Algorithmes à base de provenance pour des requêtes enrichies sur les bases de données graphes

Yann Ramusat
DI ENS, ENS, CNRS,
Université PSL & Inria
Paris, France
yann.ramusat@ens.fr

Silviu Maniu
Université Paris-Saclay, LRI, CNRS
& Inria
Paris, France
silviu.maniu@lri.fr

Pierre Senellart
DI ENS, ENS, CNRS,
Université PSL & Inria & IUF
Paris, France
pierre@senellart.com

Les bases de données orientées graphe [13] font partie de l'écosystème des SGBD appelés NoSQL, dans lesquels l'information n'est pas organisée en suivant strictement le modèle relationnel. La structure des bases de données graphe est bien adaptée à la représentation de certains types de relations dans les données et leur potentiel pour la distribution les rendent attractives pour des applications nécessitant du stockage à grande échelle et du traitement de données massivement parallèle. Des exemples d'applications naturelles de tels systèmes de bases de données sont l'analyse des réseaux sociaux [5] ou le stockage et l'interrogation du Web sémantique [2].

Les bases de données graphe peuvent être interrogées en utilisant plusieurs langages de requêtes généraux de navigation, dont une abstraction est les *requêtes régulières de chemin* (*regular path queries* ou *RPQ* en anglais) [3] (ou des généralisations de celles-ci, comme les C2RPQ), sur les chemins du graphe. Récemment, en s'appuyant sur les solutions existantes pour l'interrogation des graphes à propriétés – comme le langage Cypher [6] de Neo4j ou PGQL [15] d'Oracle – une future norme internationale pour l'interrogation de graphes à propriétés, GQL [9], est en cours d'élaboration en tant que langage de requête à part entière au côté de SQL. GQL inclura notamment un support des RPQ.

En parallèle de ces développements récents, la notion de *provenance* d'un résultat de requête [14], une notion familière dans les bases de données relationnelles, a récemment été adaptée au contexte des bases de données graphe [11], en utilisant le cadre des semi-anneaux de provenance [7]. Dans ce cadre, les arêtes d'un graphe sont annotées, en plus des propriétés usuelles, par des éléments d'un semi-anneau ; quand une requête est évaluée, le fait de traverser les chemins du graphe peut engendrer de nouvelles annotations qui dépendent des opérateurs du semi-anneau, et qui résultent en une valeur du semi-anneau associée à chaque résultat de la requête, appelée la provenance du résultat. En choisissant différents semi-anneaux, des informations différentes sur le résultat de la requête peuvent être calculées. Par exemple, quand les arêtes sont annotées avec des éléments du semi-anneau *tropical* (les nombres réels positifs ou nuls) exprimant la distance entre les nœuds, la provenance du résultat calcule la plus courte distance des chemins qui ont produit ce résultat ; quand les arêtes sont annotées par des éléments du semi-anneau de *comptage* (les entiers naturels) interprétés comme une multiplicité, la provenance du résultat calcule le nombre (qui peut être infini en cas de cycles) de manière dont chaque résultat peut être obtenu. Les propriétés sous-jacentes du

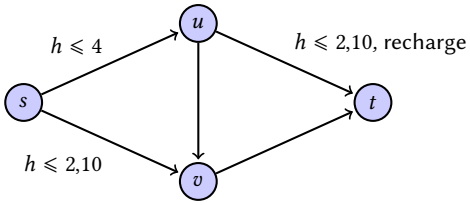

Figure 1: Réseau routier exemple représenté par un graphe avec des annotations de provenance selon deux dimensions : la hauteur h maximale (un nombre positif) qu'un véhicule doit avoir pour utiliser le segment de route, et un booléen indiquant la présence d'une station de recharge pour véhicule électrique. Quand une dimension n'est pas mentionnée, les annotations sont supposées être, respectivement, $h \leqslant \infty$ et $\neg(\text{recharge})$.

semi-anneau contrôlent directement la manière dont l'information sur les arêtes du graphe est encodée et également l'efficacité des algorithmes de traitement des requêtes.

Au-delà de ces exemples simples de semi-anneaux, le cadre de la provenance par semi-anneau permet aussi de modéliser des problèmes complexes, p. ex., où le problème d'intérêt peux être décomposé en plusieurs sous-problèmes et où la provenance du résultat ne correspond pas nécessairement à un chemin particulier dans le graphe.

Exemple 1. *Considérons l'exemple d'un réseau de transport routier modélisé comme un graphe orienté avec des annotations de provenance sur les arêtes. On peut par exemple encoder la présence de points d'intérêts (tels que des stations essence, des restaurants ou des stations de recharge électrique) comme des caractéristiques booléennes des arêtes, et les propriétés des routes (p. ex., hauteur ou poids maximal pour un tunnel ou un pont) comme des caractéristiques à valeur réelle.*

Nous allons montrer que, en utilisant la provenance par semi-anneaux, nous pouvons traiter des requêtes de graphe qui prennent en compte une multiplicité de telles caractéristiques : une paire de nœuds est valide pour ces requêtes s'il existe au moins un chemin valide pour chaque restriction entre les deux emplacements. Une application de cela serait de s'assurer que différentes catégories de véhicules (disons, un camion de grand gabarit et une voiture électrique nécessitant une recharge sur le chemin) peuvent atteindre une destination commune à partir de la même origine.

