Forage distribué des données : une comparaison entre l'agrégation d'échantillons et l'agrégation de règles

M. Aounallah*, S. Quirion*** et G. Mineau**

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

Résumé. Pour nous attaquer au problème du forage de très grandes bases de données distribuées, nous proposons d'étudier deux approches. La première est de télécharger seulement un échantillon de chaque base de données puis d'y effectuer le forage. La deuxième approche est de miner à distance chaque base de données indépendamment, puis de télécharger les modèles résultants, sous forme de règles de classification, dans un site central où l'agrégation de ces derniers est réalisée. Dans cet article, nous présentons une vue d'ensemble des techniques d'échantillonnage les plus communes. Nous présentons ensuite cette nouvelle technique de forage distribué des données où la mécanique d'agrégation est basée sur un coefficient de confiance attribué à chaque règle et sur de très petits échantillons de chaque base de données. Le coefficient de confiance d'une règle est calculé par des moyens statistiques en utilisant le théorème limite centrale. En conclusion, nous présentons une comparaison entre les meilleures techniques d'échantillonnage que nous avons trouvées dans la littérature, et notre approche de forage distribué des données (FDD) basée sur l'agrégation de modèles.

1 Introduction

Ce papier traite du problème de forage de plusieurs bases de données gigantesques et géographiquement distribuées, en présentant et en comparant deux techniques de forage de données. La première technique que nous avons examinée utilise un échantillon de taille raisonnable de chaque base de données, auxquels, une fois agrégés, nous appliquons une technique de forage de données. Cette technique relève de l'agrégation de données. Dans cette perspective, nous avons étudié les techniques d'échantillonnage existantes. Une description de ces dernières ainsi qu'une comparaison empirique sont présentées plus loin dans cet article.

La deuxième technique de forage de données, que nous introduisons (basée sur l'agrégation de modèles), se propose d'appliquer individuellement sur chaque base de

données une technique de forage de données. Les modèles résultant de ces techniques sont alors recueillis et un certain modèle agrégé est produit par une technique décrite dans ce qui suit. Dans ce travail, les modèles, qu'ils soient produits individuellement sur chaque sous-ensemble des données, soit le produit de la technique d'agrégation que nous proposons, sont représentés sous la forme d'un ensemble de règles de classification. Comme il sera expliqué dans le papier, la technique d'agrégation que nous proposons se base, d'une part, sur un *coefficient de confiance* associé à chaque règle et qui est calculé en utilisant le théorème limite central et, d'autre part, sur de très petits échantillons de chaque base de données. Ces échantillons sont employés afin de valider le coefficient statistiquement calculé.

Cet article procède comme suit. Dans la section 2, une vue d'ensemble des techniques d'échantillonnage les plus communes est présentée. Puis, dans la section 3, nous présentons notre solution au forage distribué des données (FDD) employant l'agrégation de modèles (FDD-AM). Dans la section 4, nous présentons nos résultats d'expérimentations qui nous aident à comparer ces 2 approches. Nous présentons finalement une conclusion et nos travaux futurs.

2 Échantillonnage

L'échantillonnage consiste en la création d'un échantillon représentatif d'une large base de données sous l'hypothèse qu'un classificateur entraîné sur cet échantillon n'aura pas de résultats significativement pires qu'un classificateur entraîné sur toute la base de données. Dans notre contexte, l'échantillonnage est appliqué sur chaque base répartie, générant des échantillons distincts dans chaque site. Ces derniers sont regroupés afin d'entraîner un classificateur. La littérature du forage de données est riche de plusieurs algorithmes d'échantillonnage (John et Langley 1996) (Provost et al. 1999) (Lewis et Gale 1994). En considérant comment l'échantillon est formé, tous ces algorithmes peuvent être regroupés sous trois familles : l'échantillonnage statique, dynamique et actif.

