Université Lumière Lyon 2
Institut de la communication

Librairie MIMOSA

Mixed Input Multinomial Optimization for Statistical Analysis

Par
Linh Nhi Le Dinh
Antoine Oruezabala
Béranger Thomas

Supervisé par :
Ricco Rakotomalala

Rapport présenté dans le cadre du
Master 2 SISE - 2024/2025

Table des matières

Chapitre 1

Éléments introductifs

1.1 La régression logistique

Quelle soit binaire (la variable cible possède deux modalités) ou multinomiale (la variable cible possède plus de deux modalités), le principe de la régression logistique est de maximiser la vraisemblance, ou pour le dire autrement, de minimiser la déviance.

Quel outil pour mesurer ce paramètre ? La log-vraisemblance.

1.1.1 Notations

Soit une population Ω de n individus, définie par J variables explicatives notées $\{X_1, \ldots, X_J\}$, et une variable cible Y possédant K valeurs :

Ω	Cible	X_1	$\ldots$	X_J
1	Y_1			
$\ldots$	$\ldots$			
ω	Y_ω	$X_1(\omega)$	$\ldots$	$X_J(\omega)$
$\ldots$	$\ldots$			
n	Y_K			

Dans le cas binaire, la probabilité d'un individu ω d'être positif à priori se note p par commodité, pour simplifier $p(\omega)$, lui-même une notation simplifiée de $P[Y(\omega) = +]$.

Toujours dans le cas binaire, la probabilité d'un individu ω d'être positif à posteriori, c'est-à-dire la probabilité que l'on modélisera en apprentissage supervisée, se note π par commodité, pour simplifier $\pi(\omega)$, lui-même une notation simplifiée de $P[Y(\omega) = +/X(\omega)]$.

La fonction LOGIT pour cet individu ω est :

$$\text{LOGIT}(\omega) = \ln\left[\frac{\pi(\omega)}{1-\pi(\omega)}\right] = a_0 + a_1 X_1(\omega) + \cdots + a_J X_J(\omega) \tag{1.1}$$

Soit en écriture matricielle :

$$\ln\left[\frac{\pi(\omega)}{1-\pi(\omega)}\right] = X(\omega) \times a \tag{1.2}$$

Avec $a_0, \ldots, a_J$ les paramètres que l'on souhaite estimer.

1.1.2 Algorithmes

Un algorithme utilisé pour optimiser la log-vraisemblance est celui de Newton-Raphson. Il s'agit d'une approche itérative, qui utilise la dérivée de la fonction considérée pour approcher une solution.

Il a toutefois pour défaut d'utiliser la matrice hessienne, qui peut être coûteuse en mémoire et temps de calcul.

Nous expliquerons et utiliserons un autre algorithme ici, la descente de gradient.

Il s'agit aussi d'un algorithme itératif, dont l'objectif est de trouver le minimum d'une fonction $f(x)$ en partant d'un point arbitraire x_0. On se « déplace » pour cela dans la direction opposée au gradient, c'est-à-dire que l'on « descend » le long de la pente.

Mathématiquement, on part d'une valeur arbitraire x_0, puis pour trouver x_1 on utilise la formule suivante :

$$x_{n+1} = x_n - \alpha \nabla f(x_n) \tag{1.3}$$

Avec :
- x_n la position actuelle,
- α le taux d'apprentissage,
- $\nabla f(x_n)$ le gradient de la fonction au point x_n

Les étapes sont donc les suivantes :
- choisir un point x_0 arbitraire pour commencer les calculs,
- calculer le gradient à ce point,
- mettre à jour la position en se déplaçant dans la direction opposée,
- vérifier que l'on a atteint un critère d'arrêt :
  - nombre d'itérations maximum atteint,
  - et/ou précision souhaitée (différence entre deux itérations) atteinte,

1.1.3 Champs d'application

La régression logistique n'émet pas d'hypothèse directement sur les distributions des probabilités $P(X/Y = +)$ et $P(X/Y = -)$, mais uniquement sur leur rapport.

Elle permet donc, en théorie, une application plus large, et est nommé semi-paramétrique, pour la différencier des modèles qui supposent une loi donnée sur la distribution des probabilités.

1.2 Contexte

Ce document présente la régression logistique multinomiale, par le biais d'une descente de gradient, pour des données mixtes.

