Combinaison de moindres généralisés

Fabien Torre

GRAppA, Université Charles de Gaulle - Lille III

ENST Paris : 8 juin 2005

Plan de l'exposé

Contexte : l'apprentissage supervisé

Notations et Définitions

- des exemples $x_i \in \mathcal{X}$ étiquetés (classe $y_i \in \{-1, +1\}$) ;
- des hypothèses $h \in \mathcal{H} : \mathcal{X} \to \{-1, +1\}$;
- un apprenant est un algorithme qui prend en entrée des exemples étiquetés $\{(x_i, y_i)\}$ et fournit une hypothèse $h \in \mathcal{H}$;
- on définit l'erreur ϵ de h comme la probabilité de trouver un exemple de $\mathcal{X}$ sur lequel h et le concept cible sont en désaccord.

Bases du Bagging [Breiman, 1996a]

Motivation

- Compromis biais/variance ;
- cible : les apprenants instables.

Échantillonnage

- Entrée : A un ensemble des données d'apprentissage contenant n exemples ;
- Sortie : A' un nouvel ensemble de données de même taille ;
- Tirer avec remise n exemples de A et les placer dans A'.

Bagging [Breiman, 1996a] : algorithmes

Bagging

- Entrées : un ensemble d'exemples $A = \{(x_i, y_i)\}$, un nombre d'itérations T à effectuer, un apprenant L ;
- Sortie : H le classifieur final ;
- Pour t allant de 1 à T
  - A_t =Échantillonnage(A)
  - $h_t = L(A_t)$
- retourner $H(x) = \text{sign}\left(\sum_{t=1}^{T} h_t(x)\right)$

Une application : les random forests [Breiman, 2001]

- construction stochastique de l'arbre de décision ;
- à chaque nœud, tirage aléatoire uniforme de m attributs ;
- choix du meilleur test (critère entropique) sur ces m attributs.

Apprenabilité dans le cadre PAC

$\mathcal{C}$ est une classe de concepts apprenable s'il existe un algo L tel que

- pour tout concept c de $\mathcal{C}$,
- pour toute distribution $\mathcal{D}$ sur les exemples,

L, en utilisant un oracle $EX(c, \mathcal{D})$, fournit une hypothèse $h \in \mathcal{C}$ qui, avec une probabilité $1 - \delta$, vérifie erreur$(h) \leq \epsilon$.

Deux notions d'apprenabilité

- apprenabilité forte [Valiant, 1984] :
  $\forall \epsilon, \delta : 0 < \epsilon < \frac{1}{2}$ et $0 < \delta < \frac{1}{2}$;
- apprenabilité faible [Kearns and Valiant, 1989] : $\exists \epsilon, \delta : \epsilon < \frac{1}{2}$.

Résultat

- deux preuves d'équivalence [Schapire, 1990, Freund, 1995]
- boosting d'un apprenant faible en un apprenant fort ;
- AdaBoost : algorithme boostant un apprenant faible.

Combinaison de trois votants

- $h_1 = L(EX(c, \mathcal{D}))$
- on définit $EX(c, \mathcal{D}_2)$
  - on tire à pile ou face ;
  - si pile, on appelle $EX(c, \mathcal{D})$ jusqu'à obtenir un exemple e vérifiant $h_1(e) = c(e)$;
  - si face, on appelle $EX(c, \mathcal{D})$ jusqu'à obtenir un exemple e vérifiant $h_1(e) \neq c(e)$;
  - renvoyer e.
- $h_2 = L(EX(c, \mathcal{D}_2))$
- on définit $EX(c, \mathcal{D}_3)$
  - on appelle $EX(c, \mathcal{D})$ jusqu'à obtenir un exemple e vérifiant $h_1(e) \neq h_2(e)$;
  - renvoyer e.
- $h_3 = L(EX(c, \mathcal{D}_3))$
- renvoyer $H(x) = \text{majorité}(h_1, h_2, h_3)$

L'algorithme AdaBoost

Entrées

- E un échantillon suffisant d'exemples étiquetés $\{(x_i, y_i)\}$,
- T un nombre d'étapes de boosting suffisant,
- un apprenant faible A.

Algorithme

1. initialiser les poids des exemples $w_i = \frac{1}{|E|}$
2. pour t allant de 1 à T
   1. $h_t = A(\{(x_i, y_i, w_i)\})$
   2. évaluer α_t la qualité de h_t par rapport à $\{(x_i, y_i, w_i)\}$
   3. mettre à jour les poids w_i des exemples en fonction de h_t et α_t
3. renvoyer $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . h_t(x)\right)$

Compléments : formules utilisées par AdaBoost

Confiance dans une hypothèse faible

$$\alpha_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Mise à jour des poids des exemples

$$Z_t = \sum_i^n [w_i.\exp(-\alpha_t y_i h_t(x_i))]$$

$$w_{i+1} = \frac{w_i.\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Éléments d'explication des bons résultats d'AdaBoost

- Preuves du cadre PAC [Schapire, 1990, Freund, 1995] ;
- diminution de la variance
  [Breiman, 1996b, Freund and Schapire, 1998] ;
- maximisation des marges
  [Schapire et al., 1998, Koltchinskii and Panchenko, 2002, Rätsch and Warmuth, 2002, Rätsch and Warmuth, 2003, Grove and Schuurmans, 1998, Harries, 1999] ;
- méthodes d'ensemble et leveraging
  [Dietterich, 2000b, Dietterich, 2000a, Meir and Rätsch, 2003] ;
- algorithmes de type Monte Carlo
  [Esposito and Saitta, 2003, Esposito and Saitta, 2004] ;
- SVM : présentation de Olivier Bousquet sur la plate-forme AFIA 2005.

