Apprentissage Statistique pour l'IA

Examen Théorique - Juin 2024

BUT 3

Durée : 2h00

Guillaume Metzler
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France
guillaume.metzler@univ-lyon2.fr

L'usage de la calculatrice, de l'ordinateur ou de tout autre matériel électronique n'est pas autorisé pendant la durée de cet examen. L'usage des notes des fiches de cours et des fiches personnelles est également prohibé.

Les exercices cet examen sont indépendants et peuvent être traités dans n'importe quel ordre. Il est simplement demander de bien préciser l'exercice ainsi que la question traités.

Questions de cours

On considère un échantillon $S = \{(\mathbf{x}_i, y_i)\}$ de taille m où $\mathbf{x}_i \in \mathbb{R}^d$ et $y_i \in \{-1, 1\}$.

1. On considère le problème suivant :

$$\begin{array}{ll} \min\limits_{\xi \in \mathbb{R}^m,\ (\mathbf{w},b)\in\mathbb{R}^{d+1}} & \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{m}\sum_{i=1}^m \xi_i \\ s.t. & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i, \quad \text{for all } i = 1, ..., m, \\ & \xi_i \geq 0, \quad \text{for all } i = 1, ..., m. \end{array}$$

   (a) A quel algorithme est associé ce problème d'optimisation.
   (b) Préciser le rôle de chacun des termes de la somme dans la fonction objective.
   (c) Quel est l'influence de l'hyper-paramètre C dans la fonction objective ?
   (d) Décrire les deux contraintes.
   (e) Que représentent les variables ξ_i ? Illustrer les valeurs que peut prendre cette variable sur un dessin en $2D$.
   (f) Quelle loss est souvent lié au problème associé à cet algorithme ?

2. Nous avons vu en cours que l'erreur d'un algorithme se décompose en trois termes. Donner ces trois composantes de l'erreur et décrire le sens de de ces derniers. Sur quelles composantes de l'erreur agit-on lors de l'apprentissage d'un modèle ?

3. Rappeler les deux grandes méthodes ensemblistes vues en cours. Quelle est la nature des hypothèses utilisée dans chacun des cas et sur quelles composante de l'erreur agit-on à l'aide de ces différentes approches ?

Exercice 1 : A propos du SVM

On considère le jeu de données étiquetées suivant :

y	-1	-1	-1	-1	+1	+1	+1
x_1	2	4	-1	0	1	6	5
x_2	1	3	4	7	-6	-3	-5
α	0.1	0.3	0	0.5	0.7	0	0.2

1. Rappeler les noms et expressions des trois grands noyaux présentés au cours ainsi que les hyper-paramètres liés à chaque noyaux.
2. Quels sont les avantages et inconvénients de l'utilisation d'une méthode à noyaux ? (On pourra penser à la complexité des algorithmes)

On souhaite déterminer l'étiquette de la donnée $\mathbf{x}'$ définie par $\mathbf{x}' = (1, 5)$ et on considère un noyau **polynomiale**. On prendra un polynôme de degré 2 et la constante mise en jeu dans le polynôme sera égale à 1.

3. Rappeler la règle de classification pour un SVM à noyaux.
4. Déterminer l'étiquette prédite par le modèle pour la donnée $\mathbf{x}'$ précédemment définie.

Exercice 2 : Autour des méthodes ensemblistes

Bagging

Les arbres de décision sont des modèles classiques auxquels sont appliqués la procédure de bagging, donnant ainsi naissance au bagging.

1. Rappeler, en détail, l'algorithme du bagging et l'intérêt du double échantillonnage effectué dans l'algorithme de forêt aléatoire.

On considère le jeu de données de classification suivant :

Exemples	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$
y	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1
x_1	-3	-4	2	4	-1	0	1	6	5	8
x_2	-4	-2	1	3	4	7	-6	-3	-5	9

Notre objectif est de construire une forêt constituée de deux arbres ayant chacun une profondeur de 2, *i.e.*, deux séparations (ou splits) seront effectuées.

A. Construction d'un premier arbre Pour ce premier arbre, les règles employées sont les suivantes :

- Si $x_2 \leq 3$ –> feuille de gauche
  Dans cette feuille de gauche
  — Si $x_1 \leq 0$ –> feuille de gauche, sinon feuille de droite.

