



HAL
open science

Apprentissage d'Ensembles Parallèles de Règles de Classification par Analyse de Concepts Formels

Nida Meddouri, Hela Khoufi, Monder Maddouri

► **To cite this version:**

Nida Meddouri, Hela Khoufi, Monder Maddouri. Apprentissage d'Ensembles Parallèles de Règles de Classification par Analyse de Concepts Formels. Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014, Jun 2014, France. hal-00989205

HAL Id: hal-00989205

<https://hal.science/hal-00989205v1>

Submitted on 9 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apprentissage d'Ensembles Parallèles de Règles de Classification par Analyse de Concepts Formels

Nida Meddouri¹

Hela Khoufi¹

Mondher Maddouri²

¹ Laboratoire d'Informatique, Programmation, Algorithmique et Heuristique - LIPAH
Faculté des Sciences mathématiques, physiques et naturelles de Tunis - FST

Université d'El Manar

Campus universitaire El Manar, 1060, Tunis, Tunisie

nida.meddouri@gmail.com, hela.khoufi@gmail.com

² College of Community, Hinakiyah

Taibah University - Medinah Monawara

Kingdom of Saudi Arabia

maddourimondher@yahoo.fr

Résumé

La classification supervisée est une tâche de fouille de données qui consiste à construire un classifieur à partir d'un ensemble d'instances étiquetées (étape d'apprentissage) et à prédire ensuite la classe de nouvelles instances présentées au classifieur construit (étape de la classification). Dans ce domaine, plusieurs approches ont été proposées telles que : l'Induction de l'Arbre de Décision et l'Analyse de Concepts Formels. L'apprentissage de concepts formels est basé, en général, sur la structure mathématique de treillis de Galois (ou treillis de concepts). La complexité de la génération du treillis de Galois limite les champs d'application de ces systèmes. Dans cet article, nous traitons la classification supervisée. Nous présentons une taxonomie des méthodes basées sur l'Analyse des Concepts Formels. Nous proposons une nouvelle approche qui construit seulement une partie du treillis, incluant les meilleurs concepts (concepts pertinents). Ces concepts sont utilisés comme classifieurs dans une combinaison parallèle à l'aide du vote. La méthode proposée est basée sur le Dagging de classifieurs nominaux. Les résultats expérimentaux obtenus sur des échantillons de données différentes prouvent l'intérêt de la méthode proposée.

Mots Clef

Concept formel, Méthode d'ensemble, Dagging, Règles de classification, Apprentissage automatique, Fouille de données.

Abstract

Supervised classification is a spot/task of data mining which consists at building a classifier from a set of examples labeled by their class (learning step) and then

predicting the class of new examples with a classifier (classification step). In supervised classification, several approaches were proposed such as : Induction of Decision Tree, and Formal Concept Analysis. The learning of formal concepts is based, generally, on the mathematical structure of Galois lattice (or concept lattice). The complexity of Galois lattice generation limits the application fields of these systems. In this paper, we discuss about supervised classification based on Formal Concept Analysis and we present methods based on concept lattice or sub lattice. We propose a new approach that builds only a part of the lattice including the best concepts (i.e pertinent concepts). These concepts are used as classifiers in parallèle combination using voting rule. The proposed method is based on Dagging of Nominal Classifier. Experimental results are given to prove the interest of the proposed method.

Keywords

Formal Concept, Ensemble method, Dagging, Classification Rules, Machine Learning, Data Mining.

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. La compréhension (appelé aussi intention) fait référence aux attributs nécessaires et suffisants qui caractérisent ce concept. L'extension est un ensemble d'instances qui ont permis de découvrir le concept [8].

La classification basée sur l'Analyse de Concepts Formels est une approche symbolique permettant l'extraction des corrélations et des règles selon les concepts découverts à partir des données [4]. La classification supervisée est

un processus constitué de deux étapes. Dans l'étape d'apprentissage, les informations extraites à partir d'un groupe d'objets sont organisées sous forme d'un treillis. Dans l'étape de classification, la classe de nouveaux objets est déterminée grâce aux concepts extraits. Plusieurs méthodes d'apprentissage basées sur l'*Analyse de Concepts Formels* ont été proposées : GRAND [21], CLNN&CLNB [30], IPR [15], NAVIGALA [28], CITREC[5] et récemment BFC [16] et BNC [20]. Malheureusement, les systèmes basés sur l'*Analyse de Concepts Formels* encourent quelques problèmes tels qu'une complexité exponentielle (au pire des cas), un taux d'erreur élevé et un sur-apprentissage [17] [20].

Dans la dernière décennie, un grand nombre de recherches s'est orienté vers les méthodes d'ensemble dans le but d'améliorer la performance d'un classifieur unique (généralement estimé faible) [18]. À ce propos, plusieurs méthodes d'ensemble ont été proposées [13] comme *Boosting* et *Bagging* qui permettent de générer un ensemble de classifieurs et de les combiner par les techniques de vote. En général, les méthodes d'ensemble sont basées sur l'apprentissage séquentiel ou parallèle. La différence entre les deux approches d'apprentissage provient de la façon de sélectionner les données pour l'apprentissage.

Dans l'apprentissage séquentiel comme celui utilisé dans *Boosting*, un ensemble d'instances est pris en compte dans chaque étape d'apprentissage du classifieur et les poids sont affectés à ces instances en fonction de la performance du classifieur appris. Quant à l'apprentissage parallèle comme celui dans *Bagging*, les données d'apprentissage sont tirées au hasard et avec remise de l'ensemble de données d'origine. Un tel ensemble d'apprentissage est appelé *Bootstrap*. La méthode la plus récente qui se base sur l'apprentissage parallèle est *Dagging (Disjoint samples aggregating)* [1] [25]. Elle crée un certain nombre de groupes disjoints contenant des données stratifiées à partir de l'ensemble de données d'apprentissage original. Le classifieur faible est construit sur chaque groupe. Les prédictions sont ensuite obtenues en combinant les sorties de classifieurs par vote à la majorité [27]. Cette méthode a montré son importance dans les travaux récents. Dans ce travail, nous proposons d'utiliser cette technique pour étudier les ensembles de classifieurs de type concept formel puisque aucune étude n'a porté sur les concepts formels dans le cadre de l'apprentissage parallèle.

