Le forage distribué des données : une méthode simple, rapide et efficace

Mohamed Aounallah et Guy Mineau

Département d'informatique et de génie logiciel
Pavillon Adrien-Pouliot, Université Laval
G1K 7P4, Canada
{Mohamed.Aoun-Allah, Guy.Mineau}@ift.ulaval.ca,
http://w3.ift.ulaval.ca/∼moaoa
http://www.ift.ulaval.ca/Personnel/prof/mineau.htm

Résumé. Dans cet article nous nous attaquons au problème du forage de très grandes bases de données distribuées. Le résultat visé est un modèle qui soit et prédictif et descriptif, appelé méta-classificateur. Pour ce faire, nous proposons de miner à distance chaque base de données indépendamment. Puis, il s'agit de regrouper les modèles produits (appelés classificateurs de base), sachant que chaque forage produira un modèle prédictif et descriptif, représenté pour nos besoins par un ensemble de règles de classification. Afin de guider l'assemblage de l'ensemble final de règles, qui sera l'union des ensembles individuels de règles, un coefficient de confiance est attribué à chaque règle de chaque ensemble. Ce coefficient, calculé par des moyens statistiques, représente la confiance que nous pouvons avoir dans chaque règle en fonction de sa couverture et de son taux d'erreur face à sa capacité d'être appliquée correctement sur de nouvelles données. Nous démontrons dans cet article que, grâce à ce coefficient de confiance, l'agrégation pure et simple de tous les classificateurs de base pour obtenir un agrégat de règles produit un méta-classificateur rapide et efficace par rapport aux techniques existantes.

1 Introduction

Ce papier traite du problème de forage de plusieurs bases de données gigantesques et géographiquement distribuées dans le but de produire un ensemble de règles de classification qui expliquent les groupements de données observés. Le résultat de ce forage sera donc un méta-classificateur aussi bien prédictif que descriptif. En d'autres termes, nous visons à produire un modèle qui permet non seulement de prédire la classe de nouveaux objets, mais qui permet aussi d'expliquer les choix de ses prédictions. Nous croyons que ce genre de modèles, basés sur des règles de classification, devrait aussi être facile à comprendre par des humains, ce qui est également l'un de nos objectifs. Il faut dire toutefois que nous nous plaçons dans le contexte où il est impossible de rapatrier toutes ces bases dans un même site, et ce, soit à cause du temps de téléchargement, soit à cause de l'impossibilité de traiter la base ainsi agrégée.

Dans la littérature, les techniques de forage distribué de données à la fois prédictives et descriptives sont malheureusement peu nombreuses. La plupart d'entre elles tentent de produire

un méta-classificateur sous forme d'un ensemble de règles à couverture disjointe où un objet est couvert par une et une seule règle. Nous montrerons dans cet article que cette contrainte de couverture disjointe n'est pas nécessaire pour produire un méta-classificateur fiable. Ainsi, nous proposons une technique simple où un objet peut être couvert par plusieurs règles. La relaxation de cette contrainte de couverture disjointe nous permet de produire un classificateur final rapide, sans que le taux d'erreur de celui-ci n'en souffre.

Cet article procède comme suit. Dans la section 2, une vue d'ensemble des techniques d'agrégation de modèles les plus connues est présentée. Puis, dans la section 3, nous présentons notre solution au forage distribué des données (FDD) employant l'agrégation de modèles (FDD-AM) basée sur un coefficient de confiance. Dans la section 4, nous présentons nos résultats d'expérimentations qui démontrent la viabilité de notre méthode. La section 5 compare la complexité asymptotique de notre méthode à celles rencontrées dans la littérature. Nous présentons finalement une conclusion et nos travaux futurs.

2 Techniques existantes d'agrégation de modèles

Nous présentons dans ce papier uniquement les techniques qui ont été développées dans un but de forage distribué de données. Conséquemment, nous ignorons volontiers : le système « Ruler » (Fayyad et al., 1993) (Fayyad et al., 1996) qui a été construit dans le but de regrouper plusieurs arbres de décision construits sur un même ensemble de données dans un système centralisé, le système d'apprentissage distribué (Sikora et Shaw, 1996) développé dans un cadre de gestion des systèmes d'information afin de bâtir un système apprenant distribué (*Distributed Learning System, DLS*) et l'approche de fragmentation (Wüthrich, 1995) qui utilise des règles probabilistes. En outre, nous ignorons les techniques purement prédictives telles que *bagging* (Breiman, 1996), *boosting* (Schapire, 1990), *stacking* (Tsoumakas et Vlahavas, 2002), *arbiter* et *combiner* (Chan, 1996), (Prodromidis et al., 2000).

