Apprentissage: cours 6
Théorie, concentration, borne PAC

Simon Lacoste-Julien

23 octobre 2015

Résumé

Aujourd'hui on couvre les bases théoriques pour l'analyse des algorithmes d'apprentissage de classification. On distingue l'analyse statistique traditionnelle (risque fréquentiste), de celle utilisée en informatique inspirée de l'approche PAC (Probably Approximately Correct). L'outil mathématique central qui explique l'apprentissage sont les bornes de concentrations (par exemple, borne de Chernoff). Finalement, nous allons prouver une des bornes les plus simples d'erreur de généralisation, la borne "d'Occam", qui justifie le principe du rasoir d'Occam et donne une interprétation de la régularisation en terme d'à-priori sur les hypothèses.

1 Risque fréquentiste vs. borne PAC

1.1 Rappel de définitions

Données d'entrainement : $D_n = (X_i, Y_i)_{1 \leqslant i \leqslant n}$ i.i.d. de loi P.

Algorithme d'apprentissage : $\mathscr{A} : \bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n \mapsto \mathcal{F}$

Famille d'algorithmes d'apprentissage : $(\mathscr{A}_m)_{m \in \mathcal{M}}$

Prédicteur $\widehat{f}$

Exemples : Dans ce cours par abus de notation on écrira souvent $\widehat{f}$ pour $\mathscr{A}$ et $\widehat{f}(D_n)$ pour $\mathscr{A}(D_n)$. Pour être rigoureux, il faudrait toujours utiliser $\widehat{f}_{D_n} := \mathscr{A}(D_n)$. $\widehat{f}(x; D_n)$ dénote $\widehat{f}_{D_n}$ évalué à x.

Excès de risque (Vapnik) : $\mathcal{R}_P\left(\widehat{f}(D_n)\right) - \mathcal{R}_P(f^\star)$ $\quad - -$ $\quad \mathcal{R}_P(\cdot)$ rend la dépendance sur P explicite.

Risque (Vapnik) : Le risque (au sens de Vapnik) donne l'*erreur de généralisation* de notre prédicteur – on veut le minimiser.

Consistance : Un algorithme d'apprentissage est *consistant* pour la loi P si

$$\lim_{n \to \infty} \mathbb{E}_{D_n \sim P^{\otimes n}} \left[\mathcal{R}_P(\widehat{f}) - \mathcal{R}_P(f_P^\star) \right] = 0.$$

1.2 Risque fréquentiste $\mathcal{R}_P^F(\cdot)$

En théorie de la décision statistique (traditionnelle) [à différencier de la théorie de la décision 'version apprentissage' présentée au premier cours], on veut analyser la performance d'un estimateur en moyenne sur les observations possibles. Dans le cadre de l'apprentissage automatique, les observations possibles sont les données d'entraînement D_n ; l'estimateur est l'algorithme d'apprentissage $\mathscr{A}$; et la performance est l'erreur de généralisation, i.e. le risque au sens de Vapnik : $\mathcal{R}_P(\mathscr{A}(D_n))$. La performance moyenne est l'espérance du risque de Vapnik par rapport aux données d'entraînement et s'appelle le *risque fréquentiste*[1] $\mathcal{R}_P^F(\mathscr{A}; n)$:

$$\mathcal{R}_P^F(\mathscr{A}; n) := \mathbb{E}_{D_n \sim P^{\otimes n}}[\mathcal{R}_P(\widehat{f}_n)] \quad (\widehat{f}_n = \mathscr{A}(D_n)).$$

1. Vous voyez donc le dédoublement de sens du mot risque et pourquoi nous précisons risque au sens de Vapnik vs. risque fréquentiste.

On voit donc que notre notion de consistance étudiait le comportement de $\lim_{n\to\infty} \mathcal{R}_P^F(\mathscr{A}; n)$.