2.1 Échantillonnage statique

Il s'agit d'un échantillonnage qui est effectué en ayant uniquement les informations fournies par la base de données. Il s'agit principalement d'échantillonner aléatoirement selon certains estimateurs de distribution de données (moyenne, écart-type, etc.). Ces algorithmes d'échantillonnage ne font appel à aucun autre algorithme (tel qu'un algorithme de classification automatique, par exemple) afin de produire l'échantillon. Comme présenté dans (John et Langley 1996), pour une base de données D, un échantillon initial de taille n_0 et une suite d'incréments Δn_i, il faut tout d'abord créer un ensemble initial S de taille n_0 d'éléments aléatoires de D ensuite, tant que la distribution des champs de S diffère significativement de celle de D, ajouter à S Δn_i éléments aléatoires à partir de $D \setminus S$.

2.2 Échantillonnage dynamique

L'échantillonnage dynamique diffère de l'échantillonnage statique uniquement dans le processus de validation de l'échantillon. En effet, à chaque itération de l'échantillonnage dynamique un classificateur est bâti à partir de l'échantillon et celui-ci est évalué. Si le classificateur ainsi bâti n'aboutit pas à une précision de classification satisfaisante (c'est-à-dire, la précision n'a pas encore convergé vers une précision satisfaisante), l'algorithme d'échantillonnage réitère encore une fois. Il existe trois techniques permettant la détection de la convergence : la détection locale (DL) (s'arrêter quand $précision(n_i) \leq précision(n_{i-1})$) (John et Langley 1996), l'estimation de la courbe d'apprentissage (ECA) (John et Langley 1996) et la régression linéaire avec échantillonnage local (RLEL) (Provost et al. 1999).

2.3 Échantillonnage actif

Il diffère de l'échantillonnage dynamique par la façon dont l'algorithme sélectionne les éléments à chaque itération. Dans la littérature, l'échantillonnage actif est utilisé dans le contexte où les éléments de la base de données ne sont pas lus d'un seul coup par le système d'apprentissage. En effet, ce dernier a à choisir parmi les éléments non classés et demande à un expert ou à un autre programme de les classer. Le but de l'échantillonnage actif est de minimiser le nombre d'éléments nécessaires afin d'apprendre le concept correctement. Ceci est réalisé en choisissant les éléments produisant le plus grand gain en informations (matérialisé par un score d'efficacité), contrairement à l'échantillonnage dynamique qui choisit ses éléments aléatoirement. En général, ce score d'efficacité peut être calculé soit par un classificateur probabiliste ou par un comité de classificateurs. Les différentes méthodes d'échantillonnage actif peuvent être résumées par l'algorithme de la figure 1.

1. i ← 0
2. S ← {n_0 éléments aléatoires de D}
3. Générer {C}, un ensemble de classificateurs à partir de S
4. Tant que {C} n'a pas encore convergé
   (a) i ← i + 1
   (b) Pour tout $x_j \in$ D \ S
       Calculer SE_j, le score d'efficacité, avec {C}
   (c) S ← S ∪ {Δn_i éléments choisis de D \ S en se basant sur SE}
   (d) Générer {C}, un ensemble de classificateurs, à partir de S

FIG. 1 – Algorithme de l'échantillonnage actif.

Généralement, les Δn_i éléments ajoutés à S à chaque itération sont les éléments qui ont les plus grandes valeurs de SE (« *efficiency score* »). Comme cette tâche peut être sensible aux données bruitées, une alternative intéressante est d'utiliser les valeurs de SE comme poids dans une sélection aléatoire, comme c'est proposé pour « *uncertainty*

sampling » (Saar-Tsechansky et Provost 2001). Finalement, pour des fins de gain d'efficacité, nous pouvons remplacer l'ensemble de classificateurs {C} par un classificateur unique. Dans ce cas, l'algorithme d'échantillonnage exige à ce que ce classificateur soit un classificateur probabiliste, ce qui restreint énormément le choix. Cependant, (Lewis et Gale 1994) propose une approche hétérogène où, pour une légère perte en performance, l'échantillon est bâti en utilisant un simple classificateur probabiliste, tel qu'un classificateur bayésien naïf, permettant l'ajout et la suppression d'éléments en ligne (durant l'entraînement). Ensuite, un second algorithme de classification est utilisé afin de produire le classificateur final en se basant sur l'échantillon produit.

