Propositionaliser des attributs numériques sans les discrétiser, ni les agréger

Agnès Braud*, Nicolas Lachiche*

*Université de Strasbourg, LSIIT, Pôle API, Bd Brant, 67400 Illkirch
{agnes.braud,nicolas.lachiche}@unistra.fr,
https://lsiit-cnrs.unistra.fr/fdbt-fr/index.php/Accueil

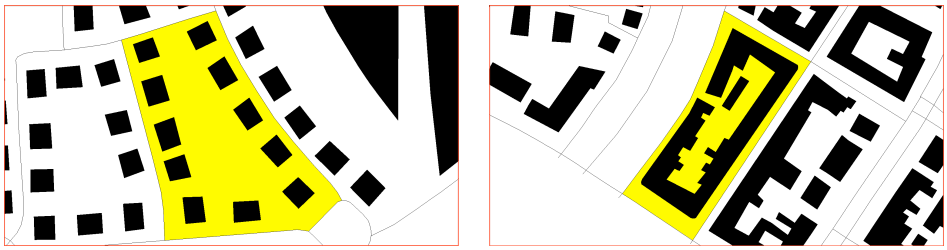
Résumé. La fouille de données relationnelles considère des données contenues dans au moins deux tables reliées par une association un-à-plusieurs, par exemple des clients et leurs achats, ou des molécules et leurs atomes. Une façon de fouiller ces données consiste à transformer les données en une seule table attribut-valeur. Cette transformation est appelée propositionalisation. Les approches existantes gèrent principalement les attributs catégoriels. Une première solution est donc de discrétiser les attributs numériques pour les transformer en attributs catégoriels. Les approches alternatives, qui gèrent les attributs numériques, consistent à les agréger. Nous proposons une approche duale de la discrétisation, qui inverse l'ordre de traitement du nombre d'objets et du seuil, et dont la discrétisation généralise les quartiles. Nous pouvons ainsi construire des attributs que les approches existantes de propositionalisation ne peuvent pas construire, et qui ne peuvent pas non plus être obtenus par les systèmes complets de fouille de données.

1 Introduction

La fouille de données relationnelles (Džeroski et Lavrač, 2001) considère des données contenues dans au moins deux tables reliées par une association un-à-plusieurs, par exemple des clients et leurs achats, ou des molécules et leurs atomes. Une façon de fouiller ces données consiste à transformer les données en une seule table attribut-valeur. Cette transformation est appelée propositionalisation (Lachiche, 2010).

Les motivations de ce travail sont liées à un problème géographique. Ce problème consiste à prédire la classe d'îlots urbains, cf. figure 1. L'îlot est caractérisé uniquement par les propriétés géométriques de son polygone : aire, élongation et convexité. Les bâtiments que l'îlot contient sont représentés par des polygones également caractérisés par les mêmes propriétés géométriques. La densité de l'îlot est aussi une propriété de l'îlot.

Les discussions avec les experts montrent que la classe dépend de conditions sur la géométrie des bâtiments et du nombre de bâtiments satisfaisant ces conditions, par exemple la classe habitat individuel dépend de la présence de petits bâtiments principalement. L'apprentissage consiste à déterminer les attributs pertinents et leurs seuils, ainsi que le nombre de ces bâtiments. Les approches existantes de fouille de données relationnelles ne permettent pas de



habitat individuel habitat collectif

FIG. 1 – *Exemple géographique : prédiction de la classe d'un îlot*

rechercher en même temps un seuil sur un attribut et un seuil sur le nombre d'objets satisfaisant cette condition, comme nous le verrons dans la section suivante. Nous proposons une approche, baptisée cardinalisation, décrite dans la section 3, qui offre cette souplesse. La section 4 présente une validation expérimentale sur des données artificielles et sur des données réelles.

2 Approches existantes

Les approches de la fouille de données relationnelles proviennent principalement de la programmation logique inductive (PLI) (Lavrač et Džeroski, 1994). Ces approches s'appuient en général sur une discrétisation pour transformer les attributs numériques en attributs catégoriels. D'autres approches s'appuient sur les bases de données relationnelles, et proposent d'utiliser les fonctions d'agrégation disponibles dans le langage SQL.

Dans ce qui suit, la table principale est la table qui contient la colonne cible. Ses lignes sont les individus. La table annexe décrit des objets liés, ou composant l'individu. On note E les objets liés à un individu i, par exemple les bâtiments d'un îlot. Le cardinal de cet ensemble, c'est-à-dire le nombre de bâtiments, est noté $|E|$. Nous utiliserons le terme anglais *features* pour appeler les attributs construits par la propositionalisation afin de les distinguer clairement des colonnes des tables originales.

