Une nouvelle approche du *boosting* face aux données bruitées

Emna Bahri*
Mondher Maddouri**

* Laboratoire Eric, Université Lyon 2, 5 avenue Pierre Mendes France, 69676 Bron Cedex
Emna.Bahri@univ-lyon2.fr,
http://eric.univ-lyon2.fr
**INSAT, zone urbaine la charguia II Tunis, 1002 Tunisie
Mondher.Maddouri@fst.rnu.tn,
http://www.insatech.net

Résumé. La réduction de l'erreur en généralisation est l'une des principales motivations de la recherche en apprentissage automatique. De ce fait, un grand nombre de travaux ont été menés sur les méthodes d'agrégation de classifieurs afin d'améliorer, par des techniques de vote, les performances d'un classifieur unique. Parmi ces méthodes d'agrégation, le *boosting* est sans doute le plus performant grâce à la mise à jour adaptative de la distribution des exemples visant à augmenter de façon exponentielle le poids des exemples mal classés. Cependant, en cas de données fortement bruitées, cette méthode est sensible au surapprentissage et sa vitesse de convergence est affectée. Dans cet article, nous proposons une nouvelle approche basée sur des modifications de la mise à jour des exemples et du calcul de l'erreur apparente effectuées au sein de l'algorithme classique d'*AdaBoost*. Une étude expérimentale montre l'intérêt de cette nouvelle approche, appelée Approche Hybride, face à *AdaBoost* et à BrownBoost, une version d'*AdaBoost* adaptée aux données bruitées.

1 Introduction Générale

L'émergence des bases de données modernes qui présentent d'énormes capacités de stockage et de gestion, associée à l'évolution des systèmes de transmission et des techniques d'acquisition automatique des données contribuent à la construction d'une masse de données qui dépasse de loin les capacités humaines à les traiter. Ces données sont des sources d'informations pertinentes qui nécessitent des outils de synthèse et d'interprétation. Les recherches se sont orientées vers des systèmes d'intelligence artificielle puissants permettant l'extraction des informations utiles et aidant à la prise des décisions. Pour une meilleure synthèse et interprétation, la fouille de données ou *data mining* est née en puisant ses outils au sein de la statistique, de l'intelligence artificielle et des bases de données. La méthodologie du data mining offre la possibilité de construire un modèle de prédiction d'un phénomène à partir d'autres phénomènes plus facilement accessibles, qui lui sont liés, en se basant sur le processus d'extraction des connaissances à partir des données qui n'est qu'un processus de classification intelligente des données. Cependant, le modèle construit peut parfois engendrer des erreurs

de classification que même une classification aléatoire n'aurait pas engendrée. Pour réduire ces erreurs, de nombreux travaux en data mining et spécifiquement en apprentissage automatique ont porté sur les méthodes d'agrégation de classifieurs. Leur but suprême est d'améliorer, par des techniques de vote, les performances d'un classifieur unique. Ces méthodes d'agrégation sont efficaces d'un point de vue compromis Biais-variance, mais aussi grâce aux trois raisons fondamentales (raison statistique, raison informatique et raison de représentation) expliquées dans l'étude menée par (Dietterich, 2000). Ces méthodes d'agrégation de classifieurs peuvent être regroupées en deux catégories : celles qui fusionnent des classifieurs prédéfinis, on trouve par exemple : le vote simple (Bauer et Kohavi, 1999), le vote pondéré (Bauer et Kohavi, 1999) et le vote à la majorité pondérée (Littlestone et Warmuth, 1994) et celles qui fusionnent des classifieurs selon les données durant l'apprentissage, on trouve des stratégies adaptatives (*boosting*) et son algorithme de base *AdaBoost* (Shapire, 1990) ou des stratégies aléatoires (*bagging*) (Breiman, 1996). Nous nous intéresserons, par la suite, au *boosting*, dans la mesure où l'étude comparative menée dans (Dietterich, 1999) montre bien que dans le cas où les données sont faiblement bruitées, *AdaBoost* est plus performant que le *bagging*, tout en semblant être immunisé contre le sur-apprentissage. En effet, *AdaBoost* essaye directement d'optimiser les votes pondérés. Cette constatation s'est traduite non seulement par le fait que l'erreur empirique sur l'échantillon d'apprentissage décroît exponentiellement avec le nombre d'itérations, mais également par le fait que l'erreur en généralisation baisse elle aussi. Cependant, cette même étude montre que, sur des données fortement bruitées, *AdaBoost* présente un taux d'erreur parfois plus important que le *bagging*. Une raison à ces mauvaises performances, mise en évidence par Dietterich, vient du fait que le *boosting* tend à augmenter le poids des exemples bruités à travers les itérations. La conséquence immédiate est le sur-apprentissage des exemples bruités. La vitesse de convergence du *boosting* se trouve également pénalisée sur ce type de données. Pour surmonter les difficultés rencontrées par le *boosting* face aux données bruitées, nous proposons une nouvelle approche qui associe les hypothèses déjà construites à l'hypothèse courante pour définir les erreurs de l'itération courante. De par sa nature, cette approche est appelée approche **Hybride**. La suite de cet article est organisée comme suit : la section 1 est consacrée à un état de l'art synthétique des principaux travaux visant à améliorer le *boosting*. En section 2, nous présentons l'amélioration du *boosting* que nous proposons. Dans la section 3, nous effectuons une large étude expérimentale visant à comparer, sur de nombreuses bases de données réelles, les performances d'*AdaBoost* et *AdaBoost* Hybride, en termes d'erreur, de rappel et de vitesse de convergence. Enfin, en section 4, nous terminons par une conclusion et des perspectives.

