Langages de requêtes interactifs pour l'exploration de données

Marie Le Guilly
Université de Lyon, Insa de Lyon, CNRS, LIRIS
Villeurbanne, France
Thèse commencée le 1er septembre 2017 sous la responsabilité de Jean-Marc Petit et Marian Scuturici

ABSTRACT

Dans le contexte actuel où l'on assiste à un déluge de données, trouver des manières de naviguer efficacement dans les bases de données s'avère un défi déterminant. En proposant des solutions basées sur des langages de requêtes connus tels que le SQL, et en empruntant des méthodes de l'apprentissage automatique, cette thèse à pour but de s'attaquer à ce défi sous l'angle du rapprochement entre bases de données et apprentissage automatique.

KEYWORDS

apprentissage automatique, SQL, bases de données

1 INTRODUCTION

Ces dernières années, le monde de la donnée à connu une véritable explosion, tant le volume que nous produisons, et que nous stockons, a augmenté. Tirer profit de ce déluge de données, que l'on désire souvent par *Big Data*, est l'un des principaux défi au cœur de la science des données actuelles. Ainsi, les systèmes de gestion de bases de données (SGBD), qui existent depuis plusieurs dizaines d'années, font actuellement face à de nouveaux défis : l'augmentation du volume de données stockées conduit à une modification des structures et paradigmes de stockage. Il n'est donc pas rare de voir des bases de données avec plusieurs centaines de tables, et/ou des tables avec plusieurs centaines d'attributs. Cela conduit à modifier la manière dont nous appréhendons les données, et donc la manière d'interroger les SGBD, notamment à travers les requêtes SQL. Au-delà de la simple consultation des données, les analystes cherchent généralement aussi à en tirer des connaissances supplémentaires, à déduire de nouvelles informations à partir des données brutes qui ont été enregistrées. C'est notamment pour cela que l'apprentissage automatique a connu un tel développement ces dernières années, car il permet justement de répondre à ce besoin d'aller au delà de la donnée en tant que telle, pour en tirer une valeur nouvelle.

Dans ce contexte, un des défis est de s'intéresser à des moyens de combiner à la fois la robustesse des SGBD à la puissance d'analyse de données des méthodes d'apprentissage automatique, pour aller plus loin dans l'exploration des volumes de données important qui sont désormais à notre disposition. Ainsi, la thèse présentée ici propose de s'intéresser à des langages de requêtes interactifs, c'est-à-dire impliquant un rôle actif de l'utilisateur, permettant d'explorer des bases de données, en s'appuyant sur des méthodes d'apprentissage automatique, et en utilisant des langages déjà existants tels que SQL. Après avoir explicité le but général de cette thèse, nous donnerons un exemple plus concret de solution, ainsi que des pistes de recherche envisagées pour la thèse.

2 CONTRIBUTIONS DE LA THÈSE

2.1 Objectif général

L'objectif général est de proposer des solutions permettant l'exploration interactive de gros volumes de données, en se basant sur des langages de requêtes tels que SQL (ou similaires), et à l'aide de techniques d'apprentissage automatique. Ces dernières doivent permettre d'extraire de la connaissance des données stockées dans les SGBD, et/ou d'y découvrir des motifs significatifs. Ainsi, cela permettrait à un analyste d'aller au delà de la donnée brute stockée dans le SGBD, en le guidant dans un processus lui permettant de faire et de tester des hypothèses sur les données en sa possession. À plus haut niveau, l'objectif est donc de combiner les forces des SGBD avec les possibilités offertes par l'apprentissage automatique, tout en fournissant une solution basée sur des éléments familiers pour les utilisateurs, tels que SQL. Il s'agit donc de sortir du paradigme actuel qui consiste souvent à extraire les données avant de les analyser dans un processus à part, et de combiner ces deux étapes pour faciliter la découverte de nouvelles connaissances liées aux données.

