Partitionnement sous contrainte de similarité

Quentin Haenn, Brice Chardin, Mickaël Baron

Partitionnement sous contrainte de similarité

Quentin Haenn
quentin.haenn@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Brice Chardin
brice.chardin@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Mickaël Baron
mickael.baron@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

RÉSUMÉ

Dans un contexte de réduction de dimensionnalité sur ses données de simulation, le gestionnaire d'un réseau de distribution d'électricité a besoin de modéliser des comportements typiques des consommateurs et des producteurs desservis. Pour cela, les critères de regroupement ou de séparation doivent être explicites et expliquables. Un des enjeux principaux est de borner l'erreur de représentation lorsqu'un élément est assimilé à un représentant type. Dans ce cadre, nous cherchons à évaluer l'intérêt d'utiliser des techniques de partitionnement de données sous contraintes de similarités, notamment avec des contraintes dites de diamètre et de rayon. Nous avons ainsi proposé et évalué des méthodes de partitionnement basées sur des algorithmes de la théorie des graphes.

MOTS CLÉS

Partitionnement de données, Contraintes de similarités, Contraintes de diamètre, Contraintes de rayon, Ensembles dominants minimaux

1 CONTEXTE INDUSTRIEL

L'intérêt principal du partitionnement de données est de pouvoir découvrir des liens sous-jacents entre les données [13]. Ces techniques sont utilisées dans de nombreux domaines, comme la biologie, la finance ou les assurances [2, 7, 13, 15]. Elles constituent non seulement un outil essentiel pour l'analyse de données, mais aussi un champ de recherche très actif. Un des grands objectifs de ces techniques de partitionnement est de pouvoir faire ressortir des représentants des différents groupes, afin de réduire le nombre de points à analyser [11]. Cependant, la majorité des algorithmes ne garantissent pas que les représentants des groupes respectent une certaine proximité entre eux et les données du groupe en question.

Le cas d'étude industriel concerne un simulateur d'état du réseau électrique. Grâce à ce simulateur, le gestionnaire du réseau peut obtenir des états projetés du réseau, en fonction des comportements des consommateurs et des producteurs. Néanmoins, compte tenu de la complexité du réseau et du nombre d'acteurs en jeu, il faut pouvoir modéliser ces comportements de manière simplifiée lors de la génération de scénarios pour le simulateur. Pour cela, les consommateurs et les producteurs sont regroupés en fonction de critères de similarité, afin de pouvoir les assimiler à un élément type. La similarité employée est, par exemple, la différence de perte en ligne par effet Joule suite au remplacement d'un consommateur ou producteur par un autre. Il devient alors nécessaire de borner l'erreur de représentation pour obtenir des garanties sur les résultats de la simulation et respecter les contraintes strictes d'exploitation du réseau.

40ème conférence sur la gestion des données (BDA), October 21-24, 2024, Orléans, France 2024.

2 PARTITIONNEMENT SOUS CONTRAINTE DE SIMILARITÉ

Pour répondre à ce besoin, les approches de partitionnement sous contrainte permettent notamment de garantir une certaine proximité entre les éléments d'un groupe. Deux tendances principales se dégagent dans la littérature : les contraintes sur les instances et les contraintes sur les groupes. Les contraintes sur les instances permettent de spécifier que deux points doivent appartenir au même groupe – « must-link »– ou à des groupes différents – « cannot-link ». Les contraintes sur les groupes sont des contraintes globales qui s'appliquent à l'ensemble des points d'un groupe. Ces contraintes sont équivalentes à généraliser et automatiser l'application de contraintes cannot-link sur les instances trop dissimilaires.

Cet article s'intéresse à la seconde catégorie. En effet, les contraintes globales introduisent une limite de dissimilarité entre les points d'un même groupe, bornant de fait l'erreur maximale admissible entre le représentant choisi et les points du groupe. Deux types de contraintes de similarité sont couramment utilisées : les contraintes de diamètre et les contraintes de rayon.

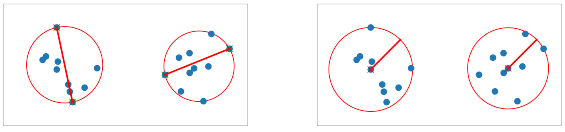
Les contraintes de diamètre imposent que la dissimilarité maximale entre les points d'un groupe soit inférieure à un seuil donné. Les contraintes de rayon, quant à elles, exigent que la dissimilarité entre chaque point du groupe et le représentant du groupe soit inférieure à un seuil donné. Ces contraintes peuvent être formulées de la manière suivante, avec C un ensemble de points (un groupe, ou *cluster*), $\mathcal{P}$ l'ensemble des groupes formant la partition, d une mesure de dissimilarité et T le seuil exigé :

- Contrainte de diamètre : $\forall C \in \mathcal{P}, \max_{a,b \in C} d(a,b) \leq T$
- Contrainte de rayon : $\forall C \in \mathcal{P}, \min_{a \in C} \max_{b \in C} d(a,b) \leq T$

Ces contraintes et les groupes résultants sont illustrés sur la Figure 1, pour un exemple de données à deux dimensions.

