Apprentissage Statistique

Examen - Contrôle de Connaissances (durée 1 heure 30)

Les notes de cours ne sont pas autorisées, l'usage d'ordinateurs ou tablettes est prohibé.

Notations. On se place dans le cadre du modèle de classification où X est un vecteur aléatoire sur $\mathbb{R}^d$, $d \geq 1$, de loi $\mu(dx)$ et Y est une variable aléatoire à valeurs dans $\{-1,\ +1\}$. On pose $\eta(X) = \mathbb{P}(Y = 1 \mid X)$, $p = \mathbb{P}\{Y = +1\ \} = \mathbb{E}[\eta(X)]$ et on suppose la v.a. $\eta(X)$ continue pour simplifier. Le risque d'un classifieur $g : \mathbb{R}^d \to \{-1,\ +1\}$ est défini par $L(g) = \mathbb{P}\{Y \neq g(X)\}$. On suppose que l'on dispose d'une collection d'exemples $\mathcal{D}_n = \{(X_1,\ Y_1),\ \ldots,\ (X_n, Y_n)\}$, copies indépendantes du couple générique (X, Y). On désigne par $\langle ., .\rangle$ et $||.||$ le produit scalaire et la norme euclidienne usuels sur $\mathbb{R}^d$. La fonction indicatrice d'un événement quelconque $\mathcal{E}$ est notée $\mathbb{I}\{\mathcal{E}\}$.

Théorie de l'Apprentissage

1 Soit $\mathcal{A}$ une classe de sous-ensembles mesurables de $\mathbb{R}^d$. Définir son coefficient d'éclatement à l'ordre n, sa dimension de Vapnik-Chervonenkis.

2 Définir le risque empirique $\widehat{L}_n(g)$ d'un classifieur g calculé sur l'échantillon d'apprentissage $\mathcal{D}_n$.

3 Expliquer en quoi consiste le principe de minimisation du risque empirique appliqué à une classe $\mathcal{G}$ de classifieurs.

Pour chacune des affirmations ci-dessous, préciser si elle est vraie ou fausse (aucune justification n'est demandée).

4 Le problème de la minimisation du risque empirique $\widehat{L}_n(g)$ est NP-difficile et la plupart des algorithmes reposent en pratique sur un relaxation du problème original.

5 Pour mettre en oeuvre la sélection de modèle, on se fonde sur une estimation de l'erreur de généralisation calculée sur un échantillon de validation, par validation croisée ou par rééchantillonnage.

6 Le classifieur minimisant le risque à coût sensitif

$$L_w(g) = 2p(1-w)\mathbb{P}\{g(X) = -1 \mid Y = +1\} + 2(1-p)w\mathbb{P}\{g(X) = +1 \mid Y = -1\},$$

avec $w \in (0, 1)$, est donné par : $\forall x \in \mathbb{R}^d$,

$$g^*(x) = 2\mathbb{I}\{\eta(x) \geq w\} - 1.$$

Algorithmes "basiques"

1. On dispose d'une métrique $D(.,.)$ sur l'espace d'entrée $\mathbb{R}^d$. Soit $k \in \{1,\ \ldots,\ n\}$. Pour le problème de la classification binaire, explicitez la règle des k-plus proches voisins fondée sur la métrique D et l'échantillon $\mathcal{D}_n$.

2. Préciser si l'assertion suivante est vraie ou fausse (aucune justification n'est demandée) : "Pour $k = 1$, l'erreur d'apprentissage de la règle des plus proches voisins est nulle".

Pour chaque affirmation ci-dessous, préciser si elle est vraie ou fausse (aucune justification n'est demandée).

3 Le modèle de la régression logistique linéaire pour la classification binaire requiert de stipuler une forme paramétrique pour la loi de X.

4 Le modèle de l'analyse discriminante linéaire pour la classification binaire requiert de stipuler que la loi de X sachant $Y = +1$ et la loi de X sachant $Y = -1$ sont des gaussiennes de même moyenne mais de matrices de covariance différentes.

5 La sortie de l'algorithme du Perceptron monocouche cesse d'évoluer au bout d'un nombre variable mais fini d'itérations quel que soit l'échantillon d'apprentissage.

6 Sans contrainte sur la profondeur maximale de l'arbre de décision, l'algorithme CART permet d'obtenir un classifieur d'erreur d'apprentissage nulle.

ALGORITHMES "AVANCÉS"

On se place toujours dans le cadre de la classification supervisée binaire déjà décrite plus haut.

1. Le problème d'optimisation résolu par l'algorithme SVM peut être formulé de façon à l'interpréter comme la minimisation d'un risque empirique pour la perte ?hinge ? $(1 + u)_+$ pénalisé.
2. L'"astuce du noyau" permet de déterminer une règle de décision affine dans l'espace de représentation (et non linéaire dans l'espace d'entrée original si le noyau n'est pas un produit scalaire dans l'espace d'entrée $\mathbb{R}^d$) sans avoir à spécifier la représentation afférente (*i.e.* "feature variables").
3. L'algorithme ADABOOST produit itérativement un classifieur $sgn(f(X))$ minimisant une version empirique de l'erreur exponentielle :

$$L_{\exp}(f) = \mathbb{E}[\exp(-Yf(X))].$$

CLUSTERING

Soit X un vecteur aléatoire sur $\mathbb{R}^d$ de loi μ et $\mathcal{D}_n = \{X_1, X_2, \ldots, X_n\}$ un n-échantillon i.i.d tiré de cette loi. Soit $K \in \{1, \ldots, n\}$ le nombre de clusters désirés.

1. L'algorithme K-means vise à déterminer une partition $C_1, \ldots, C_K$ du nuage de points maximisant
$$\sum_{k=1}^{K} \sum_{(X_i,X_j)\in C_k^2} ||X_i - X_j||^2.$$
2. L'algorithme K-means vise à déterminer une partition $C_1, \ldots, C_K$ du nuage de points minimisant
$$\sum_{1\le k\neq l\le K} \sum_{(X_i,X_j)\in C_k\times C_l} ||X_i - X_j||^2.$$
3. Les sorties de l'algorithme K-means cessent d'évoluer au bout d'un nombre fini d'itérations.