3 Le forage distribué des données utilisant l'agrégation de modèles

Afin de construire notre modèle agrégé, désormais appelé 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. La tâche de ces deux types d'agent est détaillée dans (Aounallah et Mineau 2004) ; ci-dessous nous donnons une brève description.

3.1 Tâches d'un agent mineur

La tâche d'un agent mineur est décrite par la figure 2.

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
   $R_i = \{r_{ik} \mid k \in [1..n_i]\}$ où n_i est le nombre de règles ;
2. Calculer pour chaque r_{ik} le coefficient de confiance $c_{r_{ik}}$ (voir ci-bas) ;
   où z_n est une constante qui dépend d'un degré de confiance demandé N, et σ_{E_r} est l'écart type de l'erreur E_r.
3. Extraire un échantillon aléatoire S_i à partir de DB_i.

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

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 calculer l'intervalle de confiance dans lequel nous retrouvons le vrai taux d'erreur de r, E_r, dans $N\%$ des cas, et ce, 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. Il est choisi à être 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 couvrir le pire cas.

3.2 Tâches de l'agent collecteur

La tâche de l'agent collecteur est décrite par la figure 3. Globalement, cet agent a pour tâche, d'une part, de filtrer les règles qui statistiquement vont probablement avoir un bon pouvoir prédictif, et ce, en se basant sur le coefficient de confiance des règles et, d'autre part, de valider la mesure statistique des règles gardées après la phase de filtrage en les confrontant aux échantillons récoltés par les agents mineurs.

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

1. Création de R et S comme suit :
   $R = \bigcup_i R_i$
   $S = \bigcup_i S_i$;
2. Filtrage des règles : Éliminer de R les règles ayant un coefficient de confiance inférieur à un seuil t :
   $R_t = \{r \in R \mid c_r \geq t\}$ (t est à déterminer empiriquement) ;
3. Validation des règles :
   (a) Créer une relation binaire $\mathcal{I}$ définie sur $R_t \times S$ où dans chaque intersection (r, s) nous écrivons 0 si r ne couvre pas s, sinon nous écrivons 1 si r couvre correctement s, autrement nous écrivons -1 ;
   $\mathcal{I} = \{< r, s, f(r,s) > \mid \ r \in R_t, s \in S, f(r,s) \in \{0, 1, -1\}\}$
   (b) Pour toute règle $r \in R_t$, calculer un taux d'erreur $E_r^{\mathcal{I}}(S)$ en utilisant S comme ensemble de test, c'est-à-dire, le nombre de -1 dans chaque rangée de la relation $\mathcal{I}$ divisé par le nombre de valeurs non nulles dans la même rangée ;
   (c) Construire l'ensemble de règles $R_t^{\mathcal{I}}$ (formant le méta-classificateur) en utilisant un seuil $t_{\mathcal{I}}$ comme suit :
   $R_t^{\mathcal{I}} = \left\{r \in R_t \mid E_r^{\mathcal{I}}(S) \leq t_{\mathcal{I}}\right\}$ ($t_{\mathcal{I}}$ est à déterminer empiriquement).

FIG. 3 – Algorithme détaillant les tâches d'un agent collecteur.

