par des classes (phase d'apprentissage) et ensuite prédire les classes des nouveaux exemples avec ce classifieur (phase de classification). En classification supervisée, plusieurs approches ont été proposées dont l'approche basée sur l'Analyse de Concepts Formels. L'apprentissage de Concepts Formels est basé généralement sur la structure mathématique du treillis de Galois (ou treillis de concepts). Cependant, la complexité exponentielle de génération d'un treillis de Galois a limité les champs d'application de ces systèmes. Dans cet article, nous présentons plusieurs méthodes de classification supervisée basées sur l'Analyse de Concepts Formels. Nous présentons aussi le boosting (dopage) de classifieurs, une technique de classification innovante. Enfin, nous proposons le boosting de concepts formels, une nouvelle méthode adaptative qui construit seulement une partie du treillis englobant les meilleurs concepts. Ces concepts sont utilisés comme étant des règles de classification. Les résultats expérimentaux réalisés ont prouvé l'intérêt de la méthode proposée par rapport à celles existantes.

1 Introduction

L'Analyse de Concepts Formels est une formalisation de la notion philosophique de concept, défini comme étant un couple d'extension et de compréhension du concept. La compréhension d'un concept (appelée aussi intension) fait référence aux attributs nécessaires et suffisants pour le caractériser. L'extension d'un concept est l'ensemble des exemples qui ont permis d'apprendre ce concept (Ganter et Wille, 1997).

La classification basée sur l'Analyse de Concepts Formels est une approche de fouille de données symboliques qui permet d'extraire des corrélations, des motifs et des règles, selon les concepts générés à partir des données (Carpineto et Romano, 2004). La classification supervisée est un processus composé de deux phases. Une phase d'apprentissage permet d'organiser l'information extraite d'un ensemble d'objets sous forme d'un treillis de concepts. La phase de classification permet de déterminer la classe de nouveaux objets. Une grande diversité des méthodes d'apprentissage basées sur l'Analyse de Concepts Formels a été proposée parmi lesquelles nous citons : GRAND (Oosthuizen, 1988), LEGAL (Liquiere et Mephu Nguifo, 1990), GALOIS (Carpineto et Romano, 2004), RULEARNER (Sahami, 1995), CIBLe (Njwoua et Mephu Nguifo, 1999), CLNN&CLNB (Xie et al., 2002), IPR (Maddouri, 2004), NAVIGALA (Guillas et al., 2006) et CITREC (Douar et al., 2008).

Un grand nombre de travaux en apprentissage supervisé ont porté cette dernière décennie sur les méthodes de dopage (Boosting) de classifieurs ayant pour but d'améliorer les performances d'un classifieur unique (généralement un classifieur faible) par des techniques de votes (Breiman, 1996). Deux raisons principales à cette large utilisation sont probablement : la simplicité de mise en oeuvre et également les théorèmes récemment édictés, relatifs aux bornes, aux marges, ou encore à la convergence du boosting. Le boosting est connu pour améliorer les performances de n'importe quel algorithme d'apprentissage, supposé néanmoins et a priori instable et ayant des classifieurs faibles, appelé (weak learner).

Malheureusement, les systèmes basés sur l'Analyse de Concepts Formels encourent quelques difficultés comme : la complexité exponentielle (dans le pire des cas), un taux d'erreurs élevé et le sur-apprentissage. Heureusement, les algorithmes de boosting sont connus par le pouvoir de diminuer les taux d'erreurs d'un faible classifieur.

Dans ce papier, nous proposons une nouvelle méthode basée sur l'Analyse de Concepts Formels et bénéficiant des avantages des algorithmes de boosting. Dans la section 2, nous présentons les notions de base de l'Analyse de Concepts Formels. Dans la section 3, nous décrivons la méthode proposée : le Boosting de Concept Formels. Ensuite, nous concluons en discutant les résultats expérimentaux pour prouver la validité de la méthode proposée.

2 Notions d'Analyse de Concepts Formels

Un contexte formel est un triplet $k = \langle O, P, R \rangle$, avec $O = \{o_1, o_2, \dots, o_n\}$ un ensemble fini d'éléments appelés objets (instances, exemples), $P = \{p_1, p_2, \dots, p_m\}$ un ensemble fini d'éléments appelés propriétés (attributs) et R une relation binaire définie entre O et P . La notation $(o,p) \in R$ ou bien $R(o,p) = 1$, signifie que l'objet formel o vérifie la propriété p dans la relation R (Ganter et Wille, 1997). Le contexte peut être représenté par un tableau croisé ou bien un tableau binaire comme il est présenté dans le tableau 1 (extrait de (Mephu Nguifo et Njwoua, 2005)).