Beaucoup d'algorithmes de propositionalisation, y compris des algorithmes récents, par exemple RelF (Kuzelka et Zelezný, 2009), gèrent les attributs numériques comme des attributs catégoriels. Ainsi pour traiter les attributs continus, il faut les discrétiser. Cette discrétisation se fait *a priori*, indépendamment de la construction du modèle. L'avantage est que l'on considère globalement les objets, tous individus confondus. L'inconvénient est que le choix des seuils se fait sans tenir compte de la classe, ni d'un sous-ensemble des exemples.

Les systèmes Polka (Knobbe et al., 2001) et Relaggs (Krogel et Wrobel, 2001, 2003) gèrent explicitement les attributs numériques. Ils appliquent les fonctions f d'agrégation usuelles de SQL (avg, min, max, etc.) à chaque attribut numérique A pour l'ensemble E des tuples

$$\text{agregation}(A, f, E) = f(\{v_A(t), t \in E\})$$

L'intérêt de ces approches est qu'elles résument les données et ne sont pas sujettes à l'explosion combinatoire.

3 Cardinalisation

Nous proposons une nouvelle approche pour propositionaliser en présence d'attributs numériques, sans discrétiser à l'avance, et en ne se restreignant pas à un nombre de features limité par le nombre de fonctions d'agrégation. Nous avons appelé cette approche cardinalisation pour mettre en avant qu'elle fixe la cardinalité entre la table principale et la table annexe au contraire d'approches existantes qui fixent un seuil sur le domaine de l'attribut numérique.

Les fonctions d'agrégation offrent une solution simple pour propositionaliser des attributs numériques. Cependant le nombre de fonctions proposé est petit, et ne permet pas de choisir à la fois un seuil sur l'attribut numérique et un nombre minimum d'objets correspondants.

Une alternative consiste à discrétiser l'attribut numérique afin de le transformer en attribut catégoriel. Étant donné un attribut numérique A de la table annexe, une *feature* est construite pour chaque seuil s provenant de la discrétisation. La valeur de cette feature pour un individu relié à un ensemble E de tuples de la table annexe est le nombre de tuples t dont la valeur $v_A(t)$ est inférieure au seuil s :

$$\text{agregationApresDiscretisation}(A, s, E) = |t \in E \text{ tel que } v_A(t) \leq s|$$

La construction d'un arbre de décision ne pourra choisir des seuils sur l'attribut numérique que par le choix de l'attribut le plus discriminant, puisque l'attribut est défini par le seuil.

La cardinalisation cherche à inverser les rôles des seuils sur l'attribut numérique et sur la cardinalité. Elle fixe un seuil sur la cardinalité, et laisse le programme attribut-valeur choisir le seuil sur l'attribut numérique. En fait, étant donné un attribut numérique A de la table annexe, une *feature* est construite pour chaque seuil possible k de la cardinalité, entre 1 et le nombre maximum d'objets par individu. La valeur de cette feature est la valeur minimale du seuil sur l'attribut numérique telle qu'au moins k tuples t ont une valeur $v_A(t)$ pour l'attribut numérique A inférieure à ce seuil :

$$\begin{aligned}\text{cardinalisation}(A, k, E) &= min(s \in Domaine(A) \text{ tel que } |t \in E \text{ tel que } v_A(t) \leq s| \geq k) \\ &= min(s \in Domaine(A) \text{ tel que agregationApresDiscretisation}(A, s, E) \geq k)\end{aligned}$$

Lorsqu'un îlot a moins de k objets, un seuil infini est affecté à la *feature*.

L'intérêt de la cardinalisation est de ne pas fixer le seuil sur l'attribut numérique lors de la propositionalisation, donc de laisser le programme attribut-valeur choisir le seuil pertinent, en particulier en tenant compte du contexte, par exemple dans les branches d'un arbre de décision.

La cardinalisation étant une approche duale de la discrétisation suivie d'une agrégation, une discrétisation théoriquement équivalente à la cardinalisation consiste à introduire un seuil entre chaque valeur de l'attribut numérique. L'inconvénient est que le nombre de seuils, donc de *features* produites par cette discrétisation en intervalles contenant chacun un objet, est le nombre maximum de valeurs prises par l'attribut tous objets et individus confondus. Dans le pire cas, c'est de l'ordre du nombre total d'objets de la table annexe, pour chacun de ses attributs numériques. La cardinalisation produit un nombre de features, qui est exactement le nombre maximum d'objets par individu. Il est donc possible de choisir l'approche qui produit le moins de *features* entre la cardinalisation et la discrétisation suivie d'une agrégation.