2.1 L'algorithme MIL

L'algorithme MIL (*Multiple Induction Learning*) a été initialement proposé par Williams (1990) afin de résoudre le conflit entre les règles conflictuelles (voir définition ci-dessous) dans des systèmes experts. Hall et al. (1998a,b) ont repris la technique de Williams pour agréger des arbres de décision bâtis en parallèle et préalablement transformés en règles. En outre, ils ont étendu la technique pour prendre en considération d'autres types de conflits. Le processus d'agrégation proposé par ces auteurs n'est autre qu'un regroupement des règles muni d'un processus de résolution des éventuels conflits. Il est à noter que cette résolution des conflits ne traite qu'un couple de règles conflictuelles à la fois. Deux règles se voient en situation de conflit quand leurs prémisses sont consistantes tandis qu'elles produisent deux classes différentes (Williams, 1990) (appelé conflit de type I), ou lorsque les conditions des prémisses se chevauchent partiellement (Hall et al., 1998a) (appelé conflit de type II) ou quand les règles ont le même nombre de prédicats avec des valeurs différentes associées aux conditions et classent les objets vers la même classe (Hall et al., 1998b) (appelé conflit de type III). La résolution de conflits consiste soit à spécialiser une ou les deux règles en conflit (conflits type I et II), soit à ajuster la valeur de la condition, c.-à-d., la borne de test, pour les conflits de type II et III et éventuellement fusionner les deux règles en conflit (conflit de type III). Dans certains cas

(conflits de type I et II), de nouvelles règles sont ajoutées en se basant sur les ensembles d'entraînement de celles-ci pour récupérer la couverture perdue par l'opération de spécialisation.

2.2 Le système DRL (« Distributed Rule Learner »)

La technique DRL (« Distributed Rule Learner ») (Provost et Hennessy, 1996) a été conçue et implantée en tirant avantage de la propriété du cloisonnement-invariant (Provost et Hennessy, 1994). DRL commence par partitionner les données d'entraînement E en nd sous-ensembles disjoints, assigne chacun (E_i) à une machine, et fournit l'infrastructure pour la communication entre les différents apprenants (nommé RL et roulant chacun sur une machine différente). Quand une règle répond au critère d'évaluation pour un sous-ensemble des données ($f'(r, E_i, nd) \geq c$; f' est une fonction d'évaluation d'une règle et c une constante), elle devient une candidate pour répondre au critère d'évaluation global ; la propriété de cloisonnement-invariant étendue garantit que chaque règle qui est satisfaisante sur l'ensemble des données sera acceptable au moins sur un sous-ensemble. Lorsqu'une copie locale du RL découvre une règle acceptable, elle envoie la règle aux autres machines pour mette à jour ses statistiques sur le reste des exemples. Si la règle répond au critère d'évaluation global ($f(r, E) \geq c$; f est la fonction d'évaluation principale et c une constante), elle est signalée comme règle satisfaisante. Dans le cas contraire, ses statistiques locales sont remplacées par les statistiques globales et la règle est rendue disponible pour la spécialiser encore plus. La propriété de cloisonnement-invariant garantit que chaque règle satisfaisante sur l'ensemble des données sera trouvée par l'un des RL.

2.3 Fusion d'ensembles de règles générés en parallèle

Le travail présenté par Hall et al. (1999) est un mélange des deux derniers travaux présentés ci-dessus, en d'autres termes les travaux de (Williams, 1990), (Hall et al., 1998b) et (Provost et Hennessy, 1996). Spécifiquement, à chaque règle créée lui est associée une mesure de sa « qualité » qui est basée sur la précision ainsi que le nombre et le type des exemples qu'elle couvre.

La technique proposée dans (Hall et al., 1999) est l'utilisation de ce que Provost et Hennessy (1996) proposent (voir §2.2), à une différence près où la suppression de la règle de l'espace des règles en considération ne se fait que lorsque la règle classe toutes les données des différentes bases et qu'il s'avère que sa mesure $f(r, E)$ est inférieure au seuil. Il est à noter que chaque règle ne « voyage » pas d'un site à un autre toute seule, mais bel et bien accompagnée des valeurs nécessaires pour calculer la mesure associée à chaque règle.

Toutefois, les auteurs de (Hall et al., 1999) démontrent que, dans le cas extrême, la propriété de cloisonnement-invariant risque de ne pas être satisfaite. Ainsi, ils suggèrent que la précision des règles agrégées peut être très différente de la précision des règles bâties sur l'ensemble d'entraînement entier. En outre, les auteurs soulignent qu'en cas de conflits entre règles, ces derniers peuvent être résolus, comme décrit par (Hall et al., 1998b) et (Williams, 1990).

