Projet sur le cours de statistiques bayésiennes

Abdellah MOADINE Walid IBNOUCHEIKH

6 janvier 2022

1 Introduction générale de la régression linéaire dans le paradigme bayésien :

Récapitulons brièvement la régression linéaire fréquentiste et bayésienne. La vue fréquentiste de la régression linéaire suppose que les données sont générées à partir du modèle suivant :

$$Y = B^t X + \epsilon$$

Avec la méthode des moindres carrés ordinaires (OLS), les paramètres du modèle, , sont calculés en trouvant les paramètres qui minimisent la somme des erreurs quadratiques sur les données d'apprentissage. La sortie de l'OLS est constituée d'estimations ponctuelles pour les ≪ meilleurs ≫ paramètres de modèle compte tenu des données d'apprentissage.

En revanche, la régression linéaire bayésienne suppose que les réponses sont échantillonnées à partir d'une distribution de probabilité telle que la distribution normale (gaussienne) :

$$Y \sim \mathcal{N}(B^t X,\, \sigma^2)$$

Dans les modèles bayésiens, non seulement la réponse est supposée être échantillonnée à partir d'une distribution, mais les paramètres le sont également. L'objectif est de déterminer la distribution de probabilité postérieure pour les paramètres du modèle étant donné X et y :

$$P(\beta|y, X) = \frac{P(y, X|\beta)P(\beta)}{P(y, X)}$$

Si nous avons une certaine connaissance du domaine, nous pouvons l'utiliser pour attribuer des priors pour les paramètres du modèle, ou nous pouvons utiliser des priors non informatifs : des distributions avec de grands écarts types qui ne supposent rien sur la variable. L'utilisation d'un a priori non informatif signifie que nous ≪laissons parler les données≫. Un choix préalable courant consiste à utiliser une distribution normale pour β.

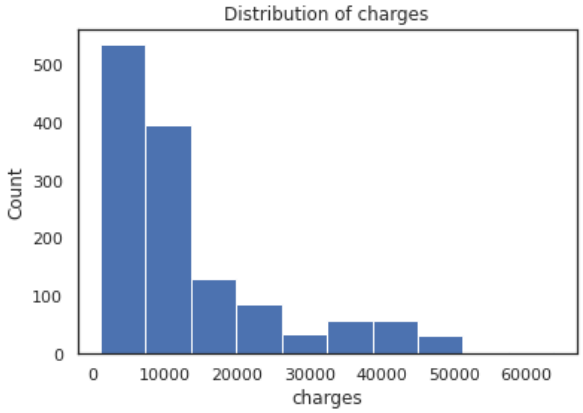
En pratique, le calcul de la distribution a posteriori exacte est difficile à calculer pour les valeurs continues et nous nous tournons donc vers des méthodes d'échantillonnage telles que Markov Chain Monte Carlo (MCMC) . Monte Carlo fait référence à la technique générale de tirage d'échantillons aléatoires, et la chaîne de Markov signifie que le prochain échantillon t+1 tiré est basé uniquement sur la valeur de l'échantillon précédent t. Le concept est qu'au fur et à mesure que nous tirons plus d'échantillons, l'approximation de la postérieure convergera finalement vers la vraie distribution postérieure pour les paramètres du modèle.

Le résultat final de la modélisation linéaire bayésienne n'est pas une estimation unique des paramètres du modèle, mais une distribution que nous pouvons utiliser pour faire des inférences sur de nouvelles observations. Cette distribution nous permet de démontrer notre incertitude dans le modèle et est l'un des avantages des méthodes de modélisation bayésienne. À mesure que le nombre de points de données augmente, l'incertitude devrait diminuer, montrant un niveau de certitude plus élevé dans nos estimations.