Dans la section 2, nous présentons les notions de base de l'*Analyse de Concepts Formels*. Dans la section 3, nous présentons un classifieur de concepts nominaux (CNC) qui se base sur l'*Analyse de Concepts Formels*. Dans la section 4, nous présentons les méthodes de classification les plus connues qui ont été développées dans cette discipline. Dans la section 5, nous proposons une nouvelle méthode de classification exploitant les avantages du *Dagging* pour générer et combiner de manière parallèle des classifieurs faibles de type CNC. Dans la section 6, une étude expé-

$\mathcal{O}-\mathcal{P}$	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	Classe
o_1	x			x				x	No
o_2	x			x					No
o_3		x		x				x	Yes
o_4			x	x				x	Yes
o_5			x		x	x	x		Yes
o_6			x		x	x			No
o_7		x			x	x			Yes
o_8	x				x			x	No
o_9	x					x	x	x	Yes
o_{10}			x	x		x	x		Yes
o_{11}	x				x		x		Yes
o_{12}		x			x				Yes
o_{13}		x	x				x	x	Yes
o_{14}			x	x					No

TABLE 1 – Illustration d'un contexte formel et spécification des attributs.

riementale est présentée pour évaluer la performance des ensembles de classifieurs générés parallèlement selon certains critères notamment le nombre et le type de classifieurs. Une étude comparative est également présentée pour montrer l'importance de l'apprentissage parallèle utilisant l'échantillonnage stratifié par rapport à l'apprentissage séquentiel utilisant l'échantillonnage aléatoire.

2 L'Analyse de Concepts Formels

Si $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$ est un ensemble fini de n instances, $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$ est un ensemble fini de m propriétés (attributs binaires) et \mathcal{R} est une relation binaire définie entre \mathcal{O} et \mathcal{P} , alors on peut présenter un contexte formel par un triplet $\langle \mathcal{O}, \mathcal{P}, \mathcal{R} \rangle$. La notation $(o_i, p_j) \in \mathcal{R}$ ou $\mathcal{R}(o_i, p_j) = 1$ désigne que l'instance o_i vérifie la propriété p_j dans la relation \mathcal{R} [8]. Un contexte est représenté par un tableau croisé ou un tableau binaire comme celui du Table 1¹ qui illustre un contexte formel de l'échantillon de données *Weather* sous format binaire.

Soit $A \subseteq \mathcal{O}$ et $B \subseteq \mathcal{P}$ deux ensembles finis. Pour ces ensembles A et B , les opérateurs $\varphi(A)$ et $\delta(B)$ sont définis comme suit [8] :

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

L'opérateur φ définit les propriétés partagées par tous les éléments de A . L'opérateur δ définit les instances qui ont les mêmes propriétés incluses dans l'ensemble B . Les opérateurs φ et δ définissent une connexion de Galois entre les ensembles \mathcal{O} et \mathcal{P} [8]. Les opérateurs de fermeture sont $A'' = \varphi \circ \delta(A)$ et $B'' = \delta \circ \varphi(B)$. Les ensembles fermés A et B sont donnés par $B = \delta \circ \varphi(B)$ et $A = \varphi \circ \delta(A)$.

Un concept formel du context $\langle \mathcal{O}, \mathcal{P}, \mathcal{R} \rangle$ est une paire (A, B) , avec $A \subseteq \mathcal{O}$, $B \subseteq \mathcal{P}$, $\varphi(A) = B$ et $\delta(B) = A$. Les ensembles

1. L'échantillon de données est disponible depuis *UCI Machine Learning Repository* [2]

A et B sont appelés, respectivement, l'*extension (domaine)* et l'*intention (co-domaine)* du concept formel.

A partir d'un contexte formel $\langle \mathcal{O}, \mathcal{P}, \mathcal{R} \rangle$, nous pouvons extraire tout les concepts possibles. L'ensemble de tous les concepts peut être organisé sous forme d'un treillis complet (dit *treillis de Galois* [8]), quand la relation d'ordre partiel ' \ll ' est défini entre les deux concepts (A_1, B_1) et (A_2, B_2) si et seulement si $(A_1 \subseteq A_2)$ et $(B_2 \subseteq B_1)$. Les concepts (A_1, B_1) et (A_2, B_2) sont dits les noeuds du treillis. Le *diagramme de Hasse* est une représentation mathématique du *treillis de Galois* [8]. La Figure 1 présente le treillis de concepts (*treillis de Galois*) du contexte formel du Table 1.

3 Classifieur des concepts nominaux (CNC)

L'algorithme d'apprentissage CNC (Algorithme 1) considère l'ensemble des instances nominales de l'ensemble d'apprentissage \mathcal{O} décrit par L attributs nominaux \mathcal{AN} (ne sont pas nécessairement binaires).

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

Le concept nominal pertinent est extrait de l'ensemble des instances en sélectionnant l'attribut nominal qui minimise le *Gain Informationnel*. CNC calcule le *Gain Informationnel (GI)* de chaque attribut à partir du contexte d'apprentissage de n instances avec :

$$GI(\mathcal{AN}, \mathcal{O}) = E(\mathcal{O}) - \sum_{j=1}^V \frac{S(v_j)}{n} E(v_j) \quad (2)$$

Le *Gain Informationnel* de l'attribut nominal AN (représenté by V valeurs différentes) est calculé en fonction de l'*Entropie* $E()$ [23] :

$$E(\mathcal{O}) = - \sum_{k=1}^{\mathcal{K}} \frac{n^k}{n} \log\left(\frac{n^k}{n}\right) \quad (3)$$

où \mathcal{K} est le nombre de classes, n est le nombre d'instances de \mathcal{O} et n^k ($k = 1.. \mathcal{K}$) est le nombre d'instances de \mathcal{O} ayant comme classe y_k .

