TP 2 : Apprentissage supervisé

L'objectif de ce Tp est d'implémenter les modèles d'apprentissage supervisé vus en cours à savoir : K-nn, NB, LDA et QLD. Pour le modèle des SVM, vous pouvez utiliser les bibliothèques existantes pour l'apprentissage et le test. Pour l'implémentation, vous avez le libre choix du langage de programmation (Python, Matlab, R, etc.).
Afin de tester, les modèles développés, vous allez utiliser la base de données Iris téléchargeable ici. Iris est un ensemble de données introduit en 1936 par Ronald Aylmer Fisher comme un exemple d'analyse discriminante. Cet ensemble contient 150 exemples de critères observés sur 3 espèces différentes d'iris de Gaspésie. Chaque exemple est composé de quatre attributs (longueur et largeur des sépales en cm, longueur et largeur des pétales en cm) et d'une classe (l'espèce).

1) k-plus proches voisins

L'algorithme des k-plus proches voisins est l'un des algorithmes d'apprentissage automatique les plus simple. Cet algorithme est considéré comme l'un des meilleurs algorithmes de fouille de données pour sa capacité à produire des classifieurs simples mais puissants. L'algorithme ci-dessous résume, l'apprentissage et le test d'une nouvelle observation.

Algorithm

Inputs : L'ensemble de données d'apprentissage $S = \{(x_i, y_i)\}_{i=1}^{n}$
Une nouvelle instance x' à classifier
Nombre de plus proche voisins k

1: **for** i=1,. . . ,n **do**
2: calculer la distance $d(x', x_i)$ entre x_i et x'
3: **end for**
4: sélectionner les k distances les plus proches de x'
5: affecter x' à la classe majoritairement représentée par ses voisins

Outputs : $\hat{y}$. la classe de x' selon les k plus proches voisins

1.1) Implémenter l'algorithme k-nn pour la prédiction d'une nouvelle observation $\boldsymbol{x^{test}} = (\boldsymbol{x_1^{test}}, \boldsymbol{x_2^{test}}, \ldots, \boldsymbol{x_i^{test}}, \ldots, \boldsymbol{x_n^{test}})$

1.2) En se basant sur la méthode de validation croisée (10-fold cross validation), évaluer les performances de l'approche des k-plus porches voisins en termes de taux de bonne classification, rappel, précision et f-mesure.
1.3) Donner la matrice de confusion.
1.4) Analyser les résultats obtenus.
1.5) Refaire 1.2 et 1.3 en utilisant une bibliothèque existante incluant le k-nn. Comparer les résultats obtenus avec ceux obtenus précédemment.
1.6) Conclure

2. Naïve Bayes

Naïve de Bayes est un modèle d'apprentissage supervisé basé sur le théorème de Bayes avec de fortes hypothèses d'indépendance entre les variables d'entrées. Le principal avantage de ce modèle réside dans la simplicité de sa mise en œuvre qui ne nécessite aucune estimation itérative compliquée des paramètres.

Soit un ensemble d'apprentissage $S = \{(\boldsymbol{x}_i, y_i)\}_{i \leq n}$ où la variable d'entrée $\boldsymbol{x_i}$ appartient à $\mathbb{R}^d$ i.e $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{il}, \ldots, x_{id})$ et la variable d'intérêt y_i appartient à un ensemble fini à K classes, $Y = \{c_1, c_2, \ldots, c_k, \ldots, c_K\}$

$$\mathrm{P}(y_i = c_k | \boldsymbol{x_i}) = \frac{\mathrm{P}(y_i = c_k)\mathrm{P}(\boldsymbol{x_i} | y_i = c_k)}{P(\boldsymbol{x_i})}$$

Dans le cadre de NB, on suppose que les variables sont indépendantes, on obtient ainsi

$$\mathrm{P}(y_i = c_k | \boldsymbol{x_i}) = \frac{\mathrm{P}(y_i = c_k) \prod_{l=1}^{d} \mathrm{P}(x_{il} | y_i = c_k)}{\sum_{k=1}^{K} \mathrm{P}(y_i = c_k) \prod_{l=1}^{d} \mathrm{P}(x_{il} | y_i = c_k)}$$