Si le nombre de features est trop grand dans le cas de la discrétisation suivie d'une agrégation, il est possible de discrétiser l'attribut numérique en intervalles plus grands. Cependant, le

défaut reste que les seuils sont fixés par rapport à la totalité des objets, tous individus confondus, par exemple pour toutes les aires des bâtiments tous îlots confondus. Si le nombre de features est trop grand dans le cas de la cardinalisation, on peut discrétiser la cardinalité.

Ainsi, au lieu de créer une *feature* par valeur possible de la cardinalité, c'est-à-dire par nombre de bâtiments dans un îlot, il est possible de discrétiser la cardinalité pour obtenir un nombre fixe de features, de façon similaire à la discrétisation de l'attribut numérique lui-même suivi de l'agrégation. Au lieu de fixer un seuil en nombre absolu sur la cardinalité, la cardinalisation discrétisée utilise un nombre relatif. En fait, cela correspond aux quantiles. Si l'on choisit quatre intervalles, les *features* correspondent au premier quartile, à la médiane, et au troisième quartile. Le k-ième n-quantile est le seuil tel qu'au moins k/n objets sont sélectionnés :

$$\begin{aligned}&\text{quantile}(A, k, n, E) = min(s \in Domaine(A) \text{ tel que}\\ &\text{agregationApresDiscretisation}(A, s, E) \geq k \times |E|/n)\end{aligned}$$

Alors que les quartiles sont implémentés dans des approches de propositionalisation, aucune ne les généralise aux quantiles. De plus, les systèmes complets de fouille de données relationnelles n'acceptent pas de biais de langage permettant de construire des agrégats correspondant aux quantiles. Nous proposons d'utiliser les quantiles dans le cadre de la propositionalisation, afin de pouvoir paramétrer le nombre de *features* générées, et ainsi obtenir une expressivité plus élevée que les approches existantes de généralisation.

4 Validation expérimentale

La validation expérimentale comprend deux tests. Le premier test consiste à générer des données artificielles dont la classe est choisie suivant un modèle connu correspondant aux biais de langages de chacune des approches afin de vérifier les écarts entre les quatre approches : agrégation après discrétisation, agrégations numériques, cardinalisation, et quantiles. Le second test consiste à comparer les approches sur des données réelles afin de vérifier leur utilité.

Données artificielles Les données artificielles ont une structure identique aux données géographiques. La table principale décrit des îlots avec une classe booléenne et des attributs numériques aire, élongation et convexité. La table bâtiment a également trois attributs numériques aire, élongation et convexité, et chaque bâtiment appartient à un et un seul îlot. Les valeurs des attributs numériques ont été générées aléatoirement entre 0 et 1000 avec une distribution uniforme. Le nombre de bâtiments par îlot a été choisi aléatoirement entre 0 et 100 suivant une distribution uniforme. Un ensemble de 1000 îlots a ainsi été construit. Cet ensemble a été dupliqué pour construire deux jeux de données dont la classe a été calculée suivant les deux modèles suivants :

Nombre absolu : premier jeu de données, au moins 15 bâtiments ont une aire strictement inférieure à 300

Nombre relatif : second jeu de données, au moins 30% des bâtiments ont une aire strictement inférieure à 300

Nous avons utilisé un arbre de décision pour vérifier s'il est possible d'apprendre exactement le modèle caché. Le tableau 1 indique le pourcentage d'îlots correctement classés en

fonction de l'approche de propositionalisation utilisée et pour chaque modèle caché. Le pourcentage indiqué est la moyenne de 10 validations croisées en 10 blocs.

	Cardinalisation	Quantiles	Agrégations	Discrétisation
Nombre absolu	99,9%	77,2%	95,6%	99,3%
Nombre relatif	90,4%	100,0%	84,8%	90,0%

TAB. 1 – *Précision de J48 par approche et pour chaque modèle caché.*

Nous observons que la cardinalisation et les quantiles permettent effectivement d'apprendre des modèles impliquant respectivement un nombre absolu et un nombre relatif de certains bâtiments. Les arbres de décision correspondants ne comportent bien qu'un nœud. Dans le cas du modèle défini par un nombre absolu, l'approche de discrétisation puis agrégation obtient une bonne précision, mais son modèle comporte 9 nœuds. Les arbres construits à partir des quantiles et des fonctions d'agrégation pour le premier problème comportent respectivement 141 et 33 nœuds. Ceux construits à partir de la cardinalisation et des fonctions d'agrégation pour le second problème ont 77 et 79 nœuds respectivement.