Chapitre 2

Étude du problème

2.1 Colinéarité et codage disjonctif

2.1.1 Colinéarité

Les problèmes de colinéarité ne se produisent que lorsque la distance euclidienne intervient dans l'algorithme de machine learning.

En régression logistique il est indispensable d'éviter ce souci.

Cela suppose de faire un codage disjonctif en enlevant une modalité.

2.1.2 Cas binaire

Dans ce cas, les coefficients qui seront indiqués se lisent en opposition à la modalité de référence, c'est-à-dire celle qui a été retirée.

Cela implique de devoir retrouver cette modalité après coup.

Chapitre 3

Librairie développée

La librairie développé porte le doux nom de MIMOSA (Mixed Input Multinomial Optimization for Statistical Analysis).

3.1 Pré-traitements

3.1.1 Séparation en jeux train et test

Enjeux : conserver la répartition des modalités de la variable cible (vérifier la cohérence des modalités).

Cela évite un one hot encoding qui ne créerait pas assez de colonnes (cas du jeu de test avec une modalité en moins par exemple).

Chapitre 4

Lexique

RÉGRESSION LOGISTIQUE ■ La régression logistique vise à prédire la probabilité qu'un événement binaire se produise (oui/non, 1/0) en fonction de variables explicatives. Elle utilise une fonction logistique (ou sigmoïde) pour transformer la combinaison linéaire des variables indépendantes en une probabilité comprise entre 0 et 1.

RÉGRESSION LOGISTIQUE MULTINOMIALE ■ Extension de la régression logistique permettant de modéliser une variable qualitative à plus de deux modalités.

Cette méthode permet d'estimer la probabilité d'appartenance à chacune des catégories en fonction de variables explicatives quantitatives et/ou qualitatives. Contrairement à la régression logistique binaire qui modélise une seule probabilité p, la régression logistique multinomiale estime simultanément les probabilités d'appartenance à toutes les catégories $(p_1, p_2, ..., p_K)$, avec la contrainte que leur somme soit égale à 1.

Le modèle utilise une transformation logistique généralisée qui garantit que les probabilités estimées restent comprises entre 0 et 1. Une catégorie est choisie comme référence, et le modèle estime les logarithmes des rapports de probabilités (log-odds) entre chaque catégorie et cette référence.

Limites : - Sensible à la multicolinéarité des variables explicatives - Nécessite un échantillon suffisamment grand, particulièrement quand le nombre de catégories augmente - Suppose l'indépendance des alternatives non pertinentes (IIA)

DESCENTE DE GRADIENT ■ La descente de gradient est un algorithme doptimisation couramment utilisé pour entraîner les modèles de machine learning et les réseaux neuronaux. Ce type dalgorithme entraîne les modèles de machine learning par réduction des erreurs entre les résultats prédits et les résultats réels. ■ Le point de départ n'est qu'un point arbitraire qui nous permet d'évaluer les performances. À partir de ce point de départ, nous allons trouver la dérivée (ou la pente) et, à partir de là, nous pourrons utiliser une ligne tangente pour observer l'inclinaison de la pente. La pente renseigne sur les mises à jour des paramètres, c'est-à-dire les poids et les biais. La pente au point de départ est plus forte, mais au fur et à mesure que de

nouveaux paramètres sont générés, elle devrait progressivement diminuer jusqu'à atteindre le point le plus bas de la courbe, dénommé point de convergence. ■ Comme pour trouver la ligne de meilleur ajustement dans la régression linéaire, l'objectif de la descente de gradient est de minimiser la fonction de coût, ou l'erreur entre y prédit et y réel. Pour ce faire, deux points de données sont nécessaires : une orientation et un taux d'apprentissage. Ces facteurs déterminent les calculs de dérivée partielle des itérations futures, ce qui lui permet d'atteindre progressivement le minimum local ou global (c'est-à-dire le point de convergence).. La fonction de perte dans la descente de gradient agit spécifiquement comme un baromètre, évaluant sa précision à chaque itération des mises à jour de paramètres. Jusqu'à ce que la fonction soit proche ou égale à zéro, le modèle continue à ajuster ses paramètres pour obtenir l'erreur la plus faible possible. ■ Il existe trois types d'algorithmes d'apprentissage par descente de gradient : la descente de gradient par lots, la descente de gradient stochastique et la descente de gradient par mini-lots.