Modèle d'apprentissage PAC

Fabien Torre

GRAppA & Mostrare

Mercredi 7 et 14 octobre 2009

Notations et oracle

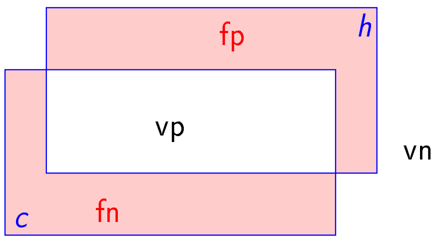
- $\mathcal{X}$ l'espace des exemples et $\mathcal{Y} = \{+1, -1\}$ leurs classes ;
- $x \in \mathcal{X}$ un exemple particulier ;
- $\mathcal{H}$ une classe de concepts définis sur $\mathcal{X}$;
- $c \in \mathcal{H}$ un concept particulier :
  - vue ensembliste : $c \subseteq \mathcal{X}$;
  - vue fonctionnelle : $c : \mathcal{X} \to \mathcal{Y}$;
- $\mathcal{D}$ une distribution sur $\mathcal{X}$;
- erreur d'une hypothèse $h \in \mathcal{H}$ visant un concept $c \in \mathcal{H}$:

$$\mathrm{erreur}(h) = \Pr_{x \in \mathcal{D}}[c(x) \neq h(x)]$$

- $\mathrm{EX}(c, \mathcal{D})$ un oracle qui tire un exemple de $\mathcal{X}$ suivant $\mathcal{D}$ et le fournit avec sa classification par le concept cible : $\langle x, c(x) \rangle$.

Notations : exemples

- $\mathcal{X}$: les points de $\mathbb{R}^2$ et $\mathcal{Y} = \{+1, -1\}$;
- $x = (70, 175)$;
- $\mathcal{H}$: les rectangles de $\mathbb{R}^2$ parallèles aux axes ;
- c un rectangle particulier :
  - vue ensembliste : les points contenus dans le rectangle c ;
  - vue fonctionnelle : test d'appartenance au rectangle c ;
- $\mathcal{D}$: distribution poids/taille chez l'homme ;
- erreur d'une hypothèse :



PAC apprenabilité [Valiant, 1984]

Définition : apprenabilité forte

$\mathcal{H}$ est PAC-apprenable s'il existe un algorithme L tel que :

- pour tout concept $c \in \mathcal{H}$,
- pour toute distribution $\mathcal{D}$ sur $\mathcal{X}$,
- pour tout paramètre d'erreur $0 < \epsilon < \frac{1}{2}$,
- pour tout paramètre de confiance $0 < \delta < \frac{1}{2}$,

L fournit une hypothèse $h \in \mathcal{H}$ qui vérifie, avec une probabilité $1 - \delta$: $\mathrm{erreur}(h) \leq \epsilon$.

Efficacement PAC-apprenable : L doit être polynomial en $\frac{1}{\epsilon}$ et $\frac{1}{\delta}$.

PAC apprenabilité : commentaires et précisions

- L a accès à l'oracle $\mathrm{EX}(c, \mathcal{D})$;
- L doit fonctionner quelle que soit la distribution ;
- L perçoit cette distribution à travers l'oracle ;
- $\mathcal{D}$ intervient aussi dans le calcul de l'erreur ;
- ϵ et δ sont fournis à L ;
- on peut choisir ϵ et δ aussi petits que voulus.

Retour aux rectangles...

- $\mathcal{X}$: les points de $\mathbb{R}^2$ et $\mathcal{Y} = \{+1, -1\}$;
- $\mathcal{H}$: les rectangles de $\mathbb{R}^2$ parallèles aux axes.

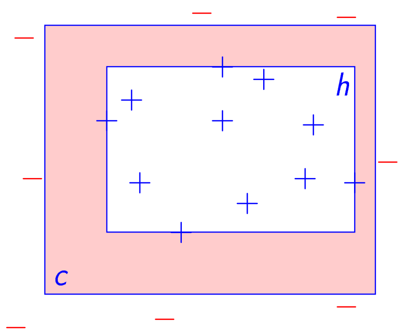
Est-ce que $\mathcal{H}$ est PAC-apprenable ?

