SOD322 - Recherche Opérationnelle et Données massives
Classification associative en grande dimension

Laurent LAM – Ilyes ER-RAMMACH

Février 2020

Table des matières

1 Introduction

Ce projet de Recherche Opérationnelle et Données Massives (SOD322) dans le cadre de l'ENSTA Paris a pour but d'implémenter et d'appliquer un algorithme de classification associative, forumulée par Bertsimas, Chang et Rudin [1]. En plus de cela, il faudra créer un processus de d'analyse de données et de traitement des variables afin de générer uniquement des coviariables binaires et discriminantes pour la tâche de classification donnée.

Nous devons traiter ici le jeu de données Adult dont le but est de prédire si une personne gagne plus de $\$50k$ par an. Les caractéristiques fournies peuvent concerner l'âge, le sexe, le niveau d'études ou encore le domaine dans lequel travaille la personne visée par exemple. Ce jeu de données comporte 32651 individus comportant chacune 14 caractéristiques additionnelles.

Les processus d'analyse, de traitement et de classification vont alors se décomposer en différentes étapes :

- **Création des caractéristiques binaires** : l'analyse des covariables vis-à-vis de la classe d'intérêt afin de transformer les covariables pertinentes en caractéristiques binaires, exploitables par l'algorithme de classification associative,
- **Génération des règles** : la génération de règles associatives et discriminantes permettant de classifier une partie de la population.
- **Tri des règles** : le tri et le filtre des règles générées afin d'utiliser uniquement les plus pertinentes successivement et de manière optimale.
- **Évaluation du classifieur** : l'évalutation des performances de classification des règles obtenues sur un jeu de données indépendant.

Le projet repose ainsi sur la réalisation successive des ces quatre étapes.

Le code source du projet se trouve sur le lien suivant :
https://github.com/laurentlam/rodm_ensta

2 Création des caractéristiques binaires

Tout d'abord, afin d'analyser le jeu de données afin de pourvoir déterminer les covariables discriminantes vis-à-vis de la classe d'intérêt et donc créer les caractéristiques binaires, il faut tout d'abord distinguer **3 types** de variables dans un jeu de données brut.

2.1 Distinction des types de variables

2.1.1 Variables numériques/continues : num

Il s'agit de covariables contenant des valeurs numériques pouvant varier énormément d'un individu à un autre, sans avoir des valeurs standardisées, (*age* ou *capital_gain* par exemple). On peut noter ici que ces variables possèdent une relation d'ordre entre les valeurs.

2.1.2 Variables nominales/catégorielles/binaires : cat

Il s'agit de covariables catégorielles dont les modalités possèdent en général une relation d'ordre entre elles. On peut donner l'exemple le niveau d'études *education* pour lequel une personne ayant un Doctorat possède nécessairement un Master au préalable ou même un diplôme du secondaire, et ainsi de suite. On peut également considérer des variables binaires comme *income* où la variable indique si la personne gagne plus de \$50k par an ou non. Il sera important de pouvoir garder cette relation d'ordre également lors de la binarisation.

2.1.3 Variables multi-classes/ : class

Il s'agit de covariables possédant différentes modalités sans pour autant qu'une hiérarchie ou une relation d'ordre semble se dégager au premier abord. On peut notamment citer l'exemple de la *race* (White, Black, Asian-Pacific-Islander parmi les modalités) ou encore de *marital_status* (Married, Divorced, Unmarried, etc.).

2.2 Conversion et Agrégation de types de variables

On obtient alors pour le jeu de données **Adult** les **covariables brutes** suivantes :

Variables	Type	Structure	Modalités
age	num	float	continues
workclass	class	str	8
fnlwgt	num	int	continues
education	cat	str	16
education_num	cat	int	16
marital_status	class	str	7
occupation	class	str	14
relationship	class	str	6
race	class	str	5
sex	cat	str	2
capital_gain	num	int	continues
capital_loss	num	int	continues
hours_per_week	num	int	continues
native_country	class	str	41
income	cat	str	2

On les pré-traite ensuite de sorte à obtenir plus facilement des covariables analysables. On supprime notamment des covariables redondantes, très reliées ou un nombre trop important de modalités

- *education* par rapport à **education_num**,
- *relationship* par rapport à **marital_status**,
- *native_country* et **race**.

Par ailleurs on va aggréger les modalités de certaines variables multi-classes afin de les simplifier.

