# GloBoost :

## Boosting de moindres généralisés

**Fabien Torre**

**GRAppA (EA 3588)**
**Université Charles de Gaulle - Lille III**

**Conférence d'Apprentissage CAp'04**
**14 juin 2004**

1. Rappels sur le boosting et sur AdaBoost
2. Présentation des moindres généralisés corrects
3. AdaBoostMG : AdaBoost et moindres généralisés

   $\Rightarrow$ le calcul de moindres généralisés corrects est un bon apprenant faible pour AdaBoost

4. GloBoost

   $\Rightarrow$ il est possible de produire aléatoirement les moindres généralisés et de leur fixer des poids a posteriori sans perte de performance

## Contexte

- 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 $\epsilon$ de $h$ comme la probabilité de trouver un exemple de $\mathcal{X}$ sur lequel $h$ et le concept cible sont en désaccord.

## Boosting

- dans le cadre PAC, un apprenant faible ($\epsilon < \frac{1}{2}$) peut être transformé en un apprenant fort ($\epsilon$ arbitrairement proche de $0$) en temps polynomial [Schapire, 1990, Freund, 1995] ;
- AdaBoost : algorithme boostant un apprenant faible.

Entrées :

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

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

## Caractéristiques d'un apprenant à utiliser avec AdaBoost

- instabilité [Breiman, 1996] ;
- $\epsilon$ 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 $\alpha_t$ et mise à jour des $w_i$ repris de [Schapire and Singer, 1999]) ;
  - avec moindres généralisés comme apprenant faible.

## Généralisation de deux exemples

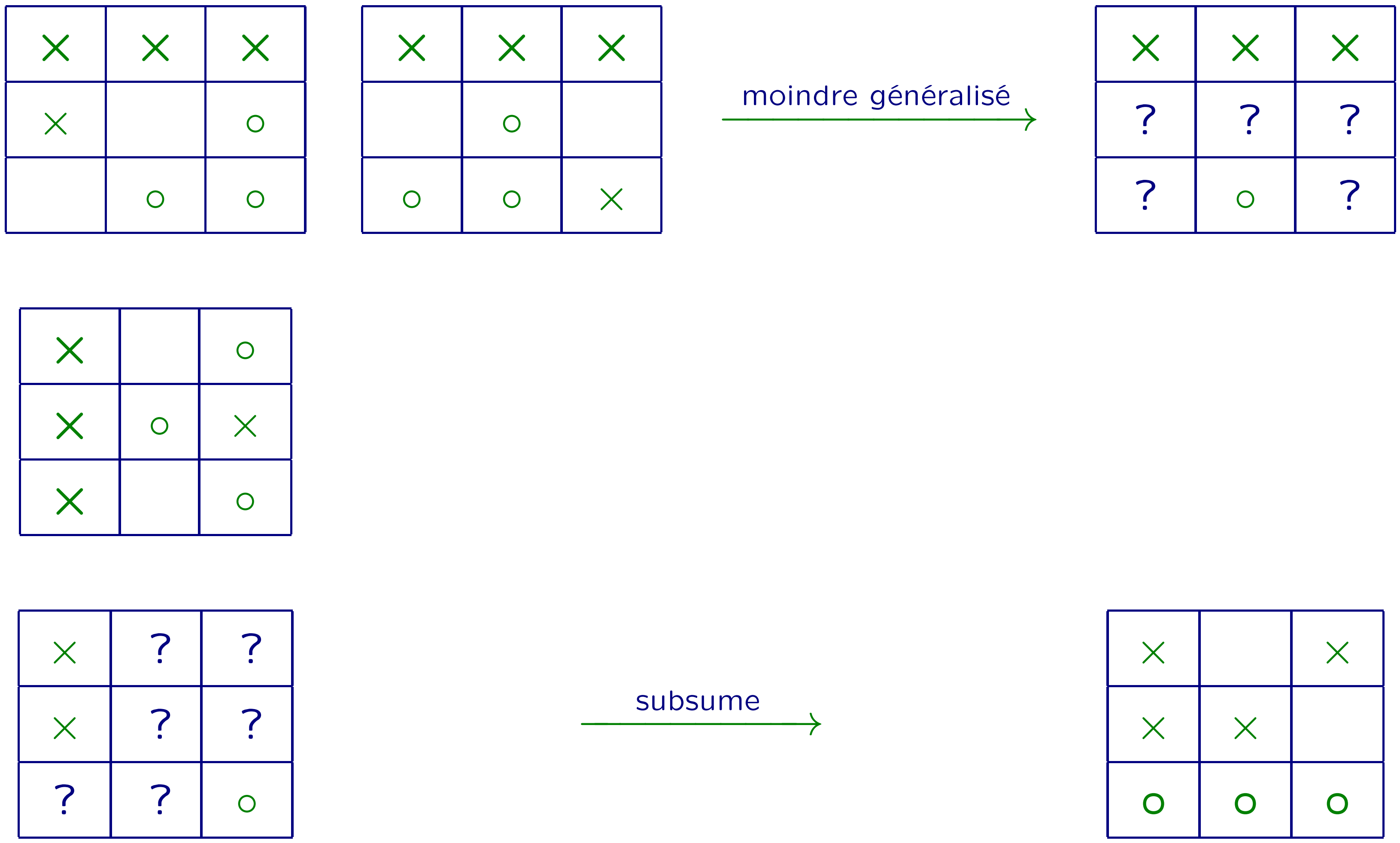
| | âge | fumeur | sexe | classe |
|---|---|---|---|---|
| $\mathbf{e_1}$ | 25 | non | homme | positif |
| $\mathbf{e_2}$ | 35 | non | femme | positif |
| $\mathbf{g_1}$ | $[25, 35]$ | non | ? | positif |