Proposition pour L

1. demander n exemples à l'oracle et constituer un échantillon d'apprentissage A ;
2. retourner h le rectangle moindre-généralisé des exemples positifs de A.

- Montrer que $\forall c, \mathcal{D}, \epsilon, \delta$ et pour n bien choisi h vérifie avec une probabilité $1 - \delta$: $\mathrm{erreur}(h) \leq \epsilon$;
- montrer que n est un polynôme de $\frac{1}{\epsilon}$ et de $\frac{1}{\delta}$;
- intuition : plus ϵ et δ sont petits, plus n sera grand.

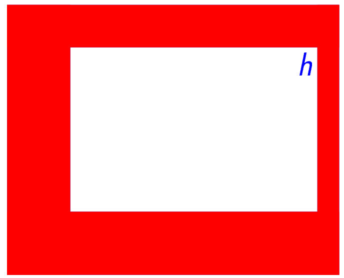
Démonstration (1) : c et h



Remarques

- h est toujours inclus dans c, car h moindre-généralisé ;
- h présente donc uniquement une erreur de type *fn*.

Démonstration (2) : borner l'erreur

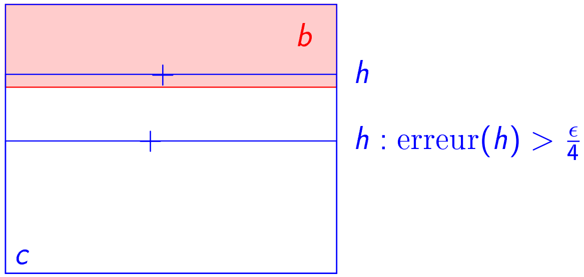


Objectifs

- limiter l'erreur globale à ϵ ;
- calculer la probabilité que l'une des bandes excède $\frac{\epsilon}{4}$.

Démonstration (3) : une bande

On définit une bande b de poids $\frac{\epsilon}{4}$ en haut de c.



Cas favorable : l'échantillon A contient un exemple (positif) dans b. L'erreur de h est inférieure à $\frac{\epsilon}{4}$ sur la bande haute.
Cas défavorable : $A \cap b = \emptyset$. L'erreur de h est supérieure à $\frac{\epsilon}{4}$ sur la bande haute. Probabilité de cette situation ?

Démonstration (4) : probabilités

- Tirer un exemple dans b : $\frac{\epsilon}{4}$;
- tirer un exemple en dehors de b : $1 - \frac{\epsilon}{4}$;
- tirer n exemples en dehors de b : $\left(1 - \frac{\epsilon}{4}\right)^n$;
- ne pas avoir d'exemple dans l'une des bandes : $\leq 4 \times \left(1 - \frac{\epsilon}{4}\right)^n$.

et on obtient finalement :

$$\Pr[\mathrm{erreur}(h) > \epsilon] \leq 4 \times \left(1 - \frac{\epsilon}{4}\right)^n \leq 4 \times \left(e^{-\frac{\epsilon}{4}}\right)^n = 4 \times e^{-\frac{n\epsilon}{4}}$$

en utilisant le fait que : $(1 - x) \leq e^{-x}$.

Démonstration (5) : calcul du n minimal

On veut $\Pr[\mathrm{erreur}(h) > \epsilon] \leq \delta$ et on résout donc :

$$\begin{array}{rrcl} & 4 \times e^{-\frac{n\epsilon}{4}} & \leq & \delta \\ \Rightarrow & e^{-\frac{n\epsilon}{4}} & \leq & \frac{\delta}{4} \\ \Rightarrow & -\frac{n\epsilon}{4} & \leq & \ln(\frac{\delta}{4}) \\ \Rightarrow & -n & \leq & \frac{4}{\epsilon} \times \ln(\frac{\delta}{4}) \\ \Rightarrow & n & \geq & -\frac{4}{\epsilon} \times \ln(\frac{\delta}{4}) \\ \Rightarrow & n & \geq & \frac{4}{\epsilon} \times \ln(\frac{4}{\delta}) \end{array}$$

Démonstration (6) : conclusions

- L doit constituer son échantillon avec $n = \frac{4}{\epsilon} \times \ln(\frac{4}{\delta})$;
- cela garantit : $\Pr[\mathrm{erreur}(h) \leq \epsilon] \geq 1 - \delta$;
- L est linéaire en $\frac{1}{\epsilon}$;
- L est logarithmique en $\frac{1}{\delta}$.