Une contrainte supplémentaire est ajoutée pour minimiser le nombre de groupes obtenus, afin d'éviter des solutions triviales où chaque point est dans son propre groupe. Cette minimisation peut-être optimale ou approchée. D'un point de vue pratique, réduire le nombre de groupes se justifie aussi par le fait que l'on cherche précisément à réduire la dimensionnalité du problème.

Hubert [12] a énoncé que le partitionnement optimal sous contrainte de diamètre était équivalent à colorer un graphe avec un



(a) Contrainte de diamètre **(b) Contrainte de rayon**

Figure 1 : Exemple de contraintes de similarité classiques

Table 1 : Principaux résultats expérimentaux

Dataset (#Classes)	MDS-APPROX			MDS-EXACT			EQW-LP			PROTOCLUST		
	#Clusters	R	Temps (s)	#Clusters	R	Temps (s)	#Clusters	R	Temps (s)	#Clusters	R	Temps (s)
Iris (3)	**3**	**1.43**	0.062 ± 0.01	**3**	**1.43**	**0.009 ± 0.00**	**3**	**1.43**	0.018 ± 0.01	4	1.24	0.026 ± 0.00
Wine (3)	4	220.05	0.029 ± 0.00	**3**	**232.08**	**0.010 ± 0.00**	**3**	**232.08**	0.014 ± 0.00	4	181.35	0.034 ± 0.00
Glass Identification (6)	7	**3.94**	**0.015 ± 0.00**	**6**	**3.94**	0.020 ± 0.00	**6**	**3.94**	0.026 ± 0.00	7	3.31	0.046 ± 0.00
Ionosphere (2)	**2**	**5.45**	**0.078 ± 0.01**	**2**	**5.45**	2.640 ± 0.05	**2**	**5.45**	0.104 ± 0.00	5	5.35	0.12 ± 0.00
WDBC (2)	**2**	**1197.42**	0.315 ± 0.01	**2**	**1197.42**	**0.138 ± 0.00**	**2**	**1197.42**	0.197 ± 0.01	3	907.10	0.402 ± 0.00
Synthetic Control (6)	8	66.59	0.35 ± 0.03	**6**	**70.11**	**0.036 ± 0.00**	**6**	**70.11**	0.143 ± 0.01	8	68.27	0.489 ± 0.00
Vehicle (4)	5	150.87	0.955 ± 0.04	**4**	**155.05**	**0.185 ± 0.00**	**4**	**155.05**	0.526 ± 0.01	6	120.97	0.830 ± 0.01
Yeast (10)	**10**	**0.423**	**2.361 ± 0.03**	**10**	**0.423**	622.87 ± 0.30	**10**	**0.423**	6.718 ± 0.02	13	0.419	2.374 ± 0.08
Ozone (2)	3	235.77	49.82 ± 1.18	**2**	**245.58**	1350.86 ± 1.5	**2**	**245.58**	26.86 ± 0.63	3	194.89	**15.32 ± 0.15**
Waveform (3)	**3**	**10.73**	**48.01 ± 0.39**	**3**	**10.73**	5559.9 ± 15.3	**3**	**10.73**	233.9 ± 1.45	6	10.47	61.27 ± 0.08

nombre minimal de couleurs. De même, le problème de rayon est équivalent à un problème de recherche d'un ensemble dominant de cardinalité minimale dans un graphe. Ces deux problèmes sont NP-difficiles.

Les problèmes de partitionnement sous contrainte de diamètre ayant été largement étudiés dans la littérature [1, 5, 6, 8, 11], nous avons choisi de nous concentrer sur les problèmes de partitionnement sous contrainte de rayon.

L'objectif principal de ce travail consiste à identifier ou développer des algorithmes de partitionnement sous contrainte de similarité et d'en évaluer l'intérêt pratique sur des données réelles d'exploitation du réseau électrique.

3 TRAVAUX ET PREMIERS RÉSULTATS

Nous avons élaboré un algorithme de partitionnement sous contrainte de rayon basé sur la recherche d'un ensemble dominant minimal dans un graphe. Deux implémentations ont été évaluées [9, 10], en fonction de l'algorithme de domination employé : une version optimale, MDS-EXACT, basée sur l'algorithme EMOS [14], et une version approchée, MDS-APPROX, basée sur l'algorithme IG [4].

Ces deux solutions sont comparées à deux algorithmes de partitionnement sous contrainte de rayon de l'état de l'art : une résolution exacte à l'aide d'un programme linéaire en nombres entiers, EQW-LP [1], et un algorithme de classification ascendante hiérarchique avec une agrégation minimax, Protoclust [3]. Le protocole d'évaluation est repris de l'état de l'art [1, 5]. Les données d'entrée de ces quatre algorithmes sont identiques, elles consistent en une matrice de dissimilarité pré-calculée et une contrainte globale de dissimilarité, c'est-à-dire un rayon maximal. Ce rayon maximal dépend du jeu de données, et a été déterminé au préalable de manière à obtenir autant de groupes que de classes réelles. Les mesures de qualité retenues sont le nombre de groupes et leur compacité, représentée par le rayon maximal R des groupes.