Supposons $A \subseteq O$ et $B \subseteq P$ deux ensembles finis. Pour les deux ensembles A et B , les opérateurs $\varphi(A)$ et $\delta(B)$ sont définis comme suit (Ganter et Wille, 1997) :

- $\varphi(A) := \{ p \mid \forall o, o \in A \text{ et } (o,p) \in R \}$.
- $\delta(B) := \{ o \mid \forall p, p \in B \text{ et } (o,p) \in R \}$.

L'opérateur φ définit les propriétés partagées par tous les éléments de A . L'opérateur δ définit les objets qui partagent les mêmes propriétés de l'ensemble B . Les deux opérateurs φ

O-P	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	CLASSE
o ₁	1	1	1	1	1	1	1	0	1
o ₂	1	1	1	1	1	1	0	1	1
o ₃	1	1	1	1	1	0	1	1	1
o ₄	1	1	1	1	0	1	0	0	1
o ₅	1	1	0	1	1	0	1	0	2
o ₆	1	1	1	0	1	0	0	1	2
o ₇	1	0	1	0	0	1	0	0	2

TAB. 1 – Illustration d'un contexte formel.

et δ définient la correspondance de Galois entre les deux ensembles A et B. Les opérateurs de fermeture sont $B'' = \delta \circ \varphi(B)$ et $A'' = \varphi \circ \delta(A)$. Enfin, les ensembles fermés A et B sont définis par $B = \delta \circ \varphi(B)$ et $A = \varphi \circ \delta(A)$.

Un concept formel du contexte $\langle O, P, R \rangle$ est un couple (A, B) avec $A \subseteq O$, $B \subseteq P$ et $\varphi(A) = B$, $\delta(B) = A$. L'ensemble A et B sont appelés respectivement le domaine (extension) et le co-domaine (intention) du concept formel.

A partir d'un contexte formel $\langle O, P, R \rangle$, on peut extraire tous les concepts possibles. L'ensemble des concepts peut être représenté en treillis complet de concepts (Appelé aussi treillis de Galois (Ganter et Wille, 1997)), quand on définit la relation d'ordre partiel ' \ll ' entre deux concepts : $(A_1, B_1) \ll (A_2, B_2)$ si et seulement si $(A_1 \subseteq A_2)$ et $(B_2 \subseteq B_1)$. Les concepts sont dits noeuds du treillis. Le diagramme de Hasse est la représentation graphique du treillis de concepts (Ganter et Wille, 1997). La Figure 1 représente le treillis de concepts associé au contexte présenté dans le Tableau 1.

3 La méthode proposée

Notre approche, basée essentiellement sur l'AdaBoostM2 (Freund et Schapire, 1996), est décrite avec plus de détails dans Algorithme 2. Initialement, l'algorithme affecte des poids égaux aux exemples d'apprentissage O. Ensuite, notre algorithme d'apprentissage proposé, débute à ce niveau (algorithme 1) afin de générer le concept pertinent comme étant un classifieur faible. Il sélectionne un autre ensemble O_t par un tirage probabiliste à partir de O.

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 minimise la fonction d'entropie de Shannon. Si plusieurs attributs ont la même entropie, on sélectionne celui qui a le support le plus important. Une fois l'attribut p^* est retenu, on cherche les exemples qui vérifient cet attribut retenu $\delta(\{p^*\})$. Ensuite, on cherche les attributs vérifiés par l'ensemble des exemples retenus précédemment (en utilisant l'opérateur de fermeture $\delta \circ \varphi(\{p^*\})$). A ce niveau là, on a construit un concept pertinent $(\delta(\{p^*\}), \delta \circ \varphi(\{p^*\}))$. On cherche la classe majoritaire associée à l'extension du concept pertinent $\delta(\{p^*\})$. De ce fait, on induit une règle de classification. La partie conclusion de la règle est formée par la classe majoritaire. La partie condition de la règle est formée par la conjonction des attributs de l'intention du concept $\delta \circ \varphi(\{p^*\})$. Dans la suite,

Générer des règles de classification par Dopage de Concepts Formels

notre algorithme utilise la règle générée pour classer l'ensemble des données d'apprentissage O .

