Séance 3: Régression logistique et réseaux de neurones

Sébastien Gadat

Laboratoire de Statistique et Probabilités
UMR 5583 CNRS-UPS

www.lsp.ups-tlse.fr/gadat

Troisième partie III

Régression logistique

Introduction - Type de données

- On cherche à expliquer une variable Y à deux modalités notées 0 ou 1 (ou plus).
- Les variables explicatives X sont *a priori* quantitatives. Il y en a p : $X \in \mathbb{R}^p$.
- On dispose d'un échantillon d'apprentissage $(X_1, Y_1), \ldots (X_n, Y_n)$.
- La régression linéaire standard n'est pas très clairement applicable à ce cas car : $\forall \beta \in \mathbb{R}^p \qquad X'\beta \in \mathbb{R}$.
- Idée de la méthode : On cherche plutôt à expliquer les probabilités

$$\pi_X = P(Y = 1|X) \qquad \text{ou} \qquad 1 - \pi_X = P(Y = 0|X)$$

- Dans le cas multiclasse, si j est une des modalités de Y, on essaye de prédire

$$\pi_X(j) = P(Y = j|X)$$

Odds et Ratio sur une variable

Quelques notions relatives à la liaison entre variables qualitatives. Elles sont utilisées couramment dans l'interprétation des modèles de régression logistique.

- Définition élémentaire : Soit Y une variable qualitative à J modalités, on appelle l'odds de j sur k la quantité :

$$\Omega_{j,k} = \frac{P(Y=j)}{P(Y=k)} = \frac{\pi_j}{\pi_k}$$

- Estimation de $\Omega_{j,k}$. Si on possède n individus dont on connaît Y, on estime

$$\hat{\Omega_{j,k}} = \frac{n_j}{n_k}$$

- Dans le cas où Y est une variable de Bernouilli :

$$\Omega_{1,2} = \frac{\pi}{1-\pi}$$

Odds et Ratio sur plusieurs variables

- Y_1, Y_2 deux variables binaires qualitatives croisées par une table de contingence

$$\begin{pmatrix} \pi_{1,1} & \pi_{1,2} \\ \pi_{2,1} & \pi_{2,2} \end{pmatrix}$$

- On définit les odds conditionnels par

$$\Omega_1 = \frac{\pi_{1,1}}{\pi_{1,2}} \qquad \Omega_2 = \frac{\pi_{2,1}}{\pi_{2,2}}$$

c'est le quotient des probabilités sur Y_2 sachant que $Y_1 = 1$

- L'odds ratio est donné par

$$\Theta = \frac{\Omega_1}{\Omega_2}$$

estimé empiriquement par

$$\hat{\Theta} = \frac{n_{1,1} n_{2,2}}{n_{2,1} n_{1,2}}$$

Interprétation de l'Odds et Ratio

À l'entrée de l'IUP SID M2, la probabilité d'être reçu pour un garçon est de 4 sur 10 alors qu'il est de 7 sur 10 pour une fille. Calculer l'odds ratio et interpréter.

De même que précédemment, on peut définir l'odds ratio croisant deux variables à J et K modalités.

Modèle

- On cherche un modèle linéaire expliquant au mieux, pour chaque individu la classe Y.
- $X\beta$ ne prend pas des valeurs qualitatives, encore moins 0 ou 1 !
- Le modèle consiste à d'estimer en fonction de X les quantités

$$\pi_X = P(Y = 1|X)$$

- Autre problème : a priori $X\beta$ parcourt $\mathbb{R}$ alors que $\pi_X \in [0; 1]$
- On cherche une bijection de $[0; 1]$ dans $\mathbb{R}$ telle que

$$g(\pi_X) \simeq X'\beta$$

pour le β choisi dans notre modèle de régression.

Modèle

- Pour trouver une bonne fonction g, on remarque alors que

$$\pi_i \simeq \phi(x_i'\beta) \qquad \text{où} \qquad \phi^{-1} = g$$

- Cela impose que ϕ soit une à valeurs entre 0 et 1. Il est assez naturel de choisir

$$\phi(t) = \frac{e^t}{1+e^t} \Longrightarrow g(\pi) = ln\left(\frac{\pi}{1-\pi}\right)$$

- La fonction g est appelée la transformée *logit* de π.
- Une telle fonction g est bien une bijection de $[0;1]$ dans $\mathbb{R}$.
- En réalité, on va chercher β tel que

$$\frac{\pi_X}{1-\pi_X} \simeq e^{X'\beta}$$

Le log de l'odds sera alors expliqué linéairement en les p variables décrivant X.