Dans ce TP, on considère le cas continue, les valeurs continues associées à chaque classe sont distribuées selon une distribution normale (ou gaussienne) :

$$\mathrm{P}(x_{il} | y_i = c_k) = \frac{1}{\sigma_{kl}\sqrt{2\pi}} \; e^{-\frac{1}{2} \; \frac{(x_{il} - \mu_{kl})^2}{\sigma_{kl}^2}}$$

2.1) Apprentissage du modèle

L'apprentissage du modèle consiste à estimer les paramètres $\hat{\pi}_k = \mathrm{P}(y_i = c_k)$, $\hat{\mu}_{kl}$ *et* $\hat{\sigma}_{kl}$:

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_{kl} = \frac{1}{n_k} \sum_{i|y_i=c_k} x_{il}$$

$$\hat{\sigma}_{kl} = \frac{1}{n_k - 1} \sum_{i|y_i=c_k} (x_{il} - \hat{\mu}_{kl})^T (x_{il} - \hat{\mu}_{kl})$$

2.2) Prédiction d'une nouvelle observation (classification)

Une fois les paramètres du modèle sont estimés, la classification d'une nouvelle observation $\boldsymbol{x'}$, consiste à calculer la probabilité a postériori P($y' = c_k|\boldsymbol{x'}$) pour toutes les classes.

En appliquant le MAP on obtient :

$$\hat{y} = \underset{y \in Y}{\text{argmax}} P(y' = c_k|\boldsymbol{x'})$$

$$= \underset{y \in Y}{\text{argmax}}\ \hat{\pi}_k \prod_{l=1}^{d} \frac{1}{\hat{\sigma}_{kl}\sqrt{2\pi}}\ e^{-\frac{1}{2}\ \frac{(x_{il} - \hat{\mu}_{kl})^2}{\hat{\sigma}_{kl}^2}}$$

2.3) Travail à effectuer

3.3.1) Implémenter l'algorithme NB pour la prédiction d'une nouvelle observation $\boldsymbol{x^{test}} = (\boldsymbol{x_1^{test}}, \boldsymbol{x_2^{test}}, \ldots, \boldsymbol{x_i^{test}}, \ldots, \boldsymbol{x_n^{test}})$

2.3.1) En se basant sur la méthode de validation croisée (10-fold cross validation), évaluer les performances du NB en termes de taux de bonne classification, rappel, précision et f-mesure.

2.3.2) Donner la matrice de confusion.

2.3.3) Analyser les résultats obtenus.

2.3.4) Refaire 2.3.2 et 2.3.3 en utilisant une bibliothèque existante incluant le NB. Comparer les résultats obtenus avec ceux obtenus précédemment.

2.3.5) Conclure

3. Analyse discriminante linéaire/quadratique (LDA/QDA)

L'analyse discriminante linéaire (linear discriminant analysis (LDA)) est une méthode d'apprentissage supervisé qui fait partie des méthodes dites linéaires). Cette méthode cherche à trouver une frontière (linéaire) de

discrimination entre les classe qui peut s'écrire sous forme d'une ou plusieurs combinaisons linéaires des covariables. Dans le cas de l'analyse discriminante linéaire on suppose toutes les classes ont la même matrice de covariance (hypothèse d'**Homoscédasticité**)

$$\Sigma = \Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = \cdots = \Sigma_K$$

Contrairement à l'analyse discriminante linéaire, dans l'analyse discriminante quadratique la frontière de discrimination est quadratique. Dans le cas de la QLD, les données ont des matrices de covariance différentes d'une classe à une autre (hypothèse d'**Hétéroscédasticité**)

$$\Sigma_1 \neq \Sigma_2 \ldots \neq \Sigma_k \neq \cdots \neq \Sigma_K$$

Dans ce qui suit, on supposons que $\boldsymbol{x_i}$ suit une loi normale multidimensionnelle (loi multi-normale) conditionnellement aux classes (y_i) :

$$\mathrm{P}(\boldsymbol{x_i}|y_i = c_k;\theta_k) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\Sigma_k}}\, e^{-\frac{1}{2}(\boldsymbol{x_i}-\boldsymbol{\mu_k})^T \Sigma_k^{-1}(\boldsymbol{x_i}-\boldsymbol{\mu_k})}$$