## Généralisation d'un exemple et d'une hypothèse

| | âge | fumeur | sexe | classe |
|---|---|---|---|---|
| $\mathbf{g_1}$ | $[25, 35]$ | non | ? | positif |
| $\mathbf{e_3}$ | 30 | oui | homme | positif |
| $\mathbf{g_2}$ | $[25, 35]$ | ? | ? | positif |
| $\mathbf{e_4}$ | 40 | non | femme | positif |
| $\mathbf{g_3}$ | $[25, 40]$ | ? | ? | 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 |
|---|---|---|
| $x_1, y_1$ | $x_5\ x_8\ \cancel{x_2}\ x_{14} \ldots$ | $g_1 = \mathsf{mg}(\{x_1, x_5, x_8, x_{14}, \ldots\})$<br>$g_1 \rightarrow y_1$ |
| $x_2, y_2$ | $\cancel{x_{14}}\ x_3\ \cancel{x_{10}}\ x_{12} \ldots$ | $g_2 = \mathsf{mg}(\{x_2, x_3, x_{12}, \ldots\})$<br>$g_2 \rightarrow y_2$ |

– Exemple : âge $\in [25, 40] \rightarrow$ positif ;

– instable : dépend de la graine et de l'ordre des exemples ;

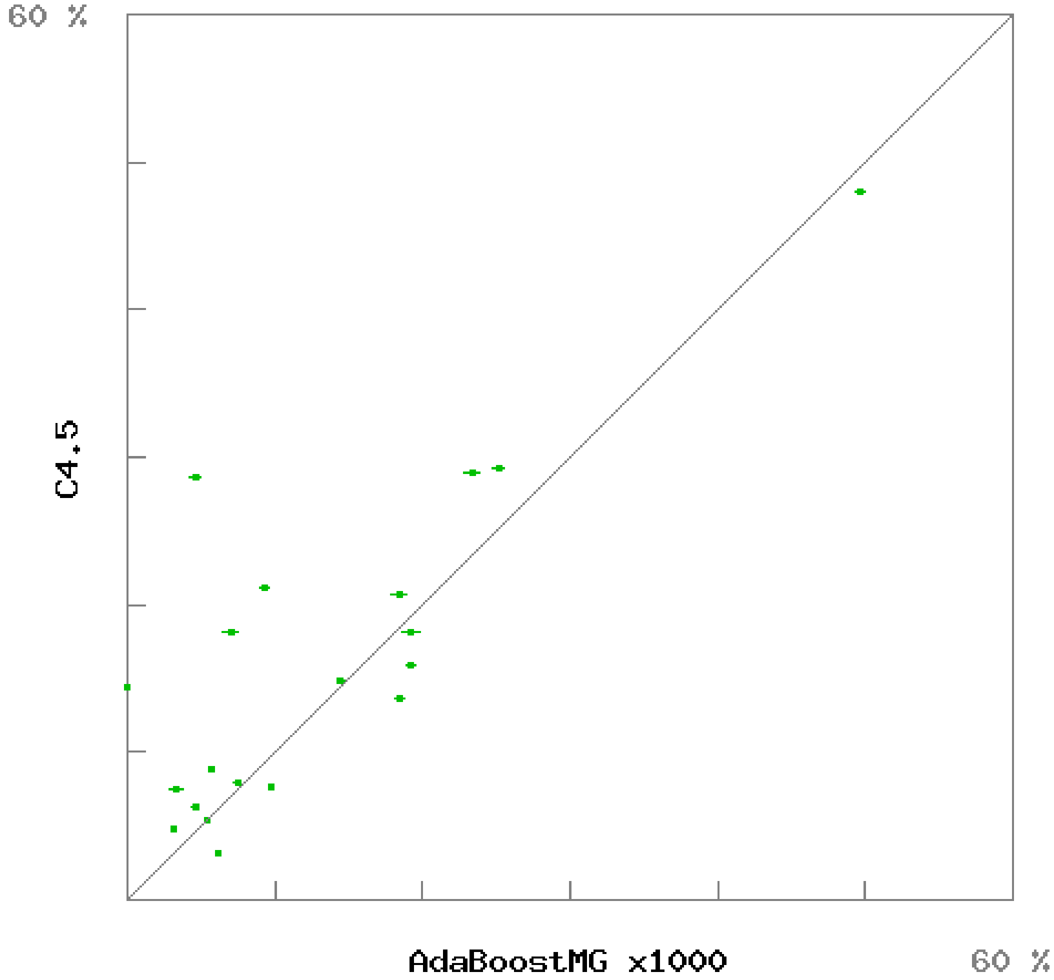
– pour un nouvel exemple en entrée, un moindre généralisé correct conclut sur une unique classe ($-1$ ou $+1$) ou s'abstient ($0$).

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.

