Université de Montpellier Année 2019-2020
HMMA303

Examen final
Durée 2h - Sans document, sans matériel - 6 novembre 2019

Exercice 1 (2 pts) Expliquer clairement comment l'on peut évaluer le taux d'erreur associé à un classifieur résultant d'une forêt aléatoire.

Exercice 2 (8 pts) Soit Y un facteur binaire à expliquer, $Y \in \{0, 1\}$ et $X > 0$ une variable explicative réelle positive. Nous supposons que $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 1/2$ et

$$X|Y = 0 \sim f_0(x) = \frac{1}{4} \exp\left(-\frac{x}{4}\right) \mathbb{I}_{x>0}$$

$$X|Y = 1 \sim f_1(x) = \exp(-x) \mathbb{I}_{x>0} .$$

1) (1 pt) Donner la loi de $Y|X = x$.

2) (2 pts) Expliciter le classifieur optimal g^1 associé à la fonction de coût élémentaire $h_1(y, d) = \mathbb{I}_{y \neq d}$.

3) (1 pt) Un classifieur alternatif g^0 consiste à affecter à la classe 0 si $x > 2.5$ et à la classe 1 sinon. Expliquer sa logique de construction.

4) (2 pts) Comparer g^1 et g^0 pour un critère bien choisi.

5) (2 pts) On considère maintenant la fonction de coût élémentaire h_2 telle que

$$h_2(y, d) = \begin{cases} 0 & \text{si} \quad y = d \\ 2 & \text{si} \quad y = 1 \quad \text{et} \quad d = 0 \\ 1 & \text{si} \quad y = 0 \quad \text{et} \quad d = 1 \end{cases} .$$

Expliciter le classifeur optimal g^2 associé à h_2, discuter de la modification induite par un changement de fonction de coût.

Exercice 3 (4 pts) Le principe général du boosting consiste à construire une famille d'estimateurs qui sont ensuite agrégés par une moyenne pondérée des estimations (en régression) ou un vote à la majorité (en classification). Les estimateurs sont construits de manière récursive : chaque estimateur est une version adaptative du précédent en donnant plus de poids aux observations mal ajustées ou mal prédites. L'estimateur construit à l'étape k concentrera donc ses efforts sur les observations mal ajustées par l'estimateur à l'étape $k - 1$.

L'algorithme adaboost développé par Freund et Schapire (1997) est le plus populaire dans la famille des algorithmes boosting. On désigne par $g(x)$ une règle de classification faible. Par faible, nous entendons une règle dont le taux d'erreur est légèrement meilleur que celui d'une règle purement aléatoire. L'idée consiste à appliquer cette règle plusieurs fois en affectant judicieusement un poids différent aux observations à chaque itération.

On se place dans le contexte de la classification supervisée binaire. On dispose d'un échantillon d'apprentissage $D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$ et d'une règle faible qui peut s'appliquer à un échantillon pondéré (par exemple les arbres CART).

Algorithme adaboost

1. Initialiser les poids $w_i = 1/n$ pour tout $i \in \{1, \dots, n\}$
2. Pour l de 1 à L
   (a) Ajuster la règle faible sur l'échantillon D_n pondéré par les poids $w_1, \dots, w_n$ on note $g_l(x)$ le classifieur issu de cet ajustement
   (b) Calculer
   $$e_l = \frac{\sum_{i=1}^n w_i \mathbb{I}_{y_i \neq g_l(x_i)}}{\sum_{i=1}^n w_i}$$
   (c) Calculer
   $$\alpha_l = \log\left(\frac{1-e_l}{e_l}\right)$$
   (d) Réajuster les poids
   $$w_i = w_i \exp\left(\alpha_l \mathbb{I}_{y_i \neq g_l(x_i)}\right)$$

Notons que la règle est faible mais respecte tout de même le fait que $e_m < 1/2$. Une fois terminées les itérations de l'algorithme adaboost l'observation x à prévoir est affectée à la classe 0 si $\sum_{l=1}^L \alpha_l g_l(x) < 1/2$ et à la classe 1 sinon.