$S()$ calcule le nombre d'apparition de la valeur v_j de l'attribut AN dans \mathcal{O} . La variation de GI dépend de $S(v_j)$ si on néglige la variation de $E(\mathcal{O})$. L'ensemble de données qui contient des instances redondantes (comme dans un échantillonnage aléatoire simple) maximise la valeur de $S(v_j)$ (si une instance contenant v_j est redondante) et minimise le GI de l'attribut correspondant. Cet effet paralysant n'apporte pas beaucoup de diversité dans les valeurs de GI pour les mêmes attributs.

Une fois que l'attribut nominal est sélectionné (AN^*), les instances associées à chaque valeur v_j de l'attribut sélectionné sont extraites (selon la Proposition 1).

Proposition 1 : A partir d'un contexte nominal (multi-valué), l'opérateur δ est donné par (4) :

$$\delta(AN^* = v_j) = \{o \in \mathcal{O} | AN^*(o) = v_j\}. \quad (4)$$

Ensuite, les attributs vérifiés par les ensembles des instances retenues précédemment peuvent être sélectionnés (en utilisant l'opérateur de fermeture $\delta \circ \varphi(AN^* = v_j)$). Pour cela, la proposition suivante est donnée :

Proposition 2 : A partir d'un contexte nominal (multi-valué), l'opérateur φ est donné par (5) :

$$\varphi(B) = \{v_j | \forall o, o \in B \text{ and } \exists AN_l \in \mathcal{AN} | AN_l(o) = v_j\}. \quad (5)$$

À ce niveau là, on peut avoir un concept pertinent correspondant à chaque valeur v_j ($\delta(AN^* = v_j)$, $\delta \circ \varphi(AN^* = v_j)$). Un classifieur faible h_t est obtenu 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 la conjonction des 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} .

Données : $\mathcal{O} = \{(o_1, y_1), \dots, (o_n, y_n)\}$: un ensemble de n instances étiquetées par $y_i \in \mathcal{K}$.

Résultat : Un classifieur h_{CNC} .

Début

A partir de \mathcal{O} , déterminer l'attribut AN^* qui maximise le *Gain Informationnel*;
 Pour AN^* , déterminer la valeur nominale qui a le support le plus important (v);
 Calculer la fermeture associée v
 ($\{\delta(AN^* = v)\}, \delta \circ \varphi(\{AN^* = v\})$);
 Déterminer la classe majoritaire y^* associée à $\delta(AN^* = v)$;
 Induire la règle de classification h_{CNC} ;

Fin

Algorithme 1 : ALGORITHME DU CLASSIFIEUR NOMINAL (CNC)

Nous avons étudié la variation des taux d'erreurs de CNC sur 15 échantillons de données². La performance de CNC est évalué selon de principe de *10 Validations Croisées* dont le principe est de diviser chaque échantillon de données en 10 sous-ensembles. A tour de rôle, chaque sous-ensemble servira pour le test et les autres sous-ensembles pour l'apprentissage [11]. Nous rapportons la variance³ des taux d'erreurs sur chaque échantillon de données : 3.12% sur *Balance Scale*, 24.1% sur *Contact Lenses*, 0.24% sur *Credit German*, 1.7% sur *Diabetes*, 1.05% sur *Ionosphere*, 2.19% sur *Iris*, 0.51% sur *Tic Tac Toe* et 0.53% sur *Transfusion*. Ces variances sont plus ou moins importantes, prouvant que *CNC* est un classifieur instable.

2. Présenter avec plus de détails dans Table 2 de la section 6.

3. Calculée sur la plateforme WEKA (plus de détails sur cette plateforme dans la section 6).

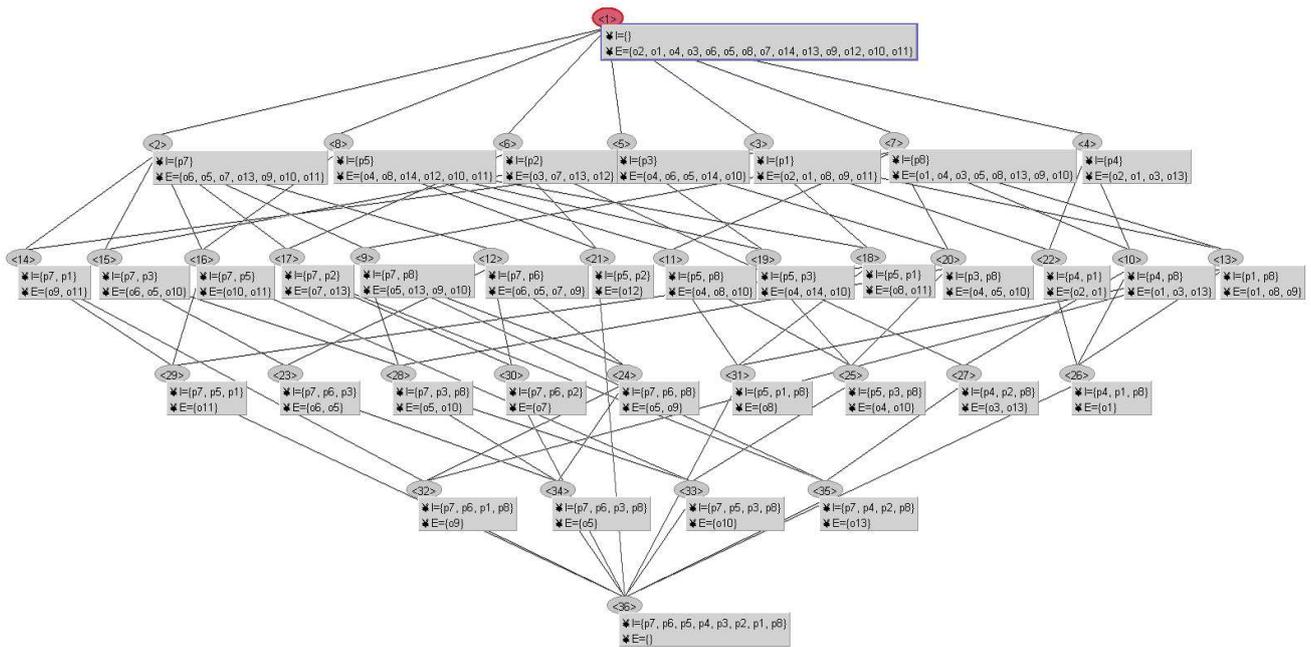


FIGURE 1 – Le treillis de Galois g n r    partir du Table 1.

