PRES LUNAM
Ecole Doctorale STIM
Sciences et Technologies de l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : LINA
Equipe : GRIM

Exploitation de clustering multiples sur un jeu de données

Dumonceaux, Frédéric
Mél : frederic.dumonceaux@univ-nantes.fr

Résumé :
Les *workflow* centrés sur les données (*data-centric*) doivent pouvoir prendre en charge des résultats d'analyses, considérés comme des données primaires en termes de fonctions de gestion de données. Nous nous focalisons sur le cas où ces résultats sont des partitions d'individus dans des agrégats distincts. L'agrégation d'un jeu de données en plusieurs groupes (ou classes) distincts est depuis longtemps une tâche centrale en fouille de données et révèle plusieurs aspects sur la structure du processus ayant mené à leur création. On considère en particulier les problèmes de partitionnement non-supervisé (*unsupervised clustering*) où aucune caractérisation sur les classes et leur nombre n'est connue *a priori* de l'analyste. Nous considérons que l'usage de tels résultats doit se faire à l'aide d'un système de gestion de partitions dédiés et nécessite donc l'étude des propriétés algébriques intrinsèques aux opérations applicables sur des partitions d'ensembles et permettant de modéliser par le biais de formules des problèmes en rapport avec la tâche de partitionnement de données.

Mots clés : *Ensemble Clustering, Partition Lattice, Data Models*

1 Contexte général

Les *workflow* centrés sur les données (*data-centric*) doivent pouvoir prendre en charge des résultats d'analyses, considérés comme des données primaires en termes de fonctions de gestion de données. Nous nous focalisons dans le cas où ces résultats sont des partitionnement dans des groupes distincts d'individus appartenant à des jeux de données. Ce besoin émerge naturellement du partitionnement de données ou *data clustering* dans l'analyse exploratoire de données scientifiques, où un nombre grandissant d'importants jeux de données et de résultats d'analyses doivent être traités.

L'agrégation d'un jeu de données en plusieurs groupes (ou classes) distincts est depuis longtemps une tâche centrale en fouille de données. La principale raison est que le partitionnement révèle plusieurs aspects sur la structure du processus ayant mené à leur création (phénomène physique, social, etc.), en séparant des sous-populations d'individus. Nous nous concentrons en particulier sur les problèmes de partitionnement non-supervisé (*unsupervised clustering*) où aucune caractérisation sur les classes et leur nombre n'est connue *a priori* de l'analyste. L'utilisation d'algorithmes de partitionnement peut être simplement modélisé : un algorithme est appelé sur un certain jeu de données, il renvoie alors une partition des individus qui sera exploité par un utilisateur ou un expert selon ses besoins. On soulignera également qu'il existe de nombreuses raisons pour lesquelles un logiciel d'analyse exploratoire puisse gérer plusieurs résultats issus d'une ou plusieurs opérations de partitionnements.

En effet, des douzaines d'algorithmes ont été proposés ces trente dernières années, effectuant diverses hypothèses sur la structure des données sous-jacentes et optimisant certains critères (parfois localement) et produisant souvent des partitions différentes. Il apparaît qu'il n'existe pas une méthode de partitionnement idéale, sa conception demeure donc un défi dans le cas général. Estimer la qualité d'une partition parmi un ensemble de résultats est également un problème ouvert et largement dépendant de l'application ciblée en règle générale. Par exemple, un jeu de données met parfois en évidence une structure naturellement hiérarchique et plusieurs partitions sont alors nécessaires pour établir le comportement « naturel » des données à se regrouper.

En outre, des applications en fouille de données peuvent elles-mêmes considérer en entrée de multiples partitions comme des instances, en plus des jeux de données eux-mêmes. Un tel schéma revient alors à considérer un ensemble de partitions (*ensemble clustering*), récemment discuté dans [1], de sorte à construire une partition qui va mieux révéler la structure des données que des partitions prises individuellement, en cherchant à combiner celles-ci. La technique décrite dans [2], soit la recherche d'une

1. Informatique

combinaison optimale dépend de la stabilité des affectations de paires d'individus à une classe unique ou *cluster* à travers l'ensemble de toutes les partitions. Dans [3], la définition d'une fonction de consensus est basée sur l'information mutuelle partagée au sein des partitions, ce qui requiert l'identification des affectations communes de paires d'individus pour un couple de classes donnée. Une autre fonction d'optimalité est la partition médiane et discutée dans [4]. Des résultats expérimentaux indiquent que la meilleur fonction de consensus n'est en réalité pas la même pour tous les jeux de données. La mise en œuvre de telles techniques nécessitent souvent des opérations élémentaires ou non, sur les partitions, comme dans le cas susmentionné des affectations sur des paires d'individus.

