Devoir maison 1 : PAC Learning

L3 Apprentissage

À rendre avant le 04/04/2012

Dans tout le problème, les algorithmes d'apprentissage prennent en entrée un ensemble S de données étiquetées (ou exemples étiquetés) : $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, avec $x_i \in X$ les valeurs des attributs (*features*) de l'exemple i, et y_i l'étiquette qui lui est associée. On supposera que l'étiquette y_i est binaire, c'est-à-dire à valeur dans $\{0, 1\}$. L'espace X est appelé espace des instances : c'est un produit cartésien d'ensembles, chacun représentant la valeur d'un attribut. Dans la suite, on notera toujours n le nombre d'attributs des instances.

Un concept est une fonction booléenne sur l'espace des instances. Une classe C de concept est un ensemble de concepts, généralement associé à une représentation particulière : la taille d'un concept c, notée $|c|$, est définie comme le nombre de bits nécessaires pour spécifier le concept dans sa représentation.

Par exemple, on peut être intéressé par classer des courriels selon qu'ils soient des *spams* ou pas. On peut se donner un ensemble d'attributs tels que la présence ou non dans le corps du courriel du mot « argent », du mot « viagra », la présence ou non de fautes d'orthographes, le fait que l'expéditeur est dans son carnet d'adresses ou pas, et la fréquence des lettres majuscules dans le corps du courriel. Ainsi, l'espace des instances serait ici $X = \{0, 1\}^4 \times [0, 1]$ (donc $n = 5$). Un concept pourrait alors être la fonction qui associe à 1 l'ensemble des exemples tels que le deuxième ou le troisième attributs sont à 1 et le cinquième attribut est inférieur à $0,5\%$.

Un algorithme $\mathcal{A}$ d'apprentissage recevant en entrée un ensemble S de données étiquetées et un ensemble C de concepts possibles doit alors produire un concept $c \in C$, *le plus susceptible d'avoir engendré l'ensemble S de données.*

Partie I : Le modèle cohérent

Un concept c est dit cohérent avec une donnée étiquetée $(x, y) \in X \times \{0, 1\}$ si $c(x) = y$. On dit qu'un algorithme $\mathcal{A}$ apprend la classe de concept C dans le modèle cohérent si, étant donné un ensemble S de données étiquetées, l'algorithme produit un concept $c \in C$ cohérent avec toutes les données de S, s'il en existe un, et répond « il n'y a pas de concept cohérent » dans le cas contraire. On dit qu'on peut apprendre la classe C dans le modèle cohérent s'il existe un tel algorithme. Si de plus, il existe un algorithme apprenant C qui s'exécute en temps polynomial en le nombre d'exemples dans S et en la taille des exemples (leur nombre d'attributs), on dit qu'on peut efficacement apprendre C dans le modèle cohérent.

Dans les questions 1 à 4, on suppose que l'ensemble des instances est $X = \{0, 1\}^n$, avec $n \geq 1$. On note $x = (x_1, \ldots, x_n) \in X$.

1. On considère dans cette question, l'ensemble C^+_{conj} de concepts représentables comme une conjonction positive (c'est-à-dire sans négation) d'attributs : par exemple, la fonction qui à $x \in X$ associe $x_1 \wedge x_3$, c'est-à-dire qui associe 1 si et seulement les attributs x_1 et x_3 sont tous les deux égaux à 1. Montrer qu'on peut efficacement apprendre la classe C^+_{conj} dans le modèle cohérent. Qu'en est-il de la classe C_{conj} de concepts représentables comme une conjonction de littéraux (un attribut ou sa négation).

2. Même question avec la classe C_{disj} de concepts représentables comme une disjonction de littéraux.

3. Pour $k \geq 2$, on appelle k-CNF la classe de concepts représentables par des formules en forme normale conjonctive telles que chaque clause admet au plus k littéraux : par exemple $x_4 \wedge (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3)$ est un concept de 2-CNF. Montrer qu'on peut efficacement apprendre la classe k-CNF pour tout $k \geq 2$ dans le modèle cohérent. Qu'en est-il des classes duales k-DNF ? Que peut-on dire pour l'union de toute ces classes, qu'on appellera CNF et DNF respectivement ?

