Algorithmique avancée: Apprentissage

Serge Haddad

LSV, ENS Paris-Saclay & CNRS & Inria

L3

Plan

Un cadre pour l'apprentissage

Classification.

- $\mathcal{X}$ est un ensemble d'objets ;
- $\mathcal{Y}$ est un ensemble d'étiquettes ;
- f de $\mathcal{X}$ vers $\mathcal{Y}$ est un *classifieur* ;
- $\mathcal{D}$ est une distribution sur $\mathcal{X}$.

Données.

Une *donnée d'apprentissage* de *taille* m est une suite $\sigma_m = \{(x_i, f(x_i)\}_{i \leq m}$ où les x_i sont des échantillons tirés de manière indépendante en suivant $\mathcal{D}$.

S_m est la variable aléatoire associée à une donnée d'apprentissage aléatoire.

Algorithme. *(f et $\mathcal{D}$ ne sont pas connus de l'algorithme)*

Un *algorithme d'apprentissage* prend en entrée une donnée d'apprentissage et renvoie un classifieur h de $\mathcal{X}$ vers $\mathcal{Y}$.

On note H_m la sortie (aléatoire) de l'algorithme sur S_m.

Capacité d'apprentissage

Erreur d'un classifieur.

Soit h un classifieur et X une variable aléatoire de distribution $\mathcal{D}$.
Alors *l'erreur de classification* $L_{\mathcal{D},f}(h)$ (notée aussi $L(h)$) est définie par :

$$L_{\mathcal{D},f}(h) = \mathbf{Pr}_{\mathcal{D}}(h(X) \neq f(X))$$

Classe d'hypothèses.

Une *classe d'hypothèses* $\mathcal{H}$ est un sous-ensemble de $\mathcal{Y}^{\mathcal{X}}$ telle que $f \in \mathcal{H}$.

$\mathcal{H}$ est connue de l'algorithme.

Capacité d'apprentissage.

$\mathcal{H}$ *peut être apprise* s'il existe un algorithme $\mathcal{A}$ tel que pour tout
$\mathcal{D}$, $f \in \mathcal{H}$ et $0 < \varepsilon, \delta < 1$, il existe un entier $m_{\mathcal{H}}(\varepsilon, \delta)$ vérifiant :

$$\exists m_{\mathcal{H}}(\varepsilon, \delta) \in \mathbb{N} \qquad \forall m \geq m_{\mathcal{H}}(\varepsilon, \delta) \qquad \mathbf{Pr}_{\mathcal{D}}(L_{\mathcal{D},f}(H_m) > \varepsilon) \leq \delta$$

Erreur empirique

Soit h et $\sigma_m = \{(x_i, f(x_i))\}_{i \leq m}$.

Alors *l'erreur empirique* $L_{\sigma_m}(h)$ de h est définie par :

$$L_{\sigma_m}(h) = \frac{1}{m} |\{i \mid h(x_i) \neq f(x_i)\}|$$

Soit $\varepsilon > 0$ et h un classifieur. Alors d'après les bornes de Hoeffding :

$$\mathbf{Pr}(L_{S_m}(h) \geq L(h) + \varepsilon) \leq e^{-2m\varepsilon^2} \quad \text{et} \quad \mathbf{Pr}(L_{S_m}(h) \leq L(h) - \varepsilon) \leq e^{-2m\varepsilon^2}$$

Par conséquent, soit $\varepsilon, \delta > 0$, h un classifieur et $m \geq \frac{1}{2\varepsilon^2} \log(\frac{2}{\delta})$:

$$\mathbf{Pr}(|L_{S_m}(h) - L(h)| \geq \varepsilon) \leq \delta$$

Apprentissage d'une classe finie

Soit $|\mathcal{H}| < \infty$. Alors ERM (Empirical Risk Minimization) renvoie $\arg\min(L_{S_m}(h))$.

ERM permet d'apprendre $\mathcal{H}$.

Preuve. On fixe ε et δ.

Soit $\mathcal{H}_B = \{h \mid L(h) > \varepsilon\}$ et $\Sigma_B = \{\sigma_m \mid \exists h \in \mathcal{H}_B \ L_{\sigma_m}(h) = 0\}$.

$\Sigma_B = \bigcup_{h \in \mathcal{H}_B} \{\sigma_m \mid L_{\sigma_m}(h) = 0\}$. D'où :

$$\mathbf{Pr}(L(H_m) > \varepsilon) \leq \sum_{h \in \mathcal{H}_B} \mathbf{Pr}(L_{S_m}(h) = 0)$$

Pour tout $h \in \mathcal{H}_B$, $\mathbf{Pr}(L_{S_m}(h) = 0) = (\mathbf{Pr}(h(X) = f(X)))^m \leq (1-\varepsilon)^m \leq e^{-m\varepsilon}$.

Par conséquent, $\mathbf{Pr}(L(H_m) > \varepsilon) \leq |\mathcal{H}_B| e^{-m\varepsilon} \leq |\mathcal{H}| e^{-m\varepsilon}$.

D'où pour tout $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}$, $\mathbf{Pr}(L(H_m) > \varepsilon) \leq \delta$.

Apprentissage d'une classe infinie

Soit $\mathcal{H} = \{h_a \mid a \in \mathbb{R}\}$ où $h_a = \mathbb{1}_{[x>a]}$ et $f = h_{a^*}$.

ERM$(\{(x_i, b_i)\}_{i \leq m}) = h_a$ avec $x_\perp \stackrel{\text{def}}{=} \max(x_i | b_i = 0) \leq a < \min(x_i | b_i = 1) \stackrel{\text{def}}{=} x_\top$.

ERM permet d'apprendre $\mathcal{H}$.

Preuve. (cas $\mathcal{D}$ continue et $\varepsilon_0 \stackrel{\text{def}}{=} \min(\mathbf{Pr}(]-\infty, a^*[), \mathbf{Pr}([a^*, \infty[)) > 0$)

On fixe $\varepsilon < \varepsilon_0$ et δ. Soit $a_\perp$ et $a_\top$ vérifiant $\mathbf{Pr}([a_\perp, a^*[) = \mathbf{Pr}(]a^*, a_\top]) = \varepsilon$.

$L(h_a) > \varepsilon$ implique : $a < a_\perp$ ou $a > a_\top$.

Notons A la variable aléatoire telle que $H_m = h_A$.

$$
\begin{aligned}
\mathbf{Pr}(L(h_A) > \varepsilon) \leq & \quad \mathbf{Pr}(A < a_\perp) + & \mathbf{Pr}(A > a_\top) \\
\leq & \quad \mathbf{Pr}(\bigwedge_{i \leq m} X_i \notin [a_\perp, a^*]) + & \mathbf{Pr}(\bigwedge_{i \leq m} X_i \notin [a^*, a_\top]) \\
= & \quad 2(1-\varepsilon)^m & \\
\leq & \quad 2e^{-m\varepsilon} & \\
\leq & \quad \delta & \text{dès que } m \geq \frac{\log(2/\delta)}{\varepsilon}
\end{aligned}
$$

Plan

La VC-dimension d'une classe

Dans la suite, $\mathcal{Y} = \{\bot, \top\}$. Soit $C = \{c_1, \ldots, c_m\} \subseteq \mathcal{X}$.

$\mathcal{H}_C$ est l'ensemble des restrictions des classifieurs de $\mathcal{H}$ à C.

$|\mathcal{H}_C| \leq 2^m$. On dit que $\mathcal{H}$ *brise* C si $|\mathcal{H}_C| = 2^m$ (tout $\mathcal{H}$ *brise* $\emptyset$).

Illustration. Soit $\mathcal{H} = \{h_a \mid a \in \mathbb{R}\}$.

Pour tout $C = \{x\}$, $\mathcal{H}$ brise C puisque $h_x(x) = \bot$ et $h_{x-1}(x) = \top$.

Pour tout $C = \{x, y\}$ avec $x < y$, $\mathcal{H}$ ne brise pas C
car il n'existe aucun a pour lequel $h_a(x) = \top$ et $h_a(y) = \bot$.

La *VC-dimension* (Vapnik, 2017 IEEE John von Neumann Medal & Chervonenkis) de $\mathcal{H}$ notée VCdim$(\mathcal{H})$ est définie par :

$$\text{VCdim}(\mathcal{H}) = \sup_C(|C| \mid \mathcal{H} \text{ brise } C)$$

Illustration.

- Si $\mathcal{H} = \{h_a \mid a \in \mathbb{R}\}$ alors VCdim$(\mathcal{H}) = 1$.
- Si $\mathcal{H} = 2^{\mathcal{X}}$ et $\mathcal{X}$ n'est pas fini alors VCdim$(\mathcal{H}) = \infty$
- Si $|\mathcal{H}| < \infty$ alors VCdim$(\mathcal{H}) \leq \lfloor \log_2(|\mathcal{H}|) \rfloor$.

$\mathrm{VCdim}(\mathcal{H}) = \infty$ (1)

Soit $\mathcal{H}$ avec $\mathrm{VCdim}(\mathcal{H}) = \infty$. Alors $\mathcal{H}$ ne peut être apprise.