2 Analyse Exploratoire des données

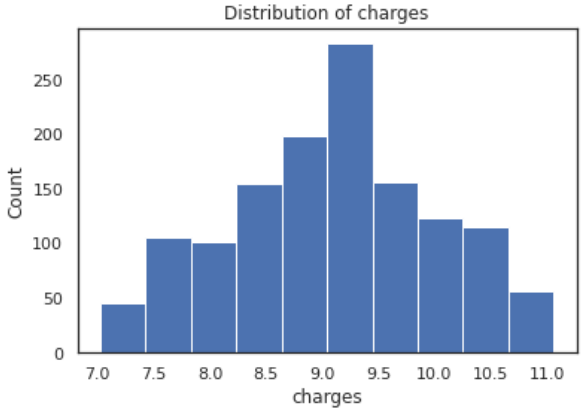
Notre jeu de données d'assurance contient 1338 observations avec 7 variables. Chaque ligne correspond à un patient, chaque colonne contenant une caractéristique différente. La colonne des frais "charges" est notre variable cible (Y), ce qui en fait une tâche de régression supervisée. Il est supervisé car nous avons un ensemble de données d'entraînement avec des cibles connues et, pendant l'entraînement, nous voulons que notre modèle apprenne à prédire les charges (Y) à partir des autres variables. Nous traiterons la variable des charges comme continues, ce qui en fait un problème de régression. Notre principale variable d'intérêt : "charges", alors examinons la distribution pour vérifier tout d'abords l'asymétrie :

FIGURE 1 – histogramme de la variable : "charges"



Le graphe montre qu'il y'a une asymétrie positive, nous résolvons pour le moment ce problème avec la normalisation.

FIGURE 2 – histogramme de la variable : $log(charges)$



Distribution des charges en fonction de la région et du sexe.

FIGURE 3 – Distribution des charges en fonction de la région

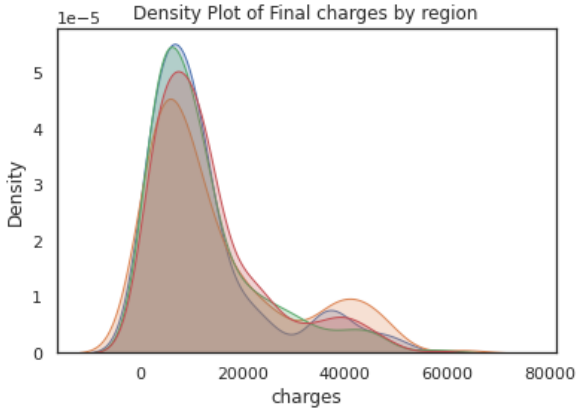
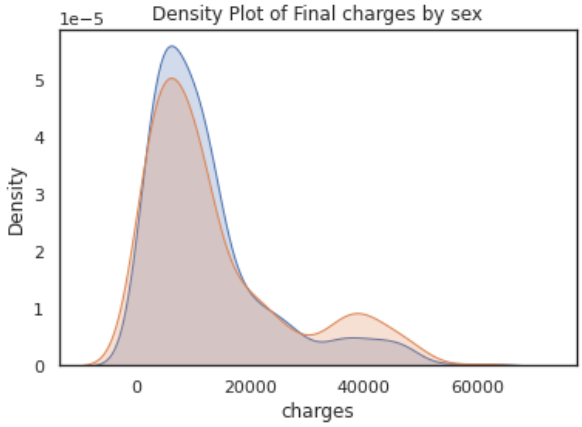


FIGURE 4 – Distribution des charges en fonction du sexe



la localisation (variable : region) et le sexe de l'individu ne semblent pas avoir un impact substantiel sur les charges médicaux.

selection des variables

Comme nous l'avons vu sur les graphiques, nous ne nous attendons pas à ce que chaque variable soit liée à la variable cible : charges, nous devons donc effectuer une sélection de variables pour choisir uniquement les variables « pertinentes ». Cela dépend du problème, mais comme nous allons faire de la modélisation linéaire dans ce projet, nous pouvons utiliser le coefficient de corrélation pour déterminer les variables les plus utiles pour prédire une note. Pour sélectionner un nombre limité de variables, on peut trouver celles qui ont la plus grande corrélation (soit négative soit positive) avec la variables "charges".

Nous avons codé une fonction qui nous a permet de :

1. Contrôler le nombre de variables corrélées qu'on voudra retenir avec la variable cible.
2. Produire un nouveau jeu de données visant la modélisation.