4 Expérimentation

Comme il a été proposé dans l'introduction, ce papier présente une comparaison entre les techniques d'échantillonnage et notre méta-classificateur. Pour ce faire, nous proposons d'effectuer les tests suivants :

1. Test 1 : Comparer les différentes techniques d'échantillonnage entre elles sur la base de la précision de prédiction afin d'en déterminer les meilleures. Seules ces dernières seront utilisées pour la comparaison avec le méta-classificateur.
2. Test 2 : Effectuer des tests du méta-classificateur en utilisant plusieurs valeurs de seuils (t et $t_{\mathcal{I}}$). Les valeurs optimales de ces seuils sont utilisées lors de la comparaison avec les techniques d'échantillonnage.
3. Test 3 : Faire des comparaisons entre les meilleures techniques d'échantillonnage et le méta-classificateur, et ce, sur la base de la précision de prédiction, sur la base de la taille requise des échantillons et sur la base du temps d'exécution.

Afin d'effectuer à nos tests, nous avons utilisé neuf 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), 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).

Afin de tester les méthodes d'échantillonnage ainsi que leur impact sur la précision d'apprentissage (i.e., test 1), nous avons subdivisé chaque base de données en un ensemble d'entraînement (/2/3) et un ensemble de test (1/3) (Pour un exemple, voir la figure 4, Première subdivision). Afin de simuler un environnement distribué (i.e., tests 2 et 3), nous avons, d'abord, subdivisé chaque base de données en deux sous-bases ayant les proportions de 1/4 et 3/4 (Fig. 4, deuxième subdivision). Le premier sous-ensemble est utilisé comme ensemble de test pour le méta-classificateur (le modèle agrégé) ou pour le classificateur bâti à partir des échantillons regroupés. Le deuxième sous-ensemble est aléatoirement subdivisé en deux, trois ou quatre sous ensembles (soit le nombre de base de données distribuées) de tailles aléatoires et qui sont à leur tour subdivisés en deux sous-ensembles ayant les proportions de 2/3 (le fichier .data de la figure 4 de chaque DB_i) et 1/3 (le fichier .test associé) respectivement pour l'ensemble d'entraînement et de test. Pour le méta-classificateur, un ensemble d'échantillons aléatoires (le fichier .sple associé) est extrait de chaque ensemble d'entraînement avec une taille de 10% la taille de la base de données et un maximum de 50 objets[1].

4.1 Test 1, comparaison des techniques d'échantillonnage

Afin de déterminer les techniques qui performeront le mieux sur nos jeux de données, nous avons comparé les deux techniques d'échantillonnage dynamique et actif en utilisant pour chacun les trois méthodes de détection de convergence cités ci-haut (*DL*, *ECA* et *RLEL*). Ces six méthodes ont été testées avec une suite d'incréments

[1] La taille maximale est nécessaire afin de borner la technique de méta-classification à des ensembles d'échantillons très petits, ce qui est en accord avec nous contraintes de base.

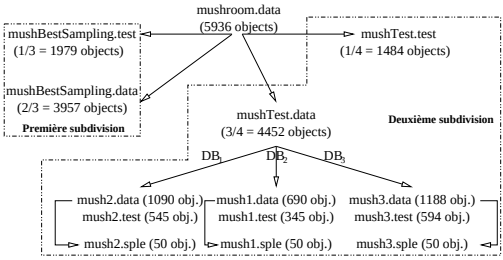

FIG. 4 – Subdivision de la base Mushroom en ensembles d'entraînement, de test et d'échantillons.

arithmétique (*Arith.*) et géométrique (*Géo.*). Nous avons aussi comparé ces méthodes, d'une part, à un échantillonnage aléatoire avec une suite d'incréments arithmétique et géométrique et, d'autre part, à un échantillon de 50 éléments tirés aléatoirement et par un « weighted uncertainty sampling », totalisant ainsi 16 méthodes d'échantillonnage.

