Sujet Impact : discrétisation, regroupement de catégories et introduction d'interactions

Adrien Ehrhardt, Christophe Biernacki, Philippe Heinrich, Vincent Vandewalle
Inria, Crédit Agricole Consumer Finance
first.last@inria.fr

Ce sujet s'inscrit dans le cadre d'une thèse CIFRE (ex-DAD) entre Inria et Crédit Agricole Consumer Finance. Finance.

Classification supervisée avec données mixtes En matières de crédit à la consommation, les instituts financiers cherchent à automatiser la décision de financement tout en ne sélectionnant que les clients susceptibles de rembourser ledit crédit. Depuis une quarantaine d'années, le *Credit Scoring* consiste à construire des modèles de classification supervisée $p_{\boldsymbol{\theta}}$ à partir des données demandées au client $\boldsymbol{x} = (x_j)_1^d$ et de l'observation du remboursement des clients passés $y \in \{0, 1\}$. Pour des raisons d'interprétabilité, la régression logistique est encore très largement utilisée. Pour des raisons pratiques (production d'une "grille" de score) et théoriques (compromis biais / variance), trois tâches de préparation des données sont généralement effectuées :

- la discrétisation des variables continues en m_j modalités : $f_j(x_j; m_j; \boldsymbol{c}_j) = \sum_{h=1}^{m_j} h \times \mathbb{1}_{]c_{j,h-1}, c_{j,h}]}(x_j)$.
- le regroupement de o_j modalités des variables catégorielles en m_j : $f_j(x_j; m_j; \boldsymbol{c}_j) = \sum_{h=1}^{m_j} h \times \mathbb{1}_{\mathbb{O}_h}(x_j)$ où $(\mathbb{O}_h)_1^{m_j}$ définit une partition de $o_j \to m_j$.
- l'introduction parsimonieuse d'interactions : $\boldsymbol{\delta} \in \{0, 1\}^{\frac{d(d-1)}{2}}$ où $\delta_{k,\ell} = 1$ si $f_k(x_k)$ et $f_\ell(x_\ell)$ interagissent au sens de la régression logistique.

Sur données discrétisées, regroupées et interagissant, la régression logistique s'écrit :

$$\text{logit}(p_{\boldsymbol{\theta}}(1|\boldsymbol{f}(\boldsymbol{x}); \boldsymbol{\delta}) = \theta_0 + \sum_{j=1}^{d} \theta_j^{f_j(x_j)} + \sum_{k<\ell} \delta_{k,\ell} \theta_{k,\ell}^{f_k(x_k), f_\ell(x_\ell)}.$$

Cette équation fait apparaître le lien étroit entre le modèle (et sa qualité) et le choix de la représentation des données. Ce constat motive une approche *representation learning* qui plongerait le choix de la représentation dans un critère de choix de modèle / fonction de coût :

$$(\boldsymbol{\theta}^\star, \boldsymbol{f}^\star, \boldsymbol{\delta}^\star) = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\boldsymbol{f}}, \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, \boldsymbol{\delta}}{\text{argmin}} \text{CRIT}(\boldsymbol{\theta}, \boldsymbol{f}, \boldsymbol{\delta})$$

Or en l'état, cette approche n'est pas tractable car l'espace des représentations $\boldsymbol{\mathcal{F}} \times \{0, 1\}^{\frac{d(d-1)}{2}}$ est discret et très combinatoire.

Première méthode de résolution proposée Nous avons proposé une première solution basée sur une interprétation de la représentation comme un ensemble de variables latentes. Au prix d'hypothèses raisonnables et d'une proposition de modèle entre les variables latentes et les variables $\boldsymbol{x}$ d'origine, on peut mettre en oeuvre un algorithme EM, classique en statistiques, pour tirer des représentations pertinentes des données au regard de l'objectif prédictif.

Deuxième méthode de résolution proposée Nous avons également proposé une solution basée sur une relaxation continue de ce problème discret, assez naturelle dans la littérature *machine learning*. L'estimation est immédiate en réinterprétant les poids d'un réseau de neurones peu profond.

Contributions de l'Impact Plusieurs directions d'étude sont possibles. D'un point de vue logiciel et concernant la première approche, il faudrait par exemple harmoniser les packages R et Python existants (gestion des manquants et des interactions), constater la similitude des résultats obtenus, étudier la possibilité d'accélérer l'algorithme (par des techniques de descente de gradient par exemple) et / ou de paralléliser le code. Concernant la seconde approche, il serait souhaitable de "robustifier" le code, conduire des tests d'optimisation des (nombreux) hyperparamètres et comparer les résultats des deux approches. Une autre direction d'étude, plus théorique, serait d'améliorer le processus de sélection des interactions en laissant la possibilité à l'utilisateur d'introduire des informations *a priori* sur les interactions qu'il considère les plus probables.