Une Algèbre de Motifs pour l'Évaluation et l'Analyse de la Complétude des Données et l'Éxactitude des Requêtes

Fatma-Zohra Hannou, Bernd Amann, Mohamed-Amine Baazizi

Une Algèbre de Motifs pour l'Évaluation et l'Analyse de la Complétude des Données et l'Éxactitude des Requêtes

Fatma Zohra Hannou
Bernd Amann
Mohamed-Amine Baazizi
firstname.lastname@lip6.fr
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6
Paris, France

KEYWORDS

information completeness, data quality, pattern model

Introduction

L'incompletude des données est un problème majeur dans les applications collectant des données dans des environnements distribués, ouverts et peu fiables. Les réseaux de capteurs et l'intégration de données sont des exemples pertinents dans lesquels des données incomplètes sont naturellement dues à des défaillances matérielles ou logicielles, à des incompatibilités de données, à des autorisations d'accès aux données manquantes, etc. Dans toutes ces situations, l'interrogation et l'aggrégation de données incomplètes peuvent conduire à des réponses manquantes et incorrectes. Nous proposons une nouvelle approche où la complétude est évaluée par rapport à des données de référence et résumée à l'aide de motifs. Les motifs de complétude ont été utilisés dans [Mot89, RKNS15] pour annoter des données avec les connaissances disponibles sur la complétude et s'avèrent être une bonne représentation pour étudier la complétude des réponses aux requêtes. Nous montrons qu'il est possible de générer des ensembles de motifs automatiquement à partir de données de référence et d'utiliser ces ensembles pour raisonner sur la qualité des réponses aux requêtes sur des données incomplètes.

Travaux connexes

Le travail précurseur de [Mot89] suggère un modèle basé sur des méta-tables décrivant des contraintes de complétude et d'exactitude sur les données . Les méta-tables sont similaires à nos tables de motif, où les nuplets servent à définir les données disponibles, valides et non valides. La complétude d'une requête est vérifiée en montrant l'existance d'une réécriture de la requête qui utilise uniquement des vues complètes. Une autre idée de ces premiers travaux est la définition d'une algèbre manipulant des méta-tables pour produire des ensembles corrects (mais non complets) de méta-tuples satisfaits par une requête en entrée. Plus récemment, [RKNS15] présente une approche consistant à associer des motifs de complétude à des tables de données et une algèbre permettant d'interroger ces motifs afin de produire des informations sur l'exhaustivité des réponses. Nous adoptons la même approche consistant à utiliser des motifs et à définir une algèbre pour manipuler ces motifs. [LNRN14] analyse différents types d'anomalies de résultats partiels engendrés par des anomalies d'accès physique observées. Les motifs de complétude des données établissent une distinction entre les anomalies de cardinalité (incomplètes, fantômes, indéterminées) et de correction (crédibles et non crédibles) à différents niveaux de granularité (entrée, opérateur, colonne, partition) et permettent d'étudier comment ces anomalies se propagent dans un plan de requête. Nous suivons la même approche en ce qui concerne la propagation de complétude en utilisant des opérateurs à la granularité de la partition (jusqu'à des tuples individuels). Notre travail se situe également dans la lignée des *modèles de complétude relative* qui utilise des données de référence (maître) pour évaluer la complétude des données [FG10]. [SKL$^+$17] introduit *m-tables* (inspiré de *c-tables* [IL88]) pour représenter les informations de complétude et une algèbre pour annoter les réponses à une requête avec des informations de certitude. D'autres travaux existants traitent de l'explication des réponses manquantes [HHT09, HH10] ou des requêtes *why-not* [TC10, BHT15]. Ces travaux supposent que les données sont complètes et se concentrent sur l'analyse et la correction des requêtes erronnées.

Motifs de complétude

Prenons l'exemple suivant où Peter souhaite étudier la fiabilité d'un réseau de capteurs de de temperature et les éventuels problèmes de qualité des données liés aux défaillances des capteurs. Le réseau de capteurs observe les températures dans un certain nombre de pièces (stockées dans une table **LOC**) et produit une table **Energy** de mesures quotidiennes. Peter veut étudier la fiabilité de ce réseau et détecter toutes les valeurs manquantes sur une période de temps **CAL**. Comparer les données collectées **Energy** aux ensembles de données de référence **LOC** et **CAL** est réalisable pour de petites tables, mais devient rapidement impossible dans un contexte réel où la table **Energy** peut contenir des milliers, voire des millions de tuples. Pour faciliter l'analyse, Peter utilise une représentation compacte basée sur un motif de partitions incomplètes, comme indiqué dans le tableau $\mathbf{P_E}$ (Tableau 1). En examinant le motif p_0, Peter peut comprendre que toutes

Table 1: Tables de gabarit $\mathbf{P_E}$ et $\overline{\mathbf{P_E}}$

$\mathbf{P_E}$	fl	ro	we	da
p_0	*	*	w1	*
p_1	f2	*	*	*
p_2	f1	r1	*	Mon
...	...	...	...	...