Par ailleurs, Hall et al. (1999) traite un nouveau type de conflit entre règles. Il s'agit d'une règle ayant un intervalle (c.-à-d., deux conditions) chevauchant un intervalle d'une deuxième règle. Dans ce cas, une règle plus générale est créée en combinant les deux règles conflictuelles et en ajustant les bornes des intervalles.

2.4 Discussion

La technique MIL souffre de plusieurs défauts. Tout d'abord, le processus de résolution de conflit ne fait que spécialiser encore plus les règles en se basant sur les ensembles d'entraînement des règles de classification. Les règles générées peuvent exhiber un faible pouvoir de classification si elles sont appliquées à de nouveaux objets, et ce, surtout dans le cas de bases d'entraînement très bruitées. En plus, si les règles sont déjà très spécifiques à un ensemble d'entraînement, cette méthode est incapable de les généraliser puisqu'elle ne fait que regrouper puis spécialiser encore plus les règles en conflit. En outre, l'adaptation de la technique de Williams afin de traiter des bases distribuées implique une augmentation du volume de données échangées entre les différents sites. En effet, d'une part, chaque règle voyage accompagnée de l'index des objets couverts et, d'autre part, en cas de conflit, tous les objets couverts par une des deux règles en conflit sont rapatriés du site d'entraînement vers le site qui résout les conflits.

Le plus important inconvénient du système DRL est le temps d'exécution. En effet, lorsqu'une règle est jugée acceptable par un site donné, elle doit passer par tous les autres sites afin de mettre à jour ses variables statistiques en fonction de leurs données. En d'autres termes, toute règle acceptable sur un site doit classer toutes les données de tous les autres sites. Ainsi, la règle doit, d'une part, « voyager » à travers tous les sites, et d'autre part, classer les données de chaque site. Si une règle n'est pas jugée satisfaisante sur l'ensemble des données, celle-ci est spécialisée et le processus recommence si la nouvelle règle est jugée localement acceptable. Il est clair que ce processus risque d'être très gourmand en temps d'exécution.

Quant au système de fusion de règles générées en parallèle, ce système est identique au précédent à une différence près ; toute règle générée dans un site donné traverse tous les autres sites afin de mettre à jour ses variables statistiques. Ainsi, le nombre de règles voyageant entre les différents sites est plus important que le nombre de règles en transite dans le système DRL. Par conséquent, il est clair que cette technique est encore plus lente que la précédente.

3 La technique d'agrégation de modèles proposée

Afin de construire notre méta-classificateur, nous proposons une architecture basée sur les agents logiciels. À cette fin, deux types d'agents sont mis en œuvre : les *agents mineurs* qui minent chaque base de données répartie et un *agent collecteur* responsable de regrouper les informations produites par les agents mineurs.

3.1 Tâches d'un agent mineur

La tâche d'un agent mineur est décrite par la figure 1.

Il est à noter que le coefficient de confiance c_r d'une règle r est calculé en utilisant le théorème limite centrale. En effet, ce théorème stipule que la somme d'un grand nombre (≥ 30) de variables aléatoires indépendantes et identiquement distribuées suit une distribution qui peut être approximée par une loi Normale. Ainsi, comme nos classificateurs sont bâtis sur un large volume de données, le taux d'erreur $E_r(T)$ d'une règle r calculé sur un ensemble de test T, disjoint de l'ensemble d'entraînement D, peut être approximé par la loi Normale au vrai taux d'erreur E_r, qui est le taux d'erreur de r appliqué à toute la population, avec l'écart-type σ_{E_r}. À l'aide du taux d'erreur $E_r(T)$ et de l'écart-type σ_{E_r} associés à une règle r, nous pouvons

Pour un agent mineur Am_i travaillant sur la base de données DB_i, faire :

1. Appliquer sur DB_i un algorithme de classification générant un ensemble de règles à couvertures disjointes. L'ensemble produit est $R_i = \{r_{ik} \mid k \in [1..n_i]\}$ où n_i est le nombre de règles;
2. Calculer pour chaque r_{ik} le coefficient de confiance $c_{r_{ik}}$ (voir ci-bas);

FIG. 1 – *Algorithme détaillant les tâches d'un agent mineur.*

calculer l'intervalle de confiance dans lequel nous retrouvons le vrai taux d'erreur de r, E_r, dans $N\%$ des cas, comme suit : $E_r \in [E_r(T) - z_n \cdot \sigma_{E_r}, E_r(T) + z_n \cdot \sigma_{E_r}]$ où la constante z_n est choisie en fonction du degré de confiance $N\%$ désiré.