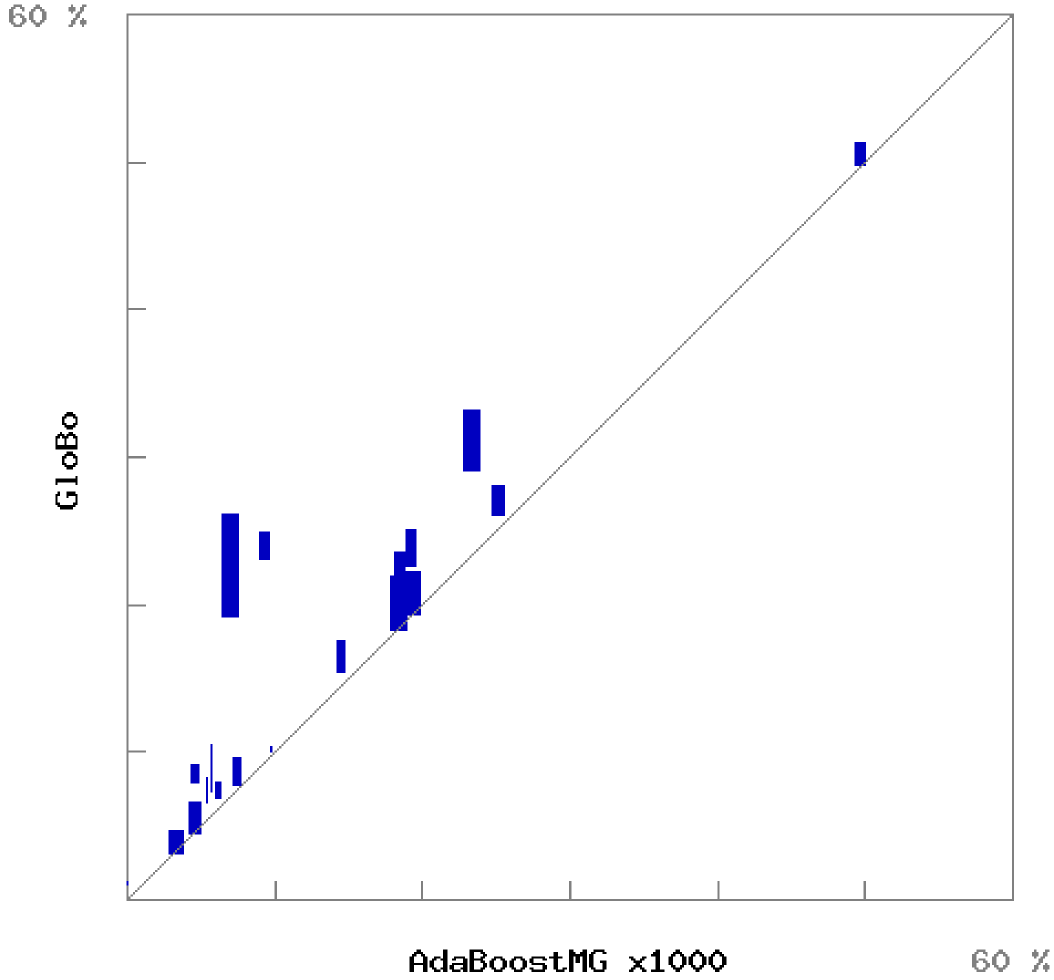
- cible = classe choisie au hasard
- graine = l'exemple de la classe cible ayant le poids le plus fort
- $P = \{x_i | y_i = \text{cible}, x_i \neq \text{graine}\}$
- $N = \{x_i | y_i \neq \text{cible}\}$
- trier $P$ par poids décroissants
- généraliser les exemples de $P$ suivant cet ordre, en maintenant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

- 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.