Algorithme 1 : L'algorithme d'apprentissage d'un concept pertinent

ENTREE : n exemples d'apprentissage $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ étiquetés $y_i \in Y$

SORTIE : Une règle de classification.

DEBUT :

1. Sélectionner les exemples par un tirage probabiliste à partir de O : O_t .
2. A partir de O_t , déterminer l'attribut qui a la valeur d'entropie la plus faible : p^* .
S'il y a plusieurs attributs ayant la même valeur d'entropie, choisir celui qui a plus de support.
3. Calculer la fermeture associée à cet attribut afin de générer le concept pertinent : $(\delta(\{p^*\}), \delta \circ \varphi(\{p^*\}))$.
4. Déterminer la classe majoritaire associée à $\delta(\{p^*\})$: y^* .
5. Induire la règle de classification h_t : la conjonction des attributs de $\delta \circ \varphi(\{p^*\})$ implique l'appartenance à la classe majoritaire y^* .

FIN.

Pour chaque itération, notre algorithme génère un classifieur h_t représenté sous forme d'une règle de production. En reprenant l'algorithme de AdaBoost.M2 (Algorithme 2), ce classifieur prédit a posteriori la classe y_i de chaque exemple o_i . Trois cas se présentent :

- si $h_t(o_i, y_i) = 1$ et $h_t(o_i, y) = 0, \forall y \neq y_i$ alors h_t a prédit correctement la classe de o_i .
- si $h_t(o_i, y_i) = 0$ et $h_t(o_i, y) = 1, \forall y \neq y_i$ alors h_t a prédit inversement la classe de o_i .
- si $h_t(o_i, y_i) = h_t(o_i, y), \forall y \neq y_i$ alors la classe de o_i est sélectionnée aléatoirement entre y et y_i .

A partir de cette interprétation, on définit la pseudo-perte du classifieur h_t via la distribution W_t comme suit : $\varepsilon_t = 0.5 \times \sum_{(o_i, y_i) \in O} W_t(o_i, o_y)(1 - h_t(o_i, y_i) + h_t(o_i, y))$.

D'où, on calcul l'erreur $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ et on met à jour les poids des exemples selon β_t et Z_t (avec Z_t une constante de normalisation choisie de façon à ce que $\sum_{i=1}^n W(o_i, y_i) = 1$). Donc, notre approche proposée modifie les poids des exemples d'apprentissage en diminuant les poids de ceux qui ont été bien classés et en augmentant les autres. Après T itérations, on obtient finalement notre classifieur via : $h_{fin}(o) = \text{argmax}_{y \in Y} \sum_{t=1}^T \log(1/\beta_t) \times h_t(o, y)$. En fait, le classifieur final h_{fin} est un ensemble de classifieurs faibles générés depuis les T itérations (chaque itération génère un classifieur faible ou met à jour le poids d'un classifieur faible).

4 Conclusion

Dans ce papier, on s'est intéressé aux approches de classification basées sur l'Analyse de Concepts Formels. On a étudié la méthode GRAND (basé sur le treillis de concepts), IPR (basé sur la couverture de concepts) et CITREC (basé sur le treillis de concepts d'un contexte réduit). Ensuite, on a étudié les notions de base du Boosting, spécialement l'AdaBoost et ses avantages.

Algorithme 2 AdaBoost.M2 (Freund et Schapire, 1996)

ENTREE :

Une base d'apprentissage de n exemples $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ étiquetés par $y_i \in Y$.

Un algorithme d'apprentissage faible /* Algorithme 1 */.

Un entier T : nombre des itérations.

SORTIE : l'hypothèse final h_{fin} : Classifieur.

DEBUT :

1 : Initialiser la distribution $W_t(o_i, y_i) = 1/|O|$ pour tout $(o_i, y_i) \in O$.2 : Pour $t=1$ à T faire2.1 : Appliquer le classifieur faible (Algorithme 1) en le traînant sur la distribution W_t .2.2 : Retenir l'hypothèse h_t .2.3 : Calculer la pseudo perte de h_t : ε_t .2.4 : Définir $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.2.5 : Mettre a jour W_t : $W_{t+1}(o_i, y_i) = \beta_t^{0.5 \times (1 + h_t(o_i, y_i) - h_t(o_i, y))} \times W_t(o_i, y_i) / Z_t$.

Fin Boucle.