Pour deux algorithmes d'apprentissage donnés $\mathscr{A}_1$ et $\mathscr{A}_2$, si $\mathcal{R}_P^F(\mathscr{A}_1; n) \leqslant \mathcal{R}_P^F(\mathscr{A}_2; n)$ pour tout n, l'analyse traditionnelle fréquentiste dira que $\mathscr{A}_1$ est meilleur que $\mathscr{A}_2$. Par contre, on considère plutôt $\mathcal{R}_P^F(\mathscr{A}; n)$ comme une fonction de P pour $P \in \mathcal{P}$, au quel cas il devient difficile de comparer les algorithmes (voir dessin au tableau). D'ailleurs, le théorème de no free lunch pour la classification que nous avons démontré dans le cours 3 implique en gros que si $\mathcal{X}$ est infini, alors pour toute paire d'algorithmes, il existe deux distributions P_1 et P_2 telle que $\mathscr{A}_1$ est meilleur que $\mathscr{A}_2$ sur P_1, mais $\mathscr{A}_2$ est meilleur que $\mathscr{A}_1$ sur P_2. Pour avoir des comparaisons non-triviales, il faut donc faire des suppositions sur $\mathcal{P}$. En statistique, un exemple d'approche est d'analyser la performance d'un algorithme d'apprentissage en considérant le pire des cas sur $\mathcal{P}$, i.e. on veut trouver un algorithme qui minimise $\sup_{P\in\mathcal{P}} \mathcal{R}_P^F(\mathscr{A}; n)$ – c'est l'approche *minimax*. Des résultats existent pour certains $\mathcal{P}$ (et par exemple, nous rappelons la consistance universelle uniforme pour la règle de la classe la plus fréquente quand $\mathcal{X}$ est fini mentionnée dans le cours 3).

1.3 Approche PAC

Un désavantage du risque fréquentiste est qu'il ne donne aucune information sur la *variance* de la performance de l'algorithme d'apprentissage sur les données d'entraînement possibles (voir dessin au tableau). En informatique, l'approche d'analyse se concentre plutôt sur des *intervalles de confiance* pour la performance de l'algorithme d'apprentissage. Plutôt que de considérer l'espérance de l'erreur de généralisation, on quantifie une *borne de queue*, c'est-à-dire, on donne une borne supérieure sur l'erreur de généralisation qui est valide avec *probablité supérieure à* $1-\delta$ (pour un petit $\delta > 0$), où l'aléat vient des données D_n possibles :

$$\mathbb{P}\Big\{\mathcal{R}_P(\mathscr{A}(D_n)) \leqslant \text{borne}(\mathscr{A}, n, \delta, P)\Big\} \geqslant 1-\delta$$

Le PAC (Probably Approximately Correct) implique que la borne n'est valide qu'avec probabilité plus grande que $1-\delta$. À noter l'analyse PAC est plus fine que celle du risque fréquentiste : avec des arguments probabilistes standards, nous pouvons transformer une borne PAC en une borne sur l'espérance de l'erreur de généralisation.

1.4 Les outils – pourquoi l'apprentissage fonctionne

En gros, nous voulons trouver une fonction f avec une petite erreur de généralisation ; rappelons :

$$\mathcal{R}_P(f) := \mathbb{E}_{(X,Y)\sim P}[\ell(f(X), Y)].$$

Nous avons accès à l'erreur empirique sur les données d'entraînement :

$$\widehat{\mathcal{R}}_n(f) = \mathbb{E}_n[\ell(f(X), Y)] = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i).$$

Puisque les données sont i.i.d, nous avons par la *la loi forte des grands nombres* :

$$\widehat{\mathcal{R}}_n(f) \xrightarrow{\text{p.s.}} \mathcal{R}_P(f).$$

Si la variance de la perte existe, nous avons même par le *théorème de la limite centrale* :

$$\sqrt{n}\left(\widehat{\mathcal{R}}_n(f) - \mathcal{R}_P(f)\right) \xrightarrow{\text{d}} \mathcal{N}(0, \text{var}_P[\ell(f(X), Y)]).$$

Par contre, si nous voulons comparer les optima de $\widehat{\mathcal{R}}_n(f)$ vs. ceux de $\mathcal{R}_P(f)$, pour être rigoureux, nous aurions besoin par exemple (condition suffisante mais non nécessaire) de la convergence *uniforme* sur une la classe de fonctions $\mathcal{F}$:

$$\sup_{f\in\mathcal{F}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}_P(f)| \xrightarrow{\text{P}} 0.$$

Ce genre de résultat est étudié par la *théorie des processus empiriques* (par exemple, *classe de Glivenko-Cantelli* et *classe de Donsker* qui généralise la loi des grands nombres et la loi du théorème central limite, respectivement, pour les processus empiriques).