4 Taxonomie des m thodes de classification bas es sur les Concepts Formels

La classification   partir des treillis de concepts, a pour but de d terminer la classe des nouvelles instances par la navigation dans le treillis. Le treillis de Galois est vu comme un espace de recherche dans lequel nous  voluons d'un niveau vers un autre en validant les caract ristiques associ es aux concepts [17]. Dans la litt rature, plusieurs m thodes de classification se basent sur un treillis complet ou un semi-treillis de concepts pour construire leurs mod les d'apprentissage.

  partir d'un treillis complet de concepts, plusieurs m thodes de classification ont  t  propos es dont les plus connus sont GRAND [21] et NAVIGALA [28]. Trois limites communes pour ces m thodes sont   signaler. D'abord, la complexit  de la g n ration du treillis (temporelle et spatiale) est exponentielle. Ensuite, la navigation dans l'immense espace de recherche est difficile [7]. En plus de cela, les donn es utilis es sont binaires. Pour ces raisons, de nombreux chercheurs se sont orient s vers la classification   partir des semi treillis de concepts.

Un demi-treillis de concepts est une structure math matique permettant de repr senter une partie du treillis de concepts d'une fa on s lective. La classification   partir de cette repr sentation reste similaire   celle   partir d'un treillis complet de concepts. La diff rence entre la classification   partir d'un treillis complet et   partir d'un demi-treillis est le nombre de concepts g n r s. En effet, les

m thodes bas es sur la classification   partir d'un demi-treillis g n rent moins de r gles de classification par rapport   celles bas es sur la classification   partir d'un treillis complet.

Les m thodes CLNN&CLNB [30], IPR [15] et CITREC [5] ont la particularit  de construire uniquement un demi-treillis de concepts ce qui permet de r duire la complexit  spatiale et temporelle (temps d'ex cution). Cependant, leur limite est une perte possible d'information dans une repr sentation condens e de donn es (CITREC) ou une reproduction partielle d'un treillis complet. Une limite d'IPR est l'induction des concepts redondants (ayant les m mes instances) [15]. Nous remarquons que pour les m thodes bas es sur la classification   partir d'un demi-treillis, les concepts construits sont choisis en fonction des crit res inappropri s (i.e. la profondeur du treillis, la couverture du contexte, etc.) [17].

Pour les m thodes de classification bas es sur l'Analyse des Concepts Formels cit es pr c demment, nous rapportons les limites communes suivantes : l'absence de l'aspect adaptatif, la manipulation des donn es binaires et une complexit   lev e. En autres, la g n ration des concepts est exhaustive ou non-contextuelle. C'est pourquoi r cemment dans le domaine de l'apprentissage supervis  bas  sur l'Analyse des Concepts Formels, certaines recherches se sont orient es vers l'int gration et l'utilisation des m thodes d'ensemble qui permettent d'am liorer la performance d'un apprenant unique (un classifieur faible) par des techniques de vote [3].

Dans ce contexte, [16] et [20] ont d velopp  la m thode

BFC (*Boosting of Formal Concepts*) basée sur l'Analyse des Concepts Formels et exploitant les avantages de l'apprentissage séquentiel du *Boosting*. Cette méthode opère à partir des données binaires seulement et se base sur AdaBoost.M2 qui est un algorithme adaptatif multi-classe du *Boosting* [6]. Initialement, BFC attribue des poids égaux aux instances d'apprentissage. Une partie des instances d'apprentissage est sélectionnée afin d'extraire le concept pertinent. BFC utilise ce concept pour classer les instances d'apprentissage et mettre à jour leurs poids (en diminuant les poids des celles bien classées et augmentant les poids des celles mal classées). BFC effectue un ré-échantillonnage de données afin d'écartier les instances bien classées et de ne considérer que les instances dures à apprendre. BFC construit d'une façon adaptative une partie du treillis à partir des concepts formels les plus pertinents. BFC fixe le nombre d'itérations à l'avance. De cette façon, le temps d'exécution peut être contrôlé and la meilleure décision peut être donnée à la fin des itérations. BNC (*Boosting of Nominal Concepts*) est une amélioration de BFC qui opère à partir des données nominales et construit adaptativement des concepts nominaux pertinents [20].

D'après [29], la mise à jour adaptative des données dans l'apprentissage séquentiel (*Boosting*) permet d'améliorer la performance de n'importe quel algorithme d'apprentissage, supposé faible. Néanmoins, les capacités de l'apprentissage séquentiel ont été remises en cause dès que les données fortement bruitées sont utilisées. Dans le cas de l'apprentissage parallèle, les données bruitées seront ignorées et le cas échéant seront réparties équiprobablement entre les différents *Boostraps* ou autres sous-ensembles de données ré-échantillonnés [3] [27]) [26]. Dans la section suivante, nous présentons la méthode d'ensemble qui se base sur l'apprentissage parallèle et qui sera exploitée pour générer nos classifieurs nominaux.

