Apprentissage: cours 8
Régression logistique

Simon Lacoste-Julien

6 novembre 2015

Résumé
On voit notre premier algorithme d'état-de-l'art pour la classification : la régression logistique.

Rappel : maximum de vraisemblance, cas génératif et conditionnel

Modèle génératif :

$$\mathcal{R}(\theta) = -\mathbb{E}[\log(p_\theta(X,Y))] \qquad \widehat{\mathcal{R}}_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log(p_\theta(x_i,y_i))$$

Modèle conditionnel :

$$\mathcal{R}(\theta) = -\mathbb{E}[\log(p_\theta(Y|X))] \qquad \widehat{\mathcal{R}}_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log(p_\theta(y_i|x_i))$$

1 Régression logistique

1.1 Motivation à partir d'un modèle génératif

On considère le problème de la classification binaire, i.e. $\mathcal{X} = \mathbb{R}^p$ et $\mathcal{Y} = \{0,1\}$. Il s'agit de modéliser $\mathbb{P}(Y|X=x)$.

Par la loi de Bayes, nous avons :

$$\begin{aligned} p(Y=1|X=x) &= \frac{p(X=x|Y=1)p(Y=1)}{p(X=x|Y=1)p(Y=1)+p(X=x|Y=0)p(Y=0)} \\ &= \frac{1}{1+\frac{p(Y=0)}{p(Y=1)}\frac{p(X=x|Y=0)}{p(X=x|Y=1)}} \\ &:= \frac{1}{1+\exp(-f(x))} \end{aligned}$$

où $f(x) = \log(\frac{p(X=x|Y=1)}{p(X=x|Y=0)}) + \log(\frac{p(Y=1)}{p(Y=0)})$ est le 'log de rapport de chance' entre la classe 1 et la classe 0 ; cette fonction caractérise complètement la probabilité postérieure $p(Y=1|X=x)$.

On modélise donc la fonction $f(x)$, ce qui conduit au modèle

$$\mathbb{P}(Y=1|X=x) = \sigma(f(x)) \qquad \text{avec} \qquad \sigma(z) = \frac{1}{1+e^{-z}}.$$

La fonction σ appelée *fonction logistique* satisfait les propriétés :
— $\sigma(-z) = 1-\sigma(z)$
— $\frac{d\sigma}{dz}(z) = \sigma(z)(1-\sigma(z)) = \sigma(z)\sigma(-z)$

À noter que plusieurs modèles génératifs différents de $p(X=x|Y)$ pourrait donner suite à la même forme pour $f(x)$. Pour la régression logistique, on suppose que $f(x)$ est une fonction linéaire dans son paramètre $\mathbf{w}$, on considère donc $f_{\mathbf{w}} : \mathbf{x} \mapsto \mathbf{w}^\top\mathbf{x}$ (pour simplifier la notation, nous absorbons une ordonnée à l'origine potentielle $b = f(\mathbf{0})$ en augmentant $\mathbf{x}$ avec la constante 1 si nécessaire : $[\mathbf{x}^\top 1]$).

Famille exponentielle

Par exemple, considérons la *famille exponentielle*

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}^\top \mathbf{T}(\mathbf{x}) - A(\boldsymbol{\eta})).$$

Elle est définit par un choix de mesure de base $h(\mathbf{x})d\mu(\mathbf{x})$ et par la 'statistique suffisante' $\mathbf{T}(\mathbf{x})$ ($A(\cdot)$ est la 'fonction de log-partition' ; et $\boldsymbol{\eta}$ est le 'paramètre naturel'). La plupart des distributions standards peuvent se mettre sous cette forme paramétrique (gaussienne, binomiale, gamma, Poisson, Dirichlet, etc. – mais *pas* la distribution uniforme sur $[0, \theta]$ par contre). Supposons que les distributions conditionnelles pour chaque classe y ont la forme : $p(X = \mathbf{x}|Y = y) = p(\mathbf{x}|\boldsymbol{\eta}_y)$; et dénotons $\pi = p(Y = 1)$ l'à-priori sur la classe 1. Nous avons alors