Interprétation des coefficients

- La régression linéaire s'exprime généralement sous la forme :

$$\frac{\pi_{x_1,\dots x_p}}{1-\pi_{x_1,\dots x_p}} \simeq e^{\alpha_0+\alpha_1 x_1+\cdots+\alpha_p x_p}$$

- Un incrément de 1 dans x_j entraîne un incrément de e^{α_i} dans l'odds ratio.
- Si $\alpha_i \simeq 0$, l'odds ratio ne dépend pas de x_i.
- Si $\alpha_i > 0$, un incrément de x_i augmente de e_i^{α} le risque (ou la chance) supplémentaire que Y vaille 1 :

$$\frac{\frac{\pi_{x_1,\dots x_i+1\dots x_p}}{1-\pi_{x_1,\dots x_i+1\dots x_p}}}{\frac{\pi_{x_1,\dots x_i\dots x_p}}{1-\pi_{x_1,\dots x_i\dots x_p}}} = e^{\alpha_i}$$

Significativité des variables

- On teste l'effet de chaque variable en examinant l'hypothèse

$$H_0 : \beta_i = 0$$

- On rejette l'hypothèse dès que la p-value de H_0 est trop faible. Les logiciels scientifiques donnent en général ces p-values.
- En effet :

$$P(Observation|H_0) = p - value$$

  Si la p-value est trop faible, c'est que H_0 est peu vraisemblable et que le coefficient β_i est non nul.
- On peut également calculer des régions de confiance pour la régression logistique sur π_X par le biais de $g^{-1}(Region \quad de \quad confiance \quad classique)$.

Estimation des coefficients

- Les coefficients sont déterminés à l'aide des données d'apprentissage.
- Problème : A priori, on ne connaît pas π_X, quel que soit le X dans l'échantillon d'apprentissage.
- $x^1, \ldots x^p$, p variables explicatives. On stratifie en I positions différentes les individus.
- Exemple : si on stratifie en 2^k positions chaque variable x^j, I vaut alors

$$I = (2^k)^p = 2^{pk}$$

Plan d'expérience dans le cadre du modèle binomial

- Pour $i \in \{1 \dots I\}$, on effectue n_i mesures de la variable Y
- y_i désigne le nombre de réalisations $Y = 1$
- On suppose que π_i est la probabilité de succès de Y sachant que les $x^1, \dots x^p$ appartiennent au groupe i :

$$\pi_i = P(Y = 1 | X \in Groupe \quad i)$$

- On suppose que les groupes sont homogènes dans leurs réalisations Y : la probabilité pour Y d'être égal à 1 au sein d'un même groupe est indépendante de la valeur de X dans ce groupe.
- Proposition En effectuant n_i mesures dans le groupe i, et en supposant tous les échantillons indépendants, si Y_i désigne le nombre de valeurs $Y = 1$, alors

$$P(Y_i = y_i) = C_{n_i}^{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i}$$

Estimation des coefficients dans le cadre du modèle binômial

- Rappelons que $\pi_i \simeq \phi(x_i'\beta)$, on utilise alors un modèle de log-vraisemblance pour trouver les coefficients β.
- Proposition La log-vraisemblance s'écrit

$$L(\beta) = \prod_{i=1}^{I} C_{n_i}^{y_i} \phi(x_i'\beta)(1 - \phi(x_i'\beta))^{n_i - y_i}$$

- On maximise cette quantité en β pour obtenir l'estimation du modèle.
- Sélection de modèle : ceci peut se faire en utilisant les méthodes de pénalisation de la séance précédente (AIC-BIC).
- Une autre technique consiste à adopter une stratégie *backward* : construire un modèle puis des modèles de plus en plus « petits »en supprimant les variables possédant le moins d'effet significatif sur Y.

Quatrième partie IV

Réseaux de neurones

Présentation

Un réseau de neurones est l'association d'objets élémentaires : les neurones formels. C'est en général l'organisation de ces neurones, le nombre de neurones, et leurs types, qui distingue ces différents réseaux.