5 Méthode proposée

Dans la littérature des méthodes d'échantillonnage de données, l'échantillonnage aléatoire stratifié a fait ses preuves [27]. Les ensembles de données disjoints et stratifiés sont les plus représentatifs de l'ensemble initial des données d'apprentissage. L'apprentissage à partir des échantillons de données stratifiées permet de générer des classifieurs plus performants que ceux générées à partir des données pondérées dans le cas de l'apprentissage séquentiel. Le *Dagging* a la particularité d'apprendre parallèlement à partir des ensembles de données stratifiées [1] [25]. Dans notre étude, nous proposons d'exploiter cette méthode pour générer des classifieurs de type CNC.

La méthode proposée se base sur le *Dagging*. Elle est décrite avec plus de détails dans l'Algorithme 2. Pour générer T classifieurs, l'algorithme d'apprentissage est exécuté T fois sur des sous-ensembles d'apprentissage disjoints et stratifiés (dites *strates* [1] [25]). Chaque *strate*

aura une répartition des instances par classe similaire à celle de l'échantillon d'apprentissage original. Ces strates ($\{O^{\Theta_1}, \dots, O^{\Theta_T}\}$) sont obtenus par un tirage aléatoire de n' instances sans remise à partir de l'échantillon d'apprentissage O , avec $n' < n$. Le principe du DNC (*Dagging Nominal Classifier*) est de considérer ces strates et de construire notre classifieur CNC sur chacune d'elles. Les classifieurs $\{h_1, \dots, h_T\}$ générés sont combinés par la règle du vote majoritaire [27].

Données :

1. T : nombre des classifieurs à générer.
2. O : données d'apprentissage de n instances avec $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ étiquetées par $y_i \in \{1, \dots, \mathcal{K}\}$.

Résultat : h_{vote} : Classifieur final.

Début

Diviser la population des instances en T strates;

Établir la liste la plus complète des instances de chaque strate;

Pour t de 1 à T **Faire**

Calculer le pourcentage P_t des instances de la $t^{ième}$ strate similaire à O ;

Sélectionner aléatoirement des instances à partir de la $t^{ième}$ strate pour créer O^{Θ_t} par respect à P_t ;

Appliquer CNC sur O^{Θ_t} pour générer h_t ;

Fin

$h_{vote} = arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T h_t(o, y)$;

Fin

Algorithme 2 : ALGORITHME DE DAGGING NOMINAL CLASSIFIER (DNC)

Dans un ensemble de données divers (comme dans un échantillonnage aléatoire stratifié), la valeur de $S(v_j)$ est minimisée et le GI de l'attribut correspondant est plus important. Ce phénomène permet une meilleure évaluation et valorisation d'un attribut dans un ensemble de données divers. Un échantillonnage aléatoire stratifié assure une présence proportionnelle de toutes les valeurs des attributs [24].

6 Étude expérimentale

Dans cette section, nous étudions la performance de la méthode DNC proposée pour voir dans quelles conditions elle peut améliorer les performances du classifieur faible CNC. En particulier, nous allons essayer de fournir des réponses aux questions suivantes : Combien faut-il générer de classifieurs par *Dagging* pour avoir des performances meilleures à celle du CNC individuel ? Est-ce que l'apprentissage parallèle est intéressant pour CNC ? Dans quels cas, le classifieur CNC peut être plus performant que les autres classifieurs ? Quel est la meilleure approche d'apprentissage adaptatif pour CNC : séquentielle ou parallèle ? Quelles sont les conditions dans lesquelles CNC se comporte mieux que les autres classifieurs ? Est-ce que la di-

Données	Instances	Attributs	Classes Nominiaux	Diversité des données
Car	1728	6	4	100%
Kr vs kp	3196	36	2	100%
Waveform	5000	40	3	100%
Optdigits	5620	64	10	100%
Nursery	12960	8	5	100%
Pendigits	10992	16	10	99.18%
Credit German	1000	20	2	98.59%
Japanese Vowels	5687	14	9	97.06%
Splice	3190	60	3	94.42%
Segment	2310	19	7	84.97%
Spambase	4601	57	2	78.26%
CMC	1473	9	3	64.96%
Solar flare	1066	12	6	34.30%
Page-blocks	5473	10	5	23.14%
Yeast	1484	8	10	22.34%

TABLE 2 – Caractéristique des échantillons de données utilisés.

versité des données d'apprentissage peut affecter les performances de Dagging de CNC ?

Pour répondre à ces questions, nous avons comparé DNC avec des méthodes de classification de la littérature comme Bayes Net, Id3, J48, Decision Stumps et CITREC [5] (basée sur l'Analyse des Concepts Formels) sur des échantillons de données extraits de "UCI Machine Learning Repository"⁴ [2].

La performance des classifieurs générés est évaluée en termes de taux d'erreurs. Pour calculer ces taux, nous rapportons la moyenne de 10 expérimentations. Chaque expérimentation s'est déroulée selon le principe de 10 validations-croisées.

Ces données ont été discrétisées avec un filtre de discrétisation implémenté sous la plateforme WEKA⁵. Ce filtre transforme les attributs numériques en attributs nominaux. Cette transformation de données est indispensable pour permettre aux CNC et DNC d'opérer à partir des données nominales. Pour chaque échantillon de données (voir Table 2), nous présentons respectivement : le nombre d'instances, le nombre des attributs nominaux (après discrétisation) et le nombre de classes. La dernière colonne présente les taux de diversité des données pour voir s'il affecte la performance du Dagging. La diversité des données est le ratio entre le nombre de différents vecteurs d'instances (les attributs) et le nombre total des vecteurs dans chaque échantillon de données [10].

