TP1 : Illustration du phénomène de sur-apprentissage

ENSTA

2020/2021

Adresse mail: mohammed.sedki@universite-paris-saclay.fr Page web: masedki.github.io

---

Objectif

L'objectif de cette séance introductive est l'illustration du phénomène de sur-apprentissage sur un jeu de données simulé autour d'un problème de classification dans $\mathbb{R}^2$. Le jeu de données provient de la librairie R appelée ElemStatLearn. Pour une installation manuelle, la librairie est disponible ici.

Lecture et découverte du jeu de données

```
require(ElemStatLearn)
require(MASS)
require(class)
data("mixture.example")
names(mixture.example)
```

L'objet mixture.example contient de nombreux sous-objets. Les objets x et y contiennent les données d'apprentissage.

- Tracer le nuage de points x en mettant comme couleur rouge si $y = 1$ et vert si $y = 0$.

L'objet xnew contient un maillage du domaine $[-2.6, 4.2] \times [-2, 2.9]$. Ce maillage nous servira de support discret pour tracer la règle de classification de modèles que nous ajusterons sur ce jeu de données.

- Visualiser ce maillage à l'aide d'un graphique.

Modèle des k plus proches voisins : knn

La fonction knn de la librairie class permet d'ajuster un modèle des k plus proches voisins sur un jeu de données d'apprentissage et calculer les probabilités de classement ainsi que les prédictions d'un jeu de données test.

- Ajuster un modèle des 15 plus proches voisins pour prédire les étiquettes de l'objet y à partir des covariables de l'objet x. Prédire les étiquettes des points du maillage xnew. Spécifier l'option nécessaire pour récupérer les proportions de vote majoritaire. Afficher le résumer du modèle à l'aide de la fonction summary.
- Extraire les proportions de votes majoritaires à l'aide de la fonction attr.
- Calculer la proportion de votes pour la classe $y = 1$ pour chaque point test du maillage.

- À l'aide de la fonction `contour`, tracer la ligne de niveau de la fonction $\hat{f}_n(x) = 0.5$ où $f(x) \approx \mathbb{P}(Y = 1|X = x)$ pour chaque point x du maillage `xnew` et $\hat{f}_n$ est l'approximation de f obtenue par le modèle des 15 plus proches voisins ajustés sur le jeu de données d'apprentissage.
- Interpréter la ligne de niveau.
- Tracer la même ligne de niveau lorsque $\hat{f}_n$ est l'approximation de f obtenue par le modèle 1 plus proches voisins ajustés sur le jeu de données d'apprentissage.
- Que peut-on dire du lien entre nombre de voisins et la complexité du modèle ?

Choix du nombre de voisins

Le code suivant permet de simuler un jeu de données supplémentaire suivant la même loi de probabilité qui a permis de simuler les 200 données contenues dans les objets `x` et `y`.

```
set.seed(123)
centers = c(sample(1:10, 5000, replace=TRUE), sample(11:20, 5000, replace=TRUE))
means = mixture.example$means
means = means[centers, ]
xtest = mvrnorm(10000, c(0,0), 0.2*diag(2))
xtest = xtest + means
ytest = c(rep(0, 5000), rep(1, 5000))
```

Nous proposons de choix le nombre de voisins optimal parmi stocker dans l'objet `K` comme suit

```
K = c(1,3,5,7,9,11,15,17,23,25,35,45,55,83,101,151)
```

- À l'aide du jeu de données d'apprentissage précédent, calculer une erreur d'apprentissage et une erreur de test basée sur `xtest` et `ytest` pour chaque valeur du nombre de voisins dans `K`.
- Tracer les deux types d'erreurs sur la même figure en fonction du nombre de voisins k.
- Commenter, interpréter et choisir la valeur optimale de k.