FIGURE 5 – titre

```
#Cette fonction nous permet de:
#*** controler le nombre de variables corrélées qu'on voudra retenir avec la variable cible.
#*** faire un nouveau jeu de données visant la modèlisation.
def format_data(data):
    labels = data['charges']
    # recodage
    data = pd.get_dummies(data)
    # correlations avec la variable charges par ordre décroissant
    plus_correlees = data.corr().abs()['charges'].sort_values(ascending=False)
    plus_correlees = plus_correlees[:4]
    data = data.loc[:, plus_correlees.index]
    X_train, X_test, y_train, y_test = train_test_split(data, labels,
                                                        test_size = 0.25,
                                                        random_state=42)
    return X_train, X_test, y_train, y_test
```

```
[75] X_train, X_test, y_train, y_test = format_data(data)
     X_train.head()
```

	charges	smoker_no	smoker_yes	age
693	2352.96845	1	0	24
1297	4340.44090	1	0	28
634	9391.34600	1	0	51
1022	42211.13820	0	1	47
178	8823.27900	1	0	46

3 Régression linéaire simple dans le paradigme bayésien et algorithme de gibbs

Le modèle bayésien part du même modèle que l'approche fréquentiste classique :

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, ..., n$$

Hypothèse du modèle

1. les erreurs ϵ_i sont iid suivant une loi normale centrée.

Sous cette hypothèse :

$$Y_i|x_i, \beta_0, \beta_1, \sigma^2 \sim \mathcal{N}(\beta_0 + \beta_1 x_i,\ \sigma^2)$$

donc, l'expression de la vraisemblance s'écrit :

$$P(Y_i|x_i, \beta_0, \beta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(Y_i - (\beta_0 + \beta_1 x_i)^2)}{2\sigma^2}$$

Nous considérons d'abord le cas du prior non informatif, nous supposons la distribution à priori de $\beta_0, \beta_1, \sigma^2$ s'écrit :

$$P(\beta_0, \beta_1, \sigma^2) \propto \frac{1}{\sigma^2}$$

ainsi, on obtient :

$$\beta_1|\sigma^2, x \sim \mathcal{N}(\hat{\beta},\ \frac{\sigma^2}{S_{xx}})$$

$$\beta_0|\sigma^2, x \sim \mathcal{N}(\hat{\beta}_0,\ \sigma^2(\frac{1}{n} + \frac{x^2}{S_{xx}}))$$

Algorithme de Gibbs

1. Partons d'une valeur $\beta_1^{(i)}$
2. itération $k \geq 1$:
   (a) on simule $\beta_{(0)}^{(i+1)} \sim p(\beta_0|\beta_1^{(i)}, x)$
   (b) on simule $\beta_{(1)}^{(i+1)} \sim p(\beta_1|\beta_0^{(i)}, x)$

3.1 Cas pratique

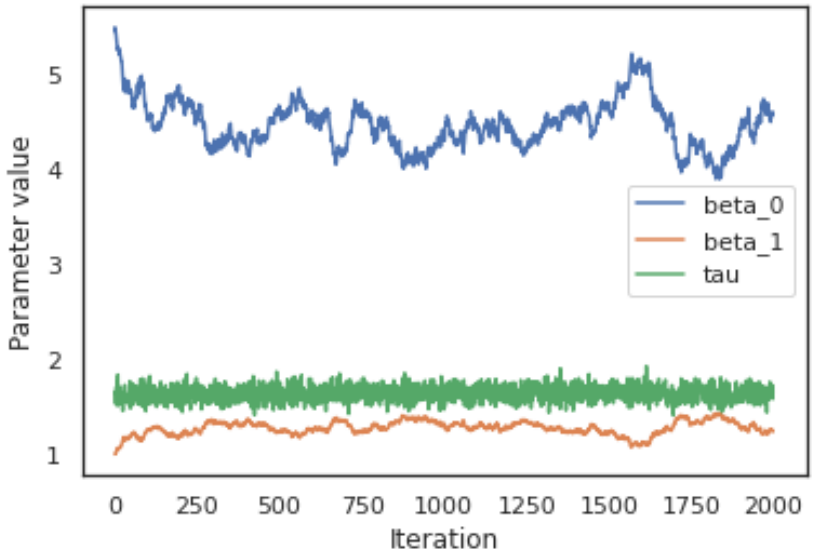
$$charges_i = \beta_0 + \beta_1 age_i + \epsilon_i, i = 1, ..., n$$