3 : $h_{fin}(o) = \arg \max_{y \in Y} \sum_{t=1}^T \log(1/\beta_t) \times h_t(o, y)$.

FIN.

Enfin, on a présenté notre approche basée sur le Boosting de Concepts Formels qui construit partiellement une partie du treillis de concepts formée seulement par des concepts formels dits pertinents. La méthode de Boosting de Concepts Formels a une complexité polynômiale meilleure que les autres méthodes basées sur ACF. Ce qui explique sa rapidité. En utilisant des bases de données connues, notre approche a montré de bonnes performances au niveau de la réduction des nombres de concepts comparée aux approches basées sur la construction du treillis. Nos futures travaux, se concentrent sur une évaluation plus poussée de cette méthode par son application à divers échantillons de données.

Dans la pratique, nous pensons qu'il est nécessaire de régulariser le nombre d'itérations pour l'apprentissage. T ne doit pas être fixé dès le début. Plusieurs critères ont été proposés en Boosting pour déterminer T. Dans le future, nous envisageons étudier ces critères et les utiliser au sein de notre méthode. Dans la pratique aussi, les outliers déstabilisent rapidement le boosting en produisant des poids grands. De ce fait, il en existe des heuristiques pour limiter les effets des exemples ou bien de détecter et éliminer les outliers. Nous envisageons étudier ces heuristiques pour améliorer la méthode proposée.

Références

Breiman, L. (1996). Bagging predictors. *Journal Machine Learning* 24, 123–140.

Carpineto, C. et G. Romano (2004). *Concept Data Analysis: Theory and Applications*. New Jersey, USA: John Wiley and Sons, Inc.

Douar, B., C. C. Latiri, et Y. Slimani (2008). Approche hybride de classification supervisée à base de treillis de galois : application à la reconnaissance de visages. In *Conférence Extraction et Gestion des Connaissances (EGC08)*, pp. 309–320.

Générer des règles de classification par Dopage de Concepts Formels

- Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Journal Information and Computation* 121, 256–285.
- Freund, Y. et R. E. Schapire (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning (ICML96)*, pp. 148–156.
- Ganter, B. et R. Wille (1997). *Formal Concept Analysis: Mathematical Foundations*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Guillas, S., K. Bertet, et J.-M. Ogier (2006). Reconnaissance de symboles bruités à l'aide d'un treillis de galois. In *Colloque International Francophone sur l'Ecrit et le Document (CIFED06)*, pp. 85–90.
- Liquiere, M. et E. Mephu Nguifo (1990). Legal : learning with galois lattice. In *Journées Françaises sur l'Apprentissage (JFA90)*, pp. 93–113.
- Maddouri, M. (2004). Towards a machine learning approach based on incremental concept formation. *Journal Intelligent Data Analysis* 8, 267–280.
- Mephu Nguifo, E. et P. Njiwoua (2005). Treillis de concepts et classification supervisée. *Journal Technique et Science Informatiques* 24, 449–488.
- Njiwoua, P. et E. Mephu Nguifo (1999). Améliorer l'apprentissage à partir d'instances grâce à l'induction de concepts : le système cible. *Revue d'Intelligence Artificielle* 13, 413–440.
- Oosthuizen, D. (1988). *The use of a Lattice in Knowledge Processing*. Thèse d'université, University of Strathclyde, Glasgow.
- Sahami, M. (1995). Learning classification rules using lattices (extended abstract). In *European Conference on Machine Learning (ECML95)*, pp. 343–346.
- Xie, Z., W. Hsu, Z. Liu, et M.-L. Lee (2002). Concept lattice based composite classifiers for high predictability. *Journal Experimental and Theoretical Artificial Intelligence* 14, 143–156.

Summary

Supervised classification is a spot/tasks of data mining which consists in building a classifier from a set of examples labeled by their class (learning step) and then to predict the class of new examples with a classifier (classification step). In supervised classification, several approaches were proposed such as: Induction of Decision Trees, and Formal Concept Analysis. The learning of formal concepts is based, generally, on the mathematical structure of Galois lattice (or concepts lattice). The complexity of generation of Galois lattice, limited the application fields of these systems. In this paper, we present several methods of supervised classification based on Formal Concept Analysis. We also present the boosting of classifiers, an emerging technique of classification. Finally, we propose the boosting of formal concepts: a new adaptive approach to build only a part of the lattice including the best concepts. These concepts are used as classification rules. Experimental results are given to prove the interest of the proposed method.