Dans certains domaines applicatifs, on cherchera à regrouper les données selon un sous-ensemble de caractéristiques et formant une partition de celles-ci. Cette tâche de classification double (*biclustering* ou *co-clustering*) est souvent employée en biologie ou encore en traitement d'images où les données peuvent être décrites sur deux dimensions. Par exemple, un bioinformaticien cherchera à regrouper des gènes selon leur expressivité, ce qui nécessiter de considérer ces derniers sur des caractéristiques particulières et mettre ainsi en évidence d'autres motifs, cela dépendant de la nature des données considérées.

Enfin, il y a actuellement un fort décalage entre l'expérimentation manuelle sur de petits jeux de données et sur les jeux de données accessibles en ligne par le biais de services et de dépôts collaboratifs. Ces derniers peuvent stocker de très importants jeux de données, chacun pouvant être associé avec plusieurs partitions les décrivant. En particulier, l'analyse scientifique à grande échelle sur des jeux de données partagés [5] gagne en efficacité par l'exploitation d'infrastructures partagées pour la coopération voire la compétition, et tire autant que possible avantage des résultats d'autrui. Cela devrait réduire considérablement l'ampleur de la tâche, les utilisations propres à la fraude ou à produire des erreurs sur les jeux de données (concernant également les résultats sous formes de partitions). Un tel cadre est à même de produire beaucoup de partitions sur un jeu de données partagé et produit par différents utilisateurs. Il peut arriver également que certains jeux de données dans un dépôt partagent des individus voire des variables, ou partageant des connexions entre eux. En physique ou dans les sciences sociales, dès qu'une nouvelle variable devient observable, il devient intéressant d'estimer quel est son incidence sur la manière dont sont classés les individus.

À travers ces cas, nous avons essayé de démontrer qu'il y a un fort besoin pour un système de gestion de données capable de manipuler des partitions sur des jeux de données à l'aide d'un modèle cohérent, de préférence algébrique, et dans une veine similaire à celle utilisée pour des données relationnelles. Bien que nous nous focalisions sur des partitions, le but principal reste conforme pour un éventail plus large de structures de données [6, 7].

2 Sur les partitions d'ensembles

Nos travaux se sont concentrés sur le problème de la conception d'un modèle logique de données dans l'optique de représenter des partitions d'ensembles afin de les manipuler à l'aide d'opérateurs dédiés. La clé du problème est ainsi de déterminer quelles sont les propriétés intrinsèques des partitions d'ensembles et offrant des capacités d'optimisations afin de déduire un modèle de calcul formel plus efficace en regard des techniques utilisées jusqu'à présent.

Il n'est donc pas surprenant de se focaliser sur la structure algébrique du treillis des partitions Π_Ω [8] qui représente l'ensemble des partitions sur un ensemble qui est un ensemble d'objets ne possédant elle-même pas de structure particulière et est considéré indépendamment d'un domaine d'application particulier. Cette algèbre repose sur une structure relationelle figurant un ordre partiel entre les différentes partitions et représenté graphiquement comme un espace réticulé. Cette relation permettant de comparer les partitions entre-elles est la relation de raffinement ($\preceq$) et est vérifiée quand chaque partie (*i.e.* classes ou *clusters*) d'une partition est incluse ($\subseteq$) dans la partie d'une autre partition. N'étant pas totalement ordonnée, il existe donc des éléments naturellement " incomparables " qui sont donc le cœur du problème et permet de nourrir de multiples interprétations selon les contextes applicatifs et donc des hypothèses faites sur la nature du résultat attendu.

Formellement, cette relation induit deux lois de compositions internes, soit les opérateurs naturels d'un treillis ($\vee, \wedge$) et induit les identités habituelles qui apparaissent basiquement au sein de toutes les structures. Pourtant, au regard des structures latticielles largement étudiées et diffusées dans la communauté scientifique comme l'algèbre de Boole, il ne détient pas naturellement de propriétés sur la composition de ses opérateurs et manque d'équivalences algébriques bien que cela n'implique pas de prime abord l'existence d'un système de réécriture de formules.