La classe des rectangles parallèles aux axes dans $\mathbb{R}^2$ est efficacement PAC-apprenable.
(VC dimension de cette classe ?)

Vecteurs booléens

Les exemples

- $\mathcal{X}_m = \{0,1\}^m$;
- $x \in \mathcal{X}_m$: vecteur booléen de taille m, $x = (a_1, a_2, \dots, a_m)$.

Hypothèses 1

- $\mathcal{H}_m$, conjonctions de littéraux ;
- $h \in \mathcal{H}_m$, conjonction de variables niées ou pas, par exemple $h = a_1 \wedge \overline{a_2}$;
- $\mathcal{H}_m$ est-il PAC apprenable ?

Hypothèses 2

- $\mathcal{H}_m$, 3-Term DNF ;
- $h \in \mathcal{H}_m$, disjonction de trois conjonctions de littéraux, par exemple $h = (a_1 \wedge \overline{a_2}) \vee (\overline{a_3} \wedge \overline{a_4}) \vee (a_1 \wedge a_3)$;
- $\mathcal{H}_m$ est-il PAC apprenable ?

PAC apprenabilité et compromis biais/variance

- Dans le modèle PAC, $c \in \mathcal{H}$, donc pas d'erreur de biais ;
- si $\mathcal{H}$ est trop riche, on a par contre une erreur importante en variance, et alors $\mathcal{H}$ peut ne pas être PAC apprenable.

Apprenabilité faible [Kearns and Valiant, 1989]

Définition : apprenabilité faible

$\mathcal{H}$ est PAC-apprenable s'il existe un algorithme L tel que :

- il existe $0 < \epsilon_0 < \frac{1}{2}$ et $0 < \delta_0 < \frac{1}{2}$,
- et pour tout concept $c \in \mathcal{H}$,
- et pour toute distribution $\mathcal{D}$ sur $\mathcal{X}$,

L fournit une hypothèse $h \in \mathcal{H}$ qui vérifie, avec une probabilité $1 - \delta_0$: $\mathrm{erreur}(h) \leq \epsilon_0$.

On demande simplement que les hypothèses h produites par L soient meilleures qu'un étiquetage purement aléatoire.

De l'apprenabilité faible à l'apprenabilité forte

On demande simplement que les hypothèses h produites par L soient meilleures qu'un étiquetage purement aléatoire.

- pour un problème à deux classes, quelle que soit la distribution, un étiquetage aléatoire a une erreur d'exactement $\frac{1}{2}$;
- les deux notions d'apprenabilité sont elles équivalentes ?
- on dit que L est un *apprenant faible* si $\forall c \in \mathcal{H}, \forall \mathcal{D}$: $\mathrm{erreur}(h) < \frac{1}{2}$ (avec une probabilité $1 - \delta_0$) ;
- peut-on transformer un apprenant faible L en un apprenant fort ? c'est-à-dire trouver L' qui soit un apprenant fort en faisant un nombre d'appels polynomial à L...

Comment *booster* (efficacement) ϵ_0 et δ_0 jusqu'à des valeurs arbitrairement petites ?

Algorithme de boosting de la confiance

Objectif : obtenir une confiance δ et une erreur $\epsilon_0 + \gamma$ avec δ et γ positifs non nuls mais arbitrairement petits.

Proposition pour L'

1. utiliser k fois l'algorithme faible L pour obtenir autant d'hypothèses faibles $h_1, h_2, \ldots, h_k$;
2. demander m exemples à l'oracle pour former un échantillon A' ;
3. fournir $h = \mathrm{ArgMin}_{h_i \in [h_1, h_2, \ldots, h_k]}(\mathrm{erreur}_{A'}(h_i))$.

