



## Combiner des apprenants: le boosting

A. Cornuéjols  
IAA

(basé sur Rob Schapire's IJCAI'99 talk)

### Types d'experts

- Un seul expert sur l'ensemble de  $\mathcal{X}$
- Un expert par sous-régions de  $\mathcal{X}$  (e.g. arbres de décisions)
- Plusieurs experts, tous sur l'ensemble de  $\mathcal{X}$
- Plusieurs experts spécialisés sur des sous-régions de  $\mathcal{X}$

Boosting 2

### Types d'experts

Domaine(s) \ Expert(s)	Sur $\mathcal{X}$	Sous-régions de $\mathcal{X}$
Un expert	<i>Classique</i>	<i>Arbres de décisions</i>
Ensemble d'experts	<i>Boosting</i>	

Boosting 3

### Types d'apprentissage

- Chaque expert apprend sur l'ensemble d'apprentissage  $S_m$
- Utilisation de différents attributs (e.g. *arbres de décisions*, *co-learning*)
- Utilisation de différents ensembles d'apprentissage pour chaque expert (e.g. *boosting*)
- Combinaison de tout cela

Boosting 4

## Prédiction de courses hippiques



Boosting 5

## Comment gagner aux courses ?

- On interroge des parieurs professionnels
- Supposons:
  - Que les professionnels ne puissent pas fournir une règle de pari simple et performante
  - Mais que face à des cas de courses, ils puissent toujours produire des règles un peu meilleures que le hasard
- Pouvons-nous devenir riche?

Boosting 6

## Idée

- Demander à l'expert **des heuristiques**
- Recueillir un ensemble de cas pour lesquels ces heuristiques échouent (**cas difficiles**)
- Ré-interroger l'expert pour qu'il fournisse des **heuristiques pour les cas difficiles**
- Et ainsi de suite...
  
- **Combiner** toutes ces heuristiques
- Un expert peut aussi bien être un **algorithme d'apprentissage peu performant** (*weak learner*)

Boosting 7

## Questions

- Comment choisir les courses à chaque étape?
  - Se concentrer sur les courses les plus "difficiles" (celles sur lesquelles les heuristiques précédentes sont les moins performantes)
- Comment combiner les heuristiques (règles de prédiction) en une seule règle de prédiction ?
  - Prendre une vote (pondéré) majoritaire de ces règles

Boosting 8

## Boosting

- **boosting** = méthode générale pour convertir des règles de prédiction peu performantes en une règle de prédiction (très) performante
- Plus précisément :
  - Étant donné un algorithme d'apprentissage "faible" qui peut toujours retourner une hypothèse de taux d'erreur  $\leq 1/2 - \gamma$
  - Un algorithme de boosting peut construire (de manière prouvée) une règle de décision (hypothèse) de taux d'erreur  $\leq \epsilon$

Boosting 9

## Illustration

- Soit  $X$  un espace d'entrée à 10 dimensions
- Les attributs sont indépendants et de distribution gaussienne
- L'étiquette est définie par :

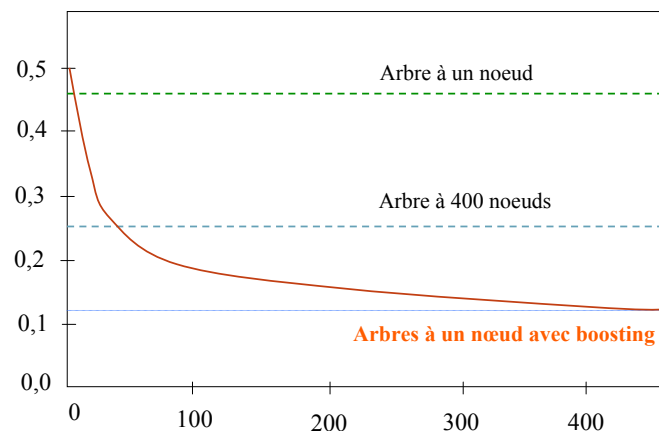
$$u = \begin{cases} 1 & \text{si } \sum_{j=1,10} x_j^2 > \chi_{10}^2(0,5) \\ -1 & \text{sinon} \end{cases}$$

avec :  $\chi_{10}^2(0,5) = 9,34$

- 2000 exemples d'apprentissages (1000+;1000-)
- 10000 exemples de test
- Apprentissage d'arbres de décision