La Table 1 présente les premiers résultats expérimentaux. Ces résultats montrent que nos algorithmes sont capables de fournir des solutions de qualité en termes d'optimalité et de compacité de la partition. MDS-APPROX permet en particulier d'obtenir de meilleures approximations que Protoclust, dans des délais raisonnables. Par leur caractère exact, MDS-EXACT et EQW-LP fournissent tous deux des résultats similaires. Néanmoins, MDS-EXACT présente des temps de traitement plus importants pour les jeux de données de grandes taille, avec jusqu'à 5560 secondes pour Waveform contre 234 secondes pour EQW-LP.

Ainsi, en fonction du jeu de données à partitionner, notre approche peut s'avérer compétitive et pourrait être une alternative intéressante aux algorithmes de l'état de l'art.

Nos travaux futurs visent à étendre MDS-EXACT avec une version itérative inspirée de Clustergraph [11], afin de déterminer automatiquement le rayon maximal à utiliser comme contrainte.

RÉFÉRENCES

[1] Jennie Andersen, Brice Chardin, and Mohamed Tribak. 2021. Clustering to the Fewest Clusters Under Intra-Cluster Dissimilarity Constraints. In *ICTAI*. https://doi.org/10.1109/ICTAI52525.2021.00036

[2] Sio Iong Ao, Kevin Yip, Michael Ng, David Cheung, Pui-Yee Fong, Ian Melhado, and Pak C. Sham. 2005. CLUSTAG : hierarchical clustering and graph methods for selecting tag SNPs. *Bioinformatics* 21, 8 (April 2005), 1735–1736. https://doi.org/10.1093/bioinformatics/bti201

[3] Jacob Bien and Robert Tibshirani. 2011. Hierarchical Clustering With Prototypes via Minimax Linkage. *J. Amer. Statist. Assoc.* 106, 495 (Sept. 2011), 1075–1084. https://doi.org/10.1198/jasa.2011.tm10183

[4] Alejandra Casado, Sergio Bermudo, Ana D. López-Sánchez, and Jesús Sánchez-Oro. 2023. An iterated greedy algorithm for finding the minimum dominating set in graphs. *Mathematics and Computers in Simulation* 207 (May 2023), 41–58. https://doi.org/10.1016/j.matcom.2022.12.018

[5] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. 2017. Constrained clustering by constraint programming. *Artificial Intelligence* 244 (March 2017), 70–94. https://doi.org/10.1016/j.artint.2015.05.006

[6] Derya Dinler and Mustafa Kemal Tural. 2016. A Survey of Constrained Clustering. In *Unsupervised Learning Algorithms*. https://doi.org/10.1007/978-3-319-24211-8_9

[7] Zhenyu Gao, Styliani I. Kampezidou, Ameya Behere, Tejas G. Puranik, Dushyanth Rajaram, and Dimitri N. Mavris. 2022. Multi-level aircraft feature representation and selection for aviation environmental impact analysis. *Transportation Research Part C : Emerging Technologies* 143 (Oct. 2022), 103824. https://doi.org/10.1016/j.trc.2022.103824

[8] Allan D. Gordon. 1996. A survey of constrained classification. *Computational Statistics & Data Analysis* 21, 1 (Jan. 1996), 17–29. https://doi.org/10.1016/0167-9473(95)00005-4

[9] Quentin Haenn, Brice Chardin, and Mickaël Baron. 2024. Clustering Under Radius Constraints Using Minimum Dominating Sets. In *27th International Symposium on Methodologies for Intelligent Systems*. Springer, Poitiers, France. https://hal.science/hal-04533921

[10] Quentin Haenn, Brice Chardin, and Mickaël Baron. 2024. MDS Clustering Experiments. *Source Code repository* (2024). https://forge.lias-lab.fr/mds_clustering.

[11] Pierre Hansen and Michel Delattre. 1978. Complete-Link Cluster Analysis by Graph Coloring. *J. Amer. Statist. Assoc.* 73, 362 (June 1978), 397–403. https://doi.org/10.1080/01621459.1978.10481589

[12] Lawrence J. Hubert. 1974. Some applications of graph theory to clustering. *Psychometrika* 39, 3 (Sept. 1974), 283–309. https://doi.org/10.1007/BF02291704

[13] Anil K. Jain Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data clustering : a review. *Comput. Surveys* 31, 3 (Sept. 1999), 264–323. https://doi.org/10.1145/331499.331504

[14] Hua Jiang and Zhifei Zheng. 2023. An Exact Algorithm for the Minimum Dominating Set Problem. International Joint Conferences on Artificial Intelligence. https://doi.org/10.24963/ijcai.2023/622

[15] Yixian Liu, Ramteen Sioshansi, and Antonio J. Conejo. 2018. Hierarchical Clustering to Find Representative Operating Periods for Capacity-Expansion Modeling. *IEEE TPWS* (May 2018). https://doi.org/10.1109/TPWRS.2017.2746379