Diversity of Answers to Conjunctive Queries (Extended Abstract)

Timo Camillo Merkl[1], Reinhard Pichler[1] and Sebastian Skritek[1]

[1]TU Wien, Austria

Abstract

Enumeration problems aim at outputting, without repetition, the set of solutions to a given problem instance. However, outputting the entire solution set may be prohibitively expensive if it is too big. In this case, outputting a small, sufficiently diverse subset of the solutions would be preferable. This leads to the Diverse-version of the original enumeration problem, where the goal is to achieve a certain level d of diversity by selecting k solutions.

In this paper, we look at the Diverse-version of the query answering problem for Conjunctive Queries and extensions thereof. That is, we study the problem if it is possible to achieve a certain level d of diversity by selecting k answers to the given query and, in the positive case, to actually compute such k answers.

Keywords

Query Answering, Diversity of Solutions, Complexity, Algorithms

1. Introduction

Answering database queries is one of the most fundamental tasks in computer science and has thus been the topic of countless theoretical analyses from numerous different perspectives. Classically, the focus of complexity studies by the database theory community has been on decision problems such as deciding if the query result is non-empty or if a given tuple is contained in the query result. In recent times, the enumeration problem (i.e., outputting all answers to a query) has played a prominent role on the research agenda, see e.g., [1, 2, 3].

It is well known that even seemingly simple problems, such as answering an acyclic Conjunctive Query, can have a huge number of solutions. Consequently, specific notions of tractability were introduced right from the beginning of research on enumeration problems [4] to separate the computational intricacy of a problem from the mere size of the solution space. However, even with these refined notions of tractability, the usefulness of flooding the user with tons of solutions (many of them possibly differing only minimally) may be questionable. If the solution space gets too big, it would be more useful to provide an overview by outputting a "meaningful" subset of the solutions.

For instance, consider a variation of the car dealership example from [5]. Suppose that I models the preferences of a customer and $\mathcal{S}(I)$ are all cars that match these restrictions. Now,

AMW 2023: 15th Alberto Mendelzon International Workshop on Foundations of Data Management

timo.merkl@tuwien.ac.at (T. C. Merkl); reinhard.pichler@tuwien.ac.at (R. Pichler); sebastian.skritek@tuwien.ac.at (S. Skritek)

0009-0003-7206-2518 (T. C. Merkl); 0000-0002-1760-122X (R. Pichler); 0000-0003-3054-7683 (S. Skritek)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3409/paper2.pdf

in a large dealership, presenting all cars in $\mathcal{S}(I)$ to the customer would be infeasible. Instead, it would be better to go through a rather small list of cars that are significantly different from each other. With this, the customer can point at those cars which the further discussion with the clerk should concentrate on.

This example suggests that we should focus on finding a small *diverse* set of answers to database queries. Due to the inherent hardness of computing maximally diverse solutions [5], the database community – apart from limited exceptions [6] – mostly focused on heuristic and approximation methods to find diverse answers (see [7] for an extensive survey). The present work takes a step forward to fill this void, broadening our understanding of the theoretical boundaries of diverse query answering while developing complementary exact algorithms. More specifically, we analyze diversity problems related to answering Conjunctive Queries (CQs) and extensions thereof.

Problem statement. To formalize the problems, we roughly follow the approach from [8] by defining the diversity of a set of answers (tuples) to be their aggregated pairwise distances. Furthermore, in the spirit of classical relational database theory, we consider the data untyped and use the Hamming distance $\Delta(\gamma, \gamma')$ as the measure of distance between two answer tuples γ, γ'. As far as the choice of an aggregator f is concerned, we impose the general restriction that it must be computable in polynomial time. Formally, for a class $\mathcal{Q}$ of queries we study the following problem Diverse-$\mathcal{Q}$:

Diverse-$\mathcal{Q}$

Instance: A database instance I, query $Q \in \mathcal{Q}$, and integers k and d.
Question: Do there exist pairwise distinct answers $\gamma_1, \dots, \gamma_k \in Q(I)$ (*diversity set*) such that $f((\Delta(\gamma_i, \gamma_j))_{1 \le i < j \le k}) \ge d$?

That is, we ask if a certain level d of diversity can be achieved by choosing k pairwise distinct answers to a given query Q over the database instance I. Since we are intuitively looking for *small* diversity sets, we usually consider k to be a problem parameter in the sense of parameterized complexity.