Boosting 10

## Illustration (cont.)



Boosting 11

## Plan

- Introduction au boosting (AdaBoost)
  - Expériences
  - Conclusion
- 
- Analyse de l'erreur en apprentissage
  - Analyse de l'erreur en généralisation basée sur la théorie des marges
  - Extensions
  - Bibliographie

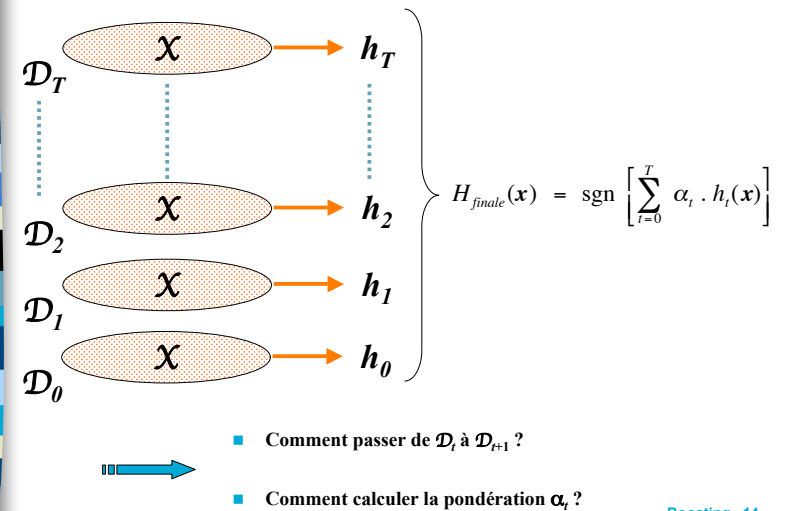
Boosting 12

## Boosting : vue formelle

- Étant donné l'échantillon d'apprentissage  $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- $y_i \in \{-1, +1\}$  étiquette de l'exemple  $x_i \in S$
- Pour  $t = 1, \dots, T$ :
  - Construire la distribution  $D_t$  sur  $\{1, \dots, m\}$
  - Trouver l'hypothèse faible ("heuristique")
 
$$h_t : S \rightarrow \{-1, +1\}$$
 avec erreur petite  $\epsilon_t$  sur  $D_t$ :  $\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$
- Retourner l'hypothèse finale  $h_{\text{final}}$

Boosting 13

## Le principe général



Boosting 14

## AdaBoost [Freund&Schapire '97]

- construire  $D_t$ : 
$$D_t(i) = \frac{1}{m}$$
- Étant donnée  $D_t$  et  $h_t$ :
 
$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

$$= \frac{D_t}{Z_t} \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(x_i))$$

où:  $Z_t$  = constante de normalisation

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) > 0$$
- Hypothèse finale :
 
$$H_{\text{final}}(x) = \text{sgn} \left( \sum_t \alpha_t h_t(x) \right)$$

Boosting 15

## AdaBoost en plus gros

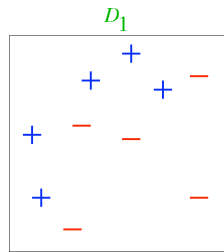
$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) > 0$$