Difficulté de choisir un apprenant faible

Caractéristiques d'un apprenant à utiliser avec AdaBoost

- instabilité [Breiman, 1996b] ;
- ϵ proche de $\frac{1}{2}$? apprenant faisant des erreurs ?
- un apprenant qui fonctionne avec AdaBoost [Freund and Schapire, 1998].

Proposition : utiliser les moindres généralisés avec AdaBoost

- instables ;
- ne se trompent pas mais s'abstiennent ;
- définition de AdaBoostMG :
  - AdaBoost classique (calcul des α_t et mise à jour des w_i repris de [Schapire and Singer, 1999]) ;
  - avec moindres généralisés comme apprenant faible.

Calcul d'un moindre généralisé

Example

Généralisation de deux exemples

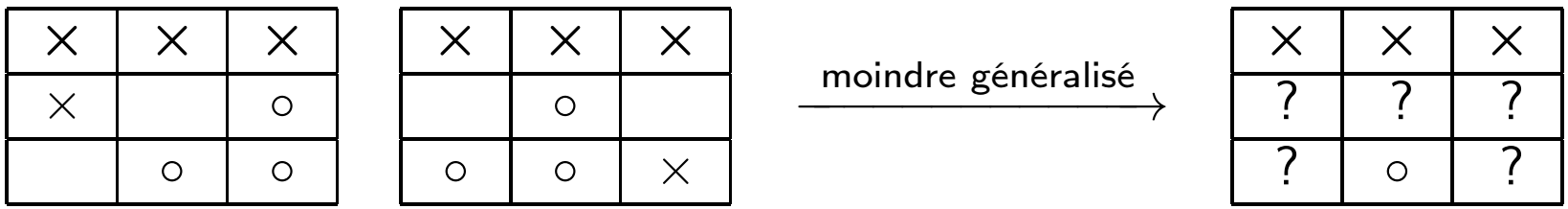
	âge	fumeur	sexe	classe
e_1	25	non	homme	positif
e_2	35	non	femme	positif
g_1	$[25, 35]$	non	?	positif

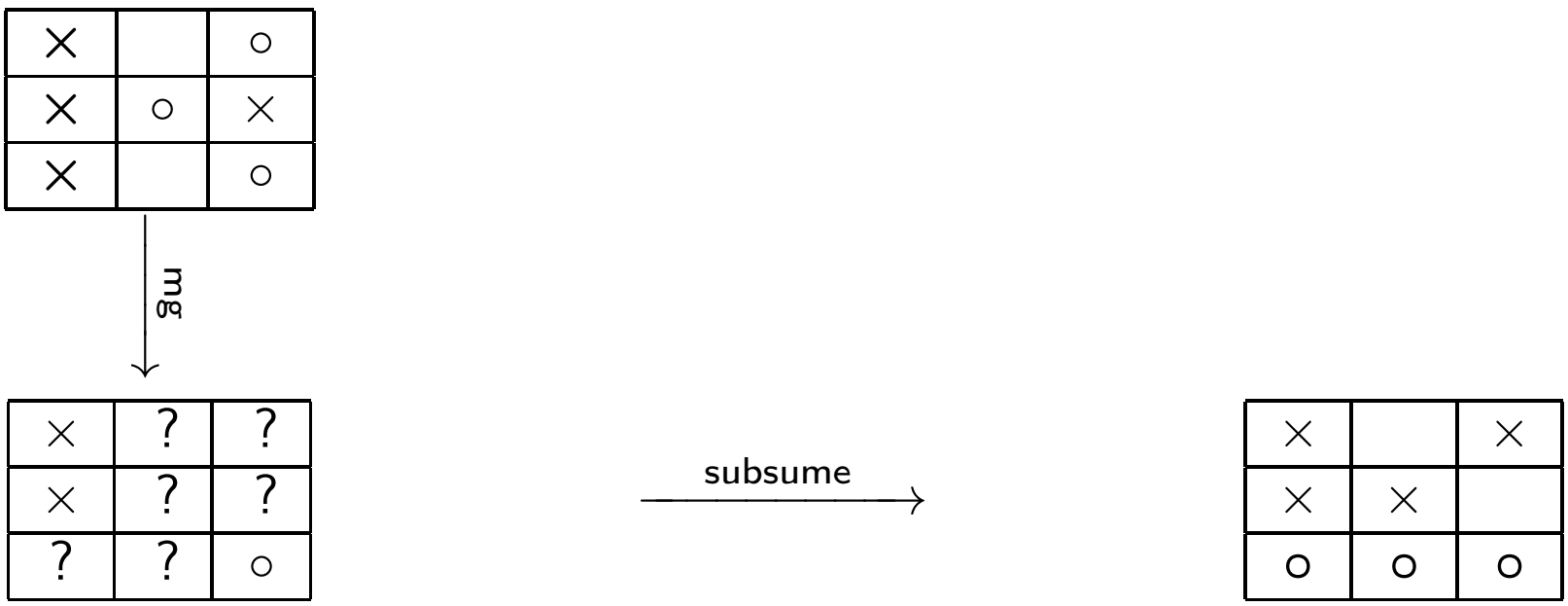
Généralisation d'un exemple et d'une hypothèse

	âge	fumeur	sexe	classe
g_1	$[25, 35]$	non	?	positif
e_3	30	oui	homme	positif
g_2	$[25, 35]$	?	?	positif
e_4	40	non	femme	positif
g_3	$[25, 40]$	?	?	positif

Moindre généralisé correct

Généralisation d'exemples d'une même classe sans couvrir aucun exemple d'une autre classe.