Tous ces résultats sont asymptotiques par contre. En apprentissage automatique, nous voulons des résultats pour un n fini. Pour ceci, nous regardons des analogues non-asymptotiques du théorème de la limite centrale : les *bornes de concentration*. Par exemple, la borne de Chernoff que vous allez prouver dans le TD à partir de l'inégalité de Markov (et qui donne l'inégalité de Hoeffding qui est plus générale).

Borne de Chernoff : Soit $\{Z_i\}_{i=1}^n$ des variables de Bernouilli i.i.d de probabilité p (i.e. $Z_i = 1$ avec probabilité p et 0 autrement). Alors [2] :

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} Z_i \geqslant p + \varepsilon\right\} \leqslant e^{-2n\varepsilon^2} \quad \forall \varepsilon \geqslant 0.$$

En appliquant cette borne pour le risque en classification binaire, on obtient :

$$\mathbb{P}\left\{\widehat{\mathcal{R}}_n(f) \geqslant \mathcal{R}_P(f) + \varepsilon\right\} \leqslant e^{-2n\varepsilon^2} \quad \forall \varepsilon \geqslant 0$$

Nous allons utiliser cette borne pour montrer la Borne d'Occam pour la classification binaire.

2 Borne d'Occam

Pour le reste du cours [3], nous considérons la classification binaire. Pour avoir un résultat simple, nous allons considérer un ensemble de fonctions $\mathcal{F}$ *dénombrable* (aussi appelé ensemble d'*hypothèses* traditionnellement en informatique). La borne d'Occam utilise un *à-priori* π sur les fonctions possibles, c'est à dire une 'distribution' (au sens Bayésien) sur les fonctions dans $\mathcal{F}$. Cette distribution π sert à définir une mesure de complexité pour les fonctions (au sens du rasoir d'Occam) via la relation :

$$|f|_\pi := \log_2 \frac{1}{\pi}. \tag{1}$$

En théorie de l'information, on peut faire correspondre un langage fixe (un code) à une distribution de probabilité sur un nombre dénombrable d'objets via la relation (1) (modulo des fonctions partie entière que l'on ignore ici pour simplifier la présentation). Sous cette perspective, $|f|_\pi$ pourrait représenter le nombre de bits nécessaires pour décrire la fonction f (en utilisant un code préfixe, comme un code de Huffman, par exemple).

Borne d'Occam. La borne d'Occam est une borne PAC uniforme pour toutes les fonctions $f \in \mathcal{F}$, pour un $\mathcal{F}$ dénombrable. Elle dit que, pour tout $\delta > 0$, avec probabilité d'au moins $1 - \delta$ sur les jeux de données D_n possibles, nous avons la borne uniforme suivante qui est valide (pour le risque de la classification binaire) :

$$\forall f \in \mathcal{F}, \quad \mathcal{R}_P(f) \leqslant \widehat{\mathcal{R}}_n(f) + \frac{1}{\sqrt{2n}}\sqrt{\ln(2) \underbrace{|f|_\pi}_{\text{complexite}} + \ln\frac{1}{\delta}}. \tag{2}$$

Commençons par prouver la borne, pour voir d'où vient l'utilité de π.

2. La borne est aussi valide dans l'autre direction (i.e. pour $\frac{1}{n}\sum_i Z_i \leqslant p - \varepsilon$). Comparez aussi avec la borne d'espérance que nous avions utilisée dans le cours 2 : $\mathbb{E}[|\frac{1}{n}\sum_i Z_i - p|] \leqslant \frac{1}{2\sqrt{n}}$.

3. Ces notes sont inspirées du cours de David McAllester disponibles à http://nagoya.uchicago.edu/~dmcallester/ttic101-07/lectures/generalization/generalization.pdf.

2.1 Preuve de la borne d'Occam

La preuve de la borne d'Occam utilise trois inégalités très simples :
— **Concentration : borne de Chernoff** :

$$\mathbb{P}\left\{\widehat{\mathcal{R}}_n(f) \geqslant \mathcal{R}_P(f) + \varepsilon\right\} \leqslant e^{-2n\varepsilon^2} \quad \forall \varepsilon \geqslant 0$$

— **Borne d'union** : $\mathbb{P}\{\exists x \text{Prop}(x)\} \leqslant \sum_x \mathbb{P}\{\text{Prop}(x)\}$
— **Inégalité de Kraft** : $\sum_{f\in\mathcal{F}} 2^{-|f|_\pi} \leqslant 1$ (valide pour les codes préfixes)
L'inégalité de Kraft devient évidente avec la définition de $|f|_\pi$ à partir d'une distribution π.