FIGURE 6 – titre

```
#tau=1/sigma^2   paramétre de précision
def gibbs(y, x, iters, init, hypers):
    assert len(y) == len(x)
    beta_0 = init["beta_0"]
    beta_1 = init["beta_1"]
    tau = init["tau"]

    trace = np.zeros((iters, 3))

    for it in range(iters):
        beta_0 = sample_beta_0(y, x, beta_1, tau, hypers["mu_0"], hypers["tau_0"])
        beta_1 = sample_beta_1(y, x, beta_0, tau, hypers["mu_1"], hypers["tau_1"])
        tau = sample_tau(y, x, beta_0, beta_1, hypers["alpha"], hypers["beta"])
        trace[it,:] = np.array((beta_0, beta_1, tau))

    trace = pd.DataFrame(trace)
    trace.columns = ['beta_0', 'beta_1', 'tau']

    return trace
```

```
[95]
y=np.log(y_train)
x=np.log(X_train.age)
iters=2000
trace = gibbs(y, x, iters, init, hypers)
```

FIGURE 7 – the progression of the samples drawn in the trace for the variable on the right



4 Régression linéaire multiple dans le paradigme bayésien à l'aide du package "pymc3"

FIGURE 8 – Modèle de régression linéaire multiple

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	10686.830	194.635	10306.171	11034.879	7.234	5.118	726.0	706.0	1.00
smoker_no	-14207.974	166.059	-14491.624	-13856.854	5.718	4.044	843.0	980.0	1.01
age	267.041	2.367	262.533	271.369	0.074	0.052	1030.0	912.0	1.00
lam	0.000	0.000	0.000	0.000	0.000	0.000	1419.0	1180.0	1.00

Interpretation

A propos de la figure 8 :

1. La variable *smokerno* est négativement liée aux charges.
2. La variable "*age*" est positivement liée aux charges.

Le côté gauche de la Figure 9 est la marginale à posteriori : les valeurs de la variable sont sur l'axe des x avec la probabilité de la variable (telle que déterminée par échantillonnage) sur l'axe des y. nous avons effectué deux chaînes de Markov Chain Monte Carlo. Du côté gauche, nous pouvons voir qu'il existe une plage de valeurs pour chaque poids. Le côté droit montre les différentes valeurs d'échantillon tirées au fur et à mesure que le processus d'échantillonnage s'exécute.

FIGURE 9 –

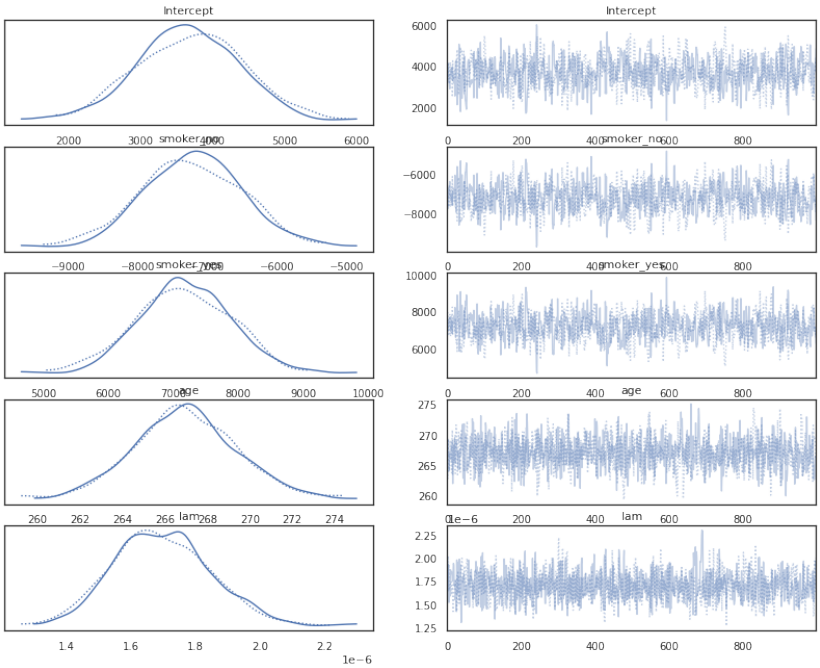


FIGURE 10 – Distribution de tous les parametres du modèle.