On the Complexity of Query Result Diversification

Ting Deng[1] Wenfei Fan[2,1]
[1]Big Data Research Center and SKLSDE Lab, Beihang University
[2]School of Informatics, University of Edinburgh
dengting@act.buaa.edu.cn, wenfei@inf.ed.ac.uk

Abstract

Query result diversification is a bi-criteria optimization problem for ranking query results. Given a database D, a query Q and a positive integer k, it is to find a set of k tuples from $Q(D)$ such that the tuples are as relevant as possible to the query, and at the same time, as diverse as possible to each other. Subsets of $Q(D)$ are ranked by an objective function defined in terms of relevance and diversity. Query result diversification has found a variety of applications in databases, information retrieval and operations research.

This paper studies the complexity of result diversification for relational queries. We identify three problems in connection with query result diversification, to determine whether there exists a set of k tuples that is ranked above a bound with respect to relevance and diversity, to assess the rank of a given k-element set, and to count how many k-element sets are ranked above a given bound. We study these problems for a variety of query languages and for three objective functions. We establish the upper and lower bounds of these problems, *all matching*, for both combined complexity and data complexity. We also investigate several special settings of these problems, identifying tractable cases.

1. Introduction

Result diversification for relational queries is a bi-criteria optimization problem. Given a query Q, a database D and a positive integer k, it is to find a set U of k tuples in the query result $Q(D)$ such that the tuples in U are as relevant as possible to query Q, and at the same time, as diverse as possible to each other. More specifically, we want to find a set $U \subseteq Q(D)$ such that $|U| = k$, and the value $F(U)$ of U is maximum. Here $F(\cdot)$ is called an *objective function*. It is defined on *sets* of tuples from $Q(D)$, in terms of a *relevance function* $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and a *distance function* $\delta_{\mathsf{dis}}(\cdot,\cdot)$, where

- for each tuple $t \in Q(D)$, $\delta_{\mathsf{rel}}(t, Q)$ is a number indicating the relevance of answer t to query Q, such that the higher $\delta_{\mathsf{rel}}(t, Q)$ is, the more relevant t is to Q; and
- for all tuples $t_1, t_2 \in Q(D)$, $\delta_{\mathsf{dis}}(t_1, t_2)$ is the distance between t_1 and t_2, such that the larger $\delta_{\mathsf{dis}}(t_1, t_2)$ is, the more diverse the answers t_1 and t_2 are.

In particular, three generic objective functions have been proposed in [17] based on an axiom system, namely, *max-sum diversification*, *max-min diversification* and *mono-objective formulation*. Each function is defined in terms of generic functions $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$, with a parameter λ specifying the tradeoff between relevance and diversity.

Query result diversification is to improve user satisfaction by remedying the over-specification problem of retrieving too homogeneous results. The diversity of query answers is measured in terms of (a) contents, to include items that are dissimilar to each other, (b) novelty, to retrieve items that contain new information not found in previous results, and (c) coverage, to advocate items in different categories [12]. It has proven effective in Web search [17, 36], recommender systems [39, 40, 41], databases [9, 25, 35], as well as in operations research and finance (see [12, 26] for surveys).

This paper investigates the complexity of result diversification analysis for relational queries. While there has been a host of work on result diversification, the previous work has mostly focused on diversity and relevance metrics, and on algorithms for computing diverse results [12, 26]. Few complexity results have been developed for query result diversification, and the known results are mostly lower bounds (NP-hardness) [3, 17, 25, 31, 36]. Furthermore, these results are established by assuming that query result $Q(D)$ is *already known*. In other words, the prior work conducts diversification in two steps: first compute $Q(D)$, and then rank k-element subsets of $Q(D)$ and find a set with the maximum $F(\cdot)$ value. The known complexity results are for *the second step only*, based on *a specific objective function* $F(\cdot)$. However, it is typically expensive to compute $Q(D)$. To avoid the overhead, one often wants to combine the two steps by embedding diversification in query evaluation, and stop as soon as top-ranked results are found based on $F(\cdot)$ (*i.e.,* early termination), rather than to retrieve entire $Q(D)$ in advance [9]. Nonetheless, the complexity of such a query result diversification process has not been studied.

This highlights the need for establishing the complexity of query result diversification, upper bounds and lower bounds, when $Q(D)$ is *not provided*, and for *different query languages* and *various objective functions*. Indeed, to develop practical algorithms for computing diverse query results, we have to understand the impact of query languages and objective functions on the complexity of result diversification. In other words, we need to know where the complexity arises.

Example 1: Consider a recommender system to help people find gifts for various events or occasions, *e.g.,* [15]. Its underlying database D_0 consists of two relations specified by:

catalog(item, type, price, inStock),
history(item, buyer, recipient, gender, age, rel, event, rating).

Here each catalog tuple specifies an item for present, its type (*e.g.,* jewelry, book), price, and the number of the item in

Proceedings of the VLDB Endowment, Vol. 6, No. 8
Copyright 2013 VLDB Endowment 2150-8097/13/06... $ 10.00.

stock. Purchase history is recorded by relation history: a history tuple indicates that a buyer bought an item for a recipient specified by gender, age and relationship with the buyer, for an event (*e.g.,* birthday, wedding, holiday), as well as rating given by the buyer in the range of [1,5].

Uncle Peter wants to use the engine to find a Christmas gift for his 14 year-old niece Grace, in the price range of [\$20, \$30]. His request can be converted to a query Q_0 defined on database D_0. The relevance $\delta_{\mathsf{rel}}(t, Q_0)$ of a tuple t returned by $Q_0(D_0)$ can be assessed by using the information from relation history, by taking into account previous presents purchased for girls of 12–16 year old by the girls' relatives for holidays, as well as the rating by those buyers. The distance (diversity) $\delta_{\mathsf{dis}}(t_1, t_2)$ between two items t_1 and t_2 returned by $Q_0(D_0)$ can be estimated by considering the differences between their types. Peter wants the system to recommend a set of 10 items from $Q_0(D_0)$ such that on one hand, those items are as fit as possible as a Christmas present for a teenage girl, and on the other hand, are as dissimilar as possible to cover a wide range of choices.

Consider the computational complexity of processing such requests. It depends on both the queries for expressing users' requests and the objective function adopted by the system.

(1) Query languages. Query Q_0 can be expressed as a conjunctive query (CQ). Nonetheless, if Peter wants a new gift that is different from previous gifts he gave to Grace, we need first-order logic (FO) to express Q_0, by using negation on relation history. In practice one cannot expect that $Q_0(D_0)$ is already computed when Peter submits his request. As remarked earlier, it is too costly to compute $Q_0(D_0)$ first and then pick a top set of k items from $Q_0(D_0)$. Instead, we want to embed result diversification in the evaluation of Q_0, and ideally, find a satisfactory set of k items *without* retrieving the entire set $Q_0(D_0)$. A question concerns what difference CQ and FO make on the complexity of processing such requests when $Q_0(D_0)$ is not necessarily available. We want to know whether the complexity is *introduced by the query languages or is inherent to result diversification.*

(2) Objective functions. Consider the objective function by max-sum diversification proposed in [17] and revised in [36]:

$$F_{\mathsf{MS}}(U) = (k-1)(1-\lambda) \cdot \sum_