Sujet de TP à rendre RT3
Implémentation de régression logistique
Enseignante : S. Toumi

17 décembre 2024

Objectif

L'objectif de ce TP est d'implémenter un test de régression logistique à partir d'un échantillon expérimental choisi arbitrairement par l'étudiant. L'étudiant devra :

- Choisir un échantillon expérimental parmi les données disponibles sur le web .
- Implémenter le calcul des coefficients β_0 et β_1 en utilisant un algorithme d'optimisation (descente de gradient ou méthode de Newton). le modèle

Consignes

1. Choix de l'échantillon expérimental

L'étudiant doit :

- Sélectionner un jeu de données contenant une y varibale binaire avec au moins une variable explicative x ;
- Donner la sources des données (exemple de sources) .
  https ://www.kaggle.com/datasets
  https ://datahub.io/collections
  https ://archive.ics.uci.edu/ml/index.php
  https ://www.openml.org/
  https ://data.world/
  https ://www.insee.fr/
  https ://www.quandl.com/
  https ://www.data.gouv.fr
  https ://data.gov/
  https ://data.worldbank.org

- Décrire brièvement l'échantillon choisi (source des données, signification des variables, etc.). (Vous vous limitez à une seule variable explicative de votre choix si les données comprennent plusieurs variables explicatives)

2. Implémentation de l'algorithme

Étapes à suivre :

1. **Modèle de régression logistique :** Le modèle de régression logistique est défini par :
$$P(y = 1 \mid x) = g(\beta_0 + \beta_1 x), \quad \text{où} \quad g(z) = \frac{e^z}{1 + e^z}$$
2. **Fonction de log-vraisemblance :** La fonction à maximiser pour des observations $(x_i, y_i)_{\{i=1,\dots N\}}$est :
$$\ell(\beta) \equiv \ln L(\beta) = \sum_{i=1}^{n} \{y_i \ln p(x_i) + (1 - y_i) \ln [1 - p(x_i)]\} .$$
3. **Méthodes d'optimisation :** Implémentez l'une des méthodes suivantes pour trouver les coefficients β_0 et β_1 :
   — *Descente de gradient :* Mettez à jour les coefficients en utilisant :
   $$\beta_j \leftarrow \beta_j - \eta \frac{\partial \ell}{\partial \beta_j}, \quad j = 0, 1$$
   où η est le taux d'apprentissage.
   — *Méthode de Newton :* Utilisez l'approximation quadratique de la fonction de log-vraisemblance pour mettre à jour :
   $$\beta \leftarrow \beta - H\ell^{-1} \nabla \ell,$$
   où ∇ est le gradient et H est la matrice hessienne.
4. **Évaluation de l'algorithme d'optimisation**
   Choisir nombre d'observation N que vous préciserez, où vous utlisez N/2 pour l'estimation des paramètres et le reste des N/2 échantillons pour la phase du test .
   Quel est l'algorithme qui vous donne un résultat meilleur ? Pourquoi (justification théorique).

   Comparez les coefficients 0 et1 otbenus à un résultat donné par un toolbox que vous préciserez du logiciel sur lequel vous allez tavailler. ?
5. **Évaluation du modèle :**
   Une fois les coefficients β calculés , évaluez le modèle en utilisant des métriques comme l'accuracy, la précision, le rappel, ou l'AUC si pertinent. justifier votre choix

A rendre

L'étudiant doit soumettre :
— Le code Python utilisé pour l'implémentation via une présentation avec Jupyther Notebook ;
— les réponses qux question précédentes dans l'ordre sous forme de commentaires (Markdown) :

— Pour les commentaires, vous pouvez utiliser Latex pour écrire des formules mathématiques lors de la déscription détaillée des algorithmes (gradient...hessienne) :

Exemple de commentaire en Markdown avec une formule en LaTeX :

$$f(x) = \int_{-\infty}^{\infty} e^{-x^2} dx$$

Ressources suggérées

— Documentation Python (modules : `numpy`, `pandas`, `matplotlib`, etc.) ;
— Cours