- On définit la covariable nominale **education_cat** à partir de *education*. On va rassembler les modalités de niveau d'études par classes larges.
  - **0** : si l'individu n'a pas atteint la diplômation du lycée
  - **1** : si l'individu n'a pas au moins un *Bachelor*
  - **2** : si l'individu a obtenu un *Bachelor*, un *Master* ou plus.
- On définit la covariable binaire **marital_status_cat** à partir de *marital_status*.
  - **1** : si l'individu est marié et si le/la partenaire est toujours présent(e)
  - **0** : sinon.
- On définit la covariable nominale **workclass_gov** à partir de *workclass* en n'attribuant que des valeurs non-nulles aux modalités liées à une classe de fonctionnaires.
  - **3** : pour le niveau *federal*
  - **2** : pour le niveau *state*
  - **1** : pour le niveau *local*
  - **0** : pour les autres emplois.
- On définit la covariable nominale **workclass_private** à partir de *workclass* en n'attribuant que des valeurs non-nulles aux modalités liées aux emplois du domaine privé.
  - **3** : pour le niveau *self-empl-inc*
  - **2** : pour le niveau *self-empl-not-inc*
  - **1** : pour le niveau *private*
  - **0** : les autres emplois.
- On définit la covariable nominale **occupation_cat** à partir de *occupation*.
  - **3** : pour *Exec-managerial*, *Prof-specialty*

— **2** : pour *Tech-support*, *Protective-serv*
— **1** : pour *Sales*, *Craft-repair*, *Transport-moving*
— **0** : pour les autres emplois.

Les 3 attributions précédentes peuvent paraître arbitraires et polémiques mais ces dernières sont le résultat d'une analyse au préalable.

Pour les covariables concernées, on construit une table de contingence entre les modalités de la covariable et la classe d'intérêt 0-1 (l'individu gagne moins de \$50k par an ou plus). On normalise ensuite chaque valeur de la table en divisant par le nombre d'individus présents pour chaque modalité et on obtient ainsi un ratio du nombre d'individus gagnant plus de \$50k/an sachant qu'il possède la modalité considérée. On peut ainsi trier de manière croissante les modalités en fonction de ce ratio et alors aggréger les modalités proches entre elles (en termes de ratio) afin de réduire le nombre de modalités distinctes.

Notons que cette pré-analyse ainsi que les suivantes dans la suite du rapport ont bien été effectuées sur le jeu de données d'entraînement uniquement.

On obtient alors les covariables toutes converties en *int* ou *float* via des variables catégorielles ou continues :

Variables	Type	Structure	Modalités
age	num	float	continues
workclass_gov	cat	int	4
workclass_private	cat	int	4
fnlwgt	num	int	continues
education_cat	cat	int	3
marital_status_cat	cat	int	2
sex	cat	int	2
capital_gain	num	int	continues
capital_loss	num	int	continues
hours_per_week	num	int	continues
income	cat	int	2

2.3 Analyse indépendante des variables

Pour réduire le nombre de covariables, on va commencer par analyser indépendamment chacune d'entre elles vis-à-vis de la classe cible d'intérêt via différents tests statistiques d'indépendance, dans le but de filtrer toutes les covariables qui ne sont pas reliées statistiquement à la classe cible.

2.3.1 Test de Student - Variables numériques

Dans le cadre des variables numériques et continues, on divise la population en deux selon la classe cible d'intérêt (ici $income = 1$ ou $income = 0$) puis on va analyser l'influence de chaque variable indépendemment des autres en comparant ces deux populations à travers le test d'indépendance de Student entre deux populations. On obtient alors la valeur de la statistique de Student ainsi que sa p-valeur. On peut également calculer un intervalle de confiance en utilisant la méthode de *bootstrap*. Ainsi on va filtrer les variables

si elles possèdent une p-valeur supérieure à 5%, c'est-à-dire si l'hypothèse d'indépendance des deux populations selon cette variable ne peut être rejetée avec un risque à 5%.

2.3.2 Test du Chi-2 - Variables nominales/de classes

Dans le cadre des variables catégorielles et des classes, on calcule tout d'abord la table de contingence de chaque variable par rapport à la classe cible d'intérêt. A partir de ce tableau de contingence, on applique le test du Chi-2 afin de mesurer la significativité statistique entre deux variables discrètes/catégorielles. On obtient ainsi la valeur de sa statistique et la p-valeur associée. De la même façon, on va filtrer les variables si elles possèdent une p-valeur supérieure à 5%.