Analyser et commenter cette procédure.

Exercice 4

1 (2 pts) On se place dans le contexte d'un problème de reconnaissance des chiffres manuscrits à partir d'image 28×28. Deux modèles implémentés en utilisant la bibliothèque `keras` de `Python` sont mis en oeuvre. On peut accéder à une description des modèles en utilisant l'attribut summary(). On obtient les résultats donnés à la page suivante.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten   (Flatten)          (None, 784)               0
_________________________________________________________________
dense   (Dense)              (None, 512)               401920
_________________________________________________________________
dense   (Dense)              (None, 512)               262656
_________________________________________________________________
dropout   (Dropout)          (None, 512)               0
_________________________________________________________________
dense   (Dense)              (None, 10)                5130
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
```

```
____________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d   (Conv2D)            (None, 28, 28, 6)         60
_________________________________________________________________
max_pooling2d (MaxPooling)   (None, 14, 14, 6)         0
_________________________________________________________________
dropout   (Dropout)          (None, 14, 14, 6)         0
_________________________________________________________________
conv2d   (Conv2D)            (None, 14, 14, 8)         440
_________________________________________________________________
max_pooling2d (MaxPooling)   (None, 7, 7, 8)           0
_________________________________________________________________
dropout (Dropout)            (None, 7, 7, 8)           0
_________________________________________________________________
flatten (Flatten)            (None, 392)               0
_________________________________________________________________
dense   (Dense)              (None, 30)                11790
_________________________________________________________________
dropout (Dropout)            (None, 30)                0
_________________________________________________________________
dense   (Dense)              (None, 10)                310
=================================================================
Total params: 12,600
Trainable params: 12,600
Non-trainable params: 0
```

Décrire clairement les modèles proposés.

Correction examen final HMMA 303 06/11/2019

Exercice 1

On utilise l'erreur Out-Of-Bag

Exercice 2

1] $\mathbb{P}(Y=1|X=x) = \left[\dfrac{e^{-x}}{e^{-x}+\frac{1}{4}e^{-\frac{x}{4}}}\right]$

2] Le classifieur optimal consiste à choisir la classe 1 ssi $\mathbb{P}(Y=1|X=x) \geq 1/2$

$\iff e^{-x} \geq \frac{1}{2}\left[e^{-x}+\frac{1}{4}e^{-\frac{x}{4}}\right]$

$\iff x \leq \frac{4}{3}\log(4)$

3] $g^0(x)$ choisit la classe dont la valeur de x est la plus proche de sa moyenne.

4] $C(g^0) = \mathbb{P}(Y \neq g^0(X))$

$= \mathbb{P}(Y=1 \cap X \geq \frac{4}{3}\log(4))$

$+ \mathbb{P}(Y=0 \cap X < \frac{4}{3}\log(4))$

$= \left(\frac{1}{2}\right) \int_{\frac{4}{3}\log(4)}^{\infty} e^{-x}\,dx$

$+ \left(\frac{1}{2}\right) \int_{0}^{\frac{4}{3}\log(4)} \frac{1}{4} e^{-\frac{x}{4}}\,dx$

$= \frac{1}{2}\left(e^{-\frac{4}{3}\log(4)} + 1 - e^{-\frac{1}{3}\log(4)} \right)$

$C(g^1) = \frac{1}{2}\left(e^{-5/2} + 1 + e^{-5/8} \right)$

3

5] On choisit la classe 1 si

$$\mathbb{P}(Y=1 \mid X=x) \geq \frac{1}{3}$$

$$\Leftrightarrow e^{-x} \geq \frac{1}{3}\left(e^{-x} + \frac{1}{4}e^{-\frac{x}{4}}\right)$$

$$\Leftrightarrow x \leq \frac{4}{3}\log(8)$$

Comme $\log(8) > \log(4)$ on choisit alors ce cas plus fréquemment la classe 1. En effet, c'est deux fois plus coûteux ici de se tromper lorsque la classe 1 est vraie.