For proving upper bounds on the complexity in terms of membership, we aim for the most general setting, i.e., the most general class of aggregators f and the biggest query class $\mathcal{Q}$. On the other hand, for lower bounds in terms of hardness proofs, we aim for the most restrictive, i.e., a concrete (but natural) aggregator f and the smallest query class $\mathcal{Q}$. In particular, the natural aggregators minimum and sum will be used to prove lower bounds while, for the upper bound, we either impose no further restriction on f or require them to be monotone in addition, i.e., $f((d_{i,j})_{1 \le i < j \le k}) \le f((d'_{i,j})_{1 \le i < j \le k})$ whenever $d_{i,j} \le d'_{i,j}$ holds for every $1 \le i < j \le k$. The corresponding diversity problems are denoted by $\mathsf{Diverse}_{\mathsf{min}}$-$\mathcal{Q}$, $\mathsf{Diverse}_{\mathsf{sum}}$-$\mathcal{Q}$, and $\mathsf{Diverse}_{\mathsf{mon}}$-$\mathcal{Q}$, respectively.

As for the query classes $\mathcal{Q}$, we start our analysis with the class CQ of Conjunctive Queries and then extend our studies to the classes UCQ and $\mathrm{CQ}^\neg$ of unions of CQs and CQs with negation. As the most general query class, we will also look at the class FO of all first-order queries. Recall that, even the question if an answer tuple exists at all is NP-complete for CQs [9]. We therefore mostly restrict our study to CQs with bounded generalized hypertree width (ghw). For $\mathrm{CQs}^\neg$, query answering remains NP-complete even if we only consider queries with $ghw \le 1$ [10].

Hence, for $\mathrm{CQ}^\neg$ we impose bounded treewidth (*tw*), a stricter structural restriction. Lastly, for arbitrary FO queries, we assume them to be fixed or, equivalently, bounded in size.

2. Preliminaries

Queries. Formally, we can consider our various query classes as fragments of first-order logic with the corresponding database being a matching finite first-order structure and answers being assignments on the free variables. CQs are formulae solely using the connectives $\{\exists, \wedge\}$, while UCQs additionally allow $\{\vee\}$ as a top-level connective and $\mathrm{CQs}^\neg$ additionally allow negated atoms.

Acyclicity and widths. (Generalized hyper-)tree decompositions and their corresponding widths are staple tools to express the acyclicity of CQs (with negation) and define large tractable query classes. A tree decompositions of a $\mathrm{CQ}^\neg$ Q is a tuple $\langle T, \chi \rangle$ such that T is a tree and $\chi\colon V(T) \to 2^{variables(Q)}$. Furthermore, for each atom A, all its variables have to appear in some $\chi(v)$ together, and for each variable x, the vertices v where x appears in $\chi(v)$ have to induce a connected subtree in T. A generalized hypertree decomposition extends a tree decomposition by a labeling $\lambda : V(T) \to 2^{atoms(Q)}$ such that, for every node v in T, every variable of $\chi(v)$ appears in some atom of $\lambda(v)$. Then the treewidth of Q and the generalized hypertree width of Q are defined as follows:

$$tw(Q) = \min_{\langle T,\chi\rangle} \max_{v} |\chi(v)| - 1, \qquad ghw(Q) = \min_{\langle T,\chi,\lambda\rangle} \max_{v} |\lambda(v)|.$$

CQs Q are called acyclic (ACQs) if $ghw(Q) = 1$ and UCQs $Q = \vee_i Q_i$ are called acyclic (UACQs) if all conjunctions Q_i are acyclic.

Parameterized Complexity. An instance of a *parameterized problem* is given as a pair (x, k), where x is the actual problem instance and k is a parameter. A problem is in FPT (*fixed-parameter tractable*), if it can be solved in time $\mathcal{O}(f(k)\cdot|x|^c)$ (for some computable function f and constant c) and in XP, if it can be solved in time $\mathcal{O}(|x|^{f(k)})$. Furthermore, there is a class W[1] in between FPT and XP, i.e., $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{XP}$. It is a generally accepted assumption in parameterized complexity theory that $\mathsf{FPT} \neq \mathsf{W}[1]$ holds – similar but slightly stronger than the famous $\mathsf{P} \neq \mathsf{NP}$ assumption in classical complexity theory.

3. Main Results

The main novel lower bounds of our work are the following hardness results:

Theorem 1. *On queries of bounded tw, the problems $Diverse_{\mathsf{sum}}$-CQ and $Diverse_{\mathsf{min}}$-CQ are W[1]-hard, and on queries of bounded size, the same problems remain NP-hard in the unparameterized case.*

The theorem follows by reduction from the (parameterized) Independent Set problem. In the case where arbitrarily large, albeit almost acyclic, queries are allowed, the reduction preserves the parameter and implies W[1]-hardness. On the other hand, getting by with bounded size

queries requires a more involved reduction, which then no longer preserves the parameter and thus, only implies unparameterized NP-hardness.

Somewhat surprisingly, dealing with UCQs is significantly harder than dealing with CQs.