	Meilleure technique	Deuxième	Troisième
Adult	Aléatoire - Géo. 82.80%	Dynamique - ECA - Géo. 81.90%	Aléatoire - Arith. 81.58%
BCW	Actif - ECA - Géo. 93.29%	Actif - RLEL - Géo. 92.35%	Dynamique - ECA - Géo. 90.53%
Chess	Actif - ECA - Géo. 93.93%	Dynamique - ECA - Géo. 93.40%	Actif - RLEL - Arith. 92.12%
Iono.	Dynamique - ECA - Géo. 80.97%	Actif - ECA - Géo. 80.80%	Actif - ECA - Arith. 80.23%
Mush.	Actif - ECA - Géo. 99.76%	Actif - ECA - Arith. 99.32%	Actif - RLEL - Géo. 98.82%
Pima.	Aléatoire - Arith./Aléatoire Géo. 75.52%	Dynamique - ECA - Géo. 73.07%	Actif - ECA - Géo. 70.83%
Tic.	Dynamique - RLEL - Géo. 75.40%	Dynamique - ECA - Géo. 75.27%	Actif - RLEL - Géo. 73.85%
Vote	Dynamique - RLEL - Arith. 96.48%	Dynamique - ECA - Géo. 96.22%	Dynamique - ECA - Arith. 95.81%
WDBC	Aléatoire - Arith. 93.80%	Dynamique - ECA - Géo. 92.64%	Actif - ECA - Géo. 92.43%

TAB. 1 – Les trois meilleures techniques sur les neuf jeux de données

Le tableau 1 présente les trois meilleures techniques d'échantillonnage parmi les 16 techniques testées en se basant sur la précision de prédiction (une moyenne de précision sur 20 essais) obtenue, sur chacun des neuf jeux de données, en utilisant l'algorithme d'apprentissage C4.5 release 8 (Quinlan 1996). De ce tableau, nous pouvons aisément observer que les deux techniques « Actif - ECA - Géo. » et « Dynamique - ECA - Géo. » apparaissent presque toujours parmi les trois meilleures techniques. Pour les jeux de données où ce n'est pas le cas (c'est-à-dire, l'une de ces deux techniques n'apparaît pas parmi les trois meilleures techniques), nous faisons appel au tableau 2 qui présente une comparaison entre la technique manquante et la meilleure technique trouvée. En nous basant sur ces deux tableaux, nous pouvons remarquer que le taux d'erreur des

méthodes d'échantillonnage « Actif/Dynamique - ECA - Géo. » est toujours dans une fourchette d'au plus 5% (dans le pire cas) par rapport à la meilleure technique, et ce, pour les neuf jeux de données. De ces résultats, nous pouvons conclure que « Actif/Dynamique - ECA - Géo. » représentent les meilleurs techniques d'échantillonnage –du moins pour les jeux d'essais qui ressemblent aux nôtres– et par conséquent, elles seront les seules techniques utilisées pour la comparaison avec le méta-classificateur.

	Technique	Différence par rapport à la meilleure technique
Adult	Actif - ECA - Géo. (79.31%)	3.49%
Mush.	Dynamique - ECA - Géo. (98.82%)	0.94%
Tic.	Actif - ECA - Géo. (73.85%)	1.54%
Vote	Actif - ECA - Géo. (95.81%)	0.67%

TAB. 2 – La différence entre les meilleures techniques d'échantillonnage et celles présentées.

4.2 Test 2, expérimentation du FDD-AM

Nous rappelons que ces tests ont pour objectif de déterminer les valeurs optimales des seuils t et $t_{\mathcal{I}}$. Pour ce faire, nous avons choisi les paramètres suivants. Pour la construction des classificateurs de base, nous avons utilisé aussi l'algorithme C4.5 release 8 qui permet la production d'ensembles de règles. Le coefficient de confiance de chaque règle est calculé par rapport à un intervalle de confiance à 95% (c'est-à-dire, $N = 95$). Pour le seuil $t_{\mathcal{I}}$, nous avons utilisé respectivement 5% et 10% ; néanmoins, ces deux valeurs ont donné exactement les mêmes résultats. En ce qui concerne le seuil t, nous avons utilisé :

a. toutes les valeurs entre 0.95 et 0.20, avec un décrément de 0.05,
b. 0.01,
c. et μ, avec $\mu = 1/nd \sum_{i=1}^{nd} \mu_i$ et $\mu_i = 1/nr_i \sum_{k=1}^{nr_i} c_{r_{ik}}$. La valeur μ est utilisé dans le but de produire une manière automatique de calculer le seuil en se basant sur la moyenne de confiance qu'on a dans les règles produites.