$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

$$H_{\text{final}}(x) = \text{sgn} \left( \sum_t \alpha_t h_t(x) \right)$$

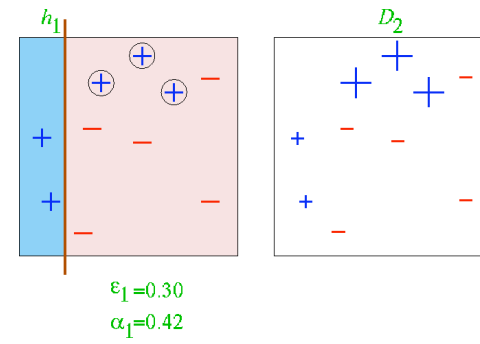
Boosting 16

## Exemple jouet



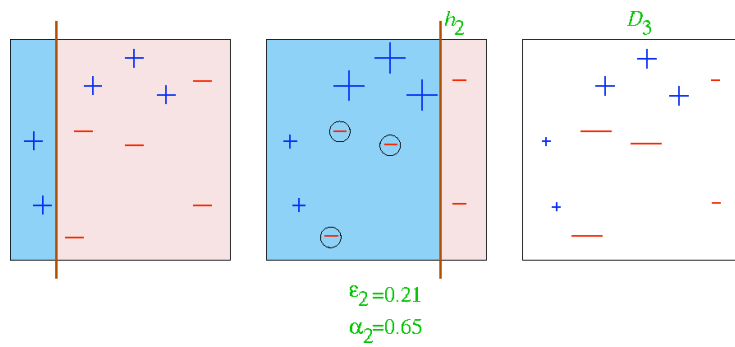
Boosting 17

## Étape 1



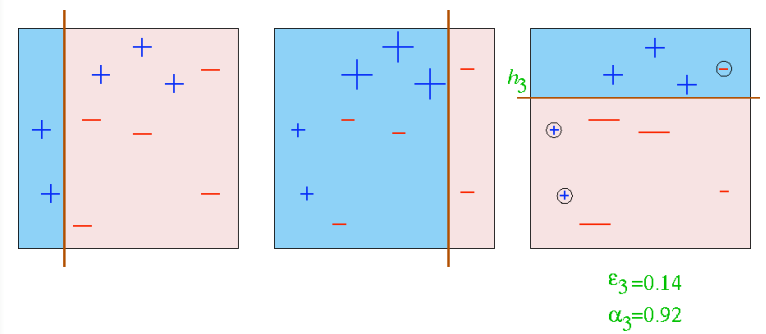
Boosting 18

## Étape 2



Boosting 19

## Étape 3



Boosting 20

## Hypothèse finale

$$H_{\text{final}} = \text{sign} \left( 0.42 \begin{array}{|c|c|} \hline & \\ \hline \end{array} + 0.65 \begin{array}{|c|c|} \hline & \\ \hline \end{array} + 0.92 \begin{array}{|c|c|} \hline & \\ \hline \end{array} \right)$$

=

Boosting 21

## Une Applet Boosting

<http://www.research.att.com/~yoav/adaboost/index.html>

Boosting 22

## Analyse théorique

- Voir [R. Meir & G. Rätsch. « An introduction to boosting and leveraging » In S. Mendelson and A. Smola (eds.) *Advanced lectures on Machine Learning*, LNCS, pp.119-184, Springer 2003]

$$R_{\text{Réel}} \leq R_{\text{Emp}} + \mathcal{O} \left( \sqrt{\frac{T \cdot d_{\mathcal{H}}}{m}} \right)$$

- Liens avec les méthodes par maximisation de la marge
- Encore en discussion

Boosting 23

## Boosting et maximisation de la marge

- **Adaboost et SVM** effectuent des recherches de classifieurs dans des espaces de grande dimension, mais :
  - Les normes utilisées sont différentes, donc les espaces explorés sont différents
  - Optimisation sous contraintes :
    - quadratique pour SVM
    - Linéaire pour boosting
  - Recherches différentes :
    - Globale par la méthode des noyaux : SVM
    - Gloutonne (1 coordonnée à la fois) pour les boosting

Boosting 24

## Avantages pratiques de AdaBoost

- (très) rapide
- simple + facile à programmer
- Une seul paramètre à régler : le nombre d'étapes de boosting ( $T$ )
- Applicable à de nombreux domaines par un bon choix de classifieur faible (neuro net, C4.5, ...)
- Pas de sur-spécialisation par la maximisation des marges
- Peut être adapté au cas où  $h_t : \mathcal{X} \rightarrow \mathcal{R}$  ; la classe est définie par le signe de  $h_t(x)$  ; la confiance est donnée par  $|h_t(x)|$
- Peut être adapté aux problèmes multi-classes où  $y_i \in \{1, \dots, c\}$  et aux problèmes multi-étiquettes
- Permet de trouver les exemples aberrants (outliers)