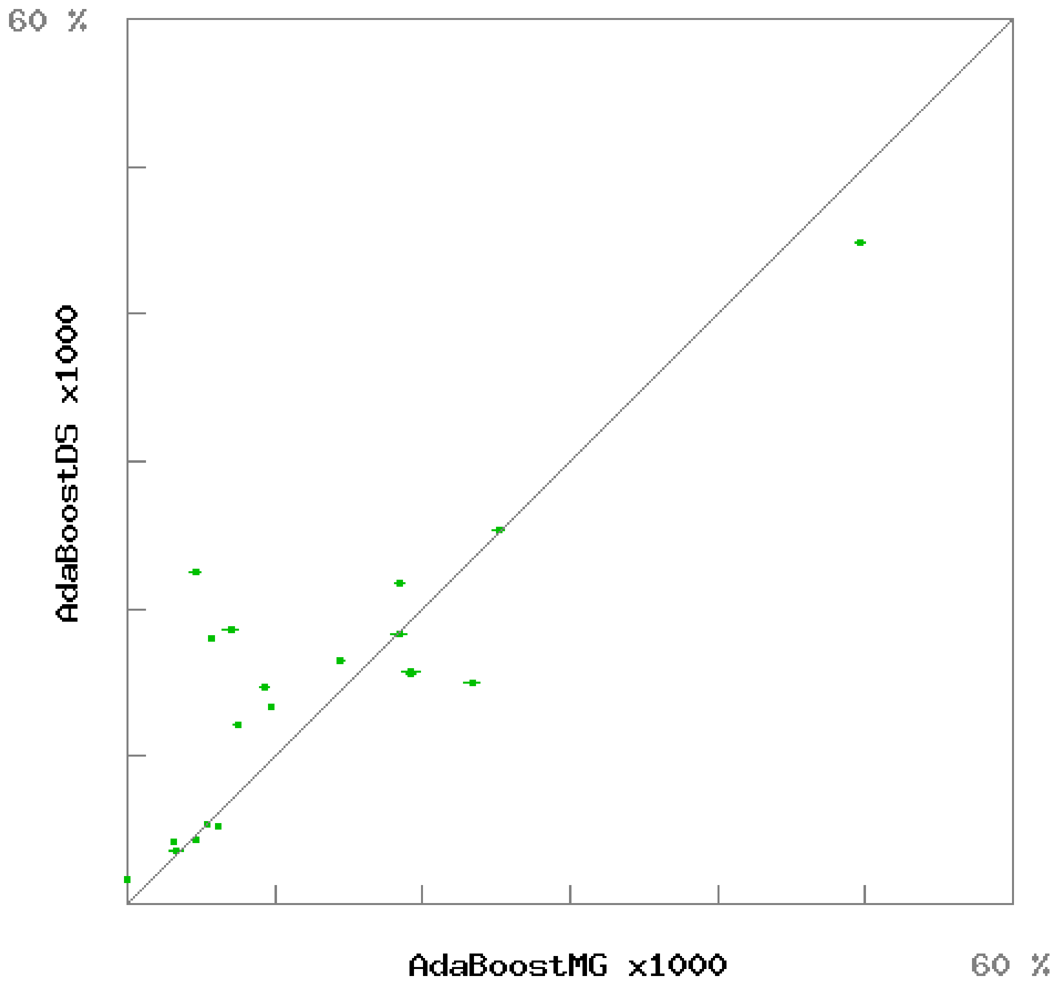

| Méthodes | Erreurs |
|---|---|
| C4.5 | 16.18 % |
| AdaBoostMG | 12.71 % |

⇒ AdaBoostMG est meilleur que C4.5.




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

⇒ AdaBoostMG est meilleur que GloBo.




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

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

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

$\Rightarrow$ 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 $\alpha_t$ aux hypothèses a posteriori.

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)