Démonstration. Pour prouver la borne d'Occam, nous allons définir qu'une fonction f est 'mauvaise' (par rapport à un jeu de données D_n) quand elle viole l'inégalité du théorème. Plus précisément, nous définissons la proposition (qui peut être vraie au fausse) :

$$\text{mauvaise}(f) := \left[\mathcal{R}_P(f) > \widehat{\mathcal{R}}_n(f) + \frac{1}{\sqrt{2n}}\sqrt{\ln(2)|f|_\pi + \ln\frac{1}{\delta}}\right]$$

Par Chernoff sur les jeux de données, nous avons :

$$\mathbb{P}\left\{\text{mauvaise}(f)\right\} \leqslant e^{-2n\left(\frac{1}{\sqrt{2n}}\sqrt{\ln(2)|f|_\pi + \ln\frac{1}{\delta}}\right)^2} = \delta 2^{-|f|_\pi}$$

En utilisant une borne d'union sur toutes les fonctions $f \in \mathcal{F}$, suivie de l'inégalité de Kraft, nous obtenons le résultat :

$$\mathbb{P}\left\{\exists f \in \mathcal{F},\ \text{mauvaise}(f)\right\} \leqslant \sum_{f\in\mathcal{F}} \delta 2^{-|f|_\pi} \leqslant \delta$$

□

On voit donc que l'utilité de π était de pouvoir sommer la borne de Chernoff sur toutes les fonctions.

2.2 Borne comme un algorithme

Après avoir dérivé une borne de généralisation PAC uniforme, cela donne naturellement un algorithme d'apprentissage (du style du rasoir d'Occam) qui minimise la borne :

$$\widehat{f}_n := \text{argmin}_{f\in\mathcal{F}} \left\{\widehat{\mathcal{R}}_n(f) + \frac{1}{\sqrt{2n}}\sqrt{\ln(2)|f|_\pi + \ln\frac{1}{\delta}}\right\}. \tag{3}$$

Définissons $\Omega(f;\delta) := \sqrt{\ln(2)|f|_\pi + \ln\frac{1}{\delta}}$, qui représente en quelque sorte une mesure de complexité pour f. Puisque la borne d'Occam est valide pour tout $f \in \mathcal{F}$, elle est aussi valide pour $\widehat{f}_n$; donc avec probabilité supérieure à $1-\delta$ sur D_n, nous avons :

$$\mathcal{R}_P\left(\widehat{f}_n\right) \leqslant \widehat{\mathcal{R}}_n\left(\widehat{f}_n\right) + \frac{1}{\sqrt{2n}}\Omega(\widehat{f}_n;\delta).$$

Cette borne est utile seulement si le côté droit est plus petit que 1 (car l'erreur de généralisation est toujours plus petite que 1). Ce sera le cas seulement si il existe dans $\mathcal{F}$ une fonction avec, à la fois, une petite erreur empirique *et aussi* une petite complexité $\Omega(\widehat{f}_n;\delta)$ par rapport à $\sqrt{n}$. Si notre à-priori π est mauvais, nous allons assigner une grande complexité $|f|_\pi$ à des fonctions qui auraient pu bien expliquer les données, ce qui fera un mauvais algorithme. Pour une distribution P donnée, nous pourrions définir π de telle sorte que $|f_P^*|_\pi$ soit minime ; l'algorithme d'apprentissage (3) serait alors très bon pour cette distribution, car bien calibré pour celle-ci. Par contre, nous voulons considérer un ensemble de P possible, et donc, il y a un analogue de no free lunch ici que nous ne pouvons pas mettre une grande probabilité à toutes les fonctions f_P^* possibles pour un large éventail de P.