6.1 Influence du nombre de classifieurs

Pour étudier les performances du Dagging en utilisant CNC, nous avons généré des ensembles de classifieurs en variant la valeur de T de 2 à 13. Nous avons rapporté les taux d'erreurs dans la Table 3. A partir de cette table, nous rapportons d'abord que la performance des ensembles de classifieurs impaires est meilleure que celle des ensembles

Données	Bayes Net	CNC	Id3	J48	Dec. Stumps	CITREC
Car	17,13%	6,31%	11,51%	18,69%	29,98%	23,79%
Kr-vs-kp	12,52%	33,95%	1,91%	3,1%	33,95%	47,78%
Waveform	19,32%	11,44%	19,9%	22,12%	48,5%	41,42%
Optdigits	7,92%	27,38%	21,12%	17,62%	56,44%	-
Nursery	9,82%	11,85%	4,88%	8,56%	33,75%	-
Pendigits	12,85%	9,66%	12,43%	16,48%	74,75%	-
German credit	23,6%	10,4%	27,9%	28,6%	30%	30%
Japanese vowels	48,51%	17,44%	44,59%	45,44%	59,13%	67,49%
Splice	5,11%	33,1%	8,31%	11,5%	27,21%	-
Segment	10,48%	6,54%	8,14%	9,35%	70,22%	-
Spambase	9,87%	9,37%	8,93%	10,3%	15,54%	-
Cmc	47,66%	29,6%	45,56%	46,23%	54,79%	55,33%
Solar-flare	1,88%	0,09%	0,47%	0,47%	0,47%	-
Page-blocks	6,47%	1,17%	3,51%	5,32%	6,8%	-
Yeast	41,18%	39,69%	42,46%	42,32%	59,3%	-
Moyenne	18,29%	16,53%	17,44%	19,07%	40,06%	44,3%

TABLE 4 – Performance du Dagging avec différentes méthodes de classification.

païres. Cela est due à la combinaison qui fonctionne mieux avec un nombre impair de classifieurs [12]. Avec plus de 8 classifieurs, Dagging produit des taux d'erreurs plus faibles. Ces ensembles ne sont pas corrélés avec la diversité des données d'apprentissage. Nous rapportons qu'avec 11 classifieurs, Dagging aura les meilleures performances. Pour tous les échantillons de données, la moyenne des erreurs obtenue avec 11 classifieurs (16.53%) est moins importante que celle obtenue avec 13 classifieurs (16.58%).

6.2 Influence du type de classifieur

Dans [20], les auteurs ont rapporté que l'apprentissage séquentiel est bénéfique pour des classifieurs de type J48 et Id3. Néanmoins CNC est parmi les pires classifieurs (de point de vue performance). Notre objectif est d'étudier le comportement de ces classifieurs dans le cas de l'apprentissage parallèle et de savoir quand est ce que CNC s'adapte mieux avec cette technique. Pour cela, des ensembles de 11 classifieurs de chaque type (ID3, Bayes Net, CNC, J48 et CITREC) ont été générés par Dagging. Les taux d'erreurs de ces ensembles sont rapportés dans la Table 4. Le signe '-' indique que la méthode ne peut pas traiter les données de l'échantillon, en raison de la consommation excessive des ressources mémoires. À partir de ces résultats, nous pouvons remarquer que CNC retient les meilleures performances pour 10 échantillons de données parmi 15. Id3 et Bayes Net sont plus performants que J48. Decision Stumps induit des taux d'erreurs élevés comparés aux autres classifieurs.

Pour les échantillons de données corrélés comme Pages blocks et Solar Flare, ayant les valeurs de diversité 23.14% et 34.3% respectivement, les taux d'erreurs des classifieurs nominaux (CNC) sont les plus faibles. Pour l'échantillon de données Yeast dont la diversité des données égale à 22.34%, les taux d'erreurs sont légèrement élevés. Ceci montre que la diversité des données n'influe pas sur la performance des ensembles de classifieurs nominaux.

À partir de ces expérimentations, nous rapportons que l'ap-

4. <http://www.ics.uci.edu/mllearn/MLRepository.html>

5. <http://www.cs.waikato.ac.nz/ml/Weka>

Données	2	3	4	5	6	7	8	9	10	11	12	13
Car	9,04%	7,41%	14,35%	8,9%	12,67%	10,3%	8,1%	6,3%	11,11%	6,31%	11,34%	7,69%
Kr-vs-kp	33,95%	33,95%	33,95%	33,95%	33,85%	33,85%	33,92%	33,92%	33,95%	33,95%	32,89%	32,88%
Waveform	13,18%	12,7%	12,58%	12,56%	12,74%	12,3%	12,14%	12,86%	13,32%	11,44%	12,02%	11,52%
Optdigits	28,35%	28,29%	28,26%	28,31%	28,35%	28,22%	33,08%	28,42%	28,86%	27,38%	28,26%	28,47%
Nursery	13,06%	14,35%	14,46%	12,89%	14,73%	14,41%	14,58%	14,21%	14,79%	11,85%	14,47%	14,77%
Pendigits	9,66%	9,68%	9,86%	11,33%	11,69%	11,71%	9,55%	11,57%	16,74%	9,66%	13,18%	11,83%
German credit	4,6%	6,4%	7,3%	7,4%	8,3%	8,8%	7,3%	9,7%	11%	10,4%	8,7%	9,2%
Japanese vowels	18,45%	16,72%	17,55%	16,6%	15,4%	15,53%	14,24%	13,84%	17,71%	17,44%	17,5%	12,76%
Splice	33,1%	33,1%	32,82%	32,13%	33,1%	31,32%	28,4%	31,57%	29,5%	33,1%	29,97%	28,53%
Segment	7,01%	6,97%	7,01%	6,36%	6,23%	6,32%	5,93%	5,5%	4,37%	6,54%	7,1%	5,58%
Spambase	7,35%	7,3%	7,28%	8,26%	8,46%	11,13%	8,22%	8,85%	7,28%	9,37%	6,52%	11,13%
Cmc	34,02%	33,34%	32,65%	29,53%	31,44%	30,62%	32,12%	30,01%	29,33%	29,6%	31,44%	31,77%
Solar-flare	0,37%	0,09%	0,09%	0,09%	0,19%	0,19%	0%	0,09%	0%	0,09%	0%	0%
Page-blocks	1,17%	1,17%	1,17%	1,17%	1,17%	1,3%	1,13%	1,17%	1,13%	1,17%	1,15%	1,13%
Yeast	40,84%	40,84%	40,84%	40,77%	40,97%	40,64%	40,44%	40,57%	40,37%	39,69%	40,84%	41,38%
Moyenne	16,94%	16,82%	17,34%	16,68%	17,29%	17,11%	16,61%	16,57%	17,30%	16,53%	17,03%	16,58%

TABLE 3 – Performances du Dagging pour CNC.