4. Une liste de décision est une suite de règles si-alors : « si ℓ_1 alors b_1, sinon si ℓ_2 alors b_2, sinon si ℓ_3 alors b_3, ..., sinon b_p », où chaque ℓ_i est un littéral et $b_i \in \{0,1\}$. On note C_{dlist} la classe de concepts représentables par une liste de décision. Par exemple, un possible concept de C_{dlist} est : « si $\neg x_1$ alors 1, sinon si x_5 alors 0, sinon 1 ». On peut voir une liste de décision comme un arbre de décision réduit à un chemin. Montrer qu'on peut efficacement apprendre C_{dlist} dans le modèle cohérent.

5. Dans cette question et dans la suivante, on suppose que l'ensemble des instances est $X = \mathbb{R}^n$. On considère l'ensemble des concepts C_{lin} représentables par une équation $w^{\mathrm{T}}x > 0$, avec $w \in \mathbb{R}^n$: ainsi, les exemples positifs et négatifs sont séparés par un hyperplan d'équation $w^{\mathrm{T}}x = 0$. En utilisant les résultats vus en cours, peut-on utiliser l'algorithme du Perceptron pour apprendre la classe C_{lin} dans le modèle cohérent ? À quelle condition sur l'ensemble d'exemples S, cet algorithme est-il efficace ? Alternativement, en utilisant le fait qu'on peut résoudre en temps polynomial un système de contraintes linéaires (programmation linéaire), montrer qu'on peut toujours efficacement apprendre la classe C_{lin} dans le modèle cohérent.

6. On considère l'ensemble des concepts C_{rect} représentables par un produit cartésien d'intervalles compacts de $\mathbb{R}$: des segments si $n = 1$, des rectangles si $n = 2$... Montrer qu'on peut apprendre efficacement cette classe dans le modèle cohérent.

Partie II : Le modèle PAC

Étant donné un concept c, on peut étiqueter toute donnée $x \in X$ à l'aide de c : $(x, c(x))$ est la donnée étiquetée associée. Étant donnée une distribution $\mathcal{D}$ sur l'espace X des instances, l'erreur $\mathsf{err}_{c^*}(c)$ d'un concept c par rapport à un concept $c^* \in C$ est la probabilité $\mathbb{P}_{x\sim\mathcal{D}}(c(x) \neq c^*(x))$ (ici, on note $\mathbb{P}_{x\sim\mathcal{D}}(A)$ la probabilité de l'évènement A étant donné que x est tiré aléatoirement selon la distribution $\mathcal{D}$).

7. On considère dans cette question la classe C^+_{conj} de concepts définie dans la question 1. Soit $\mathcal{D}$ une distribution de probabilité sur l'ensemble $X = \{0,1\}^n$, et soit c^* un concept de C^+_{conj}. Montrer que pour tout $\varepsilon, \delta \in]0,1[$, si on tire aléatoirement un ensemble S d'exemples à partir de $\mathcal{D}$, qu'on étiquette avec le concept c^*, de taille

$$m \geq \frac{1}{\varepsilon}\Big(n \ln 2 + \ln \frac{1}{\delta}\Big),$$

alors, avec probabilité au moins $1-\delta$, tout concept de C^+_{conj} cohérent avec S (c'est-à-dire cohérent avec tous les exemples étiquetés de S) a une erreur majorée par ε.

On dit qu'un algorithme $\mathcal{A}$ apprend la classe de concepts C dans le modèle PAC si pour tout $c^* \in C$, pour toute distribution $\mathcal{D}$ de probabilité sur l'espace X des instances, pour tout $\varepsilon, \delta \in]0,1[$ et pour tout polynôme p, les faits suivants sont vrais :

- l'algorithme $\mathcal{A}$ prend en entrée un ensemble S de données tirées aléatoirement en suivant la distribution $\mathcal{D}$, étiquetées à l'aide du concept c^*, de taille $p(1/\varepsilon, 1/\delta, n, |c^*|)$;
- sur cet ensemble S, $\mathcal{A}$ produit, avec probabilité au moins $1-\delta$, un concept $c \in C$ tel que $\mathsf{err}_{c^*}(c) \leq \varepsilon$;
- l'algorithme $\mathcal{A}$ s'exécute en temps polynomial en n et en la taille de S.