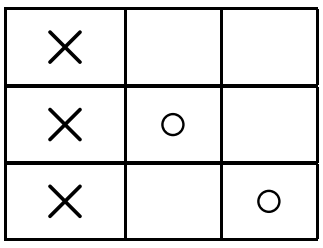


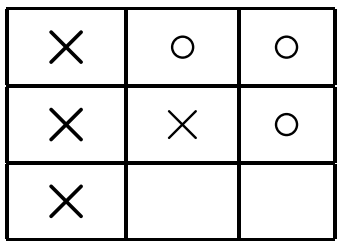
Calcul d'un moindre généralisé maximalement correct [Torre, 1999]

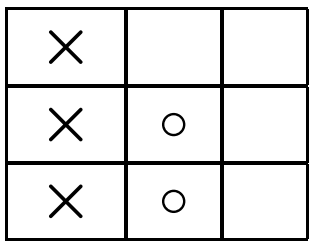
une graine et sa classe	exemples de la même classe	généralisation maximalement correcte
x_1, y_1	x_5 x_8 ~~x_2~~ $x_{14}\ldots$	$g_1 = \mathrm{mg}(\{x_1, x_5, x_8, x_{14}, \ldots\})$ $g_1 \rightarrow y_1$
x_2, y_2	~~x_{14}~~ x_3 ~~x_{10}~~ $x_{12}\ldots$	$g_2 = \mathrm{mg}(\{x_2, x_3, x_{12}, \ldots\})$ $g_2 \rightarrow y_2$

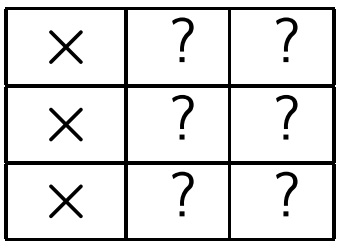
- Exemple : âge $\in [25, 40] \rightarrow$ positif ;
- instable : dépend de la graine et de l'ordre des exemples ;
- pour un exemple donné, un moindre généralisé correct conclut sur une unique classe (-1 ou $+1$) ou s'abstient (0).

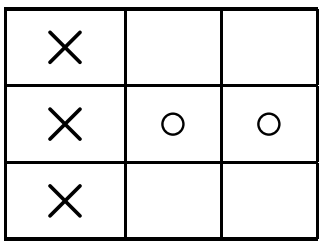
Paquets verrouillés et paquets condamnés