Les différents tests effectués sur le méta-classificateur en utilisant les valeurs de seuil t citées ci-dessus, ont montré que μ donne les meilleurs résultats. Ce résultat est prévisible puisque ce seuil n'est pas une valeur fixe, mais plutôt une valeur qui trouve un consensus entre les différents μ_i en trouvant une valeur moyenne la plus proche. Ainsi, nous pouvons conclure que les valeurs optimales des seuils t et $t_{\mathcal{I}}$, selon l'ensemble de tests effectués, sont respectivement μ et 5% ou 10%.

4.3 Test 3, comparaison entre méta-apprentissage et échantillonnage

Nous basons notre comparaison sur les résultats trouvés dans les sections 4.1 et 4.2. Par conséquent, dans cette section nous limitons notre étude à l'échantillonnage « Dynamique/Actif - ECA - Géo. » comparés à un méta-classificateur avec $N = 95$, $t = \mu$ et $t_{\mathcal{I}} = 5\%$. La comparaison est conduite sur la base du taux d'erreur en prédiction, le temps d'exécution et la taille des échantillons nécessaire dans chaque

technique. Dans le but d'évaluer l'importance des taux d'erreur obtenus par le méta-classificateur et par les techniques d'échantillonnage, nous les avons comparés au taux d'erreur de C4.5 appliqué sur toute la base de données $DB = \cup_i DB_i$. Ce test est utilisé seulement comme référence, puisque, selon nos hypothèses, nous ne pouvons pas traiter DB en entier à cause des contraintes de temps de traitement et/ou de téléchargement. Le but de ce test de référence est de juger la pertinence du taux d'erreur évalué par les techniques présentées dans cet article. Par ailleurs, nous avons choisi l'algorithme C4.5 car c'est un algorithme très utilisé sur le marché.

La figure 5 montre les différents taux d'erreur obtenus. L'algorithme C4.5 est représenté par des histogrammes noirs. L'échantillonnage « dynamique - ECA - Géo. » est représenté par des histogrammes gris clair, l'échantillonnage « actif - ECA - Géo. » est représenté par des histogrammes blancs et le méta-classificateur par des histogrammes gris foncé. La première conclusion qu'on peut tirée de ce graphique est que tous les taux d'erreur peuvent être jugés acceptables puisque qu'ils sont, la plus part du temps, dans une fourchette d'au plus 3% pires que C4.5 sauf dans trois jeux d'essais (Adult, Chess et Tic-Tac-Toe). Il est à noter qu'aucune tendance fixe n'apparaît sur ces derniers. En effet, pour le premier jeu de données l'échantillonnage actif donne le pire résultat tandis que l'échantillonnage dynamique et le méta-classificateur donne à peu près le même taux d'erreur que C4.5 ($< 1\%$). Quant aux deuxième et troisième jeux de données, nous remarquons l'inverse où l'échantillonnage actif performe correctement (1.3% pire que C4.5) et où les deux autres techniques performent moins bien.

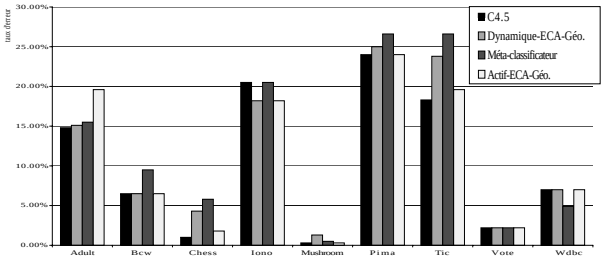

FIG. 5 – Comparaison des taux d'erreur entre le méta-classificateur et l'échantillonnage actif/dynamique ECA Géo. en considérant le taux d'erreur de C4.5 comme référence.