Preuve. Soit $\varepsilon = \frac{1}{8}$ et $\delta < \frac{1}{7}$. Supposons que $\mathcal{H}$ puisse être apprise.
Notons m la taille de l'échantillon de l'algorithme correspondant à ces seuils.
$C = \{x_1, \ldots, x_{2m}\}$ est un sous-ensemble de taille $2m$ brisé par $\mathcal{H}$.
$\mathcal{D}$ est la distribution uniforme sur l'ensemble $\{x_i\}_{i \leq 2m}$.
$\{f_k\}_{k \leq T}$, est l'ensemble des fonctions booléennes de C avec $T = 2^{2m}$.

Soit $f(\sigma_m^k)$ la fonction renvoyée par l'algorithme
pour une donnée σ_m^k de taille m, lorsque la fonction cible est f_k.
Soit S_m une donnée aléatoire de taille m pour $\mathcal{D}$. Nous allons démontrer que :

$$\max_{k \leq T} \mathbf{E}(L_{\mathcal{D}, f_k}(f(S_m^k)) \geq \frac{1}{4}$$

Soit $N = (2m)^m$ le nombre d'échantillons (avec répétition) de taille m parmi C.
On indicera ces échantillons par j : $(x_1^j, \ldots, x_m^j)$.
$\sigma_j^k \stackrel{\text{def}}{=} ((x_1^j, f_k(x_1^j), \ldots, (x_m^j, f_k(x_1^j)))$.
Par définition, $\mathbf{E}(L_{\mathcal{D}, f_k}(f(S_m^k)) = \frac{1}{N} \sum_{j \leq N} L_{\mathcal{D}, f_k}(f(\sigma_j^k))$.

$\mathrm{VCdim}(\mathcal{H}) = \infty$ (2)

Preuve (suite).

$$\begin{aligned}
\max_{k\leq T} \mathbf{E}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq & \quad \frac{1}{T}\sum_{k\leq T}\frac{1}{N}\sum_{j\leq N} L_{\mathcal{D},f_k}(f(\sigma_j^k)) \\
= & \quad \frac{1}{N}\sum_{j\leq N}\frac{1}{T}\sum_{k\leq T} L_{\mathcal{D},f_k}(f(\sigma_j^k)) \\
\geq & \quad \min_{j\leq N}\frac{1}{T}\sum_{k\leq T} L_{\mathcal{D},f_k}(f(\sigma_j^k))
\end{aligned}$$

Soit j arbitrairement choisi et $\{v_1,\ldots,v_p\} = C \setminus \{x_1^j,\ldots,x_m^j\} : p \geq m$.

$$\begin{aligned}
\frac{1}{T}\sum_{k\leq T} L_{\mathcal{D},f_k}(f(\sigma_j^k)) \geq & \quad \frac{1}{T}\sum_{k\leq T}\frac{1}{2m}\sum_{r\leq p} \mathbb{1}_{f(\sigma_j^k)(v_r)\neq f_k(v_r)} \\
\geq & \quad \frac{1}{T}\sum_{k\leq T}\frac{1}{2p}\sum_{r\leq p} \mathbb{1}_{f(\sigma_j^k)(v_r)\neq f_k(v_r)} \\
= & \quad \frac{1}{p}\sum_{r\leq p}\frac{1}{2T}\sum_{k\leq T} \mathbb{1}_{f(\sigma_j^k)(v_r)\neq f_k(v_r)} \\
\geq & \quad \min_{r\leq p}\left(\frac{1}{2T}\sum_{k\leq T} \mathbb{1}_{f(\sigma_j^k)(v_r)\neq f_k(v_r)}\right)
\end{aligned}$$

VCdim($\mathcal{H}$) $= \infty$ (3)

Preuve (fin). Soit r arbitrairement choisi.

On peut partitionner $\{f_k\}_{k \leq T}$ en $T/2$ paires qui ne diffèrent que sur v_r.

Or pour une telle paire $\{k, k'\}$, on a $\sigma_j^k = \sigma_j^{k'}$.

Par conséquent $\mathbb{1}_{f(\sigma_j^k)(v_r) \neq f_k(v_r)} + \mathbb{1}_{f(\sigma_j^{k'})(v_r) \neq f_{k'}(v_r)} = 1$.

D'où $\frac{1}{T} \sum_{k \leq T} \mathbb{1}_{f(\sigma_j^k)(v_r) \neq f_k(v_r)} = \frac{1}{2}$.

Ainsi $\max_{k \leq T}(\mathbf{E}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq \frac{1}{4}$ et $\exists k$ tel que $\mathbf{E}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq \frac{1}{4}$.

Or :

$$\begin{aligned} \mathbf{E}(L_{\mathcal{D},f_k}(f(S_m^k)) \leq & \quad \frac{1}{8}\mathbf{Pr}(L_{\mathcal{D},f_k}(f(S_m^k)) < \frac{1}{8}) + \mathbf{Pr}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq \frac{1}{8}) \\ = & \quad \frac{1}{8} + \frac{7}{8}\mathbf{Pr}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq \frac{1}{8}) \end{aligned}$$

D'où $\mathbf{Pr}(L_{\mathcal{D},f_k}(f(S_m^k)) \geq \frac{1}{8}) \geq \frac{1}{7}$
qui contredit l'hypothèse faite sur l'algorithme.

Croissance d'une classe

La *croissance* de $\mathcal{H}$ est définie par : $\tau_{\mathcal{H}}(m) = \max(|\mathcal{H}_C| \mid C \subset \mathcal{X} \wedge |C| = m)$.

Si VCdim$(\mathcal{H}) = \infty$ alors pour tout m, $\tau_{\mathcal{H}}(m) = 2^m$. Si VCdim$(\mathcal{H}) = d$ alors :

$$\boxed{\text{Pour tout } m,\ \tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}.}$$

Preuve. Il suffit de prouver (par récurrence) que pour tout C :

$$|\mathcal{H}_C| \leq |\{B \subset C \mid \mathcal{H} \text{ brise } B\}|$$

- Le cas $m = 1$ est trivial et on a d'ailleurs l'égalité.
- Soit $C = \{c_1, \ldots, c_m\}$. Notons $C' = \{c_2, \ldots, c_m\}$ et définissons $\mathcal{H}_0$ et $\mathcal{H}_1$ par :

$$\begin{aligned}
\mathcal{H}_0 = &\quad \{(y_2, \ldots, y_m) \mid (\bot, y_2, \ldots, y_m) \in \mathcal{H}_C \vee (\top, y_2, \ldots, y_m) \in \mathcal{H}_C\} \\
\mathcal{H}_1 = &\quad \{(y_2, \ldots, y_m) \mid (\bot, y_2, \ldots, y_m) \in \mathcal{H}_C \wedge (\top, y_2, \ldots, y_m) \in \mathcal{H}_C\}
\end{aligned}$$

Observons que $|\mathcal{H}_C| = |\mathcal{H}_0| + |\mathcal{H}_1|$ et $\mathcal{H}_0 = \mathcal{H}_{C'}$.

Par récurrence, on a : $|\mathcal{H}_0| \leq |\{B \subset C \mid c_1 \notin B \wedge \mathcal{H} \text{ brise } B\}|$.

Soit $\mathcal{H}' = \{h \in \mathcal{H} \mid h_{C'} \in \mathcal{H}_1\}$. $\mathcal{H}_1$ brise $B \subseteq C'$ ssi $\mathcal{H}'$ brise $B \cup \{c_1\}$.

Par conséquent (en utilisant à nouveau la récurrence) :

$$\begin{aligned}
|\mathcal{H}_1| = |\mathcal{H}'_{C'}| \leq &\quad |\{B \subset C' \mid \mathcal{H}' \text{ brise } B\}| = \quad |\{B \subset C' \mid \mathcal{H}' \text{ brise } B \cup \{c_1\}\}| \\
= &\quad |\{B \subset C \mid c_1 \in B \wedge \mathcal{H}' \text{ brise } B\}| \leq \quad |\{B \subset C \mid c_1 \in B \wedge \mathcal{H} \text{ brise } B\}|
\end{aligned}$$

En sommant les inégalités relatives à $|\mathcal{H}_0|$ et $|\mathcal{H}_1|$, on conclut.

Interlude (1)

Si VCdim$(\mathcal{H}) = d$ alors pour tout $m \geq d$,

$$\tau_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d$$

Preuve

$$\left(\frac{d}{m}\right)^d \sum_{i \leq d} \binom{m}{i} \leq \sum_{i \leq d} \left(\frac{d}{m}\right)^i \binom{m}{i} \leq \sum_{i \leq m} \left(\frac{d}{m}\right)^i \binom{m}{i} = \left(1 + \frac{d}{m}\right)^m \leq e^d$$

D'où :

$$\sum_{i \leq d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d$$

Interlude (2)

Soit $X \geq 0$ une v.a. qui vérifie pour tout $t \geq 0$, $\mathbf{Pr}(X \geq t) \leq 2be^{-\frac{t^2}{a^2}}$.
Alors $\mathbf{E}(X) \leq a(3 + \sqrt{\log(b)})$.