2.4 Analyse de corrélation des variables

Afin d'effectuer un filtrage supplémentaire pour réduire à nouveau le nombre de covariables, on va utiliser différentes méthodes de calcul de corrélation afin de ne sélectionner que les plus pertinentes. Les méthodes utilisées seront les 3 différentes méthodes de calcul de coefficient de corrélation que sont les méthodes de $Pearson$, $Spearman$ et de $Kendall$. On calcule ainsi la matrice de corrélation avec chaque méthode et on ne retient que les paires de variables corrélées à partir d'un certain seuil (par défaut à 0.7).

2.4.1 Filtrage direct

Un filtrage direct consisterait alors à conserver uniquement les covariables qui sont directement corrélées avec la classe d'intérêt cible et de filtrer le reste des variables. Le problème avec cette approche est qu'il est rare que de nombreuses covariables ou même une seule d'entre elles soit corrélée lorsqu'on regarde le coefficient de corrélation selon ces méthodes. On pourrait donc se retrouver avec un filtrage drastique, en ne conservant aucune covariable.

2.4.2 Filtrage indirect

On considère alors un filtrage indirect, en considérant tous les couples de covariables corrélées entre elles. Cette corrélation pourrait alors indiquer une redondance ou une démultiplication des mêmes informations discriminantes dans plusieurs variables ce qui pourrait déservir le modèle de classification. Pour éviter cela, on décide pour chaque couple de covariables corrélées d'en retirer une des deux. On supprime alors celle qui est la moins corrélée avec la variable cible d'intérêt. Cela conduit ainsi à un filtre léger de covariables permettant potentiellement de ne posséder que des covariables apportant chacune des informations discriminantes et uniques.

2.5 Agrégation de modalités

2.5.1 Variables nominales

Les variables nominales, avec une relation d'ordre entre les différentes modalités, possèdent souvent de nombreuses modalités qui se transformeraient en de nombreuses colonnes et variables binaires et conduiraient à de plus longues résolutions par la suite. Pour éviter cela, on va agréger différentes modalités ensemble si elles sont consécutives

et qu'elles ne comprennent qu'une même et unique population (si l'on considère à nouveau les 2 populations divisées selon la classe d'intérêt 1 ou 0). Cela permet d'agréger grandement les modalités des variables nominales.

2.5.2 Variables continues - Binning

En ce qui concerne les variables continues et numériques, il faut tout d'abord effectuer une étape de *binning* afin de créer des catégories et discrétiser et rassembler toutes ces valeurs continues. Pour cela on s'intéresse plus particulièrement à l'étendue des valeurs possibles pour la population dont la classe d'intérêt vaut 1, car elle est en général plus grande que la population opposée, qui possède des valeurs plus centrées avec une variance plus faible. On calcule alors les bordures des différentes catégories en utilisant les quantiles de cette population. On divise alors l'étendue des valeurs en un nombre maximum de catégories (par défaut 5), et on utilise les quantiles pour former des catégories avec un nombre d'individus équivalent pour cette population. Il peut arriver qu'en découpant en quantile, 2 catégories formées, trop peuplées autour d'une même valeur, se chevauchent. Cela conduit à la fusion des deux catégories. Par cette procédure on obtient une discrétisation des variables continues et la création de catégories. On peut ensuite convertir la totalité des valeurs des deux populations selon ce "découpage" en catégories. On obtient alors des covariables désormais nominales. On peut ainsi appliquer le processus d'agrégation de modalités des covariables nominales afin de réduire le nombre de modalités.

2.6 Binarisation

Il s'agit ainsi de la dernière étape de la création des caractéristiques pour la classification, en transformant les covariables désormais toutes catégorielles, de classes ou nominales en variables binaires.

2.6.1 Binarisation classique

La binarisation classique d'une covariable consiste à créer pour chaque modalité et la valeur de la colonne sera à 1 si l'individu possède cette modalité et 0 sinon. Ce processus sera alors appliqué à toutes les covariables de classes, qui ne possèdent pas de relation d'ordre entre les modalités. Pour les variables continues et les valeurs catégorielles, nous allons appliquer la binarisation ordinale qui permet de conserver une certaine relation d'ordre entre les modalités de la classe.

2.6.2 Binarisation ordinale

La binarisation ordinale consiste à prendre en compte la relation d'ordre des modalités lors de la binarisation. Pour chaque covariable nominale, on va créer $(m-1)$ colonnes si m représente le nombre de modalités de la covariable, où la colonne $i \in [[0, m-1]]$ sera activée (=1) si la valeur de la covariable k est supérieure au seuil de la colonne i.