Les risques d'échec encourus :

- que les k hypothèses produites aient chacune une erreur supérieure à ϵ_0 ; limiter ce risque à $\frac{\delta}{2}$
- que l'hypothèse choisie à la dernière étape ait en fait une erreur réelle supérieure à $\epsilon_0 + \gamma$. limiter ce risque à $\frac{\delta}{2}$

Reste à choisir k et m en conséquence...

Choix de k, le nombre d'hypothèses produites

Objectif : déterminer la valeur de k telle que la probabilité que chacune des hypothèses h_i ait une erreur supérieure à ϵ_0 soit inférieure à $\frac{\delta}{2}$.

- Probabilité qu'une h_i ait une erreur supérieure à ϵ_0 : δ_0 ;
- probabilité que toutes les hypothèses h_i aient des erreurs supérieures à ϵ_0 : ${\delta_0}^k \leq (1-\delta_0)^k \leq e^{-k\delta_0}$;

On veut que la probabilité d'un tel événement soit inférieure à $\frac{\delta}{2}$:

$$\begin{array}{rrcl} & e^{-k\delta_0} & \leq & \frac{\delta}{2} \\ \Rightarrow & -k\delta_0 & \leq & \ln(\frac{\delta}{2}) \\ \Rightarrow & -k & \leq & \frac{1}{\delta_0} \times \ln(\frac{\delta}{2}) \\ \Rightarrow & k & \geq & -\frac{1}{\delta_0} \times \ln(\frac{\delta}{2}) \\ \Rightarrow & k & \geq & \frac{1}{\delta_0} \times \ln(\frac{2}{\delta}) \end{array}$$

Choix de m, le nombre d'exemples pour choisir h

Objectif : déterminer la valeur de m telle que la probabilité que certaines erreurs de h_i mesurées sur A' dévient de plus de γ par rapport aux erreurs réelles, soit inférieure à $\frac{\delta}{2}$.

Borne de Chernoff pour une hypothèse h :

$$\Pr(|\mathrm{erreur}(h) - \mathrm{erreur}_{A'}(h)| \geq \gamma) \leq e^{-2m\gamma^2}$$

Pour une hypothèse, on veut borner cette probabilité par $\frac{\delta}{2k}$:

$$\begin{array}{rc} & e^{-2m\gamma^2} \leq \frac{\delta}{2k} \\ \Rightarrow & -2m\gamma^2 \leq \ln(\frac{\delta}{2k}) \\ \Rightarrow & -m \leq \frac{1}{2\gamma^2}.\ln(\frac{\delta}{2k}) \\ \Rightarrow & m \geq \frac{1}{2\gamma^2}.\ln(\frac{2k}{\delta}) \end{array}$$

Boosting de la confiance : bilan

La confiance peut-être boostée à volonté, avec un léger surcoût sur l'erreur.
(δ_0, ϵ_0) dépendant de L, (δ, γ) étant donnés :

1. utiliser $k = \frac{1}{\delta_0} \times ln\frac{2}{\delta}$ fois l'algorithme faible L pour obtenir autant d'hypothèses faibles $h_1, h_2, \ldots, h_k$;
2. demander $m = \frac{1}{2\gamma^2}.\ln(\frac{2k}{\delta})$ exemples à l'oracle pour former un échantillon A' ;
3. fournir $h = \mathrm{ArgMin}_{h_i \in [h_1, h_2, \ldots, h_k]}(\mathrm{erreur}_{A'}(h_i))$.

Booster l'erreur est un peu plus difficile...

Algorithme de boosting de l'erreur

1. $h_1 = L(\mathrm{EX}(c, \mathcal{D}))$;
2. définir un nouvel oracle $\mathrm{EX}(c, \mathcal{D}_2)$ comme suit :
    1. on tire une pièce à pile ou face ;
    2. si *pile*, faire $x = \mathrm{EX}(c, \mathcal{D})$ jusqu'à $h_1(x) = c(x)$;
    3. si *face*, faire $x = \mathrm{EX}(c, \mathcal{D})$ jusqu'à $h_1(x) \neq c(x)$;
    4. renvoyer x.
3. $h_2 = L(\mathrm{EX}(c, \mathcal{D}_2))$;
4. définir un nouvel oracle $\mathrm{EX}(c, \mathcal{D}_3)$ comme suit :
    1. faire $x = \mathrm{EX}(c, \mathcal{D})$ jusqu'à $h_1(x) \neq h_2(x)$;
    2. renvoyer x.
5. $h_3 = L(\mathrm{EX}(c, \mathcal{D}_3))$;
6. renvoyer $h = \mathrm{VoteMajoritaire}(h_1, h_2, h_3)$.