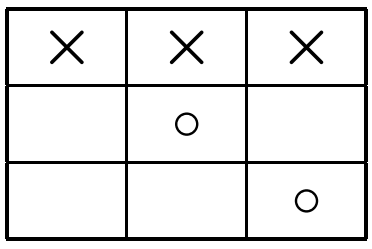


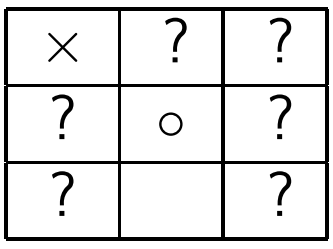






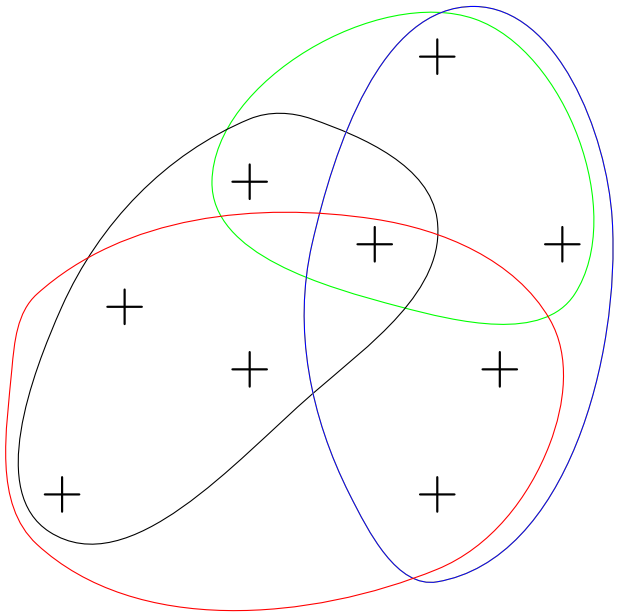


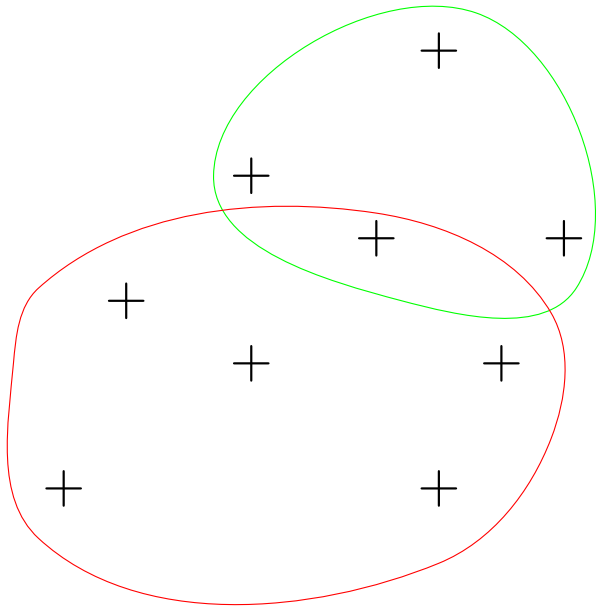




Algorithmes à base de moindres généralisés

- DLG [Webb and Agar, 1992] de la famille AQ [Michalski., 1983] ;
- RISE [Domingos, 1996]
- GloBo [Torre, 1999]
  1. calculer des moindre-généralisés en utilisant chaque exemple comme graine et les exemples de la même classe mélangés ;
  2. retenir les règles qui permettent une couverture minimale des exemples.





Définition d'un apprenant faible

Entrées : n exemples (x_i, y_i, w_i) avec leurs étiquettes et poids.
Sortie : g une généralisation maximalement correcte d'exemples de poids élevés.

- cible = classe choisie au hasard
- graine = l'exemple de la classe cible ayant le poids le plus fort
- $P = \{x_i | y_i = \text{cible}, x_i \neq \text{graine}\}$
- $N = \{x_i | y_i \neq \text{cible}\}$
- trier P par poids décroissants
- généraliser les exemples de P suivant cet ordre, en maintenant la correction vis-à-vis des exemples de N
- renvoyer l'hypothèse obtenue

Compléments : formules utilisées par Adaboost-MG

Confiance dans une hypothèse faible

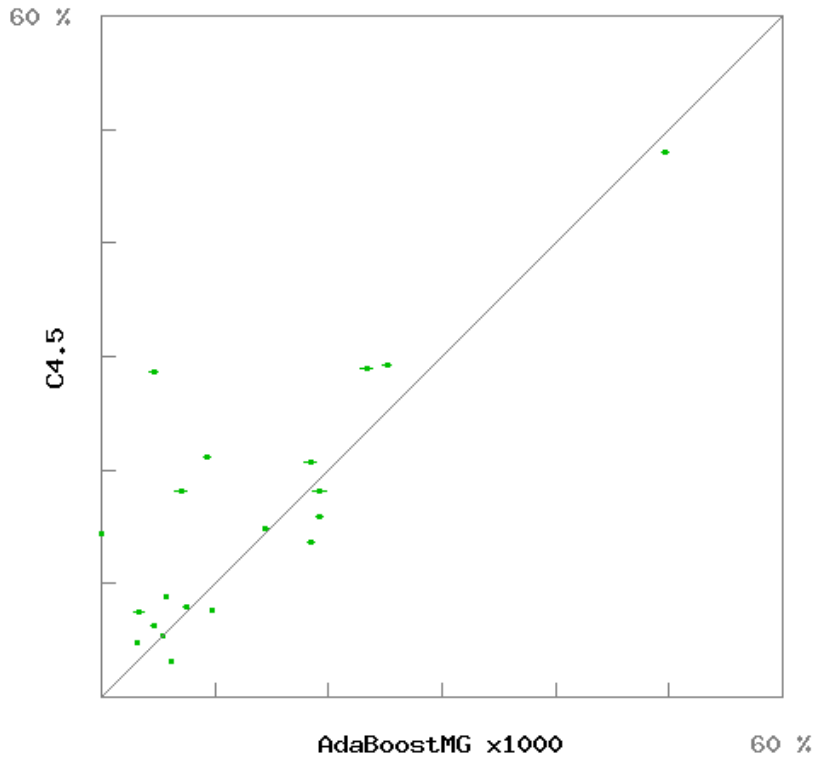
$$W_+ = \sum_{i:y_i=h_t(x_i)} w_i$$

$$\alpha_t = \frac{1}{2}\log\left(\frac{1+W_+}{1-W_+}\right)$$

Expérimentations : protocole

- 20 problèmes classiques de l'UCI [Blake and Merz, 1998] : audiology, breast-cancer, car, cmc, crx, dermatology, ecoli, glass, hepatitis, horse-colic, house-votes-84, ionosphere, iris, pima, promoters, sonar, tic-tac-toe, vowel, wine, zoo ;
- validation croisée 10 fois ;
- 10 exécutions pour les algorithmes stochastiques ;
- 1000 étapes de boosting ;
- datasets et résultats détaillés disponibles sur le web.