Preuve. Soit $t_i = a(i + \sqrt{\log(b)})$ pour $i \in \mathbb{N}$.

$$\begin{aligned}
\mathbf{E}(X) \leq & \quad a(1 + \sqrt{\log(b)}) + \sum_{i \geq 2} t_i \mathbf{Pr}(t_{i-1} < X) \\
\leq & \quad a(1 + \sqrt{\log(b)}) + 2ab \sum_{i \geq 2} (i + \sqrt{\log(b)}) e^{-(i-1+\sqrt{\log(b)})^2} \\
\leq & \quad a(1 + \sqrt{\log(b)}) + 4ab \sum_{i \geq 2} (i - 1 + \sqrt{\log(b)}) e^{-(i-1+\sqrt{\log(b)})^2} \\
\leq & \quad a(1 + \sqrt{\log(b)}) + 4ab \int_{\sqrt{\log(b)}}^{\infty} x e^{-x^2} dx \\
= & \quad a(1 + \sqrt{\log(b)}) + 2ab \left[-e^{-x^2} \right]_{\sqrt{\log(b)}}^{\infty} \\
= & \quad a(3 + \sqrt{\log(b)})
\end{aligned}$$

Croissance et erreur empirique (1)

Le résultat suivant majore (probabilistiquement) la différence entre l'erreur de classification et l'erreur empirique aléatoire à l'aide de $\tau_{\mathcal{H}}$.

Soit $h \in \mathcal{H}$ et $0 < \delta < 1$. Alors :

$$\mathbf{Pr}\left(|L_{\mathcal{D},f}(h) - L_{S_m}(h)| > \frac{\sqrt{2}}{\delta\sqrt{m}}(3 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}\right) \leq \delta$$

Preuve.

En utilisant l'inégalité de Markov,

$\mathbf{Pr}(|L_{\mathcal{D},f}(h) - L_{S_m}(h)| > \frac{\sqrt{2}}{\delta\sqrt{m}}(3 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}) \leq \mathbf{E}(|L_{\mathcal{D},f}(h) - L_{S_m}(h)|)\frac{\delta\sqrt{m}}{\sqrt{2}(3+\sqrt{\log(\tau_{\mathcal{H}}(2m))}}$

Il suffit donc de démontrer que :

$\mathbf{E}(|L_{\mathcal{D},f}(h) - L_{S_m}(h)|) \leq \frac{\sqrt{2}}{\sqrt{m}}(3 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}.$

Par croissance de l'espérance,

$\sup_{h\in\mathcal{H}} \mathbf{E}(|L_{\mathcal{D},f}(h) - L_{S_m}(h)|) \leq \mathbf{E}(\sup_{h\in\mathcal{H}} |L_{\mathcal{D},f}(h) - L_{S_m}(h)|).$

On observe que $L_{\mathcal{D},f}(h) = \mathbf{E}_{S'_m}(L_{S'_m}(h)))$

où S'_m est donnée aléatoire de taille m indépendante de S_m.

Croissance et erreur empirique (2)

Preuve (suite).

Donc, $\mathbf{E}_{S_m}(\sup_{h\in\mathcal{H}}|L_{\mathcal{D},f}(h)-L_{S_m}(h)|)=\mathbf{E}_{S_m}(\sup_{h\in\mathcal{H}}|\mathbf{E}_{S'_m}(L_{S'_m}(h)-L_{S_m}(h))|)$.

Par l'inégalité de Jensen, $|\mathbf{E}_{S'_m}(L_{S'_m}(h)-L_{S_m}(h))|\leq\mathbf{E}_{S'_m}(|L_{S'_m}(h)-L_{S_m}(h)|)$.

D'où :

$$\begin{aligned}\mathbf{E}_{S_m}(\sup_{h\in\mathcal{H}}|L_{\mathcal{D},f}(h)-L_{S_m}(h)|)\leq & \quad \mathbf{E}_{S_m}(\sup_{h\in\mathcal{H}}\mathbf{E}_{S'_m}(|L_{S'_m}(h)-L_{S_m}(h)|))\\ \leq & \quad \mathbf{E}_{S_m,S'_m}(\sup_{h\in\mathcal{H}}|L_{S'_m}(h)-L_{S_m}(h)|)\end{aligned}$$

$S_m=(X_i,f(X_i))_{i\leq m}$, $S'_m=(X'_i,f(X'_i))_{i\leq m}$, $U_{h,i}=\mathbb{1}_{f(X_i)\neq h(X_i)}$, $U'_{h,i}=\mathbb{1}_{f(X'_i)\neq h(X'_i)}$.

On a :

$$\mathbf{E}_{S_m,S'_m}(\sup_{h\in\mathcal{H}}|L_{S'_m}(h)-L_{S_m}(h)|)=\mathbf{E}_{S_m,S'_m}(\sup_{h\in\mathcal{H}}\frac{1}{m}|\sum_{i\leq m}U'_{h,i}-U_{h,i}|)$$

La distribution de $U'_{h,i}-U_{h,i}$ est la même que celle de $U_{h,i}-U'_{h,i}$.

Par conséquent, soit $sgn=(sgn_i)_{i\leq m}$ un vecteur aléatoire dont les composantes sont tirées uniformément dans $\{-1,+1\}$. On a :

$$\mathbf{E}_{S_m,S'_m}(\sup_{h\in\mathcal{H}}|L_{S'_m}(h)-L_{S_m}(h)|)=\mathbf{E}_{S_m,S'_m}\mathbf{E}_{sgn}(\sup_{h\in\mathcal{H}}\frac{1}{m}|\sum_{i\leq m}sgn_i(U'_{h,i}-U_{h,i})|)$$

Croissance et erreur empirique (3)

Preuve (fin). On va majorer $\mathbf{E}_{sgn}(\sup_{h\in\mathcal{H}} \frac{1}{m}|\sum_{i\leq m} sgn_i(U'_{h,i} - U_{h,i})|)$ par une borne indépendante de la réalisation de S_m et S'_m.

Soit $\sigma_m = (x_i, f(x_i))$ et $\sigma'_m = (x'_i, f(x'_i))$ et $C = \{x_i, x'_i\}_{i\leq m}$.

$$\mathbf{E}_{sgn}(\sup_{h\in\mathcal{H}} \frac{1}{m}|\sum_{i\leq m} sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})|)$$

$$= \mathbf{E}_{sgn}(\max_{h\in\mathcal{H}_C} \frac{1}{m}|\sum_{i\leq m} sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})|)$$

$sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})$ est soit uniforme dans $\{-1,+1\}$ soit nulle.

A l'aide des bornes de Chernoff-Hoeffding, on a :

$$\mathbf{Pr}(\frac{1}{m}|\sum_{i\leq m} sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})| \geq \rho) \leq 2e^{-\frac{m\rho^2}{2}}$$

A l'aide de la majoration union-somme, on obtient :

$$\mathbf{Pr}(\max_{h\in\mathcal{H}_C} \frac{1}{m}|\sum_{i\leq m} sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})| \geq \rho) \leq 2|\mathcal{H}_C|e^{-\frac{m\rho^2}{2}}$$

On applique alors le résultat précédent, ce qui fournit :

$$\mathbf{E}(\max_{h\in\mathcal{H}_C} \frac{1}{m}|\sum_{i\leq m} sgn_i(\mathbb{1}_{f(x_i)\neq h(x_i)} - \mathbb{1}_{f(x'_i)\neq h(x'_i)})|) \leq \frac{\sqrt{2}}{\sqrt{m}}(3 + \sqrt{\log(|\mathcal{H}_C|)})$$

$$\leq \frac{\sqrt{2}}{\sqrt{m}}(3 + \sqrt{\log(\tau_{\mathcal{H}}(2m))})$$

$\text{VCdim}(\mathcal{H}) < \infty$

Soit $\mathcal{H}$ telle que $\text{VCdim}(\mathcal{H}) = d < \infty$ et ERM soit implémentable.

Alors $\mathcal{H}$ peut être apprise.

Preuve.

Puisque $\text{VCdim}(\mathcal{H}) = d$, on sait que $\tau_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d$.

Fixons $0 < \delta, \varepsilon < 1$ et soit $z(m) = \frac{\sqrt{2}}{\delta\sqrt{m}} \left(3 + \sqrt{d \log(\frac{2em}{d})}\right)$.

On peut choisir un m tel que $z(m) \leq \varepsilon$ puisque $\lim_{m\to\infty} z(m) = 0$.

Puisque $L_{S_m}(f(S_m)) \leq L_{S_m}(f) = 0$,

par application du résultat précédent, on obtient :

$$\mathbf{Pr}(L_{\mathcal{D},f}(f(S_m)) > \varepsilon) = \mathbf{Pr}(|L_{\mathcal{D},f}(f(S_m)) - L_{S_m}(f(S_m))| > \varepsilon) \leq \delta$$

Plan

Théorème de Radon

Soit $S = \{a_1, \ldots, a_{n+2}\}$ un ensemble de $n+2$ points de $\mathbb{R}^n$.
Alors il existe une partition de $S = S_1 \uplus S_2$ telle que
les enveloppes convexes de S_1 et S_2 aient une intersection non vide.