2 Etat de l'art

En présence de données bruitées, le *boosting* présente différentes faiblesses, telles que le sur-apprentissage et la dégradation de la vitesse d'apprentissage. Diverses améliorations ont été proposées qui opèrent sur la mise à jour du poids des exemples ou parfois sur le principe même du *boosting*. De ce fait, nous allons présenter les principales méthodes ayant comme objectif l'amélioration du *boosting* par rapport à ces deux faiblesses. Cet état de l'art est effectué selon deux axes de recherches. Le premier axe regroupe les approches abordant le problème de la gestion des données bruitées, sans laquelle des phénomènes de sur-apprentissage peuvent

survenir. Le deuxième axe regroupe les approches portant sur les problèmes de vitesse de convergence.

2.1 Le sur-apprentissage

L'émergence et l'évolution des bases de données modernes contraignent aujourd'hui les chercheurs à étudier et à améliorer les capacités de tolérance au bruit du *boosting*. En effet, ces bases de données sont fortement bruitées en raison des nouvelles technologies d'acquisition de données telles que le web. En parallèle, des études telles que celles de (Dietterich, 1999), (Rätsch, 1998) et (Schapire et Singer, 1999) montrent bien qu' *AdaBoost* tend à sur-apprendre les données lorsqu'elles sont bruitées. De ce fait, un certain nombre de travaux récents ont tenté de limiter ces risques de sur-apprentissage. Les améliorations proposées se fondent essentiellement sur le fait qu'*AdaBoost* tend à augmenter le poids des exemples bruités de manière exponentielle. Deux solutions se présentent pour réduire ces données bruitées. Soit ces données sont détectées et supprimées avant l'apprentissage à l'aide d'heuristiques efficaces (Brodley et Friedl, 1996), (Wilson et Martinez, 2000). Soit ces données sont détectées tout au long du processus de *boosting*, on parle alors d'une bonne gestion de bruit. Pour ce faire, les chercheurs se sont orientés vers l'amélioration des points forts du *boosting* tels que la mise à jour des exemples mal classés, la maximisation de la marge, la signification des poids qu'*Adaboost* associe aux hypothèses et enfin le choix de l'apprenant faible.

– Modification des poids des exemples : la mise à jour adaptative de la distribution des exemples, visant à augmenter le poids de ceux mal appris par le classifieur précédent, permet d'améliorer les performances de n'importe quel algorithme d'apprentissage. En effet, à chaque itération, la distribution courante favorise les exemples ayant été mal classés par l'hypothèse précédente. De ce fait, plusieurs chercheurs ont proposé des stratégies portant sur une modification de la mise à jour des poids des exemples, pour éviter le sur-apprentissage. *Madaboost* (Domingo et Watanabe, 2000) a pour principe de borner le poids des exemples suspects par leur probabilité initiale. Il agit ainsi sur la croissance incontrôlée du poids des exemples bruités qui est à l'origine des problèmes d'*AdaBoost*. Une autre approche qui rend l'algorithme du *boosting* résistant au bruit est *BrownBoost* (McDonald et al., 2003), un algorithme basé sur le *Boost-by-Majority* en incorporant un paramètre temporel. Ainsi par une bonne évaluation de ce paramètre, *BrownBoost* est-il capable d'éviter le sur-apprentissage.
  On citera encore *Logitboost* (Schapire et Singer, 1999), qui adapte au principe d'*AdaBoost* un modèle de régression logistique. *LogitBoost* réduit au minimum son critère en employant les étapes de *Newton-like* pour adapter un modèle de régression logistique en optimisant le logarithme de la vraisemblance. Une autre approche, qui produit moins d'erreur en généralisation comparée à l'approche classique mais au prix d'une erreur d'apprentissage légèrement plus élevée, est celle de (Vladimir et Vezhnevets, 2002). En fait, sa mise à jour se base sur la diminution de la contribution des classifieurs, si cela fonctionne "trop bien" sur les données qui ont déjà été correctement classifiées. C'est pourquoi la méthode est appelée *Modest AdaBoost*. Elle force les classifieurs à être "modestes" et travaille seulement dans le domaine défini par une distribution bien déterminée et un critère d'arrêt normal. *SmoothBoost* (Servedio, 2001) essaye de réduire l'effet de sur-apprentissage par des limites imposées à la distribution produite pendant le processus de *boosting*. En particulier, lors de chaque itération, on fixe une limite de

poids. Un exemple dont le poids dépasse cette limite est considéré comme bruité et retiré de l'ensemble d'apprentissage.
Une dernière approche, IAdaBoost (Sebban et Suchier, 2003), se base sur l'idée de construire autour de chacun des exemples une mesure d'information locale permettant d'évaluer les risques de surapprentissage, en utilisant un graphe de voisinage qui permet de mesurer l'information autour de chaque exemple. Grâce à ces mesures, on calcule une fonction qui évalue la nécessité de mettre à jour l'exemple avec *AdaBoost*. Cette fonction permet de gérer à la fois les *outliers*, les recoupements de classes et les centres de *clusters*.

