APPRENTISSAGE PAR COMBINAISON DE CLASSIFIEURS ELEMENTAIRES

(« dopage » ou « Boosting »)

Pr. Fabien Moutarde
Centre de Robotique (CAOR)
MINES Paris Tech (ENSMP)
PSL Research University

`Fabien.Moutarde@mines-paristech.fr`
`http://people.mines-paristech.fr/fabien.moutarde`

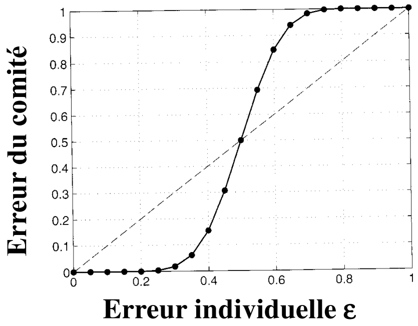
Principe essentiel : l'union fait la force

Réunir un « comité d'experts »
chacun peut se tromper, mais en combinant les avis,
on a plus de chance d'avoir la bonne prédiction !!…

Justification théorique :

- –supposons N classifieurs ***indépendants***, faisant chacun une erreur $E_{gen} = \varepsilon$
- –si on décide « à la majorité », alors on se trompe si et seulement si plus de la moitié du « comité » se trompe

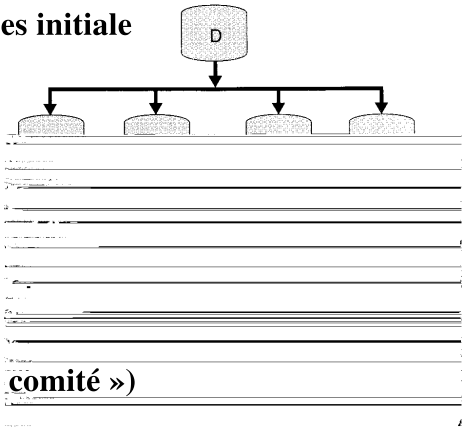
$$\Rightarrow \quad Erreur_{ensemble} = \sum_{k=N/2}^{N} C_k^N \varepsilon^k (1-\varepsilon)^{N-k}$$



Décision fortement améliorée,
(sous réserve que ε<0.5 !!)…
…et ce d'autant plus qu'on augmente le nombre N d'experts

Diverses méthodes pour produire les classifieurs à combiner

- Utiliser des algos totalement différents
- Même algo mais avec des paramètres et/ou initialisations différent(e)s
- **Faire varier l'ensemble d'apprentissage**



Base d'exemples initiale

Diverses variantes de la base
(tirages aléatoires, pondérations différentes, …)

Classifieurs élémentaires obtenus par la même procédure,
mais chacun sur une variante différente de la base

Classifieur global (« comité »)

→ Méthodes très générales, applicables avec n'importe quel algorithme « élémentaire »

Le « bagging »

Variantes de la base d'apprentissage obtenues par tirages aléatoires avec remise depuis la base initiale

(sorte de « ***bootstrap*** »→ duplication/disparition aléatoires de certains exemples selon variantes)

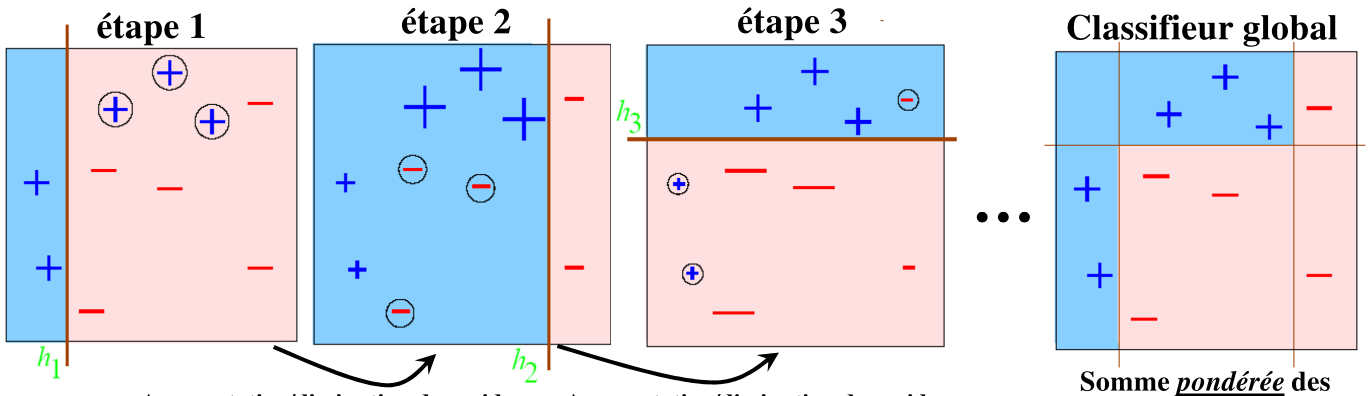
- Utile et efficace en particulier si l'algo de base utilisé est « instable » (au sens « sensible au bruit des données ») car différences entre variantes de base ➔ classifieurs élémentaires très différents
- Evite « over-fitting » (sur-apprentissage), car on « moyenne » des classifieurs construits avec différentes réalisations aléatoires des mêmes données