Le coefficient de confiance de chaque règle est déduit de l'intervalle de confiance de l'erreur. Nous l'avons défini comme étant : 1 *moins le pire taux d'erreur calculé dans* $N\%$ *des cas* : $(1 - E_r(T) - z_n\sigma_{E_r})$; en d'autres termes, 1 moins le taux d'erreur de la règle et moins la moitié de la largeur de l'intervalle de confiance de l'erreur, ainsi nous visons à prendre en compte le pire cas.

3.2 Tâches de l'agent collecteur

L'agent collecteur, quant à lui, a pour tâche de regrouper les informations produites par tous les agents mineurs. Sa tâche est détaillée par l'algorithme de la figure 2. Nous pouvons voir dans cet algorithme que l'agent collecteur passe globalement par deux phases. La première phase est la phase principale qui consiste à regrouper toutes les règles dans une même base de règles R. Cette base de règles est notre *méta-classificateur original*. La deuxième phase, optionnelle, représente une phase de raffinement par filtrage des règles. Il s'agit de supprimer de la base des règles celles qui ont un faible coefficient de confiance. En d'autres termes, il faut supprimer les règles qui, d'après la mesure de confiance, calculée statistiquement rappelons-le, n'auront vraisemblablement pas un bon pouvoir prédictif lorsque confrontées à des données nouvelles. L'ensemble de règles résultant de cette étape est le méta-classificateur R_t.

3.3 L'ensemble R comme méta-classificateur

L'ensemble R représente l'agrégation de tous les classificateurs de base. Cet ensemble de règles est utilisé comme modèle aussi bien prédictif que descriptif. D'un point de vue prédictif, la classe prédite d'un nouvel objet est la classe majoritaire prédite par les différentes règles qui le couvrent pondérée par leurs coefficients de confiance. Toutefois, en cas d'égalité des pondérations, nous proposons d'effectuer un vote à majorité simple. Ce qui revient, à peu près, à déterminer la classe votée par la majorité des classificateurs de base. Il est à noter que, contrairement à ce qui est identifié dans la littérature (voir §2), nous appelons règles en conflit seulement les règles qui couvrent un même objet différemment. Si plusieurs règles couvrent

Dans un site central, faire par l'agent collecteur Ac :

1. Étape principale, créer $R = \bigcup_{i=1...nd} R_i$ où nd est le nombre de bases distribuées ;
2. Étape optionnelle, filtrage des règles : Éliminer de R les règles ayant un coefficient de confiance inférieur à un seuil t : $R_t = \{r_{ik} \in R \mid c_{r_{ik}} \geq t\}$ (t est à déterminer empiriquement) ;

FIG. 2 – *Algorithme détaillant les tâches d'un agent collecteur.*

un même objet et prédisent la même classe, nous ne les considérons pas comme conflictuelles. Dans de rares cas, même le vote à majorité simple risque d'aboutir à une égalité. Le cas échéant, nous choisissons la classe majoritaire dans l'ensemble des bases d'entraînement.

Il est à signaler que tout objet peut être couvert par au plus nd règles –sachant que nd est le nombre de sites. Le nombre de règles n'est pas exactement égal à nd car la phase de détermination du coefficient de confiance risque dans certains cas d'échouer, et ce, à défaut d'une couverture, et par conséquent la règle en question est ignorée. Par ailleurs, en regroupant les ensembles R_i, une même règle peut apparaître dans plus d'un classificateur de base. Dans ce cas, une seule occurrence de la règle est retenue en lui attribuant un coefficient de confiance égal à la moyenne des coefficients de confiance de ses différentes occurrences.

D'un point de vue descriptif, les règles qui couvrent un objet expliquent sa classe même s'il y a eu égalité du vote à majorité simple ou pondéré. Comme le système est développé dans un but de forage de données, en d'autres termes, comme support à la prise de décision, les règles couvrant un objet sont proposées à l'utilisateur qui doit juger, de par son expertise, de leur pertinence. Le fait de présenter à un décideur plus qu'une règle afin d'expliquer la classe d'un objet a ses avantages puisque celui-ci aura une vue plus large et plus complète des « limites » de chaque classe. Nous rappelons en outre, qu'en apprentissage automatique, la limite qui définit la séparation entre différentes classes n'est généralement pas unique et par conséquent, plusieurs règles produisant une même classe peuvent représentées les « hyperplans » séparant les différentes classes, fournissant diverses vues sur ces données.