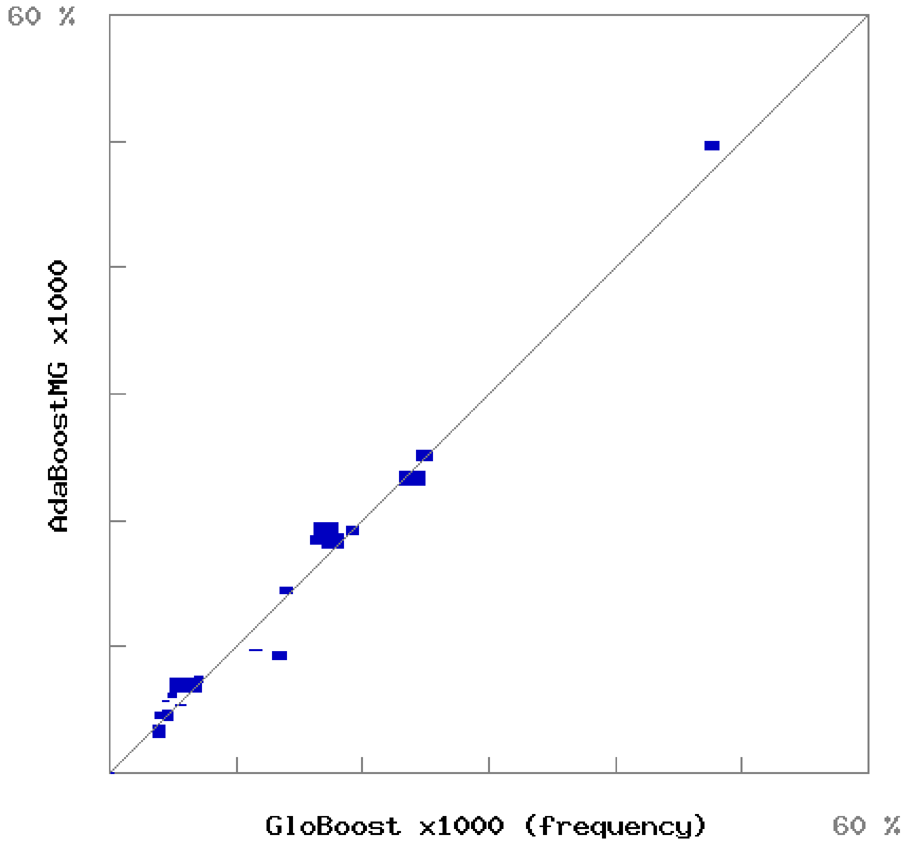
| Poids | adaboost | couverture | fréquence | uniforme |
|---|---|---|---|---|
| Erreur | 12.71 % | 13.74 % | 15.40 % | 14.51 % |

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

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

1. pour $t$ allant de 1 à $T$
   a) cible $=$ classe choisie au hasard
   b) graine $=$ un exemple de la classe cible choisi au hasard
   c) $P = \{x_i | y_i = \text{cible}, x_i \neq \text{graine}\}$
   d) $N = \{x_i | y_i \neq \text{cible}\}$
   e) mélanger $P$ aléatoirement
   f) 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 $\alpha_t$ de chaque hypothèse produite
3. retourner $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . h_t(x)\right)$

| | 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.

## 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, 1996] ;
- optimisation des marges [Schapire et al., 1997] ;
- proche des méthodes Monte Carlo [Esposito and Saitta, 2003].

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).

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.

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

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

[Esposito and Saitta, 2003] 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.

[Freund, 1995] Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2) :256–285.

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

[Kearns and Valiant, 1989] 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 and Vazirani, 1994] Kearns, M. J. and Vazirani, U. V. (1994). An Introduction to Computational Learning Theory. MIT Press.