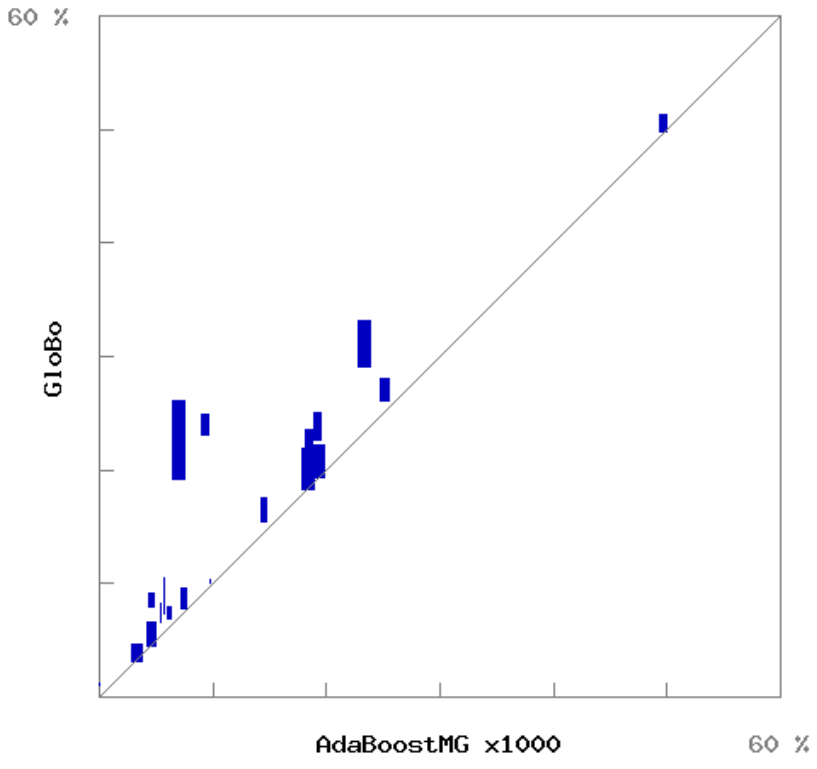
AdaBoostMG versus C4.5



Méthodes	Erreurs
C4.5	16.18 %
AdaBoostMG	12.71 %

AdaBoostMG est meilleur que C4.5.

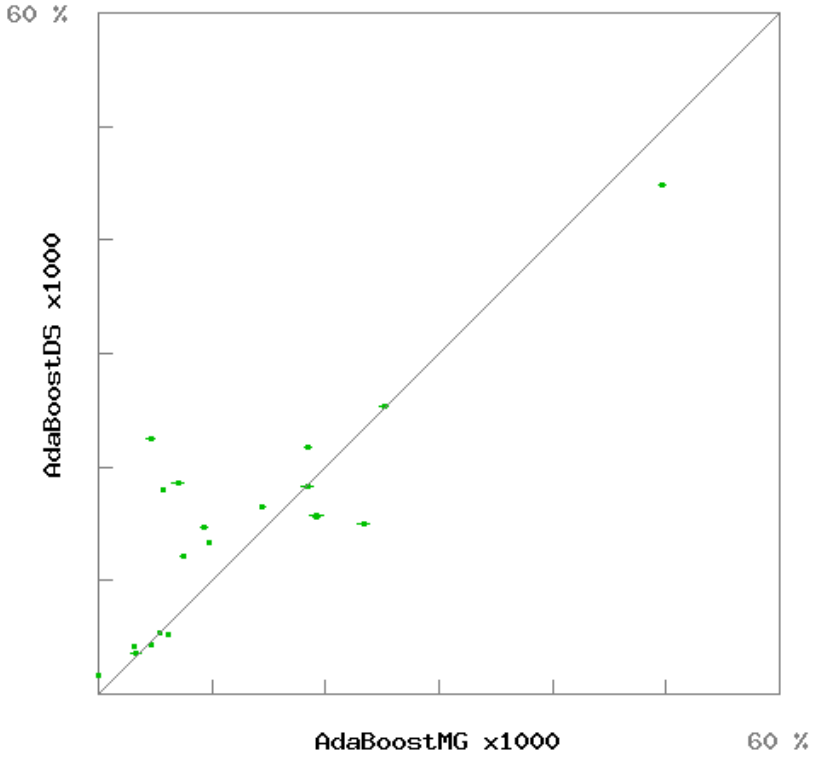
AdaBoostMG versus GloBo



Méthodes	Erreurs
C4.5	16.18 %
GloBo	16.23 %
AdaBoostMG	12.71 %

AdaBoostMG est meilleur que GloBo.

AdaBoostMG versus AdaBoostDS



Méthodes	Erreurs
C4.5	16.18 %
GloBo	16.23 %
AdaBoostDS	14.85 %
AdaBoostMG	12.71 %

Le calcul de moindres généralisés corrects est un meilleur apprenant faible pour AdaBoost que les Decision Stumps.

Sensibilité au nombre d'étapes de boosting

Étapes de boosting	AdaBoostDS	AdaBoostMG
100	14.41 %	15.57 %
1000	14.85 %	12.71 %

Constat

- Le boosting de moindres généralisés corrects s'améliore significativement avec le nombre d'étapes de boosting.
- Malheureusement, le calcul de moindres généralisés est coûteux.

Proposition

Distribuer les calculs sur différentes machines.

- produire les hypothèses indépendamment les unes des autres ;
- affecter des poids α_t aux hypothèses a posteriori.

Évaluation de nouveaux poids

Poids candidats

- couverture : le nombre d'exemples couverts par l'hypothèse
- fréquence : le nombre d'apparitions de l'hypothèse
- uniforme : 1 quelle que soit l'hypothèse

Résultats expérimentaux (les votants sont produits par AdaBoostMG)

Poids	adaboost	couverture	fréquence	uniforme
Erreur	12.71 %	13.74 %	15.40 %	14.51 %

On dispose de poids raisonnables à attribuer aux hypothèses produites indépendamment.