- Si $x_2 > 3$ –> feuille de droite
  Dans cette feuille de droite
  — Si $x_1 > 0$ –> feuille de gauche, sinon feuille de droite.

De plus, on considérera uniquement les exemples $\mathbf{x}_i$ pour $i \in [\![4, 10]\!]$ pour entraîner notre arbre et les autres exemples serviront à la validation.

2. Représenter l'arbre associé à la règle ci-dessus et l'étiquette associée à chaque feuille (en cas d'égalité, on va privilégier une prédiction positive).
3. Déterminer les performances, en *accuracy*, de ce modèle h_1, sur l'ensemble d'apprentissage et sur l'ensemble de validation.

B. Construction d'un deuxième arbre Pour ce deuxième arbre, les règles employées sont les suivantes :

- Si $x_1 < 2$ –> feuille de gauche
  Dans cette feuille de gauche
  — Si $x_2 \geq 2$ –> feuille de gauche, sinon feuille de droite.
- Si $x_1 \geq 2$ –> feuille de droite
  Dans cette feuille de droite
  — Si $x_2 \leq -2$ –> feuille de gauche, sinon feuille de droite.

De plus, on considérera uniquement les exemples $\mathbf{x}_i$ pour $i \in [\![1, 7]\!]$ pour entraîner notre arbre et les autres exemples serviront à la validation.

4. Représenter l'arbre associé à la règle ci-dessus et l'étiquette associée à chaque feuille (en cas d'égalité, on va privilégier une prédiction positive).
5. Déterminer les performances, en *accuracy*, de ce modèle h_2, sur l'ensemble d'apprentissage et sur l'ensemble de validation.

C. Performance de la forêt On considère la forêt aléatoire défini par $H_T = \frac{1}{2}(h_1 + h_2)$ et le jeu de données suivant :

y	-1	-1	+1	+1
x_1	-1	2	3	4
x_2	-5	6	-1	-2

6. Evaluer les performances, en terme d'accuracy, de la forêt aléatoire sur ce jeu de données.

Boosting

Considérons que l'on mette en place l'algorithme Adaboost avec un séparateur linéaire de la forme $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ de sorte à ce que ce dernier renvoie une valeur dans l'ensemble $\{-1, 1\}$. Les hypothèses sont apprises sur le jeu de données suivant

y	-1	1	-1	-1	-1	1	1	-1
x_1	-2	-1	-1	2	-4	-2	3	1
x_2	6	2	4	-3	-1	-2	-2	1

Au cours de la première itération de notre algorithme, les paramètres du modèle sont donnés par $\mathbf{w} = (1, 1)$ et $b = -1$ et chaque exemple a un poids égal à $\frac{1}{m}$ où m désigne le nombre d'exemples.

1. Evaluer l'erreur global de votre modèle (dans le cas où $h(\mathbf{x}) = 0$, on attribuera l'étiquette positive à la donnée.)

2. En déduire le poids du modèle nouvellement appris
3. Déterminer la pondération des exemples pour la prochaine itération de l'algorithme Adaboost.

On considère le modèle suivant :

$$H(\mathbf{x}) = \text{sign}(\alpha_1 h_1(\mathbf{x}) + \alpha_2 h_2(\mathbf{x}) + \alpha_3 h_3(\mathbf{x})),$$

où $(\alpha_1, \alpha_2, \alpha_3) = (2, 3, 1.5)$ et $h_1(\mathbf{x}) = \text{sign}(3x_1 + 2), h_2(\mathbf{x}) = \text{sign}(2x_1 + 3x_2 - 4)$ et $h_3(\mathbf{x}) = \text{sign}(-x_1 + 2x_2 - 1)$.

4. En repartant de votre connaissance de l'algorithme Adaboost, quel est le *weak learner* qui a l'erreur la plus faible ?
5. On considère les individus

$$\mathbf{x}'_1 = (1, 0),\ \mathbf{x}'_2 = (2, 3) \quad \text{et} \quad \mathbf{x}'_3 = (-3, 2)$$

Prédire l'étiquette de ces différents individus par votre méthode ensembliste.