$\overline{\mathbf{P_E}}$	fl	ro	we	da
$\overline{p_0}$	*	r2	w2	*
$\overline{p_2}$	f1	r1	*	Tue
...	...	...	...	...

les mesures sont disponibles pour la première semaine **w1** (la partition identifiée par p_0 est terminée). Le même argument est valable pour p_1, indiquant la «complétude» du deuxième étage **f2**. La table de configuration *couvre toutes* les partitions complètes de la table de données et, plus formellement, nous appelons $T = (\mathbf{Energy}, \mathbf{LOC} \times \mathbf{CAL})$ une *table contrainte* et $\mathbf{P_E}$ a *couverture stricte* de T. Cette représentation compacte permet à Peter non seulement de raisonner sur toutes les données disponibles, mais également de déduire des informations sur les mesures manquantes. Par exemple, puisque le motif $[f1, *, w2, Tue]$ n'est

couvert par aucun motif dans le tableau 1, il manque des mesures pour certaines pièces du premier étage mardi de la deuxième semaine. L'existence d'une table de référence permet également de générer le *complément* $\overline{\mathbf{P_E}}$ de toutes les partitions vides dans **Energy** (Table **??**). Ce tableau montre qu'aucune mesure n'est disponible pour la chambre $r2$ aux deux étages et pour les deux jours de la semaine $w2$.

Algèbre de motif

Une *table de motif* est une table relationnelle qui peut être interrogée à l'aide de l'algèbre relationnelle standard $\Omega = \{\sigma, \pi, \bowtie, \cup, \cap, -\}$ (et SQL). Nous étendons l'algèbre relationnelle avec deux opérateurs *unfold* et *fold* qui *spécialisent* et *généralisent* les motifs d'une table de motif P avec la garantie supplémentaire que le tableau résultant est une couverture stricte de P:

- L'opérateur *unfold* $\triangleleft_A(P, R)$ génère pour une table de motif P et la table de référence R un *équivalent* table de motifs $P' \equiv_R P$ où toutes les valeurs des attributs $a_i \in A$ sont des valeurs constantes. Notez également qu'un déploiement (unfold) complet sur tous les attributs génère D.
- L'opérateur fold $\triangleright_{a_i}$ est l'opérateur inverse de $\triangleleft_{a_i}$ et *généralise*, lorsque cela est possible, tous les sous-ensembles S qui peuvent être remplacé par un seul motif $p_{a_i:*}$ avec une valeur générique pour l'attribut a_i:

L'algèbre relationnelle étendue avec les opérateurs $fold$ et $unfold$ peut être utilisée pour définir un certain nombre d'opérateurs de niveau supérieur dans les tables de motifs:

- Génération de tables de motifs: Le pliage (fold) complet $\triangleright(P, R)$ de P génère une *couverture stricte* (et minimale) de T.
- Génération des tables de patterns pour les résultats de requête: Soit $T = (D, R)$ une table contrainte et $Q(D)$ une requête relationnelle standard sur la table de données D faisant uniquement référence aux attributs de référence de D. Il est possible d'obtenir la couverture stricte (minimale) $P^*(T')$ de la table contrainte $T' = (Q(D), Q(R))$ en repliant le résultat de query $Q(T)$ par rapport au résultat de la référence requête $Q(R)$: $P^*(T') = \triangleright(Q(D), Q(R))$ (voir les lignes pointillées rouges dans la figure 1). Une autre méthode consiste à réécrire $Q(D)$ dans un nouveau *motif de requête* $\widehat{Q}(P, R)$ sur une couverture stricte (pas nécessairement minimale) $P(T)$ de la table contrainte T telle que $\widehat{Q}$ prend en entrée le couple(P, R) et produit le nouveau couple $(P^*(T'), Q(R))$ (voir la ligne bleue continue dans la figure 1).