Selon ces résultats, nous pouvons ainsi conclure, que pour une légère perte en performance de classification[2], les techniques d'échantillonnage testées et le méta-classificateur présentent globalement des performances en classification acceptables par rapport à C4.5 appliqué à DB.

[2]La moyenne des différences par rapport à C4.5 sur les techniques (échantillonnage et méta-classificateur) est de 1.14%.

4.3.1 Comparaison des tailles

Le tableau 3 détaille la taille des bases de données ainsi que la taille des échantillons obtenus avec l'échantillonnage actif/dynamique-ECA-Géo. d'une part, et la taille des échantillons utilisés par le méta-classificateur d'autre part[3].

	$\sum$ des fichiers d'entraînement	Dynamique-ECA-Géo.	Actif-ECA-Géo.	Méta-classifier
Adult	20112	5440	4480	250
BCW	338	338	338	34
Chess	1598	630	630	145
Iono.	176	176	176	18
Mush.	2968	480	480	150
Pima.	384	374	374	39
Tic.	478	388	388	48
Vote	200	200	200	21
WDBC	285	285	285	29

TAB. 3 – Taille des bases de données et des échantillons.

Le tableau 3 montre que dans 4 cas (BCW, Iono., Vote and WDBC) la taille des échantillons issus de l'échantillonnage actif/dynamique-ECA-Géo. est la même que la taille de la base de données. Ce fait explique dans la figure 5 l'égalité du taux d'erreur des techniques d'échantillonnage à l'algorithme C4.5 pour ces 4 jeux de données. Par ailleurs, pour ces 4 jeux de données, notre méta-classificateur donne le même taux d'erreur que C4.5 dans deux cas (Iono. and Vote), 3% pire que C4.5 (BCW) et même mieux (WDBC) avec des échantillons aussi petits que 34 objets ou moins. Dans les 5 autres cas, le méta-classificateur a un taux d'erreur comparable aux techniques d'échantillonnage : meilleur que l'une des deux techniques ou pas plus mauvais que la pire technique d'échantillonnage par 2.80%. Cette performance est très intéressante puisque la taille des échantillons du méta-classificateur est nettement plus petite que celle qui est nécessaire par les techniques d'échantillonnage.

4.3.2 Comparaison du temps de traitement

Finalement, dans le but de comparer les techniques d'échantillonnage à la technique proposée de FDD-AM sur la base du temps d'exécution, nous pouvons examiner la figure 6. Notons que ces programmes ont été développés en C++, compilés par le même compilateur et exécutés sur la même machine.

Pour des raisons de présentation, nous avons omis de la figure 6 le temps d'exécution du jeu de données « adult » dont les valeurs pour l'échantillonnage dynamique/actif-ECA-Géo. et pour le méta-classificateur sont respectivement 20.078s, 38.905s et 9.852s.

Du tableau nous pouvons aisément conclure que, mis à part les jeux de données BCW et TIC, la technique de FDD-AM est toujours plus rapide et parfois beaucoup plus rapide que les deux techniques d'échantillonnage. En outre, l'analyse asymptotique illustrée en partie par la figure 6 et par le temps d'exécution de la base « adult » suggère que lorsque la taille des bases de données augmente, la technique de FDD-AM restera beaucoup plus rapide que les techniques d'échantillonnage. Ceci s'explique par

[3]Nous rappelons que les échantillons S_i utilisés afin de produire le méta-classificateur ont une taille maximale de 50 objets, mais elle peut être inférieur si la taille de DB_i est inférieur à 500.