$$\begin{aligned} f(\mathbf{x}) &= \log(\frac{p(X = x|Y = 1)}{p(X = x|Y = 0)}) + \log(\frac{p(Y = 1)}{p(Y = 0)} \\ &= (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_0)^\top \mathbf{T}(\mathbf{x}) + A(\boldsymbol{\eta}_0) - A(\boldsymbol{\eta}_1) + \log(\frac{\pi}{1-\pi}) \\ &= \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \end{aligned}$$

avec $\mathbf{w} = \begin{pmatrix} \boldsymbol{\eta}_1 - \boldsymbol{\eta}_0 \\ A(\boldsymbol{\eta}_0) - A(\boldsymbol{\eta}_1) + \log(\frac{\pi}{1-\pi}) \end{pmatrix}$ et $\boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} \mathbf{T}(\mathbf{x}) \\ 1 \end{pmatrix}$. On obtient alors un cas spécifique de la régression logistique en utilisant la transformation des données $\boldsymbol{\phi}(\mathbf{x})$. Vous allez explorer ces liens dans le cas de distributions conditionnelles gaussiennes dans le TP (modèle de discrimination linéaire de Fisher par exemple). Cela explique pourquoi la régression logistique est plus *robuste* que l'approche générative : plusieurs modèles génératifs différents donnent lieu au modèle de régression logistique, donc moins de suppositions doivent être faites sur les données.

1.2 Maximum de vraisemblance conditionnelle

Comme $-\log(p(Y = 1|X = \mathbf{x})) = \log(1 + e^{-f_{\mathbf{w}}(\mathbf{x})})$ le problème de maximisation de la vraisemblance pour le modèle conditionnel est équivalent à la minimisation du risque empirique pour la *perte logistique* définie par (pour $a = f_{\mathbf{w}}(\mathbf{x})$) :

$$\ell(y, a) = -y \log(\sigma(a)) - (1 - y) \log(\sigma(-a))$$

On a donc : $\widehat{\mathcal{R}}_n(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^{n} y_i \log(\sigma(\mathbf{w}^\top \mathbf{x}_i)) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^\top \mathbf{x}_i))$.

Comme $\frac{\partial}{\partial a}\ell(y, a) = \sigma(a) - y$, on a

$$\nabla_{\mathbf{w}} \widehat{\mathcal{R}}_n(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i(y_i - \sigma(\mathbf{w}^\top \mathbf{x}_i)) = -\frac{1}{n} \mathbf{X}^\top (\mathbf{y} - \boldsymbol{\mu}(\mathbf{w}))$$

qui ne se résout pas sous forme analytique car $\mu_i(\mathbf{w}) := \sigma(\mathbf{w}^\top \mathbf{x}_i)$ est une fonction transcendantale de $\mathbf{w}^\top \mathbf{x}_i$. On doit donc recourir à un algorithme itératif (descente de gradient ; méthode de Newton ; etc.).

Newton pour la régression logistique : moindres carrés pondérés itérés

Si on peut se permettre un algorithme quadratique en p on privilégiera l'algorithme de Newton.

La dérivée seconde de la perte logistique est $\frac{\partial^2}{\partial a^2}\ell(a, y) = \sigma(a)\sigma(-a) \geqslant 0$ (on voit donc que c'est une fonction convexe de a et donc de $\mathbf{w}$). On note $\mu_i = \sigma(\mathbf{x}_i^\top \mathbf{w}^{(t)})$, $\boldsymbol{\mu} = (\mu_i)_{1 \leqslant i \leqslant n} \in \mathbb{R}^n$ et $\mathbf{D}(\boldsymbol{\mu}) = \mathrm{Diag}\big((\mu_i(1 - \mu_i))_{1 \leqslant i \leqslant n}\big)$. La matrice hessienne du risque empirique est donc :

$$\mathbf{H}_{\mathbf{w}} \widehat{\mathcal{R}}_n(\mathbf{w}) = \frac{1}{n} \mathbf{X}^\top \mathbf{D}(\boldsymbol{\mu}) \mathbf{X}$$

L'algorithme de Newton fait donc la mise à jour suivante des paramètres :