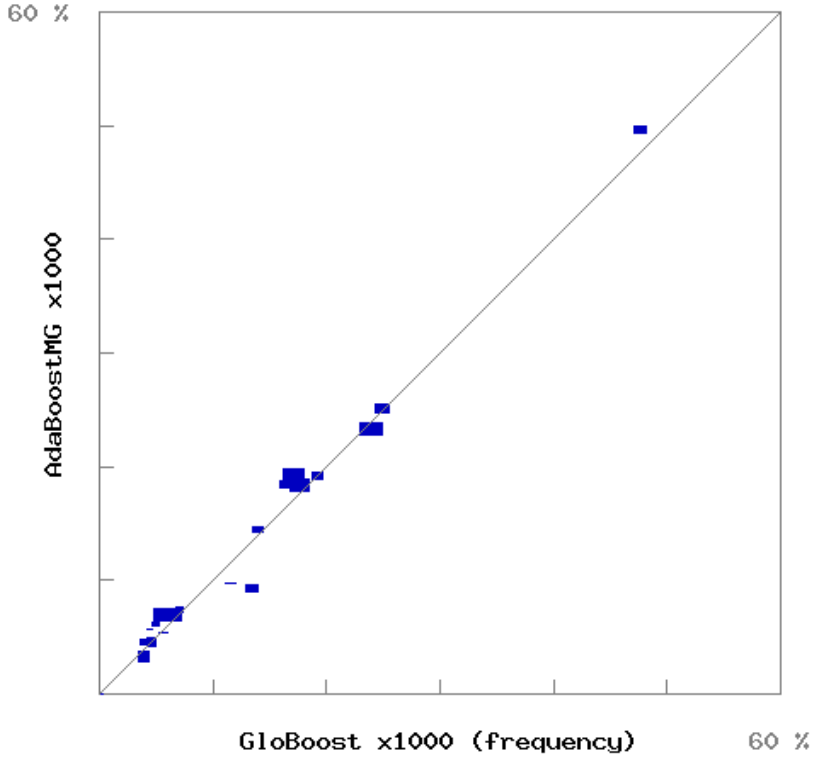
GloBoost

Entrées : n exemples (x_i, y_i) avec étiquettes ; T un nombre d'itérations.

1. pour t allant de 1 à T
   1. cible = classe choisie au hasard
   2. graine = un exemple de la classe cible choisi au hasard
   3. $P = \{x_i | y_i = \text{cible}, x_i \neq \text{graine}\}$
   4. $N = \{x_i | y_i \neq \text{cible}\}$
   5. mélanger P aléatoirement
   6. généraliser les exemples de P suivant cet ordre aléatoire, en maintenant la correction vis-à-vis des exemples de N pour obtenir h_t
2. fixer le poids α_t de chaque hypothèse produite
3. retourner $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . h_t(x)\right)$

Évaluation de GloBoost et des poids

	GloBoost			
Étapes	couverture	fréquence	uniforme	AdaBoostMG
100	16.08 %	15.76 %	15.90 %	15.57 %
1000	13.18 %	12.50 %	13.21 %	12.71 %



On peut produire les moindres généralisés aléatoirement et fixer les poids ensuite.

Confrontations

Pour chaque couple d'apprenant, on compte combien de fois l'un l'emporte sur l'autre.

	C4.5	**GloBo**	**ABDS**	**ABMG**	**GloBoost**	**Bilan**
C4.5	-	$14 \cdot 6$	$7 \cdot 12$	$6 \cdot 13$	$5 \cdot 15$	$32 \cdot 46$
GloBo	$6 \cdot 14$	-	$7 \cdot 13$	$0 \cdot 20$	$1 \cdot 19$	$14 \cdot 66$
ABDS	$12 \cdot 7$	$13 \cdot 7$	-	$7 \cdot 12$	$5 \cdot 15$	$37 \cdot 41$
ABMG	$13 \cdot 6$	$20 \cdot 0$	$12 \cdot 7$	-	$7 \cdot 13$	$52 \cdot 26$
GloBoost	$15 \cdot 5$	$19 \cdot 1$	$15 \cdot 5$	$13 \cdot 7$	-	$62 \cdot 18$

Les moindres généralisés fonctionnent avec AdaBoost

On a observé de bonnes performances lorsque le calcul de moindres généralisés corrects est utilisé comme apprenant faible de AdaBoost.

Éléments d'explication

- les moindres généralisés corrects sont à la fois fortement guidés par les données et instables ;
- les moindres généralisés corrects ne font pas d'erreur, seulement des abstentions ;
- une preuve d'apprenabilité pour les rectangles en deux dimensions utilisant les moindres généralisés comme apprenant faible [Kearns and Vazirani, 1994].

GloBoost obtient des résultats comparables à AdaBoostMG

On a des performances équivalentes à AdaBoostMG en générant les moindres généralisés corrects aléatoirement et en fixant des poids aux hypothèses a posteriori.

Éléments d'explication

- plus facile avec des moindres généralisés corrects ;
- réduction de la variance [Breiman, 1996b] ;
- optimisation des marges [Schapire et al., 1997] ;
- proche des méthodes Monte Carlo [Esposito and Saitta, 2003].

Travaux liés

Travaux de Dietterich [Dietterich, 2000b, Dietterich, 2000a]

- production d'arbres de manière aléatoire ;
- meilleur que le bagging si peu de données ;
- meilleur que le boosting en présence de bruit.