– Modification de la marge : certaines études, après une observation des algorithmes de *boosting*, ont montré que l'erreur en généralisation décroît encore une fois que l'erreur en apprentissage est stable ou même nulle. L'explication est que même si tous les exemples d'apprentissage sont déjà bien classés, le *boosting* tend à maximiser davantage les marges (Servedio, 2001), d'où les performances du *boosting*. À la suite de cette explication, certains ont cherché à modifier la marge explicitement soit en la maximisant soit en la minimisant dans le but d'améliorer les performances de *boosting* contre le sur-apprentissage. Plusieurs approches se sont succédées telles que *AdaBoostReg* (Rätsch et al., 2001) qui essaye d'identifier et d'enlever les exemples mal étiquetés de l'ensemble d'apprentissage, ou d'appliquer la contrainte de la marge maximale sur des exemples supposés mal étiquetés, en utilisant la *Soft Margin* qui est moins sensible au sur-apprentissage par rapport à la marge d' *Adaboost*. Dans l'algorithme proposé par (Friedman et al., 1998), les auteurs utilisent un schéma de pondération qui exploite une fonction des marges qui croît moins vite que la fonction exponentielle.

– Modification de poids des classifieurs : lors de l'évaluation des performances du *boosting*, des chercheurs se sont également interrogés sur la signification des poids $\alpha(t)$ qu'*AdaBoost* associe aux hypothèses produites. Ce poids est une valeur déterminée en fonction des échecs et des réussites de classification sur un échantillon bien déterminé. Cependant, ils ont noté lors d'expériences sur des données très simples que l'erreur en généralisation diminuait encore alors que l'apprenant faible avait déjà fourni toutes les hypothèses possibles. Autrement dit, lorsqu'une hypothèse apparaît plusieurs fois, elle vote finalement avec un poids, cumul de tous ses $\alpha(t)$, qui a peut-être un caractère plus absolu. De ce fait, plusieurs auteurs ont espéré approcher ces valeurs par un processus non adaptatif, tel que *Locboost* (Meir et al., 2000), une alternative à la construction de l'ensemble de représentation des hypothèses, qui permet aux coefficients $\alpha(t)$ de dépendre des données. On aura donc des poids locaux attribués à chaque exemple.

– Choix de l'apprenant faible : Plusieurs auteurs se sont intéressés au choix du classifieur de base du *boosting*. *GloBoost* (Torre, 2004) utilise un apprenant faible qui produit des hypothèses correctes. Celles-ci peuvent donc s'abstenir sur une partie des exemples mais en aucun cas se tromper sur un exemple. Il s'agit de moindres généralisés maximalement corrects. *RankBoost* (Dietterich, 1999) se base sur un apprenant faible qui accepte comme données d'entrées des attributs de préférences (*rank*) qui ne sont que des fonctions.

Certes, ces méthodes ont pu améliorer d'une façon ou d'une autre la performance du *boosting* contre le bruit. Toutefois, d'autres paramètres du *boosting* se trouvent pénalisés, tels que l'erreur d'apprentissage, le temps de détection du bruit et la vitesse de convergence.

2.2 La vitesse de convergence

En plus du problème de sur-apprentissage rencontré par le *boosting* dans les bases de données modernes déjà évoqué précédemment, il existe un autre problème qui est celui de la vitesse de convergence des algorithmes de *boosting* (spécialement *Adaboost*). En effet, en cas de présence de données fortement bruitées, l'erreur optimale de l'algorithme d'apprentissage utilisé est atteinte tardivement. En d'autres termes, *Adaboost* "perd" du temps, et donc des itérations à pondérer ces exemples qui ne méritent aucune attention, puisqu'il s'agit de bruit.
Des recherches ont été menées pour détecter les données bruitées et améliorer ainsi les performances du *boosting* en termes de convergence tel que *iBoost* (Kwek et Nguyen, 2002), qui vise à spécialiser les hypothèses faibles sur les exemples qu'elles sont supposées correctement classer. *iAdaboost* est aussi une approche qui contribue à améliorer *Adaboost* contre sa convergence. En fait, l'idée de base de l'amélioration est la modification du théorème de (Schapire et Singer, 1999). Cette modification est réalisée afin d'intégrer le risque de Bayes et de mettre en exergue les situations où certains exemples de classes différentes partagent la même représentation. Les effets de cette modification sont une convergence plus rapide vers le risque optimal et une réduction du nombre d'hypothèses faibles à construire.
Enfin, *RegionBoost* (Maclin, 1998) est une nouvelle stratégie de pondération des classifieurs. Cette pondération est évaluée au moment du vote par une technique basée sur les k plus proches voisins de l'exemple à étiqueter. Cette approche permet de spécialiser chaque classifieur sur des régions de l'ensemble d'apprentissage.