2.3 Consistence forte universelle

L'algorithme (3) est consistent (au sens de faire le mieux dans la classe $\mathcal{F}$ possible pour toute distribution P, et donc elle est 'universelle'[4]). La consistence est même forte (la convergence est presque sûre) :

$$\forall P, \quad \mathcal{R}_P\left(\widehat{f}_n\right) \xrightarrow{\text{p.s.}} \min_{f\in\mathcal{F}} \mathcal{R}_P\left(f\right)$$

La consistence forte (convergence presque sûre) implique aussi la consistence au sense du risque fréquentiste car la perte est bornée :

$$\lim_{n\to\infty} \mathbb{E}_{D_n\sim P^{\otimes n}} \mathcal{R}_P(\widehat{f}_n) = \min_{f\in\mathcal{F}} \mathcal{R}_P\left(f\right)$$

Démonstration. La technique de preuve pour la borne d'Occam (2) peut être modifié pour borner les deux directions (avec un coût de $\ln\frac{2}{\delta}$ dans la borne plutôt que $\ln\frac{1}{\delta}$) : avec probabilité d'au moins $1-\delta$ sur les jeux de données D_n possibles, nous avons :

$$\forall f\in\mathcal{F}, \quad |\mathcal{R}_P\left(f\right) - \widehat{\mathcal{R}}_n\left(f\right)| \leqslant \frac{1}{\sqrt{2n}}\Omega(f;\frac{\delta}{2}) \tag{4}$$

Nous supposons que à la fois (2) et (4) sont valides en même temps (donc avec probabilité $1-2\delta$ sur D_n, car chaque borne a une probabilité δ d'échouer). Nous avons :

$$\begin{aligned}
\mathcal{R}_P\left(\widehat{f}_n\right) &\leqslant \widehat{\mathcal{R}}_n\left(\widehat{f}_n\right) + \frac{1}{\sqrt{2n}}\Omega(\widehat{f}_n;\delta) \qquad \text{(en utilisant (2))} \\
&\leqslant \widehat{\mathcal{R}}_n\left(f\right) + \frac{1}{\sqrt{2n}}\Omega(f;\delta) \quad \forall f\in\mathcal{F} \qquad \text{(par la définition (3))} \\
&\leqslant \mathcal{R}_P\left(f\right) + 2\frac{1}{\sqrt{2n}}\Omega(f;\frac{\delta}{2}) \quad \forall f\in\mathcal{F} \qquad \text{(en utilisant (4))} \\
\mathcal{R}_P\left(\widehat{f}_n\right) &\leqslant \min_{f\in\mathcal{F}}\left\{\mathcal{R}_P\left(f\right) + 2\frac{1}{\sqrt{2n}}\Omega(f;\frac{\delta}{2})\right\} \\
&\leqslant \mathcal{R}_P\left(f_P^*\right) + 2\frac{1}{\sqrt{2n}}\Omega(f_P^*;\frac{\delta}{2})
\end{aligned}$$

où nous avons défini $f_P^* \in \operatorname{argmin}_{f\in\mathcal{F}} \mathcal{R}_P\left(f\right)$.

Maintenant, soit un $\varepsilon>0$ donné. Pour chaque n, définissons $\delta_n := \frac{1}{n^2}$. Nous avons $2\frac{1}{\sqrt{2n}}\Omega(f_P^*;\frac{\delta_n}{2}) \xrightarrow{n\to\infty} 0$. Donc pour n plus grand qu'un n_ε, nous avons $2\frac{1}{\sqrt{2n}}\Omega(f;\frac{\delta_n}{2}) \leqslant \varepsilon$. Selon nos inégalités PAC, nous avons donc que

$$\mathbb{P}\left\{\mathcal{R}_P\left(\widehat{f}_n\right) > \mathcal{R}_P\left(f_P^*\right) + \varepsilon\right\} \leqslant 2\delta_n = \frac{2}{n^2} \qquad \forall n \geqslant n_\varepsilon$$

Par la définition de f_P^* nous avons évidemment que $\mathcal{R}_P\left(\widehat{f}_n\right) \geqslant \mathcal{R}_P\left(f_P^*\right)$, et donc $\mathcal{R}_P\left(\widehat{f}_n\right) - \mathcal{R}_P\left(f_P^*\right) = |\mathcal{R}_P\left(\widehat{f}_n\right) - \mathcal{R}_P\left(f_P^*\right)|$. Ainsi, nous avons :

$$\sum_{n\geqslant n_\varepsilon} \mathbb{P}\left\{|\mathcal{R}_P\left(\widehat{f}_n\right) - \mathcal{R}_P\left(f_P^*\right)| > \varepsilon\right\} \leqslant \sum_{n\geqslant n_\varepsilon} \frac{2}{n^2} < \infty$$

et ce, pour tout $\varepsilon>0$, et donc par le lemme de Borel-Cantelli sur la convergence presque sûre, nous concluons que :

$$\mathcal{R}_P\left(\widehat{f}_n\right) \xrightarrow{\text{p.s.}} \mathcal{R}_P\left(f_P^*\right)$$