Boosting 25

## Aspects pratiques

<i>Avantages</i>	<i>Difficultés</i>
<ul style="list-style-type: none"> <li>• Un <b>meta-algorithme</b> d'apprentissage : utiliser n'importe quel algorithme d'apprentissage faible</li> <li>• En principe, <b>un seul paramètre</b> à régler (le nombre <math>T</math> d'itérations)</li> <li>• <b>Facile et aisé à programmer</b></li> <li>• <b>Performances théoriques garanties</b></li> </ul>	<ul style="list-style-type: none"> <li>• Difficile d'incorporer des connaissances a priori</li> <li>• Difficile de savoir comment régulariser</li> <li>• Le meilleur choix d'un apprenti faible n'est pas évident</li> <li>• Les frontières de décision en utilisant des méthodes parallèles aux axes est souvent très irrégulière (non interprétable)</li> </ul>

Boosting 26

## Applications

Text classification	Schapiro and Singer - Used stumps with normalized term frequency and multi-class encoding
OCR	Schwenk and Bengio (neural networks)
Natural language Processing	Collins; Haruno, Shirai and Ooyama
Image retrieval	Thieu and Viola
Medical diagnosis	Merle <i>et al.</i>
Fraud Detection	Rätsch & Müller 2001
Drug Discovery	Rätsch, Demiriz, Bennett 2002
Elect. Power Monitoring	Onoda, Rätsch & Müller 2000
<b>Fuller list:</b>	Schapiro's 2002, Meir & Rätsch 2003 review

Boosting 27

## Boosting : résumé

- La prédiction finale est issue d'une combinaison (vote pondéré) de plusieurs prédictions
- Méthode :
  - Itérative
  - Chaque classifieur dépend des précédents  
(les classifieurs ne sont donc pas indépendants comme dans d'autres méthodes de vote)
  - Les exemples sont pondérés différemment
  - Le poids des exemples reflète la difficulté des classifieurs précédents à les apprendre

Boosting 28

## Bagging

[Breiman, 96]

- **Génération de  $k$  échantillons « indépendants »** par tirage avec remise dans l'échantillon  $S_m$
- **Pour chaque échantillon**, apprentissage d'un classifieur en utilisant le même algorithme d'apprentissage
- La **prédiction finale** pour un nouvel exemple est obtenue par vote (simple) des classifieurs

Boosting 29

## Bagging (suite)

- Il est souvent dit que :
  - Le bagging fonctionne en réduisant la variance en laissant le biais inchangé
    - Seulement pour la fonction de coût quadratique
    - Pas vrai pour la fonction de perte 0-1
- Mais, encore incomplètement compris
  - Voir [Yves Grandvalet : « Bagging equalizes influence », *Machine Learning*, 55(3), pages 251-270, 2004.]

Boosting 30

## Etat de l'art (historique)

- [Valiant '84]
  - introduced theoretical PAC model for studying machine learning
- [Kearns & Valiant '88]
  - open problem of finding a boosting algorithm
- [Schapire '89], [Freund '90]
  - first polynomial-time boosting algorithms
- [Drucker, Schapire & Simard '92]
  - first experiments using boosting

Boosting 31

## Etat de l'art (suite)

- [Freund & Schapire '95]
  - introduced AdaBoost algorithm
  - strong practical advantages over previous boosting algorithms
- experiments using AdaBoost:
  - [Drucker & Cortes '95] [Schapire & Singer '98]
  - [Jackson & Cravon '96] [Maclin & Opitz '97]
  - [Freund & Schapire '96] [Bauer & Kohavi '97]
  - [Quinlan '96] [Schwenk & Bengio '98]
  - [Breiman '96] [Dietterich '98]
- continuing development of theory & algorithms:
  - [Schapire, Freund, Bartlett & Lee '97] [Schapire & Singer '98]
  - [Breiman '97] [Mason, Bartlett & Baxter '98]
  - [Grive and Schuurmans '98] [Friedman, Hastie & Tibshirani '98]

Boosting 32



## Bibliographie

- Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The elements of statistical learning. Data mining, inference and prediction.* Springer Verlag, 2001.