3 Amélioration proposée : *Adaboost Hybride*

Pour améliorer les performances d'*Adaboost* et éviter de le forcer à apprendre des exemples a priori bruités ou des exemples qui deviendraient trop difficiles à apprendre durant le processus du *boosting*, nous proposons une nouvelle approche qui s'inspire du fait qu'*Adaboost* construit, à chaque itération, des hypothèses sur un échantillon bien défini. La mise à jour et le calcul de l'erreur d'apprentissage sont faits à partir des résultats de ces seules hypothèses et n'exploitent pas les résultats fournis par les hypothèses construites aux itérations antérieures sur d'autres échantillons. Cette approche est appelée approche **Hybride** au sens où elle se base sur les hypothèses antérieures : la mise à jour des exemples à l'itération courante tiendra compte non seulement des résultats de l'itération courante mais aussi de ceux des itérations antérieures.

3.1 Pseudo code de *Adaboost Hybride*

Soit X_0 la classe à prévoir et $S = (x_1, y_1), ..., (x_n, y_n)$ un échantillon
- Pour $i = 1, 2...n$, faire
- Initialiser les poids $p_0(x_i) = 1/n$;
- Fin pour
- $t \leftarrow 0$

- Tant que $t \leq T$ faire
- Tirer un échantillon d'apprentissage S_t dans S selon les probabilités p_t.
- Construire une hypothèse h_t sur S_t par un algorithme d'apprentissage A.
- Soit ϵ_t l'erreur apparente de h_t sur S avec $\epsilon_t = \sum poids\ \ des\ exemples$ ***tel que*** $argmax(\sum_{i=1}^{t} \alpha_i h_i(x_i) \neq y_i)$. Calculer $\alpha_t = 1/2ln((1-\epsilon_t)/\epsilon_t)$.
- Pour i=1, m faire
- $P_{t+1}(x_i) \leftarrow (p_t(x_i)/Z_t)e^{-\alpha_t}$ **si** $argmax(\sum_{i=1}^{t} \alpha_i h_i(x_i)) = y_i$ **(bien classé)**
  $P_{t+1}(xi) \leftarrow (p_t(x_i)/Z_t)e^{+\alpha_t}$ **si** $argmax(\ \sum_{i=1}^{t} \alpha_i h_i(x_i)) \neq y_i$ **(mal classé)**
  (Z_t est une valeur de normalisation telle que $\sum_{i=1}^{n} p_t(x_i) = 1$)
- Fin Pour
  $t \leftarrow t+1$
- Fin tant que
- Fournir en sortie l'hypothèse finale :
  $H(x) = argmax\ \ y \in Y \sum_{t=1}^{T} \alpha_t$

La modification de l'algorithme porte sur la prise en compte de l'ensemble des itérations passées pour effectuer la prédiction courante, ce qui modifie notamment le poids des exemples et le calcul de l'erreur.

- Modification des poids des exemples : à chaque itération, on fait appel aux avis des experts déjà utilisés (hypothèses des itérations antérieures) pour mettre à jour les poids des exemples. En effet, on ne compare pas seulement la classe prédite par l'hypothèse à l'itération courante avec la classe réelle mais la somme des hypothèses pondérées depuis la première itération jusqu'à l'itération courante. Si cette somme vote pour une classe différente de la classe réelle alors une mise à jour exponentielle semblable à celle effectuée par *Adaboost* est appliquée à l'exemple mal classé. Ainsi, cette modification ne concerne-t-elle que les exemples qui sont soit mal classés soit pas encore classés. Il est donc logique de s'attendre à une amélioration de la vitesse de convergence, de même pour la réduction de l'erreur en généralisation étant donnée le lissage des hypothèses à chaque itération.
- Modification du calcul de l'erreur $\epsilon(t)$ de l'hypothèse à l'itération t : la méthode que nous proposons, prend en considération les hypothèses antérieures à l'itération courante pour former l'hypothèse courante. De ce fait, à chaque itération, l'erreur apparente $\epsilon(t)$ est le poids des exemples prédits de façon erronée par la moyenne pondérée des hypothèses des itérations antérieures. Du coup, le coefficient attribué à l'hypothèse courante, $\alpha(t)$, est lui aussi modifié, puisque ce coefficient dépend du calcul de l'erreur apparente $\epsilon(t)$. Cette modification a un effet de lissage et laisse l'algorithme à chaque itération très dépendant des autres itérations. Des résultats améliorant surtout l'erreur en généralisation sont attendus puisque le vote de chaque hypothèse (coefficient $\alpha(t)$) est calculé à partir des autres hypothèses.