Le « boosting » (dopage)

<u>Méthode *itérative* pour l'ajout des classifieurs :</u>

variantes de la base d'apprentissage obtenues par des *<u>pondérations successives</u>* des mêmes exemples

(calculées pour <u>« se focaliser » sur les exemples « difficiles »</u>, i.e. mal classés par plusieurs classifieurs déjà assemblés)



Augmentation/diminution des poids selon que bien/mal classés *<u>par h1</u>*

Augmentation/diminution des poids selon que bien/mal classés *<u>par h2</u>*

Somme *<u>pondérée</u>* des classifieurs élémentaires

L'algorithme adaBoost

<u>adaBoost (« adaptive Boosting »)</u>

- **Base d'exemples initiale :**
  $S=\{(x_1,u_1), \ldots, (x_k,u_k)\}$, **avec** $u_i \in \{+1,-1\}$, $i=1,k$
- **Poids initiaux :** $w_0(x_i) = 1/m$ **pour tout** $i=1,k$ (ou $1/2p$ pr pos et $1/2n$ pr neg)
- **Pour chaque étape (« round ») t de 1 à T, faire :**
  1. **apprendre/choisir 1 règle de classification h_t sur (S,w_t) par l'algorithme A**
  2. **calculer l'erreur pondérée de h_t sur (S,w_t) :** $\varepsilon_t = \sum_{i=1}^{k} w_t(x_i) \times \left\| h_t(x_i) - u_i \right\|$
  3. **en déduire la « fiabilité » de h_t :** $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$ ***[$\alpha_t>0$ si $\varepsilon_t<0.5$, et $\to+\infty$ si $\varepsilon_t\to 0$]***
  4. **modifier les poids, i.e. pour i de 1 à k faire :**

$$w_{t+1}(x_i) = \frac{w_t(x_i)}{Z_t} \times \begin{cases} e^{-\alpha_t}\ si\ h_t(x_i) = u_i\ (i.e.\ x_i\ bien\ classé) \\ e^{+\alpha_t}\ si\ h_t(x_i) \neq u_i\ (i.e.\ x_i\ mal\ classé) \end{cases}$$

- **Fournir en sortie le classifieur global :** $H(x) = signe\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Boosting, over-fitting, et marges

- **L'écart entre l'erreur empirique la généralisation peut être borné par :**

$$E_{gen}(H_T) < E_{emp}(H_T) + O\left(\sqrt{\frac{T\delta}{n}}\right)$$

(où T est le nombre d'étapes de boosting, n le nombre d'exemples, et δ la VC-dimension de l'espace des h_t)

⇒ **Risque over-fitting ?? (cf aussi survalorisation des exemples « ambigüs »)**

- **Heureusement une borne plus intéressante est :**

$$E_{gen}(H_T) < \Pr(m(H_T, x) \le \theta) + O\left(\sqrt{\frac{\delta}{n\theta^2}}\right)$$

indépendant de T !!

➔ **si $P(m(H_T, x) < \theta)$ très faible pour un θ assez grand, alors on a une garantie de bonne généralisation…**

Résumé sur le boosting

- **Trois principes essentiels :**
  1. **Combiner les avis/estimations de différents experts**
  2. **Modifier, avant chaque ajout d'un expert, la distribution des exemples en surpondérant au fur et à mesure les exemples mal classés**
  3. **Au final, utiliser une moyenne des « votes » des experts, *pondérée* par leurs fiabilités respectives**

- **Permet d'obtenir un très bon classifieur en associant des classifieurs très « faibles »**
- **Problème essentiel : déterminer le « weak learner », i.e. l'algo employé à chaque étape (« round ») de boosting pour choisir/entraîner le classifieur faible h_t**