Nouvel éclairage sur le co-training [Abney, 2002]

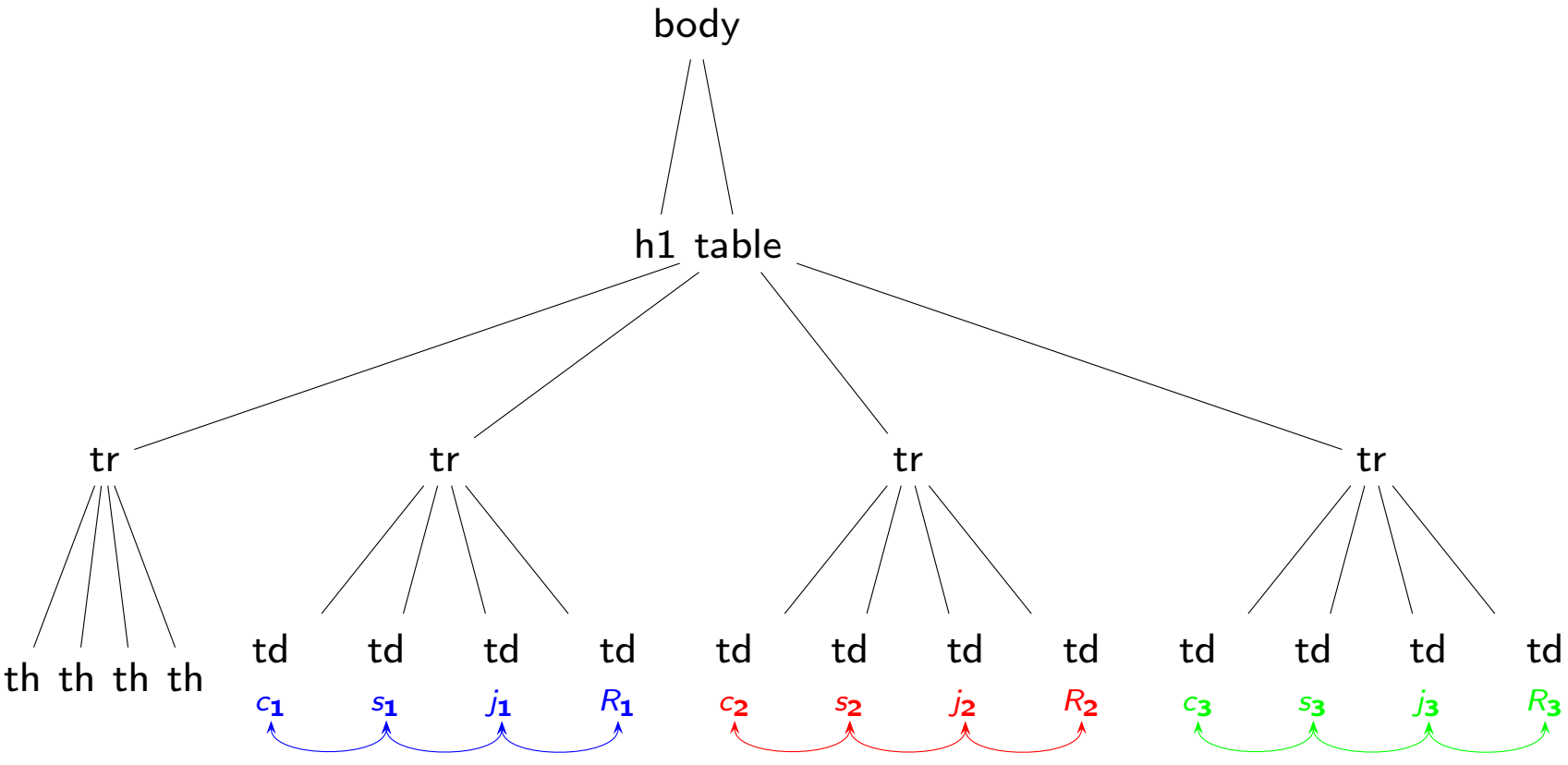
- co-training : utilisation de deux vues sur un même problème [Blum and Mitchell, 1998] avec des données non étiquetées ;
- gain si les deux vues sont indépendantes ;
- [Abney, 2002] : relâchement de l'indépendance, on peut avoir le gain en utilisant deux fois la même vues ;
- idée : maximiser l'accord entre les deux vues.

Application à l'extraction d'information...

- Tâches d'extraction dans les documents XHTML ;
- codage en attributs-valeurs ;
- classes déséquilibrées ;
- pas de bruit a priori ;
- données disponibles.

Club	Saison	Journée	Rang
Marseille	2002-2003	17	5
Monaco	2002-2003	31	1
Bordeaux	2001-2002	11	6

... vue comme un problème supervisé [Gilleron et al., 2005]



Perspectives

Améliorer les apprenants faibles

- améliorer la prise en compte des poids dans l'apprenant faible à base de moindres généralisés corrects ;
- poursuivre la parallélisation en distribuant les données ;
- accélérer les apprenants faibles à base de moindres généralisés (ajouter plus de stochastique, considérer moins d'exemples) ;
- booster des théories complètes (GloBo ou RISE [Wiratunga et al., 2002]).