4 Expérimentations

Dans cette section, nous allons comparer les résultats de notre algorithme *AdaBoostHyb* avec ceux fournis par *AdaBoost*, l'algorithme de référence et par *BrownBoost*, un algorithme connu pour être résistant face aux données bruitées. En fait, *BrownBoost* utilise la fonction de pondération suivante, qui n'est rien d'autre que la loi de probabilité d'une variable binomiale,

qui dépend du nombre d'itérations finales k (temps total d'exécution), de l'itération courante i, du nombre de fois où l'exemple a déjà été correctement étiqueté r, et enfin de la probabilité de succès $1-\gamma$ imposée à toute hypothèse faible.

$\alpha_r^i = \binom{k-i-1}{k/2-r}(1/2+\gamma)^{(k/2)-r}(1/2-\gamma)^{(k/2)-i-1+r}$

L'avantage de cette approche est que les données bruitées seront probablement détectées à un certain moment, et leur poids cessera d'augmenter. Au cours des expérimentations, la comparaison se fait à travers l'erreur en généralisation, le rappel et la vitesse de convergence. L'apprenant faible utilisé est l'algorithme C4.5 choisi suite à l'étude de (Dietterich, 1999) qui a montré que C4.5 est très sensible au bruit. Pour estimer sans biais le taux de succès théorique, nous avons fait appel à une procédure de validation croisée en 10 parties.

Afin d'évaluer le comportement d'*AdaBoostHyb* vis à vis tant des performances que de la vitesse de convergence, nous avons séparé nos expérimentations en plusieurs parties. Dans la première, où nous avons travaillé sur 15 bases de l'UCI (D.J. Newman et Merz, 1998), nous rapportons la valeur de l'erreur en généralisation et le rappel, choisis comme critère de performance. Dans la deuxième partie, nous avons bruité aléatoirement ces bases de données avec un taux de bruit de 20%, pour analyser le comportement des trois algorithmes retenus. Ce taux est choisi en référence à l'étude de (Dietterich, 1999) qui signalait les résultats décevants d'*AdaBoost* face à des données bruitées. Dans la dernière partie, nous avons établi un diagnostic de convergence de ces différents algorithmes en nous fondant sur le nombre d'itérations effectuées. Nous avons choisi de retenir des bases de données diverses. En effet, les bases de données choisies contiennent des bases ayant des valeurs manquantes (NHL, VOTE, HEPATITIS, HYPOTHYROID), d'autres ayant une classe à prédire qui comporte plusieurs modalités (Iris : classe à 3 modalités, DIABETES : 4 modalités, Zoo : classe à 7 modalités, IDS : classe à 12 modalités), d'autres ayant plusieurs attributs (IDS : 35 attributs). Le tableau 1 décrit les 15 bases de données utilisées dans nos expérimentations.

Bases de données	Nb. Inst	Attrib	Cl. Pred	Val manq
IRIS	150	4 numeric	3	non
NHL	137	8 numeric and symbolic	2	oui
VOTE	435	16 boolean valued	2	oui
WEATHER	14	4 numeric and symbolic	2	non
CREDIT-A	690	16numeric and symbolic	2	oui
TITANIC	750	3 symbolic	2	non
DIABETES	768	8 numeric	2	non
HYPOTHYROID	3772	30 numeric and symbolic	4	oui
HEPATITIS	155	19 numeric and symbolic	2	oui
CONTACT-LENSES	24	4 nominal	3	non
ZOO	101	18 numeric and boolean	7	non
STRAIGHT	320	2 numeric	2	non
IDS	4950	35 numeric and symbolic	12	non
LYMPH	148	18 numeric	4	non
BREAST-CANCER	286	9 numeric and symbolic	2	oui

TAB. 1 – *Caractéristiques des bases de données*

4.1 Comparaison en termes d'erreur en généralisation

Le tableau 2 présente les résultats obtenus pour cette partie en ayant choisi pour chacun des algorithmes d'effectuer 20 itérations. Le choix du nombre d'itérations sera expliqué dans la dernière partie des expériences. Nous avons indiqué le taux d'erreur en généralisation es-

timé pour chacun des algorithmes *AdaBoostM1*, *BrownBoost* et *AdaBoostHyb*. Nous avons par ailleurs utilisé les mêmes échantillons pour la validation croisée des différents algorithmes afin d'avoir une comparaison plus fine.

L'observation des résultats montre déjà les effets positifs de l'approche hybride. En effet, pour 14 bases sur 15, l'algorithme *AdaBoostHyb* présente un taux d'erreur inférieur ou égal à celui d'*AdaBoostM1*. C'est seulement pour la base LYMPH que notre approche donne une erreur de généralisation plus élevée que l'approche classique. Nous remarquons, aussi, des améliorations significatives de l'erreur en généralisation correspondant aux bases de données NHL, CONTACT-LENS et BREAST-CANCER. Par exemple l'erreur en généralisation de la base BREAST-CANCER passe de 45.81% à 30.41%.

De même, si nous comparons l'approche proposée avec *BrownBoost*, nous remarquons que pour 11 bases de données sur 15 l'algorithme *AdaBoostHyb* présente un taux d'erreur inférieur ou égal à *BrownBoost*. Ce gain en faveur de *AdaBoostHyb* nous montre bien qu'en exploitant les hypothèses générées aux itérations antérieures pour corriger le poids des exemples, il est possible d'améliorer les performances du *boosting*. Ceci peut être expliqué par le mode de calcul de l'erreur apparente $\epsilon(t)$ et par conséquent le calcul du coefficient du classifieur $\alpha(t)$ ainsi que par l'hybridation de l'hypothèse courante et des hypothèses antérieures.