Preuve. Soit le système d'équations défini par :

$$\sum_{i \leq n+2} \lambda_i a_i = 0 \qquad \wedge \qquad \sum_{i \leq n+2} \lambda_i = 0$$

C'est un système d'équations linéaires de $n+1$ équations à $n+2$ inconnues.

Il admet donc une solution non nulle, notée aussi $\{\lambda_i\}_{i \leq n+2}$.

Soit $I_1 = \{i \mid \lambda_i > 0\}_{i \leq n+2}$, $I_2 = \{1, \ldots, n+2\} \setminus I_1$, $S_1 = \{a_i\}_{i \in I_1}$ et $S_2 = \{a_i\}_{i \in I_2}$.

Puisque $\sum_{i \leq n+2} \lambda_j = 0$, S_1 et S_2 sont non vides.

$\sum_{i \in I_1} \lambda_i = -\sum_{i \in I_2} \lambda_i > 0$. Par conséquent :

$$\sum_{i \in I_1} \frac{\lambda_i}{\sum_{i \in I_1} \lambda_i} a_i = \sum_{i \in I_2} \frac{-\lambda_i}{\sum_{i \in I_2} -\lambda_i} a_i$$

et le point ainsi défini appartient à l'intersection des enveloppes convexes.

VC-dimension des hyperplans

Soit $\mathcal{H}$ la classe des hypothèses définie par les hyperplans affines de $\mathbb{R}^n$.
Alors VCdim$(\mathcal{H}) = n + 1$.

Preuve. Considérons $S = \{a_1, \ldots, a_{n+2}\}$ un ensemble de $n + 2$ points distincts.

Appliquons le théorème de Radon et supposons qu'il existe h telle que :

$$\forall a_i \in S_1, h(a_i) = 1 \wedge \forall a_i \in S_2, h(a_i) = 0$$

Tout $h \in \mathcal{H}$ qui affecte une valeur commune à un ensemble de points affecte cette même valeur à tout point de son enveloppe convexe.

Soit x appartenant à l'intersection des enveloppes convexes.

Alors $h(x) = 1 \wedge h(x) = 0$ ce qui est absurde.

Par conséquent VCdim$(\mathcal{H}) \leq n + 1$.

Soit $S = \{(a_1, b_1), \ldots, (a_{n+1}, b_{n+1})\}$ et le système d'équations suivant où w un vecteur de dimension n et a un scalaire sont les inconnues :

$$\forall i \leq n + 1 \; w \cdot a_i - a = (-1)^{1-b_i}$$

C'est un système d'équations linéaires de $n + 1$ équations à $n + 1$ inconnues.

Il admet donc une solution qui fournit la fonction de $\mathcal{H}$ requise.

Par conséquent VCdim$(\mathcal{H}) \geq n + 1$.

Réseaux de neurones

On dit qu'un graphe orienté $G = (V, E)$ est *en couches* de taille (n, s, r) si :

- Les sommets sont partitionnés en couches $V = \biguplus_{\ell \leq L} V_\ell$ avec $V_L = \{root\}$, $V_0 = \{1, \ldots, n\}$ et $|V \setminus V_0| = s$;
- Pour tout arc $(u, v) \in E$, il existe ℓ tel que $u \in V_\ell$ et $v \in V_{\ell+1}$;
- Tout sommet (dit *interne*) $v \in V \setminus V_0$ a un degré entrant égal à r.

Soit $\mathcal{H}$ une classe d'hypothèses pour $\mathbb{R}^r$ et G un graphe en couches.
Alors pour tout sommet interne v, $\mathcal{H}_v$ est une classe d'hypothèses pour $\mathbb{R}^n$ définie inductivement par :

- Si $v \in V_1$ et $i_1, \ldots, i_r$ sont les sommets tels que $(i_k, v) \in E$, $\mathcal{H}_v = \{h' \mid \exists h \in \mathcal{H}\ h'(x_1, \ldots, x_n) = h(x_{i_1}, \ldots, x_{i_r})\}$;
- Si $v \in V_{l+1}$ et $v_1, \ldots, v_r \in V_l$ sont les sommets tels que $(v_k, v) \in E$, $\mathcal{H}_v = \{h' \mid \exists h \in \mathcal{H}\ \exists h_1 \in \mathcal{H}_{v_1}\ \ldots \exists h_r \in \mathcal{H}_{v_r}$
$h'(x_1, \ldots, x_n) = h(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n))\}$.

On note $\mathcal{H}_G \stackrel{\text{def}}{=} \mathcal{H}_{root}$.

VC-dimension des réseaux de neurones

Soit $\mathcal{H}$ une classe d'hypothèses pour $\mathbb{R}^r$ telle que $\text{VCdim}(\mathcal{H}) = d$ et G un graphe en couches de taille (n, s, r) avec $s \geq 2$. Alors : $\text{VCdim}(\mathcal{H}_G) < 2ds\log_2(es)$.

Preuve. $h \in \mathcal{H}_G$ est défini par $(h_v)_{v \in V \setminus V_0} \in \mathcal{H}^{V \setminus V_0}$. Soit $\vec{x}_1, \ldots, \vec{x}_m \in \mathbb{R}^n$.

On peut étiqueter chaque sommet interne avec un vecteur de $\{0,1\}^m$ correspondant à l'évaluation de $\vec{x}_1, \ldots, \vec{x}_m$ qu'on nomme un *calcul*.

Le nombre de calculs différents majore le nombre de résultats possibles pour les $\vec{x}_1, \ldots, \vec{x}_m \in R^n$.

On va majorer c_s le nombre de calculs différents par $\left(\frac{em}{d}\right)^{ds}$.

Soit G' le graphe privé de $root$. G' a au plus c_{s-1} calculs différents.

Chacun de ces calculs complété par l'évaluation en $root$ d'une fonction de $\mathcal{H}$ fournit au plus $\tau_{\mathcal{H}}(m)$ calculs différents.

D'où $c_s \leq c_{s-1}\tau_{\mathcal{H}}(m)$ et par récurrence $c_s \leq \tau_{\mathcal{H}}(m)^s \leq \left(\frac{em}{d}\right)^{ds}$.

Choisissons $m = 2ds\log_2(es)$. Alors :

$$\left(\frac{em}{d}\right)^{ds} = (2es\log_2(es))^{ds} < \left((es)^2\right)^{ds} = 2^{2ds\log_2(es)} = 2^m$$

Plan

Préliminaires

Afin d'étudier la complexité de l'apprentissage,

nous partitionnons le domaine et l'ensemble des classifieurs :

- $\mathcal{X} = \biguplus_{n \in N} \mathcal{X}_n$ avec $x \in \mathcal{X}_n$ de taille polynomiale vis à vis de n ;
- $\mathcal{H} = \biguplus_{n \in N} \mathcal{H}_n$ avec $\mathcal{H}_n \subset 2^{\mathcal{X}_n}$ avec $s_n = \max(|h| \mid h \in \mathcal{H}_n)$.

Un algorithme d'apprentissage prend en entrée n, $Sample$, δ et ε.

$Sample(n)$ est une fonction d'échantillonnage aléatoire $(x, f(x))$ avec $x \in \mathcal{X}_n$

$f \in \mathcal{H}_n$ qui opère en temps polynomial vis à vis de n et de s_n.

Un algorithme d'apprentissage *efficace*
est un algorithme en temps polynomial vis à vis de n, s_n, $\frac{1}{\delta}$ et $\frac{1}{\varepsilon}$.

Transformation d'algorithmes

Soit un algorithme d'apprentissage $\mathcal{A}$ tel que :

- la garantie de précision est conditionnée par la réalisation de Ev avec $\mathbf{Pr}(Ev) \geq 1 - \delta$;
- $\mathbf{E}(T|Ev) \leq t(n, s_n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ où T est le temps d'exécution de $\mathcal{A}$ et t est un polynôme.

L'algorithme d'apprentissage (efficace) $\mathcal{A}'$ est défini ainsi :

- $\mathcal{A}'$ exécute $\mathcal{A}$ en incrémentant cpt, un compteur d'instructions exécutées ;
- Si $cpt > \frac{t(n,s_n,\frac{1}{\varepsilon},\frac{1}{\delta})}{\delta}$ alors $\mathcal{A}'$ s'arrête et renvoie un classifieur arbitraire ;
- Sinon $\mathcal{A}'$ renvoie le résultat de $\mathcal{A}$.

Alors :

- Le temps d'exécution de $\mathcal{A}'$ est borné par $C\frac{t(n,s_n,\frac{1}{\varepsilon},\frac{1}{\delta})}{\delta}$ pour un certain C ;
- En utilisant la borne de Markov, $\mathbf{Pr}(cpt > \frac{t(n,s_n,\frac{1}{\varepsilon},\frac{1}{\delta})}{\delta}|Ev) \leq \delta$;
- Par conséquent, $\mathcal{A}'$ apprend avec une précision ε et un seuil 2δ.