S'il existe un tel algorithme $\mathcal{A}$, on dit qu'on peut apprendre la classe C dans le modèle PAC.

8. Montrer qu'on peut apprendre la classe C^+_{conj} dans le modèle PAC.

9. On considère dans cette question une classe C finie, et on note $|C|$ son cardinal. Soit $\mathcal{A}$ un algorithme qui apprend la classe C dans le modèle cohérent. Montrer que, pour tout $c^* \in C$, $\mathcal{A}$ nécessite un ensemble S d'exemples étiquetés à l'aide de c^* de taille au plus

$$\frac{1}{\varepsilon}\Big(\ln |C| + \ln \frac{1}{\delta}\Big)$$

pour que le concept $c \in C$ qu'il renvoie ait une erreur $\mathsf{err}_{c^*}(c)$ inférieure ou égale à ε, avec probabilité au moins $1-\delta$. En déduire une condition suffisante sur C pour qu'on puisse l'apprendre dans le modèle PAC.

10. Donner toutes les classes de concepts de la partie I auxquelles on peut appliquer cette condition pour prouver qu'on peut les apprendre dans le modèle PAC.

Partie III : Cas non réalisable

Jusqu'à maintenant, on a considéré que l'ensemble d'exemples S que l'algorithme d'apprentissage recevait était étiqueté par un concept c^* appartenant à la classe de concept C dans lequel l'algorithme va chercher. Si on relâche cette hypothèse, on considére un ensemble de concepts C^* à apprendre, différente de la classe C dans laquelle l'algorithme va chercher. En particulier, il se peut qu'il n'y ait pas de concept dans la classe C d'erreur nulle par rapport à $c^* \in C^*$, c'est-à-dire $\inf_{c\in C} \mathsf{err}_{c^*}(c) > 0$. Pour autant, on aimerait pouvoir trouver un concept $c \in C$ *proche de l'optimal* : mais, ne connaissant pas c^*, il n'est évidemment pas possible de calculer l'erreur réelle $\mathsf{err}_{c^*}(c)$. Par contre, étant donné un ensemble d'exemples S étiquetés par c^*, on peut aisément calculer le taux d'erreur empirique $\frac{1}{m}|\{x \in S \mid c(x) \neq c^*(x)\}|$, notée $\mathsf{err}^S_{c^*}(c)$.

Il semble naturel d'espérer qu'un concept c avec un faible taux d'erreur empirique aura une erreur réelle par rapport à c^* faible. C'est ce qu'on étudie dans cette partie.

A - Inégalités de probabilité

Dans la suite, on va utiliser le théorème suivant, qu'on prouve dans la question 11.

Théorème 1 (Inégalité de Hoeffding)**.** *Soient $Y_1, \dots, Y_N$ des variables aléatoires réelles indépendantes vérifiant, pour deux suites $(a_i)_{1\leq i\leq N}$ et $(b_i)_{1\leq i\leq N}$ de nombres réels tels que $a_i < b_i$: $\forall i \quad \mathbb{P}(a_i < Y_i < b_i) = 1$. On pose $S_N = Y_1 + \cdots + Y_N$. Alors, pour tout $t > 0$,*

$$\mathbb{P}(S_N - \mathbb{E}(S_N) \geq t) \leq \exp\Big(\frac{-t^2}{2\sum_{i=1}^N (b_i - a_i)^2}\Big)\,.$$

11. Comme dans le théorème, soient $Y_1, \dots, Y_N$ des variables aléatoires réelles indépendantes vérifiant, pour deux suites $(a_i)_{1\leq i\leq N}$ et $(b_i)_{1\leq i\leq N}$ de nombres réels tels que $a_i < b_i$: $\forall i \quad \mathbb{P}(a_i < Y_i < b_i) = 1$. On pose $S_N = Y_1 + \cdots + Y_N$.