En particulier, il n'est pas distributif comme c'est le cas par exemple du corps des réelles $(\mathbb{R}, +, \times)$ et cela

a des conséquences sur les représentations alternatives utilisables et sur l'emploi de résultats permettant d'interpréter de telles structures comme des espaces topologiques (*i.e.* dualité de Marshall Stone) et qui ont des applications directes dans d'autres domaines de l'informatique théorique et appliquée.

Au-delà, il n'est pas surprenant qu'il existe une relation forte entre le concept de base d'un espace vectoriel utilisé en algèbre et le choix d'une représentation explicite pour des instances d'un très grand nombre de types de données. Une base quelconque devrait en outre permettre de représenter des objets atomiques (non-exprimables à partir d'autres) indépendamment d'une réalisation géométrique particulière de sorte à ne considérer que les interactions formelles de nature combinatoire.

Concernant les partitions d'ensembles, il est évident que celles-ci exposent deux niveaux d'imbrications et sont donc dotées de trois couches d'abstraction. Le support est simplement un ensemble d'entités pour lesquelles, une ou plusieurs sont agrégées afin de souligner les relations en cours. Quand il n'est pas pertinent de pondérer ces relations pour les différencier, cela concrétise nettement un assouplissement des contraintes sur le maintien de la cohérence des relations entre les entités au sein d'un cluster. La partition en elle-même ferme le ban et est considérée comme un tout, c'est-à-dire qu'elle représente notre objet de base au sein d'un système de gestion de bases de données (SGBD).

3 Travaux actuelles

Jusqu'à présent, nos efforts ont été concentrés sur la conception d'une représentation relationelle d'une partition d'ensemble (*i.e.* sous forme tabulaire) pour laquelle nous sommes en mesure d'implémenter un certain nombre d'opérateurs entre des partitions en employant des requêtes en algèbre relationelle et en *Datalog* et quelques méthodes d'optimisations par le biais de caractéristiques propres au langage SQL. Par ailleurs, les résultats obtenus en terme de performances sont largement décevants et ce cadre ne peut prétendre à être utilisé à plus grande échelle tandis que le nombre d'objets partitionnés augmente.

De plus, le moteur d'évaluation de requêtes ne sachant pas la présence de partitions d'ensembles ignorent donc certaines propriétés naturelles, et cela même dans des cas triviaux (*i.e.* chaque élément est *idempotent*) et justifie la conception d'un système complet *ad hoc* afin de libérer le potentiel de telles propriétés intrinsèques aux partitions d'ensembles.

Conjointement, nous élaborons une algèbre de partitions par le biais d'additions de nouveaux opérateurs à la structure de base et incluant des propriétés permettant de comparer ou d'analyser de manière plus fines des partitions et de permettre l'établissement de formules sur celles-ci et la modélisation de problèmes sous-jacents au partitionnement par consensus.

Références

[1] H.-P. Kriegel and A. Zimek. Subspace clustering, ensemble clustering, alternative clustering, multiview clustering : What can we learn from each other ? In *Proc. 1st Int. Wksp MultiClust 2010 w/ 16th ACM SIGKDD Conf. KDD 2010*, Washington, DC, USA, 2010.

[2] Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6) :835–850, 2005.

[3] Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3 :583–617, December 2002.

[4] Alexander Topchy, Anil K. Jain, and William Punch. Combining multiple weak clusterings. In *IEEE International Conference on Data Mining (ICDM'2003)*, volume 0, page 331, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[5] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. *The Fourth Paradigm : Data-Intensive Scientific Discovery.* Microsoft Research, 2009.

[6] Toon Calders, Laks V. S. Lakshmanan, Raymond T. Ng, and Jan Paredaens. Expressive power of an algebra for data mining. *ACM Trans. Database Syst.*, 31 :1169–1214, December 2006.

[7] Eduardo Ogasawara, Daniel De Oliveira, Patrick Valduriez, Daniel Dias, Fabio Porto, and Marta Mattoso. An algebraic approach for data-centric scientific workflows. In *Proceedings of VLDB*, volume 4, pages 1328–1339. VLDB Endowment, 2011.

[8] Oystein Ore. Theory of equivalence relations. *Duke Math. J.*, 9 :573–627, 1942.