Databases	AdaBoost M1	BrownBoost	AdaBoostHyb
IRIS	6.00%	3.89%	**3.00%**
NHL	35.00%	30.01%	**28.00%**
VOTE	4.36%	4.35%	**4.13%**
WEATHER	21,42%	**21.00%**	**21.00%**
CREDIT-A	15.79%	**13.00%**	13.91%
TITANIC	**21.00 %**	24.00%	**21.00%**
DIABETES	27.61%	**25.05%**	25.56%
HYPOTHYROID	0.53%	0.6%	**0.42%**
HEPATITIS	15,62%	14.10%	**14.00%**
CONTACT-LENSES	25.21%	**15.86%**	16.00%
ZOO	**7.00%**	7.23%	**7.00%**
STRAIGHT	2,40%	**2.00%**	**2.00%**
IDS	1,90%	0.67%	**0,37%**
LYMPH	19.51%	**18.54%**	20.97%
BREAST-CANCER	45.81%	31.06%	**30.41%**

TAB. 2 – *Taux d'erreurs en généralisation*

4.2 Comparaison en terme de rappel

Les résultats encourageants auxquels nous sommes parvenus, nous mènent à approfondir l'étude de cette nouvelle approche. Dans cette partie, nous essayons de connaître l'impact de notre approche sur le taux de rappel, puisque celle-ci n'améliore effectivement le *boosting* que si elle agit positivement sur le rappel.

Le tableau 3 présente les résultats obtenus pour cette partie, ayant choisi pour chacun des algorithmes d'effectuer 20 itérations comme précédemment. Nous avons indiqué le rappel pour chacun des algorithmes *AdaBoostM1*, *BrownBoost* et *AdaBoostHyb*.

Les résultats obtenus ici confirment les précédents. En effet, *AdaBoostHyb* augmente le rappel des bases de données ayant des taux d'erreur moins important. Le rappel des deux algorithmes est le même dans le cas où les taux d'erreur des bases de données sont égaux.

Si nous considérons *BrownBoost*, nous remarquons que ce dernier améliore le rappel d'*AdaBoostM1*,

pour chaque base de données (sauf la base de données TITANIC). Cependant, le rappel obtenu par notre approche est meilleur que celui obtenu par *BrownBoost*, sauf pour la base de données ZOO.

Nous constatons aussi que notre approche améliore le rappel dans le cas de la base LYMPH où l'erreur était plus importante. Nous notons alors que la nouvelle approche n'agit pas négativement sur le rappel mais elle l'améliore même lorsque l'on a une erreur de généralisation plus importante.

Databases	***AdaBoost* M1**	***BrownBoost***	***AdaBoost*Hyb**
IRIS	0,93	0.94	**0,96**
NHL	0,65	0,68	**0,71**
VOTE	0,94	0.94	**0,95**
WEATHER	0,63	0.64	**0,64**
CREDIT-A	0,84	0.85	**0,86**
TITANIC	**0,68**	0.54	**0,68**
DIABETES	0,65	0.66	**0,68**
HYPOTHYROID	0,72	0.73	**0,74**
HEPATITIS	0,69	0.70	**0,73**
CONTACT-LENSES	0,67	0.75	**0,85**
ZOO	0,82	**0.9**	0,82
STRAIGHT	0,95	0.95	**0,97**
IDS	0,97	0.97	**0,98**
LYMPH	0,54	0.62	**0,76**
BREAST-CANCER	0,53	0.55	**0,6**

TAB. 3 – *Rappel*

4.3 Comparaison sur des données bruitées

Dans cette partie, on s'est basé sur l'étude déjà faite par Dietterich (Dietterich, 1999) en ajoutant du bruit aléatoire aux données. Cet ajout de bruit de 20% est effectué, pour chacune de ces bases, en changeant aléatoirement la valeur de la classe prédite à l'aide d'un programme par une autre valeur possible de cette classe. Le tableau 4 nous montre le comportement des algorithmes vis- à-vis du bruit. Nous remarquons bien que l'approche hybride est sensible elle aussi au bruit puisque le taux d'erreur en généralisation est augmenté pour toute les bases des données. Cependant cette augmentation reste toujours inférieure à celle de l'approche classique sauf pour les bases de données telles que CREDIT-A, HEPATITIS et HYPOTHYROID. Nous avons donc étudié de près ces bases de données et nous avons noté un point commun, les valeurs manquantes. En fait, CREDIT-A, HEPATITIS et HYPOTHYROID possèdent respectivement 5%, 6%et 5,4% de valeurs manquantes.