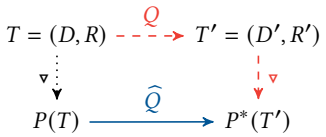

Figure 1: Requêtes de données et de signatures

- Projection sûre: Un autre problème lié à la complétude concerne les requêtes analytiques susceptibles d'agréger des informations sur des partitions incomplètes [Sho97]. Considérons, par exemple, la requête SQL avec la table de référence $R(fl, ro, we, da)$:

```
select fl, we, avg (khw) from Energy
 group by fl, nous
```

Comme indiqué précédemment, le motif de complétude du résultat peut être calculé en repliant le résultat de $\pi_{fl,we}(T) = (\pi_{fl,we}(D), \pi_{f,w}(R))$. Il est également facile de voir que des partitions incomplètes dans D conduisent à des résultats d'agrégation incorrects (par exemple, tous les résultats pour floor $f2$ et pour la semaine $w1$ sont corrects, alors que le résultat pour floor $f1$ est incorrect) . Des couvertures strictes permettent de filtrer des partitions complètes en fonction des attributs supprimés ro et da avant l'application de la projection: L'opérateur de *projection sûre* $\widehat{\pi}^*$ replie d'abord tous les motifs sur les attributs projetés et filtre toutes les dimensions incomplètes avant la projection. Cela garantit que le résultat ne contient que des motifs correspondant à des partitions complètes avec les attributs supprimés:

$$\widehat{\pi}^*_{\neg A_\pi}(P, R) = (\pi_{\neg A_\pi}(\sigma_{\theta_\pi}(\triangleright_{A_\pi}(P, R))), \pi_{\neg A_\pi}(R)) \quad (1)$$

où A_π indique les attributs supprimés par une projection et $\theta_\pi = \bigwedge_{a_i \in A_\pi}(a_i = *)$ filtre tous les motifs incomplets pour les attributs A_π.

La version complète de cet article fournit plus de détails sur la mise en œuvre et l'optimisation de cette algèbre de motif, y compris deux algorithmes efficaces pour l'opérateur fold $\triangleright$ (l'opérateur unfold peut être implémenté en SQL). La version étendu présente également des résultats expérimentaux sur les performances de la solution.

REFERENCES

[BHT15] Nicole Bidoit, Melanie Herschel, and Aikaterini Tzompanaki. Efficient computation of polynomial explanations of why-not questions. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 713–722. ACM, 2015.

[FG10] Wenfei Fan and Floris Geerts. Relative Information Completeness. *ACM Trans. Database Syst.*, 35(4):27:1–27:44, October 2010.

[HH10] Melanie Herschel and Mauricio A Hernández. Explaining missing answers to spjua queries. *Proceedings of the VLDB Endowment*, 3(1-2):185–196, 2010.

[HHT09] Melanie Herschel, Mauricio A Hernández, and Wang-Chiew Tan. Artemis: A system for analyzing missing answers. *Proceedings of the VLDB Endowment*, 2(2):1550–1553, 2009.

[IL88] Tomasz Imieliński and Witold Lipski. Incomplete information in relational databases. In *Readings in Artificial Intelligence and Databases*, pages 342–360. Elsevier, 1988.

[LNRN14] Willis Lang, Rimma V. Nehme, Eric Robinson, and Jeffrey F. Naughton. Partial results in database systems. In *International Conference on Management of Data, SIGMOD*, pages 1275–1286, Snowbird, USA, June 2014.

[Mot89] Amihai Motro. Integrity = Validity + Completeness. *ACM Trans. Database Syst.*, 14(4):480–502, December 1989.

[RKNS15] Simon Razniewski, Flip Korn, Werner Nutt, and Divesh Srivastava. Identifying the extent of completeness of query answers over partially complete databases. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 561–576, Melbourne, Victoria, Australia, May 31 - June 4 2015.

[Sho97] Arie Shoshani. OLAP and statistical databases: Similarities and differences. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 185–196. ACM, 1997.

[SKL+17] Bruhathi Sundarmurthy, Paraschos Koutris, Willis Lang, Jeffrey F. Naughton, and Val Tannen. m-tables: Representing missing data. In *20th International Conference on Database Theory, ICDT*, pages 21:1–21:20, Venice, Italy, 2017.

[TC10] Quoc Trung Tran and Chee-Yong Chan. How to conquer why-not questions. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 15–26. ACM, 2010.