Perceptron et K-Nearest Neighboors

Le but de cette séance est de comparer quelques règles simple de classification binaire.
On se place dans ce TP dans un cadre de classification binaire, avec n observations, où $Y_i \in \{-1, 1\}$ est expliqué par p régresseurs $X_i^1, \ldots, X_i^p$. Deux exemples seront traités :

- un exemple simulé, pour lequel on prendra $p = 2$, (X_i^1, X_i^2) uniformément répartis dans le carré $[0, 1] \times [0, 1]$, et la loi conditionnelle de Y_i sachant (X_i^1, X_i^2) est donnée par :

$$\begin{cases} \mathbb{P}(Y_i = 1) = \alpha & \text{si} \quad 2X_i^1 + X_i^2 < 1.5, \\ \mathbb{P}(Y_i = 1) = \beta & \text{si} \quad 2X_i^1 + X_i^2 > 1.5, \end{cases}$$

  où α et β sont deux paramètres sur lesquels on jouera un peu pour tester les méthodes ;
- un exemple de données réelles issu du livre [1] "*The elements of statistical learning : data mining, inference and prediction*" (section 4.4.2) concernant l'explication du risque d'attaques cardiaques par des facteurs comme l'âge, la consommation de tabac, etc. dont on avait récupéré les données dans le TP précédent.

On cherchera dans ce TP à comparer les mérites de trois approches simples et populaires du problème : la régression logistique, l'algorithme du perceptron de Rosenblatt et la méthodes des K plus proches voisins. Lors du TP précédent, on a construit une fonction R appelée `rexemple` prenant pour arguments n, α, et β, et renvoyant un n-échantillon $\left(X_i^1, X_i^2, Y_i\right)_{1 \leq i \leq n}$ de l'exemple simulé sous la forme de liste de trois éléments : `list(x1,x2,y)`, où `x1,x2` et `y` sont des tableaux. On avait également étudié la régression logistique sur l'exemple simulé et sur les données réelles.

- Perceptron -

1. Programmer et essayer l'algorithme du perceptron tel qu'il est décrit dans la section 4.5 de [1]. On codera une version qui prend comme arguments : les observations $\left(X_i^1, X_i^2, Y_i\right)_{1 \leq i \leq n}$, le pas (de gradient) ρ, le nombre d'époques (ou de passages sur la base de données entière), et un vecteur décrivant l'hyperplan d'initialisation. La sortie renverra un vecteur codant l'hyperplan séparateur.
2. Comparer visuellement avec le résultat obtenu par régression logistique.
3. Visualiser l'évolution de la droite de séparation au fil des itérations.
4. Montrer (numériquement) sa convergence quand les données sont séparables c'est-à-dire par exemple si $\alpha = 1$ et $\beta = 0$. Que se passe-t-il dans les autres cas ? Tester sur un exemple à quatre points qui ne peuvent pas être séparés
5. Illustrer que même dans le cas séparable le nombre de points mal classés ne décroît pas forcement à chaque itération.
6. Que donne l'algorithme du perceptron sur les données réelles ?

Questions optionnelles (à traiter si le reste du TP est fini) :

7. Étudier numériquement la vitesse de convergence dans le cas suivant : les X_i sont des points uniformément repartis sur les segments $\{0\} \times [0, M]$ (alors les Y_i valent -1) ou bien sur le segment $\{\delta\} \times [0, M]$ (alors les Y_i valent -1). De plus la proportion de de 1 est égal à $1/2$. On regardera l'impact de δ, M et n sur le temps de convergence du perceptron.

- "K-Nearest Neighbors" -

8. Appliquer la méthode k-NN aux données simulées, et visualiser le classifieur obtenu pour différentes valeurs de K (utiliser la fonction `knn`).
9. A partir de quelle taille d'échantillon obtient-on un classifieur de bonne qualité ?
10. Comment choisir K ? Mettre en oeuvre une sélection par validation croisée.
11. Traiter ensuite le cas réel de l'explication du risque d'attaques cardiaques.

12. Récupérer sur le site http://www-stat.stanford.edu/~tibs/ElemStatLearn/ les données de la base ZIPCODE, et comprendre ce qu'elles contiennent.
13. Appliquer la méthode k-NN (version multi-classe) aux données issues de la base ZIPCODE avec différents choix de $K \geq 1$. Estimer la matrice de confusion $(\mathbb{P}\{C_K(X) = i, Y = j\})_{i,\,j}$ associée au classifieur C_K ainsi obtenu. Proposer une méthode pour choisir K et la mettre en oeuvre.

Références

[1] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. http://www-stat.stanford.edu/~tibs/ElemStatLearn/. 1