4 Expérimentation

Afin d'effectuer nos tests, nous avons utilisé dix jeux de données tirés de la banque de données de l'UCI (Blake et Merz, 1998) et dont la taille varie de 351 objets à 45222 objets. Il s'agit des bases : adult, chess end-game (King+Rook versus King+Pawn), Crx, house-votes-84, ionosphere, mushroom, pima-indians-diabetes, tic-tac-toe, Wisconsin Breast Cancer (BCW)(Mangasarian et Wolberg, 1990) and Wisconsin Diagnostic Breast Cancer (WDBC). La subdivision de ces bases afin de simuler des bases distribuées est bien détaillée dans (Aounallah et al., 2005).

Pour des fins de comparaison, nous utilisons l'algorithme C4.5 appliqué sur la totalité des données. Le résultat, l'ensemble de règles R', représente le cas idéal où toutes les données peuvent être regroupées dans un site central. Les résultats obtenus avec C4.5 sont seulement à titre de référence et nous supposons qu'en pratique il n'est pas possible de regrouper les données dans un même site. L'algorithme C4.5 est aussi utilisé pour construire les classificateurs de base.

4.1 Les taux d'erreur obtenus avec R et R_t

Nous commençons par regrouper les ensembles de règles R_i afin de créer le méta-classificateur original R. Le tableau 1 représente le taux d'erreur de R' pour chaque ensemble de test avec son intervalle de confiance à 95% ainsi que ceux de R. L'avant dernière colonne représente une comparaison entre les taux d'erreur des ensembles R et R' ; nous y trouvons :

- « Empire » (resp. « Améliore ») qui signifie que R est statistiquement à 95% du temps pire (resp. mieux) que R' du point de vue du taux d'erreur de classification. La dernière colonne indique la valeur absolue de cette différence.
- « $\approx$ » indiquant que R est statistiquement comparable à R'.

Ce tableau montre bien que dans 8 cas sur 10, le taux d'erreur de R est comparable à celui de R' et même dans les deux autres cas la différence n'est pas très importante. Toutefois, cet excellent résultat pourrait être la conséquence de bases distribuées très riches en informations. Afin de vérifier si tel est le cas, nous avons appauvri les bases de données BD_i en y introduisant du bruit, et ce, en inversant l'attribut de classe[1] pour 10%, 20%, 25% et 30% des objets.

Base	R'	Borne inf	Borne sup	R	R vs. C4.5	Val. de la diff.
Adult	14.7%	14.1%	15.3%	14.5%	$\approx$	
BCW	7.0%	3.2%	10.8%	4.7%	$\approx$	
Chess	0.9%	0.2%	1.6%	2.5%	Empire	1.6%
Crx	18.4%	12.5%	24.3%	20.2%	$\approx$	
Iono.	20.5%	12.1%	28.9%	20.5%	$\approx$	
Mush.	0.0%	0.0%	0.0%	0.3%	Empire	0.3%
Pima.	23.4%	17.4%	29.4%	27.6%	$\approx$	
Tic.	18.3%	13.4%	23.2%	21.3%	$\approx$	
Vote	3.0%	0.1%	5.9%	3.0%	$\approx$	
WDBC	6.3%	2.3%	10.3%	4.9%	$\approx$	

TAB. 1 – *Les taux d'erreur du méta-classificateur original R comparés à ceux de C4.5.*

En utilisant les bases appauvries (50 jeux de données différents : 10 bases de départ et 4 bases appauvries de chacune), nous constatons que les taux d'erreur de R sont toujours aussi comparables à ceux de R'. Par ailleurs, R arrive à produire de meilleurs taux d'erreur (statistiquement, avec un taux de confiance de 95%) que R', et ce, au fur et à mesure que le bruit augmente dans les bases.

Quant aux taux d'erreur de R_t, avec $t = 0.01$, seuil optimal sur les 50 bases tel qu'évalué empiriquement, ils sont sensiblement les mêmes (ou comparables statistiquement avec un taux de confiance de 95%) que ceux de R pour les 50 jeux de données.

[1] Il est à noter que toutes les bases utilisées ont deux classes.

4.2 Le nombre de règles formant le méta-classificateur

Le tableau 2 représente le nombre de règles formant le classificateur obtenu : R', R, R_t. Il est clair de ce tableau que nos méta-classificateurs R et R_t ont un nombre de règles raisonnable qui est même dans certains cas inférieur au nombre de règles de notre classificateur de référence. Ce résultat est très encourageant puisque nos méta-classificateurs ne sont ni plus difficiles ni plus faciles à interpréter que R'.