a) Pour une variable aléatoire réelle Y, d'espérance nulle et telle que $\mathbb{P}(c \leq Y \leq d) = 1$ pour deux réels $c < d$, montrer que pour tout $s > 0$,

$$\mathbb{E}\big(\mathrm{e}^{sY}\big) \leq \exp\Big(\frac{s^2(d-c)^2}{8}\Big)\,.$$

Indication : commencer par utiliser la convexité de la fonction $s \mapsto \mathrm{e}^{sy}$.

b) Prouver alors que pour tout $t > 0$

$$\mathbb{E}\big(S_N - \mathbb{E}(S_N) \geq t\big) \leq \exp\Big(-st + \frac{s^2 \sum_{i=1}^N (b_i - a_i)^2}{8}\Big)\,.$$

c) Conclure en utilisant l'inégalité de Markov.

12. Appliquer cette inégalité dans le cas d'une suite $X_1, \dots, X_N$ de variables aléatoires i.i.d. suivant une loi de Bernoulli de paramètre p. En notant $\overline{X}_N = \frac{X_1+\cdots+X_N}{N}$, en déduire que pour tout $\varepsilon > 0$

$$\mathbb{P}(|\overline{X}_N - p| \geq \varepsilon) \leq 2\mathrm{e}^{-2N\varepsilon^2}\,.$$

B - Application au modèle PAC dans le cas non réalisable

13. Soit C une classe de concepts finie sur un ensemble d'instances X. Soit $\mathcal{D}$ une distribution sur X et c^* un concept à apprendre. Pour tout $\varepsilon, \delta \in]0,1[$, montrer que si on tire un ensemble d'exemples S à partir de la distribution $\mathcal{D}$ de taille

$$m \geq \frac{1}{2\varepsilon^2}\Big(\ln|C| + \ln\frac{2}{\delta}\Big)\,,$$

alors, avec probabilité au moins $1-\delta$, tous les concepts c de C vérifient $|\mathsf{err}_{c^*}(c) - \mathsf{err}^S_{c^*}(c)| \leq \varepsilon$.

14. Pour $k \geq 2$, soit C_k la classe de fonctions booléennes sur $X = \{0,1\}^n$ représentées par des formules de la forme : $\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k$, avec φ_i une conjonction de littéraux. Il est possible de démontrer (mais ce n'est pas demandé ici) qu'à moins que NP = P, il n'y a pas d'algorithme polynomial apprenant la classe C_k dans le modèle consistent : c'est en effet un problème NP-difficile. Après avoir clairement expliqué la signification de la phrase, montrer qu'on peut cependant apprendre la classe C_k par la classe d'hypothèses k-CNF dans le modèle PAC. Plus généralement, donner une condition suffisante pour qu'on puisse apprendre une classe C^* par une classe d'hypothèses C dans le modèle PAC.

Partie IV : Ensemble infini de concepts

Dans cette partie, on cherche à généraliser les résultats précédents dans le cas où la classe C de concepts est infinie. Pour un ensemble fini d'exemples $S = \{x_1, \dots, x_m\} \subseteq X$, on note $C(S)$ l'ensemble des m-uplets d'étiquettes des exemples de S par des concepts de C : $C(S) = \{(c(x_1), \dots, c(x_m)) \mid c \in C\} \subseteq \{0,1\}^m$. De plus, pour un entier naturel m, on note $C[m]$ le nombre maximal de façons distinctes d'étiqueter m exemples par des concepts de C : $C[m] = \max\{|C(S)| \mid S \subseteq X, |S| = m\}$.

A - Cas réalisable

Si besoin, on pourra utiliser sans démonstration le théorème suivant :

Théorème 2 (Inégalité de Chernoff)**.** *Soient $Y_1, \dots, Y_N$ des variables aléatoires réelles indépendantes identiquement distribuées à valeurs dans $\{0,1\}$ avec $p = \mathbb{E}(Y_i)$ pour tout i. On pose $\overline{Y}_N = \frac{Y_1 + \cdots + Y_N}{N}$. Alors, pour tout $t > 0$,*

$$\mathbb{P}\big(\overline{Y}_N \leq p(1-t)\big) \leq \mathrm{e}^{-Npt^2/2}\,.$$

Dans cette section, on fixe un ensemble d'instances X quelconque muni d'une distribution de probabilités $\mathcal{D}$, une classe de concepts C et un concept à apprendre c^*. Soient $m \geq 0$ et $\varepsilon, \delta \in]0,1[$. Dans la suite, S et S' seront toujours des ensembles de taille m d'exemples tirés aléatoirement en suivant la distribution $\mathcal{D}$. On note $S = \{x_1, \dots, x_m\}$ et $S' = \{x'_1, \dots, x'_m\}$.