Une autre sémantique possible pour la provenance par semi-anneaux est de vérifier que tous les chemins entre deux nœuds vérifient (ou excluent) certaines propriétés (p. ex., absence de péages ou présence de

stations essence sur la route) fournissant ainsi à des administrateurs des informations cruciales sur l'état global des itinéraires entre deux points.

Ceci est illustré en figure 1, un réseau routier dans lequel certains segments de routes ont des restrictions sur la hauteur des véhicules ; c'est une première dimension de provenance. La deuxième dimension indique s'il existe une station de recharge électrique sur le segment de route – dans notre exemple, ce n'est le cas que pour une seule arête.

Dans nos recherches préliminaires antérieures [11], nous avons généralisé trois algorithmes existants d'une large gamme de la littérature en informatique au calcul de la provenance de requêtes régulières de chemin, dans le cadre de provenance par semi-anneaux. Pris ensemble, ces trois généralisations recouvrent une grande classe de semi-anneaux utilisés pour la provenance, chacun conduisant à un compromis entre complexité en temps et généralité. Nous avons également conduit des expériences suggérant que ces approches sont complémentaires et applicables en pratique pour divers types d'indications de provenance, même sur des réseaux de transports relativement grands.

Dans les recherches résumées ici, et décrites en détail en [12], nous étendons ce travail en :

- Introduisant un nouvel algorithme, MultiDijkstra, pour les semi-anneaux commutatifs *0-clos* (ou *absorptifs*). Cet algorithme, qui généralise l'algorithme de Dijkstra et exploite les propriétés des treillis distributifs, comble partiellement un fossé entre deux classes de semi-anneaux qui était non traité dans nos recherches antérieures. Les requêtes de l'exemple 1 font partie de cette classe et ont fortement motivé notre intérêt pour développer de nouveaux algorithmes. Les expériences que nous avons conduites démontrent que notre nouvel algorithme passe à l'échelle de très grands réseaux contenant des dizaines de millions de nœuds, apportant une amélioration notable à l'état de l'art du calcul de provenance dans les bases de données graphe.
- Établissant un résumé précis, sous la forme d'une taxonomie, des algorithmes utilisés dans notre contexte, ainsi que de leur complexité et des propriétés attendues des semi-anneaux sous-jacents utilisés pour les annotations de provenance. Nous analysons également les similarités avec des classes de semi-anneaux utilisés soit pour le calcul de provenance de requêtes de l'algèbre relationnelle [8] soit pour celui de programmes Datalog [4].
- Accomplissant un ensemble complet d'expériences sur des données du monde réel démontrant le temps de calcul de la provenance sur des graphes, avec une grande de variété de semi-anneaux et de cas d'utilisations. Nous observons également que les paramètres de topologie du graphe, comme la *largeur d'arbre* [10] semblent avoir un impact plus important sur l'efficacité des algorithmes que des paramètres basés sur la distance tels que la *highway dimension* [1]. L'implémentation de tous les algorithmes que nous utilisons pour ces expériences est librement disponible sur https://bitbucket.org/smaniu/graph-provenance/src/master/.

Pour plus de détails sur ce travail, se référer à [12].

REMERCIEMENTS

Ce travail a été financé par le gouvernement français sous gestion de l'Agence Nationale de la Recherche comme partie du programme « Investissements d'avenir », référence ANR-19-P3IA-0001 (Institut 3IA PRAIRIE).

RÉFÉRENCES

[1] Ittai Abraham, Amos Fiat, Andrew V. Goldberg, and Renato Fonseca F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *SODA*, pages 782–793, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[2] Marcelo Arenas and Jorge Pérez. Querying semantic web data with sparql. In *PODS*, pages 305–316, New York, 2011.

[3] Pablo Barceló. Querying graph databases. In *PODS*, pages 175–188, New York, 2013. ACM.

[4] Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen. Circuits for Datalog Provenance. In *ICDT*, pages 201–212, 2014.

[5] Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *KDD*, pages 57–66, New York, 2001. ACM.

[6] Nadime Francis, Andrés Taylor, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, and Petra Selmer. Cypher : An evolving query language for property graphs. In *SIGMOD*, pages 1433–1445, 2018.

[7] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*, pages 31–40, New York, 2007. ACM.

[8] Todd J. Green and Val Tannen. The semiring framework for database provenance. In *PODS*, page 93–99, New York, NY, USA, 2017. Association for Computing Machinery.

[9] ISO SC32 / WG3. Graph Query Language GQL. https://www.gqlstandards.org/.

[10] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In *ICDT*, Lisbon, Portugal, 2019.

[11] Yann Ramusat, Silviu Maniu, and Pierre Senellart. Semiring provenance over graph databases. In *TaPP*, 2018.

[12] Yann Ramusat, Silviu Maniu, and Pierre Senellart. Provenance-based algorithms for rich queries over graph databases, 2021. À paraître.

[13] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. O'Reilly Media, 2013.

[14] Pierre Senellart. Provenance and probabilities in relational databases: From theory to practice. *SIGMOD Record*, 46(4), 2017.

[15] Oskar van Rest, Sungpack Hong, Jinha Kim, Xuming Meng, and Hassan Chafi. PGQL : A property graph query language. In *GRADES*, pages 7 :1–7 :6, New York, NY, USA, 2016. ACM.