3.1) Apprentissage du modèle

L'apprentissage du modèle consiste à estimer les paramètres $\hat{\pi}_k = \mathrm{P}(y_i = c_k)$, $\hat{\boldsymbol{\mu}}_k$ *et* Σ_k :

Dans les deux cas (LDA et QAD) $\hat{\pi}_k$ et $\hat{\boldsymbol{\mu}}$ sont estimés de la même phase

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i|y_i=c_k} x_i$$

Dans le cas de LDA $\hat{\Sigma}$ est estimé comme suit :

$$\hat{\Sigma} = \hat{\Sigma}_k = \frac{1}{\mathrm{n} - \mathrm{K}} \sum_{k=1}^{K} \sum_{i|y_i=c_k} (x_i - \hat{\mu}_k)^T (x_i - \hat{\mu}_k)$$

Dans le cas de QDA $\hat{\Sigma}$ est estimé comme suit :

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i|y_i=c_k} (x_i - \hat{\mu}_k)^T (x_i - \hat{\mu}_k)$$

3.2) Prédiction d'une nouvelle observation (classification)

Une fois les paramètres du modèle sont estimés, la classification d'une nouvelle observation $\boldsymbol{x'}$, consiste à calculer la probabilité a postériori $\mathrm{P}(y' = c_k|\boldsymbol{x'})$ pour toutes les classes.

En appliquant le MAP on obtient :

$$\hat{y} = \underset{y \in \mathrm{Y}}{\operatorname{argmax}}\, P(y' = c_k|\boldsymbol{x'})$$
$$= \underset{y \in \mathrm{Y}}{\operatorname{argmax}} \frac{\mathrm{P}(y=c_k)\mathcal{N}(\boldsymbol{x},\hat{\mu}_k,\hat{\Sigma}_k)}{\sum_{g=1}^{K} \mathrm{P}(y=c_g)\mathcal{N}(\boldsymbol{x},\ \hat{\mu}_g,\hat{\Sigma}_k)}$$

3.3) Travail à effectuer

3.3.2) Implémenter les algorithmes LDA et QDA pour la prédiction d'une nouvelle observation $\boldsymbol{x^{test}} = (\boldsymbol{x_1^{test}}, \boldsymbol{x_2^{test}}, \ldots, \boldsymbol{x_i^{test}}, \ldots, \boldsymbol{x_n^{test}})$

3.3.3) En se basant sur la méthode de validation croisée (10-fold cross validation), évaluer les performances de LDA et QDA en termes de taux de bonne classification, rappel, précision et f-mesure.

3.3.4) Donner la matrice de confusion.

3.3.5) Analyser les résultats obtenus.

3.3.6) Refaire 3.3.2 et 3.3.3 en utilisant une bibliothèque existante incluant le LDA et QDA. Comparer les résultats obtenus avec ceux obtenus précédemment.

3.3.7) Conclure

4. Séparateurs à Vaste Marge

Dans cette partie, l'objectif est d'utiliser une bibliothèque existante et proposer un SVM pour classifier les types de fleurs de la base Iris.

4.1) En se basant sur la méthode de validation croisée (10-fold cross validation), proposer un SVM pour classifier les types de fleurs de la base Iris et évaluer ses performances en termes de taux de bonne classification, rappel, précision et f-mesure

4.2) Justifier le choix des paramètres du SVM

4.3) Donner la matrice de confusion.
4.4) Analyser les résultats obtenus.
4.5) Comparer les résultats des différentes approches précédentes (k-nn, NB, LDA, QDA et SVM).
4.6) Proposer une méthodologie pour améliorer les résultats obtenus précédemment.