	Adult	BCW	Chess	Crx	Iono.	Mush.	Pima.	Tic.	Vote	Wdbc
C4.5	523	10	31	25	7	24	21	69	5	11
R	592	50	54	23	11	11	30	77	10	18
R_t	482	33	54	20	9	11	26	64	6	17

TAB. 2 – *Le nombre de règles formant chaque ensemble de règles*

5 Évaluation asymptotique

Dans cette section nous comparons la complexité asymptotique de nos méta-classificateurs R et R_t à ceux présentés dans la section 2. Pour ce faire, nous notons par n la taille maximale de l'ensemble d'entraînement dans un site, m le nombre d'attributs dans la base de données, k : le nombre maximum de valeurs par attribut, l : le nombre maximum de prédicats (littéraux) dans une règle, p : le nombre maximum de règles produites des n objets d'entraînement, d : le nombre de sites et n' : la taille maximale de l'ensemble de test dans un site quelconque.

5.1 Coût de la technique proposée

La technique proposée, rappelons-le, fonctionne sur deux phases : une phase distante accomplie par des agents mineurs et une phase centralisée achevée par l'agent collecteur.

5.1.1 Coût des tâches de l'agent mineur

Les tâches d'un agent mineur sont détaillées dans la figure 1. Globalement, il s'agit de bâtir le classificateur de base (tâche 1) et de calculer le coefficient de confiance de chaque règle (tâche 2).

Le coût de la tâche 1 est le coût de l'application de l'algorithme C4.5. Il est bien connu que ce coût est de l'ordre de $O(m^2n)$.

Le coût de la tâche 2 se résume au calcul de la couverture de chaque règle. Ce coût est le suivant :

- Le nombre de tests à faire afin de savoir si une règle couvre un objet est l (le nombre de prédicats dans une règle).
- Coût de déterminer la couverture de toutes les règles (au nombre de p) sur un site ayant n' objets dans l'ensemble de test est : $n' \times l \times p$.

Donc, le coût total des tâches d'un agent mineur est donc $O(m^2n + n'lp)$.

5.1.2 Coût des tâches de l'agent collecteur

Les tâches de l'agent collecteur sont détaillées dans la figure 2. Globalement, il s'agit de regrouper toutes les règles issues des différents agents mineurs dans un même ensemble R (tâche 1) et d'en extraire celles qui ont un coefficient de confiance inférieur à un certain seuil afin d'avoir l'ensemble R_t (tâche 2).

Le coût de la tâche 1 peut être considéré comme négligeable. Le coût de la tâche 2 est égal au nombre de règles dans R. En considérant que le nombre de règles issues d'un seul site est p et le nombre de sites est d, le coût de cette tâche est dp.

Ainsi, le coût total de notre méta-classificateur R_t est $O(m^2n + n'lp + dp)$.

Puisque le nombre de sites d est constant, le terme dp peut être remplacé par p et celui-ci peut être négligé devant $n'lp$ et par conséquent le coût de R_t est au pire de l'ordre de $O(m^2n + n'lp)$ qui n'est autre que le coût de l'agent mineur. Ainsi, le temps de regroupement et de filtrage des règles ne présente asymptotiquement aucun surcoût par rapport au temps nécessaire pour produire (en parallèle) les classificateurs de base.

5.2 Coût des techniques existantes

5.2.1 L'algorithme MIL

Afin de résoudre les conflits (de type 1 (Williams, 1990)), l'algorithme MIL a besoin de rapatrier sur un même site tous les objets couverts par les règles en conflit. Dans le pire cas, toutes les règles issues d'un site B provoquent des conflits avec les règles du site A. Afin de résoudre ces conflits, il faut récupérer tous les objets couverts par les règles issues du site B. En d'autres termes, il faut récupérer l'ensemble d'entraînement du site B. Ceci risque d'être très lent, voire même irréalisable, vu les hypothèses que l'on s'est fixées au départ, à savoir que nous nous plaçons dans le contexte où il est impossible de transférer toute une base de données d'un site à un autre.

Ainsi, comme la quantité de données qui transite d'un site à un autre n'est pas bornée, cet algorithme est dans le pire cas non comparable aux autres algorithmes, car il viole l'une des hypothèses de l'apprentissage distribué. Il sera par conséquent ignoré durant notre comparaison.

5.2.2 Le système DRL

Pour simplifier la comparaison, nous supposons que la complexité de l'algorithme utilisé pour construire l'ensemble de règles (RL) est la même que celle de l'algorithme C4.5 que nous utilisons dans notre technique (voir ci-dessus), malgré que, d'après (Hall et al., 1999), C4.5 est plus rapide que RL.

Cette technique se base sur une fonction d'évaluation de chaque règle. Si cette fonction est évaluée sur un ensemble indépendant de l'ensemble d'entraînement, son coût est le même que celui du calcul de notre coefficient de confiance (voir ci-dessus).