Comprendre les modes de production

- trouver une explication théorique aux bons résultats de la production aléatoire ;
- définir de meilleurs poids calculables a posteriori pour AdaBoost et pour GloBoost.


Abney, S. (2002).
Bootstrapping.
In Charniak, E. and Lin, D., editors, Proceedings of the Fortieth Annual Meeting of the Association for Computational Linguistics, pages 360–367. Morgan Kaufmann Publishers, San Francisco.


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


Blum, A. and Mitchell, T. (1998).
Combining labeled and unlabeled data with co-training.
In COLT : Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers.


Breiman, L. (1996a).
Bagging predictors.
Machine Learning, 24(2) :123–140.


Breiman, L. (1996b).
Bias, variance, and arcing classifiers.
Technical Report 460, Statistics Department, University of California.


Breiman, L. (2001).
Random forests.
Machine Learning, 45(1) :5–32.


Dietterich, T. G. (2000a).
Ensemble methods in machine learning.
In Kittler, J. and Roli, F., editors, First International Workshop on Multiple Classifier Systems, volume 1857 of Lecture Notes in Computer Science, pages 1–15. Springer-Verlag.


Dietterich, T. G. (2000b).
An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization.
Machine Learning, 40(2) :139–158.

Domingos, P. (1996).
Unifying instance-based and rule-based induction.
Machine Learning, 24(2) :141–168.

Esposito, R. and Saitta, L. (2003).
Monte Carlo Theory as an Explanation of Bagging and Boosting.
In Gottlob, G. and Walsh, T., editors, Proceeding of the Eighteenth International Joint Conference on Artificial Intelligence, pages 499–504. Morgan Kaufman.

Esposito, R. and Saitta, L. (2004).
A monte carlo analysis of ensemble classification.
In Greiner, R. and Schuurmans, D., editors, Proceedings of the twenty-first International Conference on Machine Learning, pages 265–272, Banff, Canada. ACM Press, New York, NY.

Freund, Y. (1995).
Boosting a weak learning algorithm by majority.

Information and Computation, 121(2) :256–285.


Freund, Y. and Schapire, R. E. (1998).
Discussion of the paper Arcing Classifiers by Leo Breiman.
The Annals of Statistics, 26 :824–832.


Gilleron, R., Marty, P., Tommasi, M., and Torre, F. (2005).
Statistical classification for wrapper induction.
Dagstuhl Seminar : Machine Learning for the Semantic Web.


Grove, A. J. and Schuurmans, D. (1998).
Boosting in the limit : Maximizing the margin of learned ensembles.
In AAAI/IAAI, pages 692–699.


Harries, M. (1999).
Boosting a strong learner : evidence against the minimum margin.
In Proc. 16th International Conf. on Machine Learning, pages 171–180. Morgan Kaufmann, San Francisco, CA.


Kearns, M. and Valiant, L. G. (1989).
Cryptographic limitations on learning Boolean formulae and finite automata.
In Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pages 433–444.


Kearns, M. J. and Vazirani, U. V. (1994).
An Introduction to Computational Learning Theory.
MIT Press.


Koltchinskii, V. and Panchenko, D. (2002).
Empirical margin distributions and bounding the generalization error of combined classifiers.
Annals of Statistics, 30(1) :1–50.


Meir, R. and Rätsch, G. (2003).
An introduction to boosting and leveraging.
In Mendelson, S. and Smola, A., editors, Advanced Lectures on Machine Learning, number 2600 in LNAI, pages 119–184. Springer-Verlag.


Michalski., R. S. (1983).
A Theory and Methodology of Inductive Learning, pages 83–134.
Springer-Verlag.


Rätsch, G. and Warmuth, M. (2002).
Maximizing the margin with boosting.
In Kivinen, J. and Sloan, R. H., editors, Proceedings of the Annual Conference on Computational Learning Theory (COLT), volume 2375 of Lecture Notes in Computer Science, pages 334–350. Springer-Verlag.


Rätsch, G. and Warmuth, M. K. (2003).
Efficient margin maximizing with boosting.
submitted to Journal of Machine Learning Research (JMLR).


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


Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1997).
Boosting the margin : a new explanation for the effectiveness of voting methods.
In Proc. 14th International Conference on Machine Learning (ICML), pages 322–330. Morgan Kaufmann.


Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998).
Boosting the margin : a new explanation for the effectiveness of voting methods.
The Annals of Statistics, 26 :1651–1686.


Schapire, R. E. and Singer, Y. (1999).
Improved boosting algorithms using confidence-rated predictions.
Machine Learning, 37(3) :297–336.


Torre, F. (1999).

GloBo : un algorithme stochastique pour l'apprentissage supervisé et non-supervisé.
In Sebag, M., editor, Actes de la Première Conférence d'Apprentissage, pages 161–168.


Valiant, L. G. (1984).
A theory of the learnable.
Communications of the ACM, 27 :1134–1142.


Webb, G. I. and Agar, J. W. M. (1992).
Inducing diagnostic rules for glomerular disease with the DLG machine learning algorithm.
Artificial Intelligence in Medicine, 4 :419–430.


Wiratunga, N., Craw, S., and Rowe, R. (2002).
Learning to adapt for case-based design.
In Craw, S. and Preece, A. D., editors, Proceedings of the 6th European Conference on Advances in Case-Based Reasoning, volume 2416 of Lecture Notes in Computer Science, pages 421–435. Springer-Verlag.