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

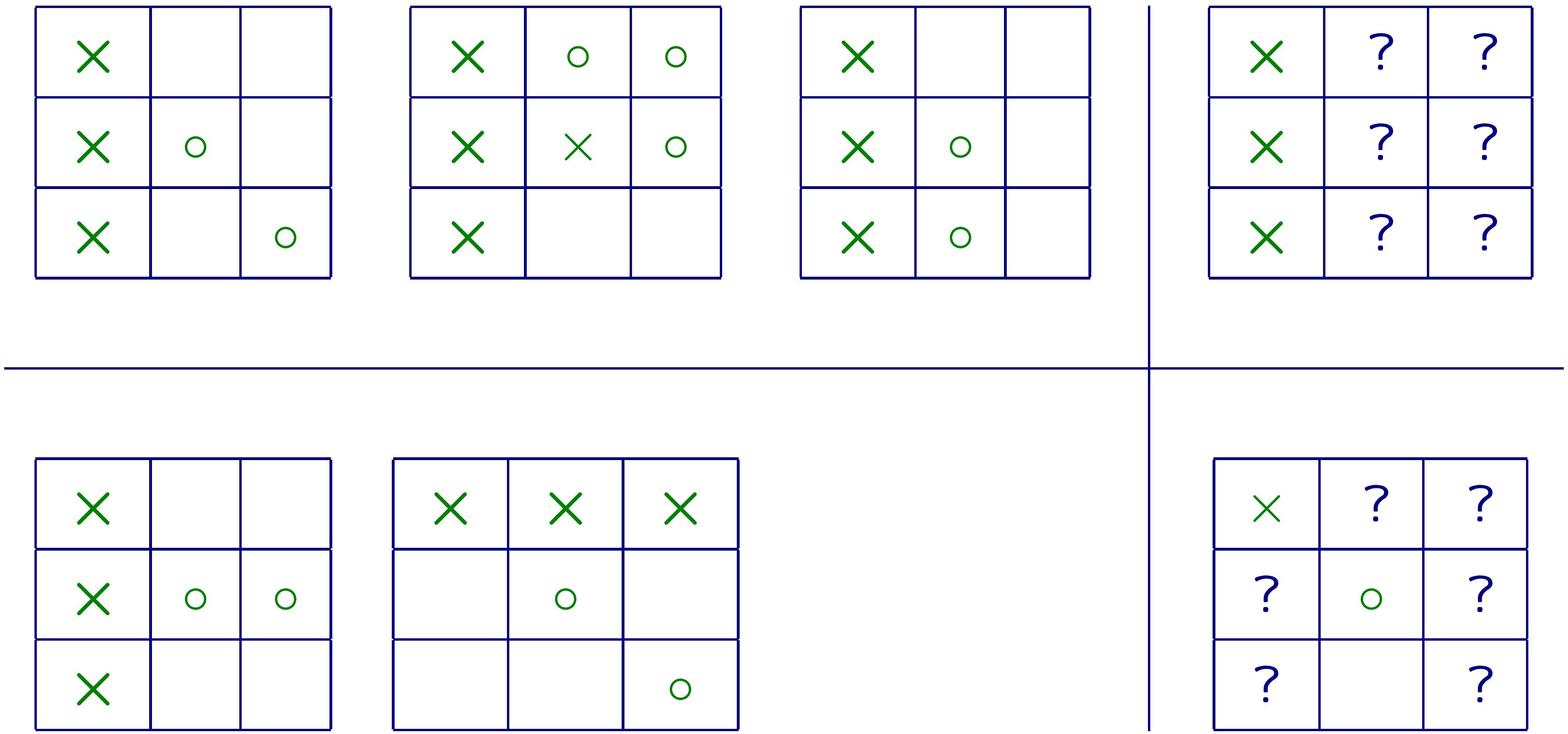
[Schapire et al., 1997] Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1997). Boosting the margin : a new explanation for the effectiveness of voting methods. In Proc. 14th International Conference on Machine Learning (ICML), pages 322–330. Morgan Kaufmann.

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

[Torre, 1999] 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, 1984] Valiant, L. G. (1984). A theory of the learnable. Communications of the ACM, pages 1134–1142.

# Paquets verrouillés et paquets condamnés



## Bilan

- les moindres généralisés fonctionnent avec AdaBoost ;
- GloBoost permet des performances équivalentes.

## Pourquoi ça marche ?

- à la fois fortement guidé par les données et instable ;
- une preuve d'apprenabilité [Kearns and Vazirani, 1994].

## Perspectives

- explication théorique ;
- distribuer les données ;
- trouver de meilleurs poids ;
- accélérer les apprenants faibles à base de moindres généralisés.

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

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

$L$ 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

Étant donnés

- $E$ un échantillon suffisant d'exemples étiquetés,
- $T$ un nombre d'étapes de boosting suffisant,
- un apprenant faible $A$.

1. donner un poids de $\frac{1}{|E|}$ à chaque exemple disponible
2. pour $t$ allant de 1 à $T$

   a) utiliser $A$ sur la distribution actuelle pour obtenir $h_t$

   b) évaluer $\alpha_t$ la qualité de $h_t$

   c) mettre à jour les poids des exemples en fonction de $h_t$

3. renvoyer $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t . h_t(x)\right)$