Commentaires, objectif, notations

- $\mathcal{D}_2$ donne un poids global de 0.5 aux exemples bien classés par h_1 et 0.5 aux mal classés ;
- il s'en suit que h_1 a une erreur de 0.5 sur $\mathcal{D}_2$ et que L, apprenant faible, ne peut pas apprendre h_1 sur $\mathcal{D}_2$;
- par construction, chaque h_i amène une information différente.

On doit montrer que l'hypothèse h fournie par L' vérifie :

$$\mathrm{erreur}(h) < \epsilon_0$$

c'est-à-dire que L' fait strictement mieux que L.

Soient $\beta_1 = \mathrm{erreur}_{\mathcal{D}}(h_1)$, $\beta_2 = \mathrm{erreur}_{\mathcal{D}_2}(h_2)$ et $\beta_3 = \mathrm{erreur}_{\mathcal{D}_3}(h_3)$.

D'une distribution à l'autre (1) : formules de passage

Considérons les exemples mal classés par h_1 :

$\mathcal{D}$	$\rightarrow$	$\mathcal{D}_2$
β_1	$\rightarrow$	$\frac{1}{2}$
$\mathcal{D}[x] = 2.\beta_1.\mathcal{D}_2[x]$	$\rightarrow$	$\mathcal{D}_2[x] = \frac{1}{2.\beta_1}.\mathcal{D}[x]$

... et les exemples bien classés par h_1 :

$\mathcal{D}$	$\rightarrow$	$\mathcal{D}_2$
$1 - \beta_1$	$\rightarrow$	$\frac{1}{2}$
$\mathcal{D}[x] = 2.(1 - \beta_1).\mathcal{D}_2[x]$	$\rightarrow$	$\mathcal{D}_2[x] = \frac{1}{2.(1-\beta_1)}.\mathcal{D}[x]$

D'une distribution à l'autre (2) : sur un exemple

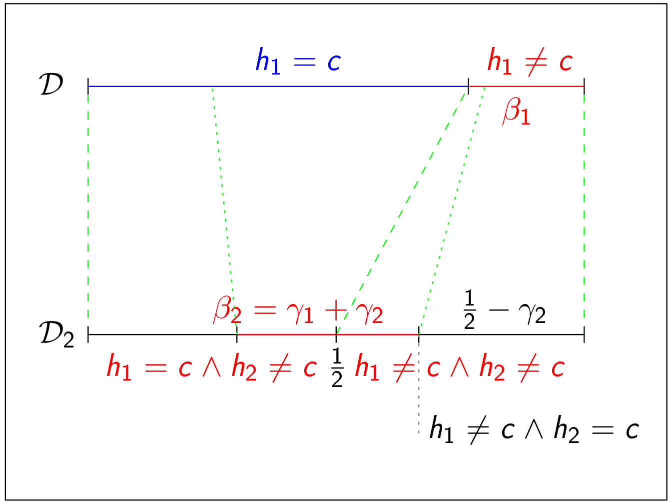
Cinq exemples, distribution uniforme

$$\mathcal{D} : \frac{1}{5} \quad \frac{1}{5} \quad \frac{1}{5} \quad \frac{1}{5} \quad \frac{1}{5}$$

- trois sont bien classés par h_1, deux en erreur : $\beta_1 = \frac{2}{5}$;
- facteur multiplicateur des mal classés : $\frac{1}{2.\beta_1} = \frac{5}{4}$;
- facteur multiplicateur des bien classés : $\frac{1}{2.(1-\beta_1)} = \frac{5}{6}$;

$$\mathcal{D}_2 : \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{4} \quad \frac{1}{4} \quad \underbrace{\frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6}}_{=\frac{1}{2}} \quad \underbrace{\frac{1}{4} \quad \frac{1}{4}}_{=\frac{1}{2}}$$