Par exemple si l'on souhaite binariser une covariable dont les modalités sont 0, 1, 2, 3, on va obtenir 3 colonnes dont les valeurs associées seront :
- 0 : 0|0|0
- 1 : 1|0|0
- 2 : 1|1|0

— 3 : 1|1|1

Cette manière de binariser permet de conserver une notion d'ordre au sens où dès lors que la modalité est plus élevée que la valeur seuil de la colonne, cette dernière sera activée et vaudra 1. Le classifieur peut ainsi prendre en compte cette dépendance entre les colonnes ce qui enlève l'inconvénient d'indépendance lors de la binarisation.

2.7 Variables discriminantes obtenues

Après analyse et traitement des données on obtient finalement les **14 covariables binaires** suivantes :

Variables	Modalités
age	24+ 31+ 39+ 49+
sex	Male
hours_per_week	32+ 40+ 45+
education_cat	High-School diploma or more Bachelor or more
workclass_private	Private or more Self-empl-not-inc or more Self-empl-inc
marital_status_cat	Married
income	+$50k/year

3 Génération et tri de règles

Nous avons implémenté l'algorithme de génération de règles à partir des notes de cours. L'algorithme de tri des règles était déjà présent.

3.1 Configuration et Paramètres

Voici la configuration de la machine utilisée :

— **Processeur** : Intel Core i7-9700K CPU @ 3.60GHz

— **Mémoire vive** : 64Go

— **Système d'exploitation** : Linux Ubuntu 20.04.1 LTS

ainsi que les versions des langages et modules utilisés :

— **Julia** : 1.4.1

— **Python** : 3.8.5

— **CPLEX** : 12.10.0

Concernant les paramètres des l'algorithmes, tous les paramètres par défaut ont été conservés pour la génération et le tri des règles excepté le nombre d'itérations lors de l'algorithme de génération. Il a été abaissé à 3 (au lieu de 5) afin de générer moins de règles, et pouvoir trier les règles obtenues plus facilement et en un temps de résolution plus raisonnable.

3.2 Règles obtenues de règles

L'algorithme de génération a ainsi fourni **117 règles** pour un temps de calcul de l'ordre de **92 secondes**. L'algorithme de tri et de filtre des règles a permis de réduire ce nombre de règles à **17 règles** pour un temps de calcul de **300 secondes**, ce qui correspond bien à la durée limite accordée à la résolution et au tri des règles.

Voici les règles finales :

Cible	Covariables													
Salaire	Age				Sexe	Heures/sem			Éducation		Profession			Marié ?
+50k$/an	24+	31+	39+	49+	♂	32+	40+	45+	1	2	1	2	3	Marié ?
1	1	0	0	1	1	0	0	1	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0	1	0	1	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	1	0	1	0	0	0	1
1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	1	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.3 Procédure d'évaluation

Afin d'évaluer la performance des différentes étapes de traitement de données ainsi que de l'algorithme de classification, on utilise différentes métriques :

- **Precision** par classe : Proportion des prédictions pertinentes de la classe parmi l'ensemble des prédicions de la classe,
- **Recall** par classe : Proportion des prédictions pertinentes de la classe parmi l'ensemble des individus de la classe,
- **Average Recall** : Moyenne du *Recall* de chaque classe considérée,
- **Average Precision** : Moyenne de la *Precision* de chaque classe considérée
- **Weighted Average Recall** : Moyenne pondérée du *Recall* de chaque classe considérée,
- **Weighted Average Precision** : Moyenne pondérée de la *Precision* de chaque classe considérée

Pour la procédure de validation, on divise le jeu de données en 2 populations :

- le jeu de données **d'entraînement**, correspondant *1/32* de la totalité des données pour *adult.csv* (2/3 pour les autres jeux de données). Toutes les analyses, traitement et conversion des données s'effectueront à partir de ce jeu de données.

- le jeu de données **de test**, correspondant au reste du jeu de données. La création des mêmes covariables binaires sera appliquée à ces données et l'algorithme de classification y sera appliquée. Cela permet de quantifier et d'évaluer la capacité de généralisation de la méthode et du modèle afin de ne pas avoir un algorithme ou une méthode trop spécifique au jeu de données d'entraînement.

