Examen (CC) : Statistique pour "Big Data"

Jeudi 07 février 2019 - Durée 3h

Nous allons utiliser pour ce contrôle un jeu de données contenant des temps d'attente concernant des requêtes effectuées sur internet. On mesure pour chaque requête le temps d'attente, noté `temps` ainsi que deux variables quantitatives, nommées `X1` et `X2`.

Nous allons élaborer un modèle de régression (linéaire généralisée) qui permettra de modéliser ce temps d'attente.

Les programmes et commentaires de la partie II devront être envoyés par email à mon adresse à la fin de l'examen sous format pdf ou texte. On pourra ajouter des graphiques pour illustrer ses choix.

Partie I : un peu de théorie

1. On considère une loi exponentielle de paramètre $\lambda > 0$ (on rappelle que la densité de cette loi est $f(y, \lambda) = \lambda \exp(-\lambda y)$, si $y \geq 0$ et $f(y, \lambda) = 0$ si $y < 0$).

Montrer que l'estimateur du maximum de vraisemblance du paramètre λ obtenu à partir d'un échantillon $Y_1, \ldots, Y_n$ issu d'une loi exponentielle de paramètre λ est

$$\hat{\lambda}_n = \frac{1}{\overline{Y}_n}.$$

2. On suppose que les observations sont effectuées de manière séquentielle. Montrer que l'algorithme suivant est un algorithme de descente de gradient stochastique permettant d'estimer de manière séquentielle λ (par approximation séquentielle du minimum de l'opposé de la log-vraisemblance). Pour $n \geq 1$ et $\hat{\lambda}_1 = Y_1$, l'algorithme est défini par

$$\hat{\lambda}_{n+1} = \hat{\lambda}_n - \gamma_n \left(Y_{n+1} - \frac{1}{\hat{\lambda}_n} \right)$$

avec une suite de pas d'apprentissage γ_n qui vérifie $\sum_{n \geq 1} \gamma_n^2 < \infty$ et $\sum_{n \geq 1} \gamma_n$ diverge.

3. Proposer un meilleur algorithme séquentiel d'estimation de λ (en remarquant que l'estimateur obtenu dans la question 1 dépend simplement de $\overline{Y}_n$ et que $\overline{Y}_n$ peut être mis à jour très simplement).

4. On suppose maintenant qu'on dispose de p variables explicatives $X_1, \ldots, X_p$ mesurées aussi de manière séquentielle, en même temps que Y. On suppose par ailleurs que la loi conditionnelle de Y sachant $X_1, \ldots, X_p$ est une loi exponentielle de paramètre λ défini par

$$\ln \lambda \ = \ \beta_0 + \sum_{j=1}^{p} \beta_j X_j$$

Il s'agit donc du cadre du modèle linéaire généralisé. L'objectif est maintenant d'estimer de manière séquentielle les cœfficients de la régression (linéaire généralisée) $\beta_0, \ldots, \beta_p$.

On observe $(Y_n, X_{n1}, \ldots, X_{np}), n \geq 1$. Expliquez pourquoi l'algorithme suivant est un algorithme de gradient stochastique d'estimation de $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{p+1}$. Pour $n \geq 1$,

$$\widehat{\boldsymbol{\beta}}_{n+1} = \widehat{\boldsymbol{\beta}}_n - \gamma_n \left(\exp\left(\widehat{\boldsymbol{\beta}}_n' \widetilde{\mathbf{X}}_{n+1} \right) Y_{n+1} - 1 \right) \widetilde{\mathbf{X}}_{n+1}$$

avec $\widehat{\boldsymbol{\beta}}_1 = (Y_1, 0, \ldots, 0) \in \mathbb{R}^{p+1}$ et $\widetilde{\mathbf{X}}_n = (1, X_{n1}, \ldots, X_{np}) \in \mathbb{R}^{p+1}$.

Partie II : programmation des algorithmes avec le langage $\mathbb{R}$

Télécharger le fichier situé à l'adresse ci dessous et décompressez le :
`http://cardot.perso.math.cnrs.fr/MIGS/temps.csv.zip`

Importer les données dans $\mathbb{R}$ en tapant les commandes suivantes :

```
### Chargement des données
library(readr)
temps_data <- read_csv("temps.csv")
View(temps)
```

1. Tracer l'histogramme de la variable `temps` et argumenter le choix de la loi exponentielle de la question 1 (partie I) pour modéliser le temps d'attente.

2. Programmer en $\mathbb{R}$ (ou bien avec `Rcpp`) l'algorithme décrit dans la question I.2 pour l'estimation du paramètre λ associé à la distribution de la variable `temps`.
Calibrer les pas d'apprentissage γ_n de façon à obtenir une bonne estimation.
Indication : on pourra paramétrer γ_n *comme dans le cours :* $\gamma_n = C/(\phi + n^\alpha)$ *et considérer* $C \in \{0.001, 0.01, 0.05, 0.1, 1, 2\}$, $\alpha = 2/3$ *et* $\phi = 0$.

Le procédé de moyennisation permet-il d'améliorer la qualité des estimations ?
Comparer avec l'estimateur du maximum de vraisemblance de la question I.1.

3. On souhaite tenir compte maintenant des variables `X1` et `X2` pour modéliser le temps d'attente. Programmer en $\mathbb{R}$ (ou bien avec `Rcpp`) l'algorithme décrit dans la question I.3, en introduisant éventuellement une étape de moyennisation.

Pour calibrer les pas d'apprentissage γ_n, on pourra calculer la valeur de la log-vraisemblance ou une erreur de prédiction du temps d'attente sur les 1000 dernières observations de la table `tempc`.