Nous constatons alors que notre amélioration perd son effet avec l'accumulation de deux types de bruit : les valeurs manquantes et le bruit artificiel, bien que l'algorithme *AdaBoostHyb* améliore les performances d'*AdaBoost* contre le bruit sur le reste des bases de données. Considérant *BrownBoost*, nous remarquons qu'il améliore l'erreur en généralisation de toute les bases de données en comparaison avec *AdaBoostM1*. Cependant, *BrownBoost* donne des résultats meilleurs que l'approche hybride sur seulement 6 bases de données. Notre approche donne les meilleurs résultats avec les 9 autres bases de données. Ces résultats nous encouragent à étudier en détails le comportement de notre approche vis-à-vis du bruit.

Bases de Données	***AdaBoost* M1**	***BrownBoost***	***AdaBoost*Hyb**
IRIS	33.00%	**26.00%**	28.00%
NHL	45.00%	40.00%	**32.00%**
VOTE	12.58%	**7.00%**	7.76%
WEATHER	25.00%	22%	**21%**
CREDIT-A	22.56%	**20.99%**	24.00%
TITANIC	34.67%	28.08%	**26.98%**
DIABETES	36.43%	32.12%	**31.20%**
HYPOTHYROID	0.92%	**0.86%**	2.12%
HEPATITIS	31.00%	**27.38%**	41.00%
CONTACT-LENSES	33%	30.60%	**25%**
ZOO	18.84%	14.56%	**11.20%**
STRAIGHT	3.45%	**2.79%**	2.81%
IDS	2.40%	1.02%	**0.50%**
LYMPH	28.73%	24.57%	**24.05%**
BREAST-CANCER	68.00%	50.98%	**48.52%**

TAB. 4 – *Erreur en généralisation des données bruités*

4.4 Comparaison de la vitesse de convergence

Dans cette partie, on va s'intéresser au nombre d'itérations à partir duquel les algorithmes convergent, c'est à dire où le taux d'erreur se stabilise. Le tableau 5 nous montre que l'approche hybride permet à *AdaBoost* de converger plus vite. En effet, le taux d'erreur d'*AdaBoostM1* ne se stabilise pas, même à la $1000^{ième}$ itération, alors que *AdaBoostHyb* converge à la $20^{ième}$ itération ou même avant. C'est pour cette raison qu'on a choisi pour la première partie 20 itérations pour effectuer la comparaison en termes d'erreur et de rappel. Ces résultats sont aussi valables pour la base de données HEPATITIS. En fait, cette base de données est riche de valeurs manquantes (taux de 6%). Ces valeurs manquantes présentent toujours un problème de convergence pour les algorithmes d'apprentissage. De plus, ces même résultats se manifestent sur des bases de données de différents types (plusieurs attributs, classe à prédire ayant K modalités, tailles importantes). Ceci nous laisse penser que grâce à la façon de calculer l'erreur apparente en tenant compte des hypothèses antérieures, l'algorithme atteint plus rapidement la stabilité. Finalement, nous remarquons que *BrownBoost*, ne converge pratiquement pas même après 1000 itérations ce qui confirme le problème de vitesse de convergence de *BrownBoost*.

-	*AdaBoostM1*				*BrownBoost*				*AdaBoosthyb*			
Nb. iterations	10	20	100	1000	10	20	100	1000	10	20	100	1000
Iris	7,00	6,00	5,90	5,85	3.96	3.89	3,80	3,77	3,50	3,00	3,00	3,00
Nhl	37,00	35,00	34,87	34,55	30,67	30,01	29,89	29,76	31,00	28,00	28,00	28,00
Weather	21,50	21,42	21,40	14,40	21,10	21,00	20,98	21,95	21,03	21,00	21,00	21,00
Credit-A	15,85	15,79	15,75	14,71	13,06	13,00	12,99	12,97	14,00	13,91	13,91	13,91
Titanic	21,00	21,00	21,00	21,00	24,08	24,00	23,89	23,79	21,00	21,00	21,00	21,00
Diabetes	27,70	27,61	27,55	27,54	25,09	25,05	25,03	25,00	25,56	25,56	25,56	25,56
Hypothyroid	0,60	0,51	0,51	0,50	0,62	0,60	0,59	0,55	0,43	0,42	0,42	0,42
Hepatitis	16,12	15,60	14,83	14,19	14,15	14,10	14,08	14,04	14,03	14,00	14,00	14,00
Contact-Lenses	26,30	24,80	24,50	16,33	15,90	15,86	15,83	15,80	16,00	16,00	16,00	16,00
Zoo	7,06	7,00	7,00	7,00	7,25	7,23	7,19	7,15	7,00	6,98	7,00	7,00
Straight	2,50	2,46	2,45	2,42	2,12	2,00	1,98	1,96	0,42	0,42	0,42	0,42
IDS	2,00	1,90	1,88	1,85	0,7	0,67	0,65	0,63	0,7	0,67	0,65	0,63
Lymph	19,53	19,51	19,51	19,50	18,76	18,54	18,50	18,45	18,76	18,54	18,50	18,45
Breast-Cancer	45,89	45,81	45,81	45,79	31,10	31,06	31,04	31,00	31,10	31,06	31,04	31,00

TAB. 5 – *Comparaison de la vitesse de convergence*

5 Conclusion

Dans cet article, nous avons proposé une amélioration d'*AdaBoost* qui se fonde sur l'exploitation des hypothèses déjà construites aux itérations précédentes. Les expérimentations et les résultats trouvés montrent que cette approche améliore les performances d'*AdaBoost* en taux d'erreur, en rappel et en vitesse de convergence. Cependant, il s'est avéré que cette même approche est sensible elle aussi au bruit.