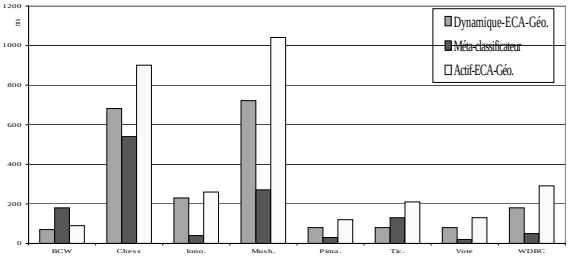

FIG. 6 – Comparaison du temps d'exécution entre le méta-classifier et les techniques d'échantillonnage actif/dynamique-ECA-Géo.

la méthode de détection de convergence très coûteuse en temps d'exécution pour les techniques d'échantillonnage dynamique et actif qui bâtissent un classificateur à chaque itération comparativement au méta-classificateur qui bâti pour chaque base de donnée distribuée (DB_i), en parallèle, un seul classificateur. Ainsi, nous pouvons extrapoler que pour de très grandes bases de données le temps d'exécution des techniques d'échantillonnage efficaces est de loin supérieur au temps d'exécution du méta-classificateur

5 Conclusion

L'objectif de ce papier est de faire une comparaison entre les techniques d'échantillonnage existantes et une nouvelle technique de forage distribué des données (FDD) par agrégation de modèles. Pour ce faire, nous avons présenté un survol des techniques d'échantillonnage les plus communes ainsi qu'une brève description de la nouvelle technique de FDD. En outre, nous avons conduit quelques expériences afin de :

1. déterminer les meilleures techniques d'échantillonnage ;
2. déterminer les paramètres optimaux du méta-classificateur ;
3. et comparer les meilleures techniques d'échantillonnage au méta-classificateur utilisant les paramètres optimaux trouvés précédemment, et ce, selon trois aspects différents : la précision de prédiction, le temps d'exécution et la taille des échantillons.

Comme résultats, ce sont l'échantillonnage dynamique/actif-ECA-Géo. qui sont presque toujours les meilleures techniques pour les jeux de données utilisés. La comparaison de ces dernières au méta-classificateur a révélé les conclusions suivantes :

- Toutes les techniques (échantillonnage et méta-classificateur) ont globalement des taux d'erreur comparables.
- Une comparaison de ces techniques avec l'algorithme C4.5, bâti sur l'agrégation de toutes les bases et utilisé comme référence, a prouvé que le taux d'erreur de ces techniques est généralement acceptable (avec une légère perte en performance de 1.14% en moyenne).

– Le méta-classificateur est plus rapide que les techniques d'échantillonnage.
– Le méta-classificateur nécessite des échantillons de beaucoup plus petite taille que ceux requis par les techniques d'échantillonnage.

En conclusion, si l'exigence en précision permet une légère perte ($< 1.5\%$ tel qu'évaluée du moins avec nos jeux de données), il apparaît que le méta-classificateur peut représenter une bonne solution pour le forage des bases de données distribuées car le temps de calcul requis est considérablement moindre. De surcroît, l'architecture multi-agents utilisé sous-tend la parallélisation de la technique et permettrait grâce à la hiérarchisation des agents collecteur encore plus d'adaptabilité à de très grandes bases de données. Des applications qui se servent de base de données transactionnelles, telle 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.

Références

Aounallah, M. et Mineau, G. (2004). Rule confidence produced from disjoint databases : a statistically sound way to regroup rules sets. *IADIS international conference, Applied Computing 2004*, pages II27 – II31, Lisbon, Portugal.

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

John, G. et Langley, P. (1996). Static Versus Dynamic Sampling for Data Mining. In Simoudis, E., Han, J., and Fayya, U. M., editors. *Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining*, pages 367–370, Portland, Oregon. AAAI/MIT Press.

Lewis, D. D. et Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin,IE. Springer Verlag, Heidelberg, DE.

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

Provost, F., Jensen, D. et Oates, T. (1999). Efficient progressive sampling. In Chaudhuri, S. and Madigan, D., editors. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 23–32, N.Y. ACM Press.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4 :77–90.

Saar-Tsechansky, M. et Provost, F. (2001). Active sampling for class probability estimation and ranking.