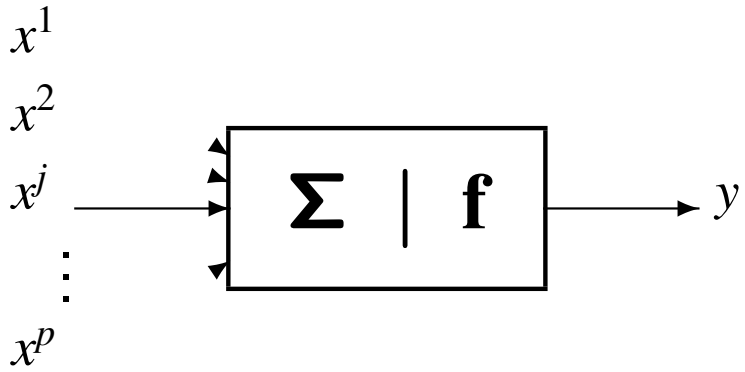

FIG.: Représentation d'un neurone formel.

Il est parfois repris le vocabulaire issu de la biologie : synapses, dendrites, axone et noyau.

Présentation

- Le neurone formel est un modèle qui se caractérise par un état interne $s \in \mathcal{S}$, des signaux d'entrée $x_1, \ldots, x_p$ et une fonction de transition d'état

$$s = h(x_1, \ldots, x_p) = f\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right).$$

- β_0 : biais du neurone
- Combinaison affine est déterminée par un *vecteur de poids* $[\beta_0, \ldots, \beta_p]$ associé à chaque neurone et dont les valeurs sont estimées dans la phase d'apprentissage ("mémoire" ou "connaissance répartie" du réseau).

Fonctions de transition

- $f(x) = x$ (neurone linéaire)
- $f(x) = 1/(1+e^x)$ (neurone sigmoïde)
- $f(x) = \mathbf{1}_{\mathbf{0};+\infty}$ (fonction de seuillage)
- $f(x) = 1$ avec la probabilité $1/(1+e^{-H(x)})$ (neurone stochastique)
- La plupart des problèmes d'apprentissage utilisent les deux premières fonctions de transition.
- La fonction représentant vraisemblablement le mieux la réalité est la fonction de seuillage. Mais elle n'est pas différentiable et donc peu adaptée pour les statisticiens.

Architecture du perceptron

- Réseau composé de couches successives
- Pas de connexion entre des réseaux de la même couche
- Couche d'entrée $\mathcal{C}_1$: un neurone par variable x^j
- Couche sortie : fournit la prédiction y
- Plusieurs couches cachées
- Chaque neurone de la couche cachée $\mathcal{C}_k$ est connecté en entrée à chaque neurone de la couche précédente $\mathcal{C}_{k-1}$.
- y est donc calculé par le biais de

$$y = \phi(x^1, \ldots x^p, \beta) \qquad \text{où} \qquad \beta = (\beta_{j,k,l})$$