Apprentissage de conjonctions

$\mathcal{X}_n = 2^n$ et $\mathcal{H}_n = \{\ell_1 \wedge \ldots \wedge \ell_k \mid \forall j \leq k\ \ell_j \in \{x_i, \overline{x}_i\}_{i \leq n}\}$.

$m \leftarrow \lceil \frac{2n}{\varepsilon}(\log(2n) + \log(\frac{1}{\delta})) \rceil$
$E \leftarrow \{x_i, \overline{x}_i\}_{i \leq n}$
For j **from** 1 **to** m **do**
  $(\mathbf{v}, b) \leftarrow Sample(n)$
  If b **then**
    For i **from** 1 **to** n **do**
      If $\mathbf{v}[i]$ **then** $E \leftarrow E \setminus \{\overline{x}_i\}$ **else** $E \leftarrow E \setminus \{x_i\}$
Return$(\bigwedge_{\ell \in E} \ell)$

Soit ψ, la formule à apprendre.

L'algorithme maintient un ensemble de littéraux qui contient ceux de ψ.

Après un échantillon *positif* $(\mathbf{v}, \top)$, il retire les littéraux contredisant l'évaluation.

D'où pour tout $\mathbf{v}$, $\mathbf{v} \models \bigwedge_{\ell \in E} \ell$ implique $\mathbf{v} \models \psi$.

Analyse de l'algorithme

Analyse de la précision

Soit $\mathbf{v}_n$ une donnée aléatoire et ℓ un littéral. On note $p(\ell) = \mathbf{Pr}(\mathbf{v}_n \not\models \ell)$.
On dit que ℓ est *pertinent* si $p(\ell) > \frac{\varepsilon}{2n}$ et ℓ n'apparait pas dans ψ.

Supposons que E ne contienne pas de littéraux pertinents,
alors l'erreur commise est majorée par $\sum_{\ell \in E} p(\ell) \leq 2n\frac{\varepsilon}{2n} = \varepsilon$.

Analyse du seuil

Soit ℓ un littéral pertinent, $\mathbf{Pr}(\ell \in E) \leq (1 - \frac{\varepsilon}{2n})^m$.
La probabilité qu'un littéral pertinent appartienne à E est donc majorée par :

$$2n(1 - \frac{\varepsilon}{2n})^m \leq 2ne^{-\frac{m\varepsilon}{2n}}$$

Il faut choisir m tel que :

$$2ne^{-\frac{m\varepsilon}{2n}} \leq \delta$$

Soit :

$$m \geq \frac{2n}{\varepsilon}(\log(2n) + \log(\frac{1}{\delta}))$$

Apprentissage de formules 3CNF

Une formule 3CNF est une conjonction de clauses disjonctives de 3 littéraux :

$$\psi = \bigwedge_{j \leq k} \ell_j^1 \vee \ell_j^2 \vee \ell_j^3$$

Le nombre de clauses est majoré par $(2n)^3$.

Nous réduisons l'apprentissage d'une formule 3CNF
à l'apprentissage d'une conjonction de variables.

Pour chaque clause $\ell \vee \ell' \vee \ell''$, on définit une variable $y_{\ell,\ell',\ell''}$.

On adapte l'algorithme précédent ainsi :

- Initialement E ne contient que les variables $y_{\ell,\ell',\ell''}$ et pas leur négation ;
- Pour chaque donnée $\mathbf{v} \in 2^n$, $\mathbf{v} \models y_{\ell,\ell',\ell''}$ ssi $\mathbf{v} \models \ell \vee \ell' \vee \ell''$;
- A la fin de l'algorithme, on renvoie la formule $\bigwedge_{y_{\ell,\ell',\ell''} \in E} \ell \vee \ell' \vee \ell''$.

Analyse. Comme précédemment avec $p(\ell \vee \ell' \vee \ell') = \mathbf{Pr}(\mathbf{v}_n \not\models \ell \vee \ell' \vee \ell')$
et $\ell \vee \ell' \vee \ell'$ *pertinente* si $p(\ell \vee \ell' \vee \ell') > \frac{\varepsilon}{(2n)^3}$ et $\ell \vee \ell' \vee \ell'$ n'apparait pas dans ψ.

La classe RP

Un problème appartient à RP s'il existe une machine de Turing $\mathcal{M}$ non déterministe opérant en temps polynomial telle que :

- si l'instance du problème est négative alors tous les calculs renvoient $\bot$;
- si l'instance est positive alors au moins la moitié des calculs renvoient $\top$.

PTIME $\subseteq$ RP $\subseteq$ NP et on conjecture fortement que RP $\neq$ NP.

On peut remplacer $\frac{1}{2}$ par $\frac{1}{pol(n)}$ où pol est un polynôme et n la taille de l'instance. $\mathcal{M}'$ exécute $pol(n)$ fois $\mathcal{M}$ et renvoie $\top$ si l'une des réponses est $\top$.

Formulation probabiliste. Un problème appartient à RP s'il existe un algorithme probabiliste opérant en temps polynomial tel que :

- si l'instance du problème est négative alors toute exécution renvoie $\bot$;
- si l'instance est positive alors la probabilité que l'algorithme renvoie $\top$ est supérieure ou égale à $\frac{1}{pol(n)}$.

Consistance de donnée (1)

Soit $\mathcal{H} = \biguplus_{n\in\mathbb{N}} \mathcal{H}_n$ telle que $(s_n)_{n\in\mathbb{N}}$ soit polynômialement bornée et qui peut être apprise efficacement.

Le problème de *consistance d'une donnée*

- prend en entrée n, une donnée $\sigma = ((x_i, y_i))_{i\leq m} \in (\mathcal{X}_n \times \{\bot, \top\})^m$
- et décide s'il existe $f \in \mathcal{H}_n$ telle que pour tout i, $y_i = f(x_i)$.

Le problème de consistance d'une donnée appartient à RP.

Preuve.

Soit $\mathcal{A}$, l'algorithme d'apprentissage efficace de $\mathcal{H}$.
Fixons $\delta = \frac{1}{2}$ et $\varepsilon = \frac{1}{2m}$. $\mathcal{A}$ opère en temps polynomial par rapport à n et m.
Soit l'algorithme de décision $\mathcal{A}'$ défini ainsi :

- Il prend en entrée une donnée $((x_i, y_i))_{i\leq m}$;
- Il exécute $\mathcal{A}$ avec les paramètres spécifiés plus haut en choisissant la distribution uniforme $\mathcal{U}$ sur $(x_i)_{i\leq m}$.
- Soit h la fonction renvoyée $\mathcal{A}$. $\mathcal{A}'$ renvoie $\top$ si pour tout i, $h(x_i) = y_i$.

Consistance de donnée (2)

Preuve (suite).
Etablissons la correction (probabiliste) de cet algorithme.

- Soit σ n'est pas consistante. alors quelque soit la fonction h renvoyée, il existe un i, tel que $h(x_i) \neq y_i$.

- Soit σ est consistante et f telle que pour tout i, $y_i = f(x_i)$.
  Dans ce cas, $\mathcal{A}$ apprend la fonction f avec la distribution $\mathcal{U}$ sur $(x_i)_{i \leq m}$.
  Avec une probabilité au moins $\frac{1}{2}$, h renvoyée par $\mathcal{A}$ vérifie $L_{\mathcal{U},f}(h) \leq \frac{1}{2m}$.
  S'il existait un i tel que $f(x_i) \neq h(x_i)$ alors $L_{\mathcal{U},f}(h) \geq \frac{1}{m}$.
  Donc avec probabilité au moins $\frac{1}{2}$, pour tout i, $h(x_i) = f(x_i) = y_i$.

Formules 3-term DNF et consistance (1)

Une formule 3-term DNF est une disjonction de trois conjonctions de littéraux.

$$\psi = \bigvee_{p \leq 3} \bigwedge_{j \leq k_p} \ell_j^p$$

Le problème de la consistance d'une donnée
associé aux formules 3-term DNF est NP-complet.

Preuve. Nous réduisons le problème de la 3-coloration d'un graphe au problème de la consistance d'une donnée associé aux formules 3-term DNF.

Soit $G = (V, E)$ tel que $V = \{1, \ldots, n\}$ et un ensemble de couleurs $\mathcal{C} = \{R, B, J\}$.

Le problème de la 3-coloration consiste en l'existence

d'une coloration $col : V \to \mathcal{C}$ telle que pour tout $\{u, v\} \in E$, $col(u) \neq col(v)$.

$\mathcal{X}_n = \{\bot, \top\}^n$.

Soit $\bot_i$ tel que $\bot_i[i] = \bot$ et $\bot_i[j] = \top$ pour $i \neq j$.

Soit $\bot_{i,j}$ tel que $\bot_{i,j}[i] = \bot_{i,j}[j] = \bot$ et $\bot_{i,j}[k] = \top$ pour $k \notin \{i, j\}$.

Alors $\sigma = \sigma^+ \uplus \sigma^-$ avec $\sigma^+ = \{(\bot_i, \top)\}_{i \in V}$ et $\sigma^- = \{(\bot_{i,j}, \bot)\}_{\{i,j\} \in E}$.

Cette réduction s'opère en temps polynomial.