Lorsqu'une règle satisfait au critère d'évaluation local, elle est envoyée à tous les autres sites afin de mettre à jour ses statistiques en fonction de leurs données. Ainsi, cette règle, ayant l prédicats, doit classer tous les objets de tous les autres sites, au nombre de dn. Le coût associé à cette opération pour une règle est $O(ldn) = O(ln)$.

Dans le pire cas, toutes les règles peuvent satisfaire au critère d'évaluation local. Ainsi, le coût de mettre à jour les statistiques des règles d'un site donné, au nombre de p, est $O(lnp)$. Dans le pire cas, chaque site devrait classer les règles des d autres sites. Ainsi, ce coût est aussi de l'ordre de $O(dlnp) = O(lnp)$.

Si une règle ne satisfait pas le critère d'évaluation global, elle est renvoyée à son site de départ pour qu'elle soit spécialisée encore plus. Nous notons par α le coût de cette opération. Puis, si la nouvelle règle satisfait toujours le critère d'évaluation local, le processus est réitéré. On aura par conséquent, un autre coût de l'ordre de $O(ln)$ (c'est le coût d'une règle classant les données de tous les autres sites).

En conclusion, le coût global de cette technique est de l'ordre de $O(m^2n + n + lnp + \alpha + ln) = O(m^2n + lnp + \alpha)$.

5.2.3 Fusion d'ensembles de règles générées en parallèle

L'étude de complexité au pire cas de cette technique est sensiblement la même que le système DRL puisqu'il s'agit exactement de la même technique augmentée par un processus de résolution de conflit selon l'algorithme MIL. Par conséquent, globalement, la complexité de cette technique est pire ou égale à la complexité de la technique précédente.

5.2.4 Comparaison

La complexité de notre technique lorsque la validation est réalisée en considérant les échantillons comme ensemble de test est de l'ordre de $O(m^2n + ln'p)$. La complexité du système DRL ainsi que la technique de fusion de règles en parallèle est de l'ordre de $O(m^2n+lnp+\alpha)$. Ainsi, les trois techniques ont sensiblement la même complexité asymptotique à un terme près qui est dans notre technique fonction de la taille de l'ensemble de test dans un site, et dans le système DRL, fonction de la taille de l'ensemble d'entraînement. Comme la taille de l'ensemble d'entraînement est généralement plus importante que la taille de l'ensemble de test, notre technique est dans ce cas asymptotiquement plus rapide que le système DRL.

6 Conclusion

L'objectif de ce papier est de faire une comparaison entre les techniques existantes d'agrégation de modèles dans un but de forage distribué de données (FDD), d'une part, et une version simplifiée de notre technique de FDD (Aounallah et Mineau, 2004) (Aounallah et al., 2004, 2005) d'autre part. Pour ce faire, nous avons présenté un survol des techniques d'agrégation de modèles existantes les plus comparables à la nôtre ainsi qu'une description de la version simplifiée de notre technique de FDD.

Les expériences menées ont démontré que notre technique performe d'un point de vue prédiction aussi bien, ou même mieux, qu'un classificateur bâti sur la totalité des données, utilisé comme point de référence. Par ailleurs, nos méta-classificateurs sont toujours de tailles comparables au classificateur centralisé de référence.

Une étude asymptotique démontre, en outre, que nos techniques sont asymptotiquement comparables ou plus rapides que les techniques existantes de FDD.

En conclusion, nous avons démontré qu'un méta-classificateur bâti par la simple agrégation des classificateurs de bases, formés par des ensembles de règles, auxquelles est attribué un coefficient de confiance, démontre un bon pouvoir de prédiction, est de taille raisonnable, est aussi rapide ou plus rapide que les techniques d'agrégation de modèles existantes et il est, d'après (Aounallah et Mineau, 2004) (Aounallah et al., 2005) plus rapide que les techniques d'échantillonnage. Il apparaît que notre méta-classificateur peut représenter une bonne solution pour le forage des bases de données distribuées, très compétitif aux techniques existantes. De surcroît, l'architecture multi-agents utilisé sous-tend la parallélisation de la technique et permettrait grâce à la hiérarchisation des agents collecteurs encore plus d'adaptabilité à de très grandes bases de données. Des applications qui se servent de base de données transactionnelles, tel le e-commerce, pourraient alors bénéficier du FDD. Des expérimentations sur le terrain sont à venir bien que les résultats préliminaires tels que présentés dans cet article sont encourageants. La technique proposée, du moins pour certaines expérimentations et selon l'étude de complexité faite à la section 5, propose une technique de FDD beaucoup plus simple que celles proposées à la section 2.