Nous avons effectué une étude comparative de notre approche Hybride avec *BrownBoost*, une approche connue pour son amélioration d'AbaBoost face au bruit. Les résultats trouvés montrent que l'approche Hybride donne des résultats meilleurs que *BrownBoost* en termes de rappel et de vitesse de convergence sur les 15 bases de données. Ils montrent aussi que *BrownBoost* donne des taux d'erreur meilleurs sur certains bases de données. Avec les données bruitées, on arrive à la même conclusion. Vu les résultats encourageants, une étude théorique de la convergence est envisagée pour justifier les résultats expérimentaux.
Une autre perspective qui nous semble importante consiste à améliorer cette approche contre les données bruitées en se basant soit sur les graphes de voisinage soit sur des paramètres de mise à jour efficaces. Enfin, une dernière perspective de ce travail consiste à étudier le *boosting* d'apprenant générant plusieurs règles (à chaque itération). En effet, le problème de ce type d'apprenant est la production, à chaque itération, de règles qui peuvent être conflictuelles.

Références

Bauer, E. et R. Kohavi (1999). An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine Learning 24*, 173–202.

Breiman, L. (1996). Bagging predictors. *Machine Learning 26*, 123–140.

Brodley, C. E. et M. A. Friedl (1996). Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pp. 799–805.

Dietterich, T. G. (1999). An experimental comparison of three methods for constructing ensembles of decision trees : bagging, boosting, and randomization. *Machine Learning*, 1–22.

Dietterich, T. G. (2000). Ensemble methodes in machine learning. *First International Workshop on Multiple ClassifierSystems*, 1–15.

D.J. Newman, S. Hettich, C. B. et C. Merz (1998). Uci repository of machine learning databases.

Domingo, C. et O. Watanabe (2000). Madaboost : A modification of adaboost. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pp. 180–189. Morgan Kaufmann, San Francisco.

Friedman, J., T. Hastie, et R. Tibshirani (1998). Additive logistic regression : a statistical view of boosting. *Dept. of Statistics, Stanford University Technical Report.*.

Kwek, S. et C. Nguyen (2002). iboost : Boosting using an instance-based exponential weighting scheme. *hirteenth European Conference on Machine Learning*, 245–257.

Littlestone, N. et M. K. Warmuth (1994). The weighted majority algorithm. In *Information and computation*, Volume 24, pp. 212–261.

Maclin, R. (1998). Boosting classifiers regionally. *In AAAI/IAAI*, 700–705.

McDonald, R., D. Hand, et I. Eckley (2003). An empirical comparison of three boosting algorithms on real data sets with artificial class noise. *In Fourth International Workshop on Multiple Classifier Systems*, 35–44.

Meir, R., R. El-Yaniv, et S. Ben-David (2000). Localized boosting. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pp. 190–199. Morgan Kaufmann, San Francisco.

Rätsch, G. (1998). Ensemble learning methods for classification. *Master's thesis,Dep of computer science, University of Potsdam*.

Rätsch, G., T. Onoda, et K.-R. Müller (2001). Soft margins for adaboost. *Mach. Learn. 42*(3), 287–320.

Schapire, R. E. et Y. Singer (1999). Improved boosting algorithms using confedence rated predictions. *Machine Learning 37*(3), 297–336.

Sebban, M. et H.-M. Suchier (2003). Étude sur amélioration du boosting : réduction de l'erreur et accélération de la convergence. *Journal électronique d'intelligence artificielle*. submitted.

Servedio, R. A. (2001). Smooth boosting and learning with malicious noise. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, Volume 2111, pp. 473–489. Springer, Berlin.

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

Torre, F. (2004). Globoost : Boosting de moindres généralisés. Technical report, GRAppA - Université Charles de Gaulle - Lille 3.

Vladimir, A. et Vezhnevets (2002). Modest adaboost : Teaching adaboost to generalize better. *Moscow State University*.

Wilson, D. R. et T. R. Martinez (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning 38*(3), 257–286.

Summary

The reduction of the error in generalization is one of the principal motivations of research in machine learning. This is more important as the cost induced by a bad classification is high. Thus a great number of work is carried out on the methods of aggregation of classifiers in order to improve, by techniques of vote, the performances of a single classifier. Among these methods of aggregation, we find the *boosting* which is most practical thanks to the adaptive update of the distribution of the examples aiming at increasing in an exponential way the weight of the badly classified examples. However, this method is blamed following on training, and the speed of convergence especially with noise. In this study, we propose a new approach and modifications carried out on the algorithm of *AdaBoost*. We will demonstrate, by exploiting assumptions generated with the former iterations to correct the weights of the examples, that it is possible to improve the performances of the *boosting*. An experimental study shows the interest of this new approach, called hybrid approach.