Formules 3-term DNF et consistance (2)

Preuve (suite). Supposons qu'il existe une fonction de coloration col.
Soit $\varphi = \varphi_R \vee \varphi_B \vee \varphi_J$ avec pour $C \in \mathcal{C}$, $\varphi_C = \bigwedge_{col(i) \neq C} x_i$.

- Pour tout $i \in V$, $\perp_i \models \varphi_{col(i)}$;
- Pour tout $\{i, j\} \in E$ et $C \in \mathcal{C}$, $\perp_{i,j} \not\models \varphi_C$ car soit $col(i) \neq C$ soit $col(j) \neq C$.

σ est donc consistante avec φ.

Supposons que σ soit consistante avec une formule $\varphi = \varphi_R \vee \varphi_B \vee \varphi_J$.
Définissons la fonction col ainsi :
Si $\perp_i \models \varphi_R$ alors $col(i) = R$ sinon si $\perp_i \models \varphi_B$ alors $col(i) = B$ sinon $col(i) = J$.
Puisque σ est consistante avec φ pour tout i, $\perp_i \models \varphi_{col(i)}$.

Supposons qu'il existe $\{i, j\} \in E$ tel que $C \stackrel{\text{def}}{=} col(i) = col(j)$.
Puisque $\perp_i \models \varphi_C$, x_i et $\overline{x}_k$ pour $k \neq i$ n'apparaissent pas dans φ_C.
Puisque $\perp_j \models \varphi_C$, x_j n'apparaît pas dans φ_C.
Par conséquent $\perp_{i,j} \models \varphi_C$ ce qui contredit la consistance de σ.

Conséquence.

Sauf si RP = NP, une formule 3-term DNF ne peut être apprise efficacement.

Un paradoxe apparent

Une formule 3-term DNF se transforme en temps polynomial

en une formule équivalente 3CNF en distribuant les $\wedge$ sur les $\vee$:

$$\bigvee_{p\leq 3} \bigwedge_{j\leq k_p} \ell_j^p \equiv \bigwedge_{j_1\leq k_1, j_2\leq k_2, j_3\leq k_3} \ell_{j_1}^1 \vee \ell_{j_2}^2 \vee \ell_{j_3}^3$$

Or les formules 3CNF peuvent être apprises efficacement !

Explication.

En choisissant une classe d'hypothèses au moins aussi expressive

que la classe originelle et dont la syntaxe se prête plus à l'apprentissage,

on peut augmenter l'efficacité de l'apprentissage.

Plan

Apprentissage faible

$\mathcal{H}$ peut être faiblement (et efficacement) apprise
s'il existe des polynômes positifs p, q, et un algorithme d'apprentissage (efficace) telle qu'en notant H la sortie de l'algorithme alors :

$$\mathbf{Pr}\left(L(H) > \frac{1}{2} - \frac{1}{p(n, s_n)}\right) \leq 1 - \frac{1}{q(n, s_n)}$$

Un résultat profond et difficile.

La capacité d'apprentissage faible (efficace) est « équivalente »
à la capacité d'apprentissage (efficace).

Robert Schapire et Yoav Freund, concepteurs d'ADABOOST

Prix Gödel 2003 et Prix Paris Kanellakis 2004.

La preuve qui suit est issue de la thèse de Robert Schapire.

Augmentation de la confiance

De **WeakLearn** avec erreur : $\frac{1}{2} - \frac{1}{p(n,s_n)}$ et confiance : $\frac{1}{q(n,s_n)}$

à **MidLearn** avec erreur : $\frac{1}{2} - \frac{1}{2p(n,s_n)}$ et confiance : $1 - \delta$

Présentation.

- k appels à **WeakLearn** pour obtenir k classifieurs
- Génération de m échantillons pour évaluer l'erreur empirique des classifieurs
- Choix du « meilleur » classifieur

$k \leftarrow \lceil q(n, s_n) \log(\frac{2}{\delta}) \rceil$; $m \leftarrow \lceil 8p(n, s_n)^2 \log(\frac{4k}{\delta}) \rceil$; $opt \leftarrow 1$

For i **from** 1 **to** k **do** $h[i] \leftarrow \mathbf{WeakLearn}(n, Sample)$; $cpt[i] \leftarrow 0$

For j **from** 1 **to** m **do**

$(x, y) \leftarrow Sample(n)$

For i **from** 1 **to** k **do if** $y \neq h[i](x)$ **then** $cpt[i] \leftarrow cpt[i] + 1$

For i **from** 2 **to** k **do if** $cpt[i] < cpt[opt]$ **then** $opt \leftarrow i$

Return$(h[opt])$

Analyse de l'algorithme

Soit $\varepsilon_n = \frac{1}{2} - \frac{1}{p(n,s_n)}$ et $\delta_n = 1 - \frac{1}{q(n,s_n)}$

$$\mathbf{Pr}(\min_{i \leq k}(L(h[i])) > \varepsilon_n) \leq \delta_n^k \leq \frac{\delta}{2} \text{ par le choix de } k.$$

Soit S_m l'échantillon aléatoire de taille m.

Par le choix de m et application des bornes de Hoeffding pour tout i,

$$\mathbf{Pr}(|L_{S_m}(h[i]) - L(h[i])| > \frac{1}{4p(n,s_n)}) \leq \frac{\delta}{2k}$$

D'où par application de la majoration union-somme

$$\mathbf{Pr}(\max_{i \leq k}(|L_{S_m}(h[i]) - L(h[i])|) > \frac{1}{4p(n,s_n)}) \leq \frac{\delta}{2}$$

Soit $h[optr]$ le classifieur avec l'erreur minimale avec probabilité $1 - \delta$:

$$\begin{aligned} L(h[opt]) = & (L(h[opt]) - L_{S_m}(h[opt])) + (L_{S_m}(h[opt]) - L_{S_m}(h[optr])) \\ & + (L_{S_m}(h[optr]) - L(h[optr])) + L(h[optr]) \\ \leq & \frac{1}{4p(n,s_n)} + \frac{1}{4p(n,s_n)} + \frac{1}{2} - \frac{1}{p(n,s_n)} = \frac{1}{2} - \frac{1}{2p(n,s_n)} \end{aligned}$$

Echantillonnages spécifiques

$Sample2$ prend en entrée un classifieur h_1 et $Sample$.

Il affecte aléatoirement b. Si $b = \top$ (resp. $\bot$), $Sample2$ appelle $Sample$ jusqu'à ce que h_1 et f soient égales (resp. différentes) sur l'échantillon.

$Sample2(h_1, Sample)(n)$

$b \leftarrow \mathbf{Random}(0.5)$

While true do

  $(x, y) \leftarrow Sample(n)$

  If $(b \wedge y = h_1(x)) \vee (\neg b \wedge y \neq h_1(x))$ **then return**(x, y)

$Sample3$ prend en entrée deux classifieurs h_1, h_2 et $Sample$.

$Sample3$ appelle $Sample$ jusqu'à ce que h_1 et h_2 soient différentes sur l'échantillon.

$Sample3(h_1, h_2, Sample)(n)$

While true do

  $(x, y) \leftarrow Sample(n)$

  If $h_1(x) \neq h_2(x)$ **then return**(x, y)

Un classifieur par majorité

Soit trois classifieurs h_1, h_2, h_3.

Alors le classifieur $\mathbf{maj}(h_1, h_2, h_3)$ renvoie la réponse majoritaire.

Si $h_1, h_2, h_3 \in \mathcal{H}_n$ on n'a pas nécessairement $\mathbf{maj}(h_1, h_2, h_3) \in \mathcal{H}_n$.

Soit $\mathbf{Learn}$ un algorithme d'apprentissage.

Alors le (pseudo-)algorithme $\mathbf{MajLearn}$ est défini ainsi :

$h_1 \leftarrow \mathbf{Learn}(n, Sample)$
$h_2 \leftarrow \mathbf{Learn}(n, Sample2(h_1, Sample))$
$h_3 \leftarrow \mathbf{Learn}(n, Sample3(h_1, h_2, Sample))$
Return$(\mathbf{maj}(h_1, h_2, h_3))$

Observation. $\mathbf{MajLearn}$ ne se termine pas nécessairement.

Analyse du pseudo-algorithme (1)

Notons $\mathcal{D}_i$ la distribution de la $i$ème exécution (d'où $\mathcal{D}_1 = \mathcal{D}$) et $\beta_i = L_{\mathcal{D}_i,f}(h_i)$.