15. Notons B l'évènement : « $\exists c \in C$ avec $\mathsf{err}^S_{c^*}(c) = 0$ mais $\mathsf{err}_{c^*}(c) > \varepsilon$ ». On note également B' l'évènement : « $\exists c \in C$ avec $\mathsf{err}^S_{c^*}(c) = 0$ mais $\mathsf{err}^{S'}_{c^*}(c) > \varepsilon/2$ ». Montrer que si $m > \frac{8}{\varepsilon}$, alors $\mathbb{P}(B' \mid B) \geq \frac{1}{2}$.

16. On considère RandomSwap le processus suivant

> Pour i de 1 à m :
> Lancer une pièce équilibrée ;
> Si FACE, alors échanger x_i et x'_i.
> Appeler T et T' les nouveaux ensembles d'exemples obtenus.

Notons B'' l'évènement : « $\exists c \in C$ avec $\mathsf{err}^T_{c^*}(c) = 0$ mais $\mathsf{err}^{T'}_{c^*}(c) > \varepsilon/2$ ». Montrer que $\mathbb{P}(B'') = \mathbb{P}(B')$.

17. Fixons un concept $c \in C$. Notons F l'évènement « $\mathsf{err}^T_{c^*}(c) = 0 \wedge \mathsf{err}^{T'}_{c^*}(c) > \varepsilon/2$ » (qui dépend des choix de S, S' et des bits aléatoires de RandomSwap). Montrer que $\mathbb{P}(F \mid S, S') \leq 2^{-\varepsilon m/2}$.

18. En déduire que $\mathbb{P}(B'') \leq C[2m]2^{-\varepsilon m/2}$.

19. Trouver ainsi une borne inférieure sur le nombre m permettant d'assurer que si on tire aléatoirement un ensemble S de m exemples en suivant la distribution $\mathcal{D}$ et qu'on les étiquette avec c^*, avec probabilité au moins $1 - \delta$, toutes les hypothèses c de C cohérentes avec S ont une erreur $\mathsf{err}_{c^*}(c) \leq \varepsilon$.

B - Cas non réalisable

20. Généraliser le résultat de la question 19 au cas non réalisable.

C - Dimension de Vapnik-Chervonenkis

Un ensemble d'exemples S est dit pulvérisé par une classe de concepts C si $|C(S)| = 2^{|S|}$. La dimension de Vapnik-Chervonenkis de C, dénotée $\mathrm{VCdim}(C) \in \mathbb{N} \cup \{\infty\}$, est définie comme la taille du plus grand ensemble d'exemples S pulvérisé par C.

21. Quelle est la dimension de Vapnik-Chervonenkis de C_{lin} et C_{rect} ?

22. En utilisant le théorème suivant (sans démonstration)

Théorème 3 (Lemme de Sauer)**.** *Si $\mathrm{VCdim}(C) = d$ alors pour tout $m > d$, on a $C[m] \leq \left(\frac{em}{d}\right)^d$.*

montrer que si C est un ensemble de concepts de dimension de Vapnik-Chervonenkis d, $\mathcal{D}$ une distribution sur l'ensemble des instances X, c^* un concept à apprendre, $\varepsilon, \delta \in]0,1[$, alors si on tire un ensemble S d'exemples à partir de $\mathcal{D}$ de taille

$$m \geq \frac{8}{\varepsilon}\Big(d \ln \frac{16}{\varepsilon} + \ln \frac{2}{\delta}\Big)\,,$$

alors, avec probabilité au moins $1 - \delta$, toutes les hypothèses c de C cohérentes avec S ont une erreur $\mathsf{err}_{c^*}(c) \leq \varepsilon$.

23. En déduire qu'on peut apprendre les classes C_{lin} et C_{rect} dans le modèle PAC.