Données réelles Nous considérons trois jeux de données géographiques correspondant à des quartiers différents en périphérie d'une grande agglomération. Les sept classes d'îlots ont des répartitions variant de 0 à 44,7% entre les classes, et entre 0 et 44,7% pour une même classe selon le quartier. De plus, il est probable que les caractéristiques d'une classe donnée, par exemple habitat collectif, diffèrent d'un quartier à l'autre. Les experts imposent que le modèle construit soit compréhensible. Le pourcentage d'îlots correctement classés a été évalué par la moyenne de 10 validations croisées en 10 blocs, cf. tableau 2.

	Cardinalisation	Quantiles	Agrégations	Discrétisation
Quartier 1	93,3%	90,9%	92,2%	92,3%
Quartier 2	90,0%	92,4%	90,7%	90,1%
Quartier 3	93,4%	93,1%	92,6%	93,6%
Quartier 4	90,6%	92,9%	92,9%	90,1%

TAB. 2 – *Précision de J48 par approche et par quartier.*

Nous observons que la cardinalisation, les quantiles et les agrégations obtiennent des précisions différentes selon les quartiers. Leurs biais de langage offrent bien une expressivité différente. Les différences entre les approches sont moins marquées que sur les données artificielles, cf. tableau 1. Ceci s'explique par le fait que les modèles sont plus complexes. En effet, ils comportent respectivement 30, 150 et 60 nœuds environ pour les quartiers 1, 2 et 3, avec peu d'écarts entre les trois approches. Les résultats sont globalement très bons pour des problèmes comprenant 7 classes à prédire. Les bâtiments ne sont décrits que par leurs géométries en deux dimensions, c'est-à-dire vus du dessus, comme c'est le cas pour les photographies aériennes. Il est parfois impossible de différencier certains bâtiments comme des écoles et des immeubles. Il semble difficile d'améliorer sensiblement ces résultats.

5 Conclusion

Nous proposons d'ajouter les quantiles aux outils de la propositionalisation, et plus généralement une approche duale de la discrétisation qui ne s'applique que si le nombre maximum d'objets reste d'un ordre compatible avec le nombre de colonnes que l'on peut gérer, mais qui permet de laisser le programme attribut-valeur choisir le meilleur seuil sur l'attribut numérique et le meilleur seuil sur le nombre d'objets correspondants en même temps.

Remerciements

Ce travail a été réalisé dans le cadre du projet ANR GeOpenSim, et les données préparées par les laboratoires LIVE de l'université de Strasbourg et COGIT de l'IGN.

Références

Džeroski, S. et N. Lavrač (Eds.) (2001). *Relational data mining*. Springer.

Knobbe, A. J., M. de Haas, et A. Siebes (2001). Propositionalisation and aggregates. In L. De Raedt et A. Siebes (Eds.), *PKDD*, Volume 2168 of *LNCS*, pp. 277–288. Springer.

Krogel, M.-A. et S. Wrobel (2001). Transformation-based learning using multirelational aggregation. In *ILP*, Volume 2157 of *LNCS*, pp. 142–155. Springer.

Krogel, M.-A. et S. Wrobel (2003). Facets of aggregation approaches to propositionalization. In *Work-in-progress session of the 13th Int. Conf. on Inductive Logic Programming*.

Kuzelka, O. et F. Zelezný (2009). Block-wise construction of acyclic relational features with monotone irreducibility and relevancy properties. In A. P. Danyluk, L. Bottou, et M. L. Littman (Eds.), *ICML*, Volume 382 of *ACM Int. Conf. Proceeding Series*, pp. 72. ACM.

Lachiche, N. (2010). *Encyclopedia of Machine Learning* (C. Sammut and G. I. Webb ed.)., Chapter Propositionalization. Springer.

Lavrač, N. et S. Džeroski (1994). *Inductive Logic Programming : Techniques and Applications*. Ellis Horwood.

Summary

Relational Data Mining deals with data stored in at least two tables linked by a one-to-many relationship, as in the case of customers and their purchases, or molecules and their atoms. An approach for mining these data consists in transforming them into a single attribute-value table. This transformation is called propositionalisation. Existing approaches mainly deal with categorical attributes. A first solution is then to discretise numeric attributes in order to transform them into categorical ones. Alternative approaches dealing with numeric attributes consist in aggregating them. We propose an approach dual to discretisation that reverses the processing of objects and thresholds, and whose discretisation generalises quartiles. We are thus able to construct attributes that existing approaches cannot, and that cannot be obtained by full-fledgedo relational systems either.