- si $h_1(x) = f(x)$ alors $\mathbf{Pr}_{\mathcal{D}_2}(X = x) = \mathbf{Pr}(b = \top)\mathbf{Pr}_{\mathcal{D}}(X = x | h_1(X) = f(X)) = \frac{1}{2}\frac{\mathbf{Pr}_{\mathcal{D}}(X=x)}{1-\beta_1}$
- sinon $\mathbf{Pr}_{\mathcal{D}_2}(X = x) = \frac{1}{2}\frac{\mathbf{Pr}_{\mathcal{D}}(X=x)}{\beta_1}$

Par conséquent, pour tout $\mathcal{X}' \subseteq \mathcal{X}$, on a :

$$\begin{aligned}\mathbf{Pr}_{\mathcal{D}}(X \in \mathcal{X}') = \quad & 2(1-\beta_1)\mathbf{Pr}_{\mathcal{D}_2}(X \in \mathcal{X}' \wedge h_1(X) = f(X)) \\ & + 2\beta_1\mathbf{Pr}_{\mathcal{D}_2}(X \in \mathcal{X}' \wedge h_1(X) \neq f(X)) \end{aligned} \quad (1)$$

Soit $g(\beta) = 3\beta^2 - 2\beta^3$. g est bijective et croissante de $[0, 0.5]$ dans $[0, 0.5]$ avec $g(\beta) < \beta$ pour tout $0 < \beta < 0.5$.

Si $\max(\beta_1, \beta_2, \beta_3) \leq \beta \leq \frac{1}{2}$ alors $L_{\mathcal{D},f}(h) \leq g(\beta)$.

Preuve. Observons que (les probabilités étant exprimées par rapport à $\mathcal{D}$) :

$$\begin{aligned} L_{\mathcal{D},f}(h) = & \quad \mathbf{Pr}(h_1(X) \neq f(X) \wedge h_2(X) \neq f(X)) \\ & + \mathbf{Pr}(h_3(X) \neq f(X) \mid h_1(X) \neq h_2(X))\mathbf{Pr}(h_1(X) \neq h_2(X)) \\ = & \quad \mathbf{Pr}(h_1(X) \neq f(X) \wedge h_2(X) \neq f(X)) + \beta_3\mathbf{Pr}(h_1(X) \neq h_2(X)) \\ \leq & \quad \mathbf{Pr}(h_1(X) \neq f(X) \wedge h_2(X) \neq f(X)) + \beta\mathbf{Pr}(h_1(X) \neq h_2(X)) \end{aligned}$$

Analyse du pseudo-algorithme (2)

Observons que $\beta_2 = \gamma_1 + \gamma_2$ avec :

$$\gamma_1 = \mathbf{Pr}_{\mathcal{D}_2}(h_1(X) = f(X) \wedge h_2(X) \neq f(X)) \leq \frac{1}{2}$$

$$\gamma_2 = \mathbf{Pr}_{\mathcal{D}_2}(h_1(X) \neq f(X) \wedge h_2(X) \neq f(X)) \leq \frac{1}{2}$$

A l'aide de l'équation 1, $\mathbf{Pr}(h_1(X) = f(X) \wedge h_2(X) \neq f(X)) = 2(1-\beta_1)\gamma_1$.

Par définition de $\mathcal{D}_2$, $\mathbf{Pr}_{\mathcal{D}_2}(h_1(X) \neq f(X) \wedge h_2(X) = f(X)) = \frac{1}{2} - \gamma_2$.
A l'aide de l'équation 1, $\mathbf{Pr}(h_1(X) \neq f(X) \wedge h_2(X) = f(X)) = 2\beta_1(\frac{1}{2} - \gamma_2)$.

D'où $\mathbf{Pr}(h_1(X) \neq h_2(X)) = 2(1-\beta_1)\gamma_1 + 2\beta_1(\frac{1}{2} - \gamma_2)$.

De même à l'aide de l'équation 1, $\mathbf{Pr}(h_1(X) \neq f(X) \wedge h_2(X) \neq f(X)) = 2\beta_1\gamma_2$.

Par conséquent,

$$\begin{aligned}
L_{\mathcal{D},f}(h) &\leq 2\beta_1\gamma_2 + \beta(2(1-\beta_1)\gamma_1 + 2\beta_1(\frac{1}{2} - \gamma_2)) \\
&= \beta_1\beta(1-2\gamma_1) + 2\beta_1\gamma_2(1-\beta) + 2\gamma_1\beta \\
&\leq \beta^2(1-2\gamma_1) + 2\beta\gamma_2(1-\beta) + 2\gamma_1\beta \\
&= \beta^2 + 2\beta(1-\beta)(\gamma_1+\gamma_2) \\
&\leq 3\beta^2 - 2\beta^3
\end{aligned}$$

Interlude

Soit $k \geq \lceil \log_{\frac{11}{8}}(\frac{p(n,s_n)}{4}) \rceil + \lceil \log_2(\log_{\frac{4}{3}}(\frac{1}{\varepsilon})) \rceil$. Alors $g^{(k)}(\frac{1}{2} - p(n,s_n)) \leq \varepsilon$.

Preuve. On remarque que :

$$\frac{1}{2} - g(\alpha) = \frac{1}{2} - 3\alpha^2 + 2\alpha^3 = (\frac{1}{2} - \alpha)(1 + 2\alpha - 2\alpha^2)$$

Lorsque $\frac{1}{4} \leq \alpha \leq \frac{1}{2}$, $(1 + 2\alpha - 2\alpha^2) \geq \frac{11}{8}$.

Par conséquent pour $k' \geq \lceil \log_{\frac{11}{8}}(\frac{p(n,s_n)}{4}) \rceil$, $g^{(k')}(\frac{1}{2} - p(n,s_n)) \leq \frac{1}{4}$.

On remarque que $g(\alpha) \leq 3\alpha^2$ et par conséquent pour tout k, $g^{(k)}(\alpha) \leq \frac{(3\alpha)^{2^k}}{3}$.

Lorsque $\alpha \leq \frac{1}{4}$, cela implique que $g^{(k)}(\alpha) \leq (\frac{3}{4})^{2^k}$.

Par conséquent pour $k'' \geq \lceil \log_2(\log_{\frac{4}{3}}(\frac{1}{\varepsilon})) \rceil$, $g^{(k'')}(\alpha) \leq \varepsilon$.

On note $N = 3^{\lceil \log_{\frac{11}{8}}(\frac{p(n,s_n)}{4}) \rceil + \lceil \log_2(\log_{\frac{4}{3}}(\frac{1}{\varepsilon})) \rceil}$.

N croît polynômialement en fonction de n, s_n et $\frac{1}{\varepsilon}$.

Diminution de l'erreur

De **MidLearn** avec erreur $\frac{1}{2} - \frac{1}{p(n,s_n)}$ et confiance $1-\delta$

à **StrongLearn** avec erreur ε et confiance $1-\delta$

$\delta' \leftarrow \frac{\delta}{2N}$; **Return RecLearn**$(n, Sample, \delta', \varepsilon)$

RecLearn$(n, Sample', \delta', \varepsilon')$

If $\varepsilon' \geq \frac{1}{2} - \frac{1}{p(n,s_n)}$ **then Return MidLearn**$(n, Sample', \delta')$

$\varepsilon'' \leftarrow g^{-1}(\varepsilon')$

$h_1 \leftarrow \mathbf{RecLearn}(n, Sample', \delta', \varepsilon'')$; $err_1 \leftarrow \mathbf{ErrEmp}(h_1, Sample', \frac{\varepsilon'}{3}, \delta')$

If $err_1 \leq \frac{2\varepsilon'}{3}$ **then Return**(h_1)

$h_2 \leftarrow \mathbf{RecLearn}(n, Sample2(h_1, Sample'), \delta', \varepsilon'')$

$\tau \leftarrow \frac{(1-2\varepsilon')\varepsilon'}{8}$; $err_2 \leftarrow \mathbf{ErrEmp}(h_2, Sample', \tau, \delta')$

If $err_2 \leq \varepsilon - \tau$ **then Return**(h_2)

$h_3 \leftarrow \mathbf{RecLearn}(n, Sample3(h_1, h_2, Sample'), \delta', \varepsilon'')$

Return maj(h_1, h_2, h_3)

ErrEmp$(h^*, Sample^*, \varepsilon^*, \delta^*)$ évalue l'erreur empirique de h^* vis à vis de $Sample^*$ avec précision ε^* et confiance $1-\delta^*$.

Garantie probabiliste

D'après l'étude de la fonction g, le nombre d'appels à $\mathbf{RecLearn}$ est borné par N.

$\mathbf{RecLearn}$ fait au plus deux appels à $\mathbf{ErrEmp}$ ou un appel à $\mathbf{MidLearn}$.

Soit Ev, l'évenement correspondant à la satisfaction de la précision lors de tous ces appels. Alors $\mathbf{Pr}(Ev) \geq 1 - \delta$.

Analyse de la précision *conditionnée par la réalisation* de Ev :

La fonction renvoyée par $\mathbf{RecLearn}$ a une erreur bornée par ε'.

Preuve.

- Cas de base $\varepsilon' \geq \frac{1}{2} - \frac{1}{p(n,s_n)}$: établi par hypothèse sur $\mathbf{MidLearn}$.
- Induction
  - Si h_1 est renvoyée alors $err_1 \leq \frac{2\varepsilon'}{3}$, d'où $L_{\mathcal{D},f}(h_1) \leq \varepsilon'$.
  - Si h_2 est renvoyée alors $err_2 \leq \varepsilon' - \tau$, d'où $L_{\mathcal{D},f}(h_2) \leq \varepsilon'$.
  - Par induction, pour tout $i \leq 3$, $L_{\mathcal{D}_i,f}(h_i) \leq g^{-1}(\varepsilon')$.
    D'où si $\mathbf{maj}(h_1, h_2, h_3)$ est renvoyée alors $L_{\mathcal{D},f}(\mathbf{maj}(h_1, h_2, h_3)) \leq \varepsilon'$.