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{H}^{-1}\nabla_{\mathbf{w}}\widehat{\mathcal{R}}_n(\mathbf{w}_t) \\ &= \mathbf{w}_t + n(\mathbf{X}^\top\mathbf{D}_t\mathbf{X})^{-1}\frac{1}{n}\mathbf{X}^\top(\mathbf{y}-\boldsymbol{\mu}_t) \\ &= (\mathbf{X}^\top\mathbf{D}_t\mathbf{X})^{-1}[\mathbf{X}^\top\mathbf{D}_t\mathbf{X}\mathbf{w}_t + \mathbf{X}^\top(\mathbf{y}-\boldsymbol{\mu}_t)] \\ &= (\mathbf{X}^\top\mathbf{D}_t\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{D}_t\mathbf{z}_t\end{aligned}$$

qui est la solution à l'équation normale pour le problème de moindres carrés *pondérés* par $\mathbf{D}_t$ avec cible $\mathbf{z}_t = \mathbf{X}\mathbf{w}_t + \mathbf{D}_t^{-1}(\mathbf{y}-\boldsymbol{\mu}_t)$, c'est à dire, la minimisation sur $\mathbf{w}$ de :

$$\frac{1}{n}\sum_i \frac{1}{2\sigma_i^2}((\mathbf{z}_t)_i - \mathbf{w}^\top\mathbf{x}_i)^2$$

où la pondération est $\sigma_i^2 = (\mu_i(1-\mu_i))^{-1}$. Comme la pondération change avec t, l'algorithme de Newton pour la régression logistique se nomme donc aussi la méthode des "moindres carrés pondérés itérés" (algorithme IRLS : Iterated Reweighted Least Squares). Comparez avec l'algorithme de Newton pour la régression linéaire standard qui converge en une itération (avec l'équation normale non-pondérée).

Comparez aussi la descente de gradient (de pas γ) pour la régression linéaire vs. la régression logistique :

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma\frac{1}{n}\sum_i(y_i - \mathbf{w}_t^\top\mathbf{x}_i)\ \mathbf{x}_i \qquad \text{(régression linéaire)}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \gamma\frac{1}{n}\sum_i(y_i - \sigma(\mathbf{w}_t^\top\mathbf{x}_i))\ \mathbf{x}_i \qquad \text{(régression logistique)}$$

Régularisation La régression logistique a les même possibilités de surapprentissage que la régression linéaire (en particulier quand $p > n$ où la matrice hessienne n'est plus inversible). Il faut donc aussi régulariser, par exemple avec un terme $\lambda\|\mathbf{w}\|^2$ à rajouter dans l'objectif à minimiser.

2 Complément sur la régression linéaire

En haute dimension, i.e., quand $p > n$, le prédicteur de la régression linéaire peut se calculer plus efficacement qu'avec la formule issue des équations normales.

Exercice 1. **(Lemme d'inversion de matrice)** Soit $\mathbf{X} \in \mathbb{R}^{n\times p}$ tel que $\mathbf{I} + \mathbf{X}^\top\mathbf{X}$ est inversible. Quel est la complexité de l'inversion matricielle d'une matrice $p\times p$ en général ? Si $p > n$, comment calculer $(\mathbf{I}+\mathbf{X}^\top\mathbf{X})^{-1}$ plus efficacement ?

En se basant sur le résultat de l'exercice (ou avec mes dérivations alternatives au tableau !), on a :

Proposition 1. *Soit $\hat{f}_\lambda$ le prédicteur de la régression linéaire régularisée pour une matrice de design $\mathbf{X}$ et un vecteur de variables de sortie $\mathbf{y}$. Dénotons $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$ la matrice de Gram des données. On a*

$$\hat{f}_\lambda : \mathbf{x}' \mapsto \mathbf{y}^\top(\mathbf{K} + n\lambda\mathbf{I})^{-1}\mathbf{X}\,\mathbf{x}'.$$

Cela vient du fait que l'on peut montrer que $\hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$ puisse aussi s'écrire comme $\hat{\mathbf{w}} = \mathbf{X}^\top\boldsymbol{\alpha} = \sum_i \alpha_i\mathbf{x}_i$ où

$$\boldsymbol{\alpha} = (\mathbf{K} + n\lambda\mathbf{I})^{-1}\mathbf{y}.$$

À noter que la prédiction $\hat{\mathbf{w}}^\top\mathbf{x}' = \sum_i \alpha_i\mathbf{x}_i^\top\mathbf{x}'$ ne dépend sur les données $\mathbf{x}_i$ qu'au travers du produit scalaire. Cela conduit à *l'astuce du noyau*, qui va être couvert plus en détails au prochain cours avec le *théorème du représentant*.