Architecture du perceptron : un exemple à une couche cachée

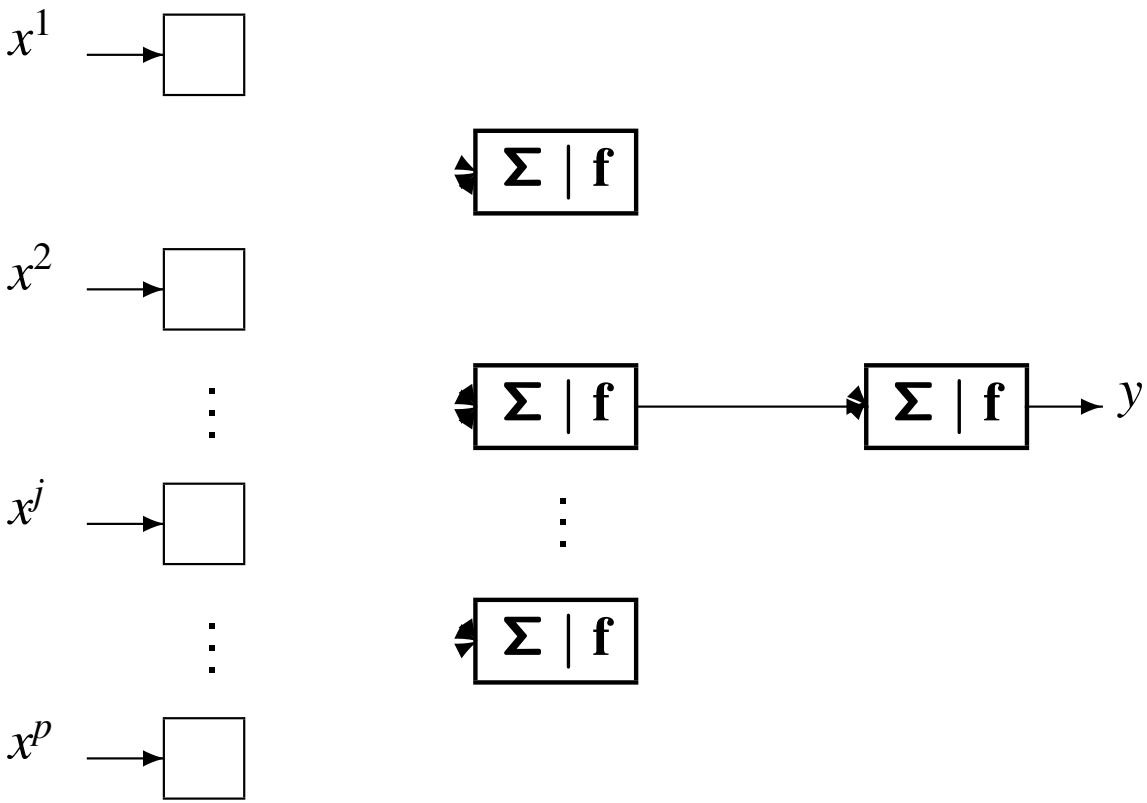

Fig 1 : Un exemple de perceptron avec une couche cachée et une couche de sortie.

Apprentissage du perceptron

- On dispose de $(x_i^1, \ldots x_i^p, y_i)$, n observations de variables explicatives $X^1, \ldots X^p$ et de Y, à prédire.
- On cherche $\hat{\beta}$ solution du problème :

$$\hat{\beta} = \arg\min_b Q(b) \qquad \text{où} \qquad Q(b) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \phi(x_i^1, \ldots x_i^p, b)\right]^2$$

- C'est un problème hautement non linéaire, le plus souvent, on adopte une méthode de descente de gradient.

Apprentissage du perceptron

- En tout point $\mathbf{b}$, le gradient de Q pointe dans la direction de l'erreur croissante.
- Il suffit donc de se déplacer en sens contraire, l'algorithme est itératif modifiant les poids de chaque neurone selon :

$$b_{jk\ell}(i) = b_{jk\ell}(i-1) + \Delta b_{jk\ell}(i)$$

- la correction $\Delta b_{jk\ell}(i)$ est proportionnelle au gradient et à l'erreur attribuée à l'entrée concernée $\varepsilon_{jk\ell}(i)$ et incorpore un terme d'"inertie" $\alpha b_{jk\ell}(i-1)$ permettant d'amortir les oscillations du système :

$$\Delta b_{jk\ell}(i) = -\tau \varepsilon_{jk\ell}(i) \frac{\partial Q}{\partial b_{jk\ell}} + \alpha b_{jk\ell}(i-1)$$

Utilisation du perceptron multicouches

- On fixe la variable de sortie et celles d'entrées.
- On définit un nombre de couches cachées (en général 1 ou 2).
- On définit le nombre maximum d'itérations pour la recherche de $\hat{\beta}$, l'erreur maximale tolérée et le pas.
- Pour limiter le sur-apprentissage : nombre de couches faible, terme de pénalisation inséré dans Q, ...
- Inconvénients : Temps de calcul important, aspect boîte noire, minimum local de Q
- Avantages : applicable aux situations non linéaires, pondération d'interactions possibles

Paramètres de réglages du réseau de neurone

- Nombre de couches (en général au maximum 2 ou 3)
- Nombre de neurones par couche 10 au maximum
- Taille du pas de l'algorithme de descente de gradient (à régler de façon sensible)
- Nombre d'itérations pour l'apprentissage des poids par descente de gradient
- Attention : parfois, l'erreur "remonte" lorsqu'on fait trop d'itérations
- Optimisation de ces paramètres par validation croisée