D'une distribution à l'autre (3) : notations complètes



Décomposition de l'erreur

Si h finale se trompe c'est que l'un des deux cas est survenu :

1. h_1 et h_2 sont d'accord mais se trompent ;
2. h_1 et h_2 ne sont pas d'accord et h_3 se trompe.

ce qui se traduit formellement par :

$$\begin{array}{rcl} \text{erreur}_{\mathcal{D}}(h) & = & \Pr_{\mathcal{D}}[h_1(x) \neq c(x) \wedge h_2(x) \neq c(x)] \\ & + & \Pr_{\mathcal{D}}[h_3(x) \neq c(x) | h_1(x) \neq h_2(x)].\Pr_{\mathcal{D}}[h_1(x) \neq h_2(x)] \end{array}$$

On va développer explicitement, en vue de comparer à ϵ_0.

Décomposition de l'erreur : cas 1

h_1 et h_2 sont d'accord mais se trompent.

Par définition :

$$\Pr_{\mathcal{D}_2}[h_1(x) \neq c(x) \wedge h_2(x) \neq c(x)] = \gamma_2$$

et par application de la formule de passage de $\mathcal{D}_2$ vers $\mathcal{D}$:

$$\Pr_{\mathcal{D}}[h_1(x) \neq c(x) \wedge h_2(x) \neq c(x)] = 2.\beta_1.\gamma_2$$

Décomposition de l'erreur : cas 2

h_1 et h_2 ne sont pas d'accord et h_3 se trompe.

Par définition de $\mathcal{D}_3$:

$$\Pr_{\mathcal{D}}[h_3(x) \neq c(x) | h_1(x) \neq h_2(x)] = \Pr_{\mathcal{D}_3}[h_3(x) \neq c(x)] = \beta_3$$

On décompose la situation où h_1 et h_2 diffèrent :

$$\begin{array}{rcl} \Pr_{\mathcal{D}}[h_1(x) \neq h_2(x)] & = & \Pr_{\mathcal{D}}[h_1(x) = c(x) \wedge h_2(x) \neq c(x)] \\ & + & \Pr_{\mathcal{D}}[h_1(x) \neq c(x) \wedge h_2(x) = c(x)] \end{array}$$

Décomposition de l'erreur : cas 2.1 et cas 2.2

$$\begin{array}{rrcl} & \Pr_{\mathcal{D}_2}[h_1(x) = c(x) \wedge h_2(x) \neq c(x)] & = & \gamma_1 \\ \Rightarrow & \Pr_{\mathcal{D}}[h_1(x) = c(x) \wedge h_2(x) \neq c(x)] & = & 2.(1-\beta_1).\gamma_1 \end{array}$$

$$\begin{array}{rrcl} & \Pr_{\mathcal{D}_2}[h_1(x) \neq c(x) \wedge h_2(x) = c(x)] & = & \frac{1}{2} - \gamma_2 \\ \Rightarrow & \Pr_{\mathcal{D}}[h_1(x) \neq c(x) \wedge h_2(x) = c(x)] & = & 2.\beta_1.(\frac{1}{2} - \gamma_2) \end{array}$$

Calcul exact de l'erreur

$$\begin{array}{rcl} \mathrm{erreur}_{\mathcal{D}}(h) & = & \Pr_{\mathcal{D}}[h_1(x) \neq c(x) \wedge h_2(x) \neq c(x)] \\ & + & \Pr_{\mathcal{D}}[h_3(x) \neq c(x) | h_1(x) \neq h_2(x)].\Pr_{\mathcal{D}}[h_1(x) \neq h_2(x)] \\ & & \\ & = & 2.\beta_1.\gamma_2 + \beta_3.\left[2.(1-\beta_1).\gamma_1 + 2.\beta_1.(\frac{1}{2} - \gamma_2)\right] \\ & = & 2.\beta_1.\gamma_2 + 2.\beta_3.\gamma_1 - 2.\beta_1.\beta_3.\gamma_1 + \beta_1.\beta_3 - 2.\beta_1.\beta_3.\gamma_2 \\ & = & \beta_1.(2.\gamma_2 - 2.\beta_3.\gamma_1 + \beta_3 - 2.\beta_3.\gamma_2) + 2.\beta_3.\gamma_1 \\ & = & \beta_1.[2.\gamma_2 + \beta_3.(1-2.\beta_2)] + 2.\beta_3.\gamma_1 \end{array}$$