Analyse de la complexité

Objectif. Soit T le temps aléatoire d'exécution.

1. Etablir que $\mathbf{E}(T \mid Ev)$ est polynomialement borné ;
2. Appliquer la transformation en un algorithme efficace d'apprentissage puisque $\mathbf{Pr}(Ev) \geq 1 - \delta$.

Observation. Puisqu'il y a un nombre polynomial d'appels à $\mathbf{RecLearn}$, il suffit d'analyser la complexité d'un appel à $\mathbf{RecLearn}$.

Cette complexité est polynomiale en nombre d'appels à $Sample'$ (passée en paramètre), $Sample2(h_1, Sample')$ et $Sample3(h_1, h_2, Sample')$.

Nous allons établir que :

- le nombre moyen d'appels à $Sample'$ par $Sample2(h_1, Sample')$ et à $Sample3(h_1, h_2, Sample')$ est polynomialement borné ;
- le temps moyen d'un appel à un $Sample'$ passé en paramètre est polynomialement borné.

Analyse de $Sample2$ et $Sample3$ (1)

Soit $\mathcal{D}'$ la distribution de $Sample'$, $\mathcal{D}'_2$ la distribution de $Sample2(h_1, Sample')$ et $\mathcal{D}'_3$ la distribution de $Sample3(h_1, h_2, Sample')$.

- Si $Sample2$ est appelée alors $\frac{\varepsilon'}{3} \leq L_{\mathcal{D}',f}(h_1) \leq \varepsilon''$.

Par conséquent le nombre moyen d'appels à $Sample'$ par $Sample2$ est borné par $\max(\frac{3}{\varepsilon'}, \frac{1}{1-\varepsilon''})$.

- Si $Sample3$ est appelée alors $\varepsilon' - \tau \leq L_{\mathcal{D}',f}(h_2) \leq \varepsilon''$.

$$\boxed{\mathbf{Pr}_{\mathcal{D}'}(h_1(X) \neq h_2(X)) \geq \frac{\varepsilon'}{4}}$$

D'où le nombre moyen d'appels à $Sample'$ par $Sample3$ est borné par $\frac{4}{\varepsilon'}$.

Preuve.

Introduisons $\beta_1 = L_{\mathcal{D}',f}(h_1)$, $\beta_2 = L_{\mathcal{D}'_2,f}(h_2)$ et $e = L_{\mathcal{D}',f}(h_2)$:

$$w_{00} = \mathbf{Pr}_{\mathcal{D}'}(h_1(X) = h_2(X) \neq f(X))$$
$$w_{01} = \mathbf{Pr}_{\mathcal{D}'}(h_1(X) \neq h_2(X) = f(X))$$
$$w_{10} = \mathbf{Pr}_{\mathcal{D}'}(h_2(X) \neq h_1(X) = f(X))$$
$$w_{11} = \mathbf{Pr}_{\mathcal{D}'}(h_1(X) = h_2(X) = f(X))$$

$$e = w_{00} + w_{10},\ \beta_1 = w_{00} + w_{01} \text{ d'où } w_{10} - w_{01} = e - \beta_1$$
$$w_{00} + w_{01} + w_{10} + w_{11} = 1$$

Analyse de *Sample2* et *Sample3* (2)

Preuve (suite).

$$
\begin{aligned}
1-\beta_2 &= \mathbf{Pr}_{\mathcal{D}'_2}(h_1(X)\neq h_2(X)=f(X)) + \mathbf{Pr}_{\mathcal{D}'_2}(h_2(X)=h_1(X)=f(X)) \\
&= \frac{1}{2\beta_1}\mathbf{Pr}_{\mathcal{D}'}(h_1(X)\neq h_2(X)=f(X)) + \frac{1}{2(1-\beta_1)}\mathbf{Pr}_{\mathcal{D}'}(h_2(X)=h_1(X)=f(X)) \\
&= \frac{w_{01}}{2\beta_1} + \frac{w_{11}}{2(1-\beta_1)} \\
&= \frac{w_{01}}{2\beta_1} + \frac{1-\beta_1-w_{10}}{2(1-\beta_1)}
\end{aligned}
$$

D'où : $4\beta_1(1-\beta_1)(1-\beta_2) = 2(1-\beta_1)w_{01} + 2\beta_1(1-\beta_1-w_{10})$

Après simplification : $2(1-\beta_1)w_{01} - 2\beta_1 w_{10} = 2\beta_1(1-\beta_1)(1-2\beta_2)$

Puis à : $(1-2\beta_1)(w_{01}+w_{10})2\beta_1 w_{10} + w_{01} - w_{10} = 2\beta_1(1-\beta_1)(1-2\beta_2)$

D'où : $(1-2\beta_1)(w_{01}+w_{10})2\beta_1 w_{10} = e - \beta_1 + 2\beta_1(1-\beta_1)(1-2\beta_2)$

$$
\begin{aligned}
w_{01}+w_{10} &= \frac{e-\beta_1+2\beta_1(1-\beta_1)(1-2\beta_2)}{1-2\beta_1} \\
&= \beta_1 + \frac{e-4\beta_1\beta_2(1-\beta_1)}{1-2\beta_1} \\
&\geq \beta_1 + \frac{e-4\beta_1\varepsilon''(1-\beta_1)}{1-2\beta_1} \stackrel{\text{def}}{=} \varphi(\beta_1)
\end{aligned}
$$

Analyse de $Sample2$ et $Sample3$ (3)

Preuve (fin).

Nous allons minorer $\varphi(\beta_1)$ pour tout $\beta_1 \in [0, \varepsilon'']$. $\lim_{\beta_1 \to -\infty} \varphi(\beta_1) = -\infty$ et :

$$\varphi'(\beta_1) = \frac{(4 - 8\varepsilon'')\beta_1^2 - (4 - 8\varepsilon'')\beta_1 + (1 - 4\varepsilon'' + 2e)}{(1 - 2\beta_1)^2}$$

Le numérateur décroit sur $]-\infty, \frac{1}{2}]$ et sa limite en $-\infty$ est ∞.

Il a donc au plus un zéro sur $]-\infty, \frac{1}{2}]$

et cet éventuel zéro est un maximum de φ sur cet intervalle.

On conclut que le minimum de $\varphi(\beta_1)$ sur $[0, \varepsilon'']$ est atteint soit en 0 soit en ε''.

$\varphi(0) = e \geq \varepsilon' - 2\tau = (\frac{3}{4} + \frac{\varepsilon''}{2})\varepsilon' \geq \frac{3}{4}\varepsilon'$ et :

$$\begin{aligned}
\varphi(\varepsilon'') &= \varepsilon'' + \frac{e - 4\varepsilon''^2(1 - \varepsilon'')}{1 - 2\varepsilon''} \\
&= \frac{\varepsilon'' - 6\varepsilon''^2 + 4\varepsilon''^3 + e}{1 - 2\varepsilon''} \\
&\geq \frac{\varepsilon'' - 6\varepsilon''^2 + 4\varepsilon''^3 + (\frac{3}{4} + \frac{\varepsilon''}{2})(3\varepsilon''^2 - 2\varepsilon''^3)}{1 - 2\varepsilon''} \\
&= \frac{\varepsilon''}{4}(4 - 7\varepsilon'' + 2\varepsilon''^2)
\end{aligned}$$

Puisque $4 - 7\varepsilon'' + 2\varepsilon''^2 \geq 1$ lorsque $\varepsilon'' \leq \frac{1}{2}$, $\varphi(\varepsilon'') \geq \frac{\varepsilon''}{4} \geq \frac{\varepsilon'}{4}$

Analyse de $Sample'$

Considérons la distribution $Sample'$ d'un appel **RecLearn** de profondeur i.

Par induction, le nombre moyen d'échantillonnages de $Sample$

pour effectuer un échantillonnage de $Sample'$ est majoré par $\prod_{0\leq j<i} \frac{4}{g^{-j}(\varepsilon)}$.

Puisque $g(x) \leq 3x^2$, $g^{-1}(y) \geq \sqrt{y/3} \geq \frac{\sqrt{y}}{3}$.

Par induction, $g^{-i}(y) = g^{-1}(g^{-(i-1)}(y)) \geq \left(\frac{y^{2^{-(i-1)}}/3}{3}\right)^{1/2} = y^{2^{-i}}/3$.

Par conséquent,

$$\prod_{0\leq j<i} \frac{4}{g^{-j}(\varepsilon)} \leq \prod_{0\leq j<i} \frac{12}{\varepsilon^{2^{-j}}} \leq \frac{12^i}{\varepsilon^2} \leq \frac{12^{B(n)}}{\varepsilon^2}$$

avec $B(n) = \log_{\frac{11}{8}}(\frac{p(n,s_n)}{4})\rceil + \lceil \log_2(\log_{\frac{4}{3}}(\frac{1}{\varepsilon}))\rceil$.