Theorem 2. *On UACQs and when the size of the sought-after diversity set k is at most two, the problems Diverse$_{\mathsf{sum}}$-UCQ and Diverse$_{\mathsf{min}}$-UCQ remain NP-hard.*

Intuitively, this comes down to the fact that even if all conjuncts of a query are acyclic, the interconnectedness of two or more conjuncts can be arbitrarily cyclic. Furthermore, when searching for diverse answers to UCQs, it is not sufficient to solve the conjuncts independently.

Complementing the W[1]-hardness result, we establish the following memberships:

Theorem 3. *On queries of bounded ghw, the problem Diverse-CQ, and on queries of bounded tw, the problem Diverse-CQ$^\neg$ is in XP.*

This runtime can be achieved by dynamic programming algorithms which maintain data structures storing all possible k-tuples of partial solutions together with all possible pairwise Hamming distances. This data structure is then propagated along the (generalized hyper-)tree decompositions. In the end, a witnessing diversity set can be extracted from the root. Crucially, the size of the data structure is exponential only in k (and the (generalized hyper-)tree width).

For queries of bounded size, an even faster algorithm is developed.

Theorem 4. *On queries of bounded size, the problem Diverse$_{\mathsf{mon}}$-FO is in FPT.*

Such a runtime is achieved by a kernelization algorithm. Essentially, first, all (polynomially many) query answers are computed. Then, the answers are grouped by all possible subsets of the columns, and whenever a group exceeds a certain threshold, insufficiently diverse members of the group are discarded.

Table 1 summarizes the discussed results.

Query class	Lower Bound	Upper Bound
bounded *ghw* CQ	W[1]	XP
bounded *tw* CQ$^\neg$	W[1]	XP
UACQ	NP* for $k = 2$	
bounded size FO	NP*	FPT

Table 1
Classification of Diverse-$\mathcal{Q}$. * unparameterized.

4. Conclusion and Future Work

In this work, we took a fresh look at the Diversity problem of query answering. Looking at different classes of queries from CQs to FO queries, we provide a fairly complete picture of complexity results. Nevertheless, numerous directions for future work exist, including:

Restricting the database. We primarily focused on restrictions on the queries. However, restricting the database is also a viable option with far-reaching implications in related areas. Our full paper contains some preliminary results [11].

Further parameterization. Exploring further parameterizations besides the size of the diversity set seems interesting. For instance, choosing the diversity threshold d lead to intriguing results in [12].

Alternative distance measure. The Hamming distance might not a suitable distance measure in all situation. It might thus be interesting to consider replacing it by more specific distance measures derived from domain specific metrics.

Acknowledgments

This work has been funded by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT2201, 10.47379/VRG18013, 10.47379/NXT22018]; and the Christian Doppler Research Association (CDG) JRC LIVE.

References

[1] A. Amarilli, L. Jachiet, M. Muñoz, C. Riveros, Efficient enumeration for annotated grammars, in: PODS, ACM, 2022, pp. 291–300.
[2] Y. Kobayashi, K. Kurita, K. Wasa, Linear-delay enumeration for minimal steiner problems, in: PODS, ACM, 2022, pp. 301–313.
[3] C. Lutz, M. Przybylko, Efficiently enumerating answers to ontology-mediated queries, in: PODS, ACM, 2022, pp. 277–289.
[4] D. S. Johnson, C. H. Papadimitriou, M. Yannakakis, On generating all maximal independent sets, Inf. Process. Lett. 27 (1988) 119–123.
[5] E. Hebrard, B. Hnich, B. O'Sullivan, T. Walsh, Finding diverse and similar solutions in constraint programming, in: AAAI, AAAI Press / The MIT Press, 2005, pp. 372–377.
[6] T. Deng, W. Fan, On the complexity of query result diversification, ACM Trans. Database Syst. 39 (2014) 15:1–15:46.
[7] K. Zheng, H. Wang, Z. Qi, J. Li, H. Gao, A survey of query result diversification, Knowl. Inf. Syst. 51 (2017) 1–36.
[8] L. Ingmar, M. G. de la Banda, P. J. Stuckey, G. Tack, Modelling diversity of solutions, in: AAAI, AAAI Press, 2020, pp. 1528–1535.
[9] A. K. Chandra, P. M. Merlin, Optimal implementation of conjunctive queries in relational data bases, in: STOC, ACM, 1977, pp. 77–90.
[10] M. Samer, S. Szeider, Algorithms for propositional model counting, J. Discrete Algorithms 8 (2010) 50–64.
[11] T. C. Merkl, R. Pichler, S. Skritek, Diversity of answers to conjunctive queries, in: ICDT, volume 255 of *LIPIcs*, 2023, pp. 10:1–10:19.
[12] F. V. Fomin, P. A. Golovach, L. Jaffke, G. Philip, D. Sagunov, Diverse pairs of matchings, in: ISAAC, 2020, pp. 26:1–26:12.