Borne sur l'erreur

$$\begin{array}{rcl} \mathrm{erreur}_{\mathcal{D}}(h) & = & \beta_1.[2.\gamma_2 + \beta_3.(1-2.\beta_2)] + 2.\beta_3.\gamma_1 \\ & \leq & \epsilon_0.[2.\gamma_2 + \beta_3.(1-2.\beta_2)] + 2.\beta_3.\gamma_1 \\ & \leq & 2.\gamma_2.\epsilon_0 + \beta_3.\epsilon_0.(1-2.\beta_2) + 2.\beta_3.\gamma_1 \\ & \leq & \beta_3.[\epsilon_0.(1-2.\beta_2) + 2.\gamma_1] + 2.\gamma_2.\epsilon_0 \\ & \leq & \epsilon_0[\epsilon_0.(1-2.\beta_2) + 2.\gamma_1] + 2.\epsilon_0.\gamma_2 \\ & \leq & {\epsilon_0}^2 - 2.{\epsilon_0}^2.\beta_2 + 2.\epsilon_0.(\gamma_1 + \gamma_2) \\ & \leq & {\epsilon_0}^2 - 2.{\epsilon_0}^2.\beta_2 + 2.\epsilon_0.\beta_2 \\ & \leq & {\epsilon_0}^2 + 2.\beta_2.(\epsilon_0 - {\epsilon_0}^2) \\ & \leq & {\epsilon_0}^2 + 2.\epsilon_0.(\epsilon_0 - {\epsilon_0}^2) \\ & \leq & 3.{\epsilon_0}^2 - 2.{\epsilon_0}^3 < \epsilon_0 \end{array}$$

L' booste effectivement l'erreur de L !

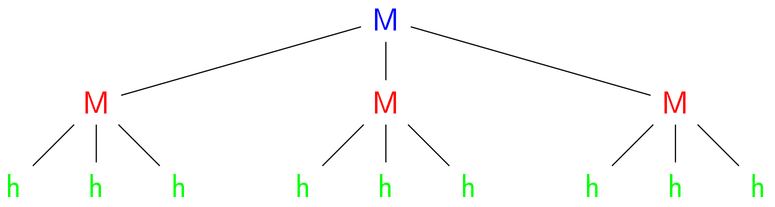
Premier algorithme de boosting [Schapire, 1990]

On note $g(\epsilon) = 3.\epsilon^2 - 2.\epsilon^3$.

$\mathrm{ApprentissageFort}(\epsilon, \mathrm{EX}(c, \mathcal{D}))$:

1. si $\epsilon \geq \epsilon_0$, retourner $L(\mathrm{EX}(c, \mathcal{D}))$;
2. $\epsilon' = g^{-1}(\epsilon)$;
3. $h_1 = \mathrm{ApprentissageFort}(\epsilon', \mathrm{EX}(c, \mathcal{D}))$;
4. définir $\mathcal{D}_2$ en fonction de $\mathcal{D}$ et de h_1, comme précédemment ;
5. $h_2 = \mathrm{ApprentissageFort}(\epsilon', \mathrm{EX}(c, \mathcal{D}_2))$;
6. définir $\mathcal{D}_3$ en fonction de $\mathcal{D}$, h_1 et h_2, comme précédemment ;
7. $h_3 = \mathrm{ApprentissageFort}(\epsilon', \mathrm{EX}(c, \mathcal{D}_3))$;
8. retourner $h = \mathrm{VoteMajoritaire}(h_1, h_2, h_3)$.

Premier algorithme de boosting : bilan



Critiques

- algorithme triplement récursif ;
- trois types d'hypothèses sont manipulés : les h apprises par L, les votes majoritaires de trois h et les votes majoritaires de trois votes majoritaires ;
- chaque L fait appel à l'oracle et ne partage pas ses exemples.

Bibliographie I

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.

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

Bibliographie II

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