3.4 Résultats numériques

On obtient les résultats numériques suivants :

Résultats sur le jeu d'entraînement		
Classe	Precision	Recall
0	0.72	0.54
1	0.91	0.96
Average	0.82	0.75
Weighted Average	0.88	0.89

Résultats sur le jeu de test		
Classe	Precision	Recall
0	0.72	0.52
1	0.91	0.96
Average	0.81	0.74
Weighted Average	0.87	0.88

4 Questions d'ouvertures

4.1 Application à un autre jeu de données

On applique la méthode à un autre jeu de données : Acute Inflammations. Le jeu de données "Acute Inflammations" ou alors *diagnosis* possède **120 individus** dont la classe d'intérêt est **la présence d'inflammation** pour **6 covariables** dont 5 sont d'entre elles sont catégorielles binaires. La dernière covariable est numérique. L'analyse des données et le traitement des covariables conduit à la création de **7 caractéristiques binaires**. La méthode permet de générer **81 règles** en **14 secondes** pour seulement **5 règles** après tri et filtre, effectué en **80 secondes**.

Cible	Covariables						
Inflammation	Température				Douleurs	Troubles	Douleurs
	36.98+	37.26+	37.74+	40.4+	lombaires	urinaires	urinaires
1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	0
1	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

On obtient les résultats numériques suivants :

Résultats sur le jeu d'entraînement		
Classe	Precision	Recall
0	0.58	1.00
1	1.00	0.85
Average	0.79	0.92
Weighted Average	0.93	0.88

Résultats sur le jeu de test		
Classe	Precision	Recall
0	0.86	1.00
1	1.00	0.97
Average	0.93	0.99
Weighted Average	0.98	0.98

Les résultats sont surprenants au sens où **les performances sur le jeu de test sont meilleures que celles sur le jeu d'entraînement**. Mais cela pourrait s'expliquer par la **faible quantité de données** ou alors par le fait que **les individus les plus diversifiés**

et les plus difficiles à classer soient dans le jeu d'entraînement directement alors que les individus dans le jeu de test sont plus classiques ou faciles à classifier.

4.2 Analyse de l'influence du temps accordé au tri de règles

En augmentant le temps accordé au tri de règles, on obtient des règles finales différentes pour chaque temps différent.

Temps accordé (secondes)	Règles triées
300	16
600	15
1500	13
3000	13

Voici les résultats de *Precision* et *Recall* par classe :

Temps accordé (secondes)	Classe	Precision		Recall	
		Train	Test	Train	Test
300	0	0.79	0.77	0.5	0.47
	1	0.91	0.9	0.97	0.97
600	0	0.77	0.74	0.51	0.47
	1	0.91	0.9	0.97	0.97
1500	0	0.79	0.76	0.5	0.47
	1	0.91	0.9	0.97	0.97
3000	0	0.79	0.76	0.5	0.46
	1	0.91	0.9	0.97	0.97

Voici les résultats de *Precision* et *Recall* en moyenne :

Temps accordé (secondes)	Moyenne	Precision		Recall	
		Train	Test	Train	Test
300	Simple	0.85	0.83	0.74	0.72
	Pondérée	0.89	0.88	0.89	0.89
600	Simple	0.84	0.82	0.74	0.72
	Pondérée	0.88	0.87	0.89	0.88
1500	Simple	0.85	0.83	0.73	0.72
	Pondérée	0.89	0.87	0.89	0.88
3000	Simple	0.85	0.83	0.74	0.71
	Pondérée	0.89	0.87	0.89	0.88

On peut voir qu'**augmenter le temps accordé au tri des règles ne permet pas généralement d'augmenter les performances** mais peut permettre de réduire le nombre de règles en sortie. Il s'agirait de réduire le nombre de règles au maximum sans pour autant perdre significativement en performances.

Cependant **réduire le temps accordé au tri peut conduire à un temps insuffisant pour la résolution du problème** et donc conduire à un fichier de règles obtenu naïf avec une unique règle qui s'applique à tous les individus.

5 Conclusion

Dans ce projet on a pu implémenter un processus entier d'analyse de pertinence et de traitement des données ainsi que l'algorithme de classification associative. On a ensuite pu tester les performances de cet algorithme de classification sur le jeu de données **Adult** mais aussi sur celui de **Acute Inflammations**. On a également comparé les performances de l'algorithme lorsque l'on modifie le temps accordé au tri et au filtre des règles.

Références

[1] D. Bertsimas, A. Chang, and C. Rudin. An integer optimization approach to associative classification. *Advances in Neural Information Processing Systems*, 4 :3302–3310, 01 2012.