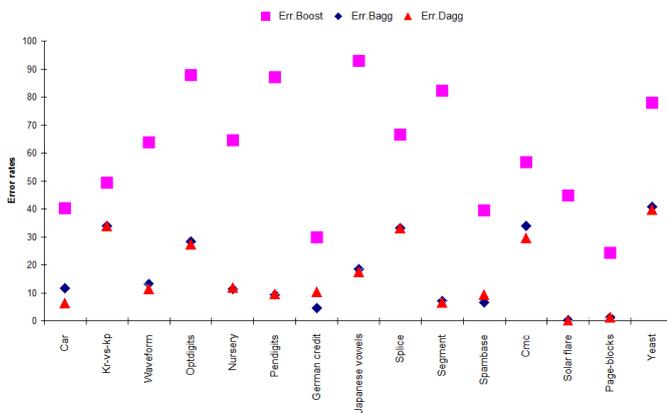


FIGURE 2 – Taux d’erreurs du Boosting, Bagging et Dagging utilisant CNC

prentissage parallèle est le plus intéressant pour notre classifieur nominal CNC. Les autres méthodes de classification peuvent être plutôt utilisées dans l’apprentissage séquentiel (comme indiqué dans [20]).

6.3 Comparaison des méthodes d’ensemble

D’après la littérature, la relation entre les classifieurs de type concept formel et les méthodes d’ensemble n’est pas claire : par exemple pour notre classifieur CNC, nous ne savons pas s’il est préférable d’utiliser *Boosting*, *Bagging* ou *Dagging* pour un problème de classification donné.

D’après [19], nous avons rapporté que CNC n’est pas assez performant dans le cadre de l’apprentissage séquentiel sur des échantillons de données de différentes tailles. Pour voir ce qui est meilleur pour CNC, l’apprentissage séquentiel ou parallèle des ensembles de 11 classifieurs nominaux ont été générés par *Boosting*, *Bagging* et *Dagging* sur 15 échantillons de données (Table 2). Figures 2 et 3 présentent les performances de ces 3 méthodes d’ensemble, en termes de taux d’erreurs et de temps d’apprentissage. Ces performances sont présentées verticalement alors les dif-

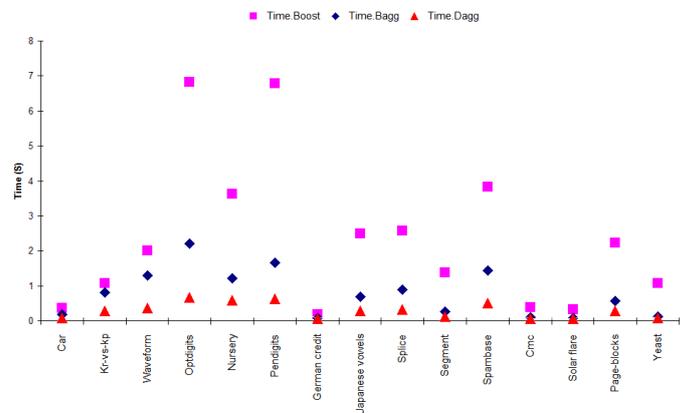


FIGURE 3 – Temps d’apprentissage du Boosting, Bagging et Dagging utilisant CNC

férents échantillons de données sont présentés horizontalement. Ces résultats montrent que *Dagging* est meilleure que *Boosting* puisqu’elle a généré des taux d’erreurs beaucoup plus faibles. Cependant, elle est proche de *Bagging*.

Tous les taux d’erreurs du *Boosting* sont plus élevés que ceux du *Dagging* (Figure 2). En outre, *Boosting* est 6 fois plus lent que *Dagging*. Nous rapportons aussi que *Dagging* est 2 fois plus rapide que *Bagging* (Figure 3).

D’après ces expérimentations, l’apprentissage des ensembles de 11 classifieurs nominaux par *Dagging* est plus intéressant que ceux par *Bagging*. D’après notre étude, l’apprentissage parallèle par *Dagging* des classifieurs basés sur l’Analyse des Concepts Formels est plus intéressant que l’apprentissage séquentiel.

7 Conclusion

Dans cet article, nous nous sommes intéressés par l’apprentissage parallèle des classifieurs de type concept nominal basé sur l’Analyse des Concepts Formels. Nous avons présenté un classifieur de concept nominal (CNC)

qui est meilleur que les classifieurs basés sur l'Analyse des Concepts Formels [20]. Nous avons proposé d'améliorer sa performance en utilisant les méthodes d'ensemble. D'une part, les travaux récents ont encouragé l'utilisation des méthodes d'ensemble pour les classifieurs linéaires. D'autre part, il n'existe aucune étude sur l'apprentissage parallèle pour les classifieurs basés sur les concepts formels. En particulier, nous avons présenté une nouvelle variante de *Dagging* pour générer et combiner des ensembles de classifieurs CNC qui semble être un classifieur faible.

Nous recommandons l'apprentissage parallèle par *Dagging* pour les classifieurs de type *concept formel* et l'apprentissage séquentiel pour les autres types de classifieurs. Dans l'apprentissage parallèle, peu de classifieurs sont suffisants pour obtenir une meilleure performance.

Des améliorations peuvent être apportées au niveau de la combinaison. DNC utilise le vote majoritaire pour la combinaison des classifieurs. Une variété des méthodes de vote existent déjà. Une étude de ces méthodes sera bénéfique pour améliorer la performance du classifieur CNC [22].