Cet objectif s'inscrit dans une tendance générale visant à rapprocher les bases de données et l'apprentissage, qui est un sujet existant depuis un certain temps, mais qui prend un nouvel essor avec le contexte actuel, et qui fait écho à [5]. On peut par exemple citer un article de Imielinski et Mannila, qui en 1996, argumentait en faveur de la nécessité d'un langage proche de SQL qui soit relatif à la fouille de données [7]. Cela peut également rappeler le travail de Luc de Raedt sur les bases de données inductives permettant de stocker à la fois données et relations [10]. Dans un autre registre, des travaux récents se sont intéressés à des manières d'aider un utilisateur à formuler ses requêtes pour parvenir aux données désirées, parmi lesquels on peut citer [2, 9, 11].

2.2 Complétion de requêtes SQL

Dans ce cadre général, un premier élément de solution plus spécifique a commencé à être développé, et nous ammené à définir un concept nouveau et prometteur, la complétion de requêtes SQL. La complétion est un processus relativement connu, notamment dans la recherche d'information avec les moteurs de recherche sur internet qui suggèrent automatique la suite des mots-clés lors des requêtes de leurs utilisateurs. Si l'on s'intéresse plus spécifiquement au cas du SQL, certains outils de complétion existent, mais il s'agit d'un cas limité que l'on pourrait qualifier de syntaxique. Ainsi, des outils avancés tels que *SQL Complete* de **Devart** [1] ou *SQL Prompt* de **Redgate** [2] se limitent à compléter les requêtes SQL de leurs utilisateur avec des éléments syntaxiques du langage SQL, ou des éléments du schéma de la base interrogée. Bien qu'utile, cette forme

1. https://www.devart.com/dbforge/sql/sqlcomplete/
2. http://www.red-gate.com/products/sql-development/sql-prompt/

de complétion ne permet que d'accélérer l'écriture de la requête en évitant à l'utilisateur de devoir rechercher des syntaxes ou d'avoir à les écrire en entier. En revanche, cette complétion ne s'intéresse pas à la requête en elle-même, ni aux données qu'elle est susceptible de renvoyer. Pourtant, l'écriture d'une requête en elle-même peut s'avérer difficile, non pas sytaxiquement, mais pour identifier la manière pertinente d'accéder aux données recherchées. Ainsi, il n'est pas rare que plus de temps soit consacré à l'écriture de la requête qu'à son traitement par le SGBD [8].

L'un de premiers axes de la thèse s'intéresse à la proposition d'une forme de complétion de requête SQL que l'on peut qualifier de sémantique, c'est à dire basée sur la formulation globale de la requête et sur les donnée qu'elle renvoie. Ce travail s'inscrit dans la continuité d'un projet de fin d'étude ayant eu lieu de février à juin 2017, qui a permis de faire émerger ce problème, d'apporter des premières définitions d'une telle complétion sémantique, et de proposer une solution décrite dans [6]. Ce travail a fait l'objet d'un dépôt de brevet relatif à la solution développée pour proposer des complétions de requêtes SQL [3].

L'idée générale et de proposer à l'utilisateur d'effectuer une première requête, qui peut être très générale (et par exemple renvoyer tout les tuples de la base), et de proposer des clauses supplémentaire *pertinentes* à rajouter à cette requête, chaque suggestion explorant un espace de données possiblement pertinent, et permettant de guider l'exploration des données de la base. Le critère de pertinence de la requête est un élément essentiel, dont la démonstration devra passer par une confrontation avec des développeurs SQL.

Le procédé, illustré sur la figure 1, se base sur l'enchainement de deux méthodes d'apprentissage automatique, mais dont l'utilisation est masqué à l'utilisateur, puisqu'il n'a qu'à donner le début d'une requête SQL pour en obtenir la complétion qui est directement utilisable par le SGBD. Il permet de proposer un type particulier de complétions dont la définition exacte est développé dans [6].

Outre la facilitation de l'écriture de la requête ; un tel procédé permet d'apporter des informations pertinentes à un analyste vis-à-vis de ces données, en lui indiquant des zones pouvant être pertinentes. Ainsi, en plus des résultats auxquelles elles mènent, les complétions contiennent de l'information dans leur syntaxe même, car les clauses ajoutées sont une manière d'expliciter des motifs identifiés par le clustering dans les données. Il s'agit de plus d'un bon exemple de la manière dont les domaines des bases de données et de l'apprentissage automatique peuvent collaborer pour tirer encore plus profit des données générées dans le contexte actuel du *Big Data*.