Références

Aounallah, M. et G. Mineau (2004). Rule confidence produced from disjoint databases : a statistically sound way to regroup rules sets. In *IADIS international conference, Applied Computing 2004*, Lisbon, Portugal, pp. II–27 – II31.

Aounallah, M., S. Quirion, et G. Mineau (2004). Distributed Data Mining vs Sampling Techniques : a Comparison. In *Advances in Artificial Intelligence, 17th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2004*, Number 3060 in Lecture Notes in Artificial Intelligence, London, Ontario, Canada, pp. 454–460. Springer-Verlag.

Aounallah, M., S. Quirion, et G. Mineau (2005). Forage distribué des données : une comparaison entre l'agrégation d'échantillons et l'agrégation de règles. *Revue des Nouvelles Technologies de l'Information, extraction et gestion des connaissances 1*(E-3), 43–54.

Blake, C. et C. Merz (1998). UCI repository of machine learning databases. http ://www.ics.uci.edu/∼mlearn/MLRepository.html.

Breiman, L. (1996). Bagging predictors. *Machine Learning*.

Chan, P. K.-W. (1996). *An extensible meta-learning approach for scalable and accurate inductive learning*. Ph. D. thesis, Columbia University.

Fayyad, U., N. Weir, et S. Djorgovski (1993). Skicat : A machine learning system for automated cataloging of large scale sky surveys. In *Machine Learning : Proceedings of the Tenth International Conference*, San Mateo, CA, pp. 112–119. Morgan Kaufmann.

Fayyad, U. M., S. G. Djorgovski, et N. Weir (1996). *Advances in Knowledge Discovery and Data Mining*, Chapter Automating the analysis and cataloging of sky surveys, pp. 471–493. Menlo Park, California : AAAI Press/The MIT Press.

Hall, O. L., N. Chawla, et W. K. Bowyer (1998a). Combining Decision Trees Learned in Parallel. In *Working notes of KDD*.

Hall, O. L., N. Chawla, et W. K. Bowyer (1998b). Decision tree learning on very large data sets. In *IEEE International Conference on Systems, Man, and Cybernetics, 1998*, Volume 3, pp. 2579–2584.

Hall, O. L., N. Chawla, et W. K. Bowyer (1999). Learning rules from distributed data. In *Workshop on Large-Scale Parallel KDD Systems (KDD99). Also in RPI, CS Dep. Tech. Report 99-8*, pp. 77–83.

Mangasarian, O. L. et W. H. Wolberg (1990). Cancer diagnosis via linear programming. *SIAM News 23*(5), 1–18.

Prodromidis, A. L., P. K. Chan, et S. J. Stolfo (2000). Meta-learning in distributed data mining systems : Issues and approaches. In H. Kargupta et P. Chan (Eds.), *Advances in Distributed and Parallel Knowledge Discovery*, pp. 81–113. Menlo Park, CA / Cambridge, MA : AAAI Press / MIT Press. chap. 3 part II.

Provost, F. J. et D. N. Hennessy (1994). Distributed machine learning : Scaling up with coarse-grained parallelism. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pp. 340–347.

Provost, F. J. et D. N. Hennessy (1996). Scaling up : Distributed machine learning with cooperation. In *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 74–79.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*(2), 197–227.

Sikora, R. et M. Shaw (1996). A Computational Study of Distributed Rule Learning. *Information Systems Research 7*(2), 189–197.

Tsoumakas, G. et I. Vlahavas (2002). Distributed Data Mining of Large Classifier Eensembles. In I. Vlahavas et C. Spyropoulos (Eds.), *Proceedings Companion Volume of the Second Hellenic Conference on Artificial Intelligence*, Thessaloniki, Greece, pp. 249–256.

Williams, G. J. (1990). *Inducing and Combining Decision Structures for Expert Systems*. Ph. D. thesis, The Australian National University.

Wüthrich, B. (1995). Probabilistic knowledge bases. *IEEE Transactions on Knowledge and Data Engineering 7*(5), 691–698.

Summary

This paper deals with the problem of mining very large distributed data bases, where a model that is both predictive and descriptive, called meta-classifier, can be produced. To do this, we propose to mine independently each database remotely. Then we have to gather the produced models (called base classifiers), knowing that each mining process will produce a predictive and descriptive model, represented for our needs by a set of classification rules. In order to guide the aggregation of the final rule set, which is the union of the individual rule sets, a confidence coefficient is assigned to each rule of each set. This coefficient, computed by statistical means, represents the confidence which we can have in each rule according to its cover and its error rate. We prove in this paper that, thanks to this confidence coefficient, the aggregated rule set which is the simple aggregation of base classifiers, represents a fast and reliable meta-classifier compared to existing techniques.