Au niveau de notre algorithme d'apprentissage CNC, d'autres mesures de pertinence peuvent être utilisées pour sélectionner le meilleur attribut puisque le concept de la comité des mesures a montré sa fiabilité dans des recherches récentes [9], [14]. Nous pouvons étudier la combinaison des mesures de pertinence pour adopter une comité de mesure à notre classifieur.

Références

- [1] S.B. Kotsiantis Anyfantis D., M. Karagiannopoulos and P.E. Pinelas. Local dagging of decision stumps for regression and classification problems. In *Proceedings of 15th IEEE Mediterranean Conference on Control and Automation*, pages 1–6, 2007.
- [2] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.
- [3] L. Breiman. Bagging predictors. *Journal of Machine Learning*, 24(2) :123–140, 1996.
- [4] C. Carpineto and G. Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Journal of Machine Learning*, 24(2) :95–122, 1996.
- [5] B. Douar, C. C. Latiri, and Y. Slimani. Approche hybride de classification supervisée à base de treillis de galois : application à la reconnaissance de visages. In *Actes des 8èmes Journées Franco-phones en Extraction et Gestion des Connaissances*, volume E-11 of *Revue des Nouvelles Technologies de l'Information*, pages 309–320. Cépaduès-Éditions, 2008.
- [6] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.
- [7] Huaiyu Fu, Huaiguo Fu, Patrick Njiwoua, and Engelbert Mephu Nguifo. A comparative study of fca-based supervised classification algorithms. In *Proceeding of Second International Conference on Formal Concept Analysis*, pages 313–320, 2004.
- [8] G. Stumme Ganter B. and R. Wille. *Formal Concept Analysis : Foundations and Applications*. Springer, 2005.
- [9] F. Guillet and H. J. Hamilton. *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*. Springer, 2007.
- [10] Mehdi Salkhordeh Haghghi, Abedin Vahedian, and Hadi Sadoghi Yazdi. Creating and measuring diversity in multiple classifier systems using support vector data description. *Applied Soft Computing*, 11(8) :4931–4942, 2011.
- [11] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial Intelligence*, pages 1137–1143. Morgan Kaufmann, 1995.
- [12] L. I. Kuncheva, C. J. Whitaker, and R. P. W. Duin. Limits on the majority vote accuracy in classifier fusion. *Journal of Pattern Analysis and Applications*, 6 :22–31, 2003.
- [13] Ludmila I. Kuncheva. *Combining Pattern Classifiers : Methods and Algorithms*. Wiley-Interscience, 2004.
- [14] P. Lenca, P. Meyer, B. Vaillant, and S. Lallich. On selecting interestingness measures for association rules : User oriented description and multiple criteria decision aid. *European Journal of Operational Research*, 184(2) :610–626, 2008.
- [15] M. Maddouri. Towards a machine learning approach based on incremental concept formation. *Journal of Intelligent Data Analysis*, 8(3) :267–280, 2004.
- [16] N. Meddouri and M. Maddouri. Boosting formal concepts to discover classification rules. In *Proceeding of the 22rd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, volume 5579 of *Lecture Notes in Computer Science*, pages 501–510. Springer, 2009.
- [17] N. Meddouri and M. Maddouri. Classification methods based on formal concept analysis. In *Proceedings of the 6th International Conference on Concept Lattices and Their Applications*, pages 9–16, 2009.
- [18] Nida Meddouri, Hela Khoufi, and Mondher Maddouri. Vers un critère d'arrêt de boosting basé sur la diversité des classifieurs. In *Actes du 19ème Congrès en INformatique des ORganisations et Systèmes d'Information et de Décision*, pages 111–126, 2011.
- [19] Nida Meddouri, Hela Khoufi, and Mondher Maddouri. Diversity analysis on boosting nominal concepts. In *PAKDD (1)*, pages 306–317, 2012.
- [20] Nida Meddouri and Mondher Maddouri. Adaptive learning of nominal concepts for supervised classification. In *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6276 of *Lecture Notes in Computer Science*, pages 121–130. Springer, 2010.
- [21] G. D. Oosthuizen. *The use of a lattice in knowledge processing*. PhD thesis, University of Strathclyde, Glasgow, Scotland, UK, 1988.
- [22] M. Oueslati, H. Zouari, M. Maddouri, and L. Heutte. Étude de la combinaison de classifieurs dans le cadre du boosting. In *Actes de la 6ième édition des ateliers de travail sur le Traitement et Analyse de l'Information : Méthodes et Applications*, 2005.
- [23] J. R. Quinlan. Induction of decision trees. *Journal of Machine Learning*, 1(1) :81–106, 1986.
- [24] G. Saporta. *Probabilités, analyses des données et statistiques*. Editions Technip, 2011.
- [25] Kotsiantis S.B. and D. Kanellopoulos. Combining bagging, boosting and dagging for classification problems. In *Proceedings of the 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, volume 4693 of *Lecture Notes in Computer Science*, pages 493–500. Springer, 2007.
- [26] M. Skurichina and R. P. W. Duin. Bagging for linear classifiers. *Journal of Pattern Recognition*, 31(7) :909 – 930, 1998.
- [27] Kai Ming Ting and Ian H. Witten. Stacking bagged and dagged models. In *In Proc. 14th International Conference on Machine Learning*, pages 367–375, 1997.
- [28] M. Visani, K. Bertet, and J.-M. Ogier. Navigala : an original symbol classifier based on navigation through a galois lattice. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(04) :449–473, 2011.
- [29] Manfred K. Warmuth, Karen A. Glocer, and Gunnar Rätsch. Boosting algorithms for maximizing the soft margin. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2007.
- [30] Z. Xie, W. Hsu, Z. Liu, and M. L. Lee. Concept lattice based composite classifiers for high predictability. *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3) :143–156, 2002.