2.3 Extension à d'autres problématiques

La complétion de requête SQL telle que présentée ici ne constitue pas le seul sujet d'étude de la thèse, et plusieurs pistes de travail additionnels peuvent être envisagés. Outre le bien connu SQL, on pourra envisager d'étendre l'étude à d'autres langages tels que le *CQL* (continuous query language) pour les flux de données [1], ou au *RQL* [4] qui concerne les implications. De plus, on pourra s'intéresser à des familles de requêtes particulière, telles que les skyline query [3].

3. Brevet numéro FR1757682 déposé en France le 14 août 2017

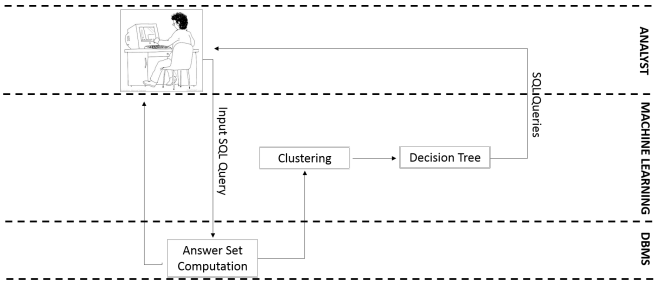

Figure 1: Processus de complétion d'une requête SQL

3 CONCLUSION

L'objectif principal de cette thèse est donc de proposer des solutions permettant d'explorer des données en se greffant à des langages de requête, en se basant sur des méthodes d'apprentissage automatique. Ces solutions ont pour but de mettre en lumière des faits nouveaux sur les données, et de permettre une participation active des analystes explorant les données. La complétion de requête SQL est une première piste prometteuse étudiée, ouvrant la voir à d'autres pistes de recherche potentielles.

RÉFÉRENCES

[1] Arvind Arasu, Shivnath Babu, and Jennifer Widom. 2006. The CQL continuous query language : semantic foundations and query execution. *The VLDB Journal—The International Journal on Very Large Data Bases* 15, 2 (2006), 121–142.
[2] Angela Bonifati, Radu Ciucanu, and Slawek Staworko. 2014. Interactive Join Query Inference with JIM. *Proc. VLDB Endow.* 7, 13 (Aug. 2014), 1541–1544. https://doi.org/10.14778/2733004.2733025
[3] S. Borzsony, D. Kossmann, and K. Stocker. 2001. The Skyline operator. In *Proceedings 17th International Conference on Data Engineering*. 421–430. https://doi.org/10.1109/ICDE.2001.914855
[4] Brice Chardin, Emmanuel Coquery, Marie Pailloux, and Jean-Marc Petit. 2014. RQL : A sql-like query language for discovering meaningful rules. In *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 1203–1206.
[5] Surajit Chaudhuri. 1998. Data mining and database systems : Where is the intersection ? (1998).
[6] Marie Le Guilly. 2017. SQL Query Completion. *Rapport de fin d'études, département informatique INSA Lyon* (2017).
[7] Tomasz Imielinski and Heikki Mannila. 1996. A Database Perspective on Knowledge Discovery. *Commun. ACM* 39, 11 (Nov. 1996), 58–64. https://doi.org/10.1145/240455.240472
[8] Arnab Nandi and H. V. Jagadish. 2011. Guided Interaction : Rethinking the Query-Result Paradigm. *PVLDB* 4, 12 (2011), 1466–1469. http://dblp.uni-trier.de/db/journals/pvldb/pvldb4.html#NandiJ11
[9] Olga Papaemmanouil, Yanlei Diao, Kyriaki Dimitriadou, and Liping Peng. 2016. Interactive Data Exploration via Machine Learning Models. *IEEE Data Eng. Bull.* 39, 4 (2016), 38–49. http://sites.computer.org/debull/A16dec/p38.pdf
[10] Luc De Raedt. 2002. A Perspective on Inductive Databases. *SIGKDD Explorations* 4 (2002), 69–77.
[11] Yanyan Shen, Kaushik Chakrabarti, Surajit Chaudhuri, Bolin Ding, and Lev Novik. 2014. Discovering Queries Based on Example Tuples. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14)*. ACM, New York, NY, USA, 493–504. https://doi.org/10.1145/2588555.2593664