□

4. Attention ! À noter que la consistence n'est pas *uniforme* (sinon, cela contredirait le 'no free lunch' !).

2.4 Exemple avec régularisation

Pour donner un exemple concret et faire le lien avec la régularisation standard, considérons $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ et $\mathcal{F}$ l'espace des frontières de décision linéaires passant par 0 : $\mathcal{F} := \{f_w \mid w \in \mathcal{W}\}$ où $f_w : x \mapsto \text{sign}(\langle w, x\rangle)$ et $\mathcal{W}$ est un sous-ensemble dénombrable de $\mathbb{R}^d$. Par exemple, on pourrait décider d'encoder chaque coordonnée de w sous la forme de $q.\underbrace{d_1 \cdots d_k}_{k \text{ décimales}}$ où q est un nombre entier ; pour un k fixe donné, le nombre de possibilités est dénombrables. On considère maintenant l'à-priori $\pi(w) := \frac{1}{Z_k} 2^{-\|w\|^2}$, où Z est une constante à déterminer en sommant sur le nombre dénombrable de possibilités dans $\mathcal{W}$. À noter que souvent ces constantes ne sont pas très importantes, car elles apparaissent dans un facteur log. Par exemple, avec cet à-priori, nous avons :

$$|f_w|_\pi = -\log_2 \pi(w) = \|w\|^2 + \log_2 Z_k$$

Pour avoir l'ordre de grandeur de Z_k :

$$\begin{aligned} Z_k = \sum_{w \in \mathcal{W}} 2^{-\|w\|^2} = \prod_{i=1}^{d} \left(\sum_{w_i \in \mathcal{W}_i} 2^{-w_i^2} \right) &= (\sum_{u \in \mathbb{Z}} 2^{-u^2/10^{2k}})^d \\ &\leqslant \left(2 \sum_{u \in \mathbb{N}} \left(2^{-\frac{1}{10^{2k}}} \right)^{u^2} \right)^d \\ &\leqslant \left(2 \sum_{u \in \mathbb{N}} \left(2^{-\frac{1}{10^{2k}}} \right)^{u} \right)^d = \left(\frac{2}{1 - 2^{-\frac{1}{10^{2k}}}} \right)^d \end{aligned}$$

Avec un peu de calcul, on peut montrer que :

$$\log_2 Z_k \approx d(1 + 2k \log_2(10) + \log_2 \frac{1}{\ln(2)}) = O(d \cdot k).$$

Nous avons donc que la pénalité de complexité ressemble à :

$$|f_w|_\pi = \|w\|^2 + C\, d\, k$$

Et la minimisation de la borne d'Occam donne une minimisation de risque empirique régularisé :

$$\widehat{f}_n := \text{argmin}_{w \in \mathcal{W}} \left\{ \widehat{\mathcal{R}}_n\,(\,f_w\,) + \frac{1}{\sqrt{2n}} \sqrt{\ln(2)(\|w\|^2 + Cdk) + \ln \frac{1}{\delta}} \right\}.$$

On voit déjà l'effet de la dimension d dans la borne.

À noter que en général, la minimisation du risque empirique est NP-difficile, alors cet algorithme n'est pas réaliste d'un point de vue computationnel. Des techniques analogues à la borne d'Occam peuvent être développées avec une version convexe (en w) du risque empirique à droite (avec des 'pertes convexes de substitut'). Vous allez voir des algorithmes inspirés de cette approche dans le cours sur les pertes convexes.

Finalement, à noter que pour un ensemble $\mathcal{F}$ non-dénombrable de fonctions, la borne d'Occam se généralise en définissant une *mesure* à-priori sur les fonctions, remplaçant la somme dans l'inégalité de Kraft par une intégrale. Cela donne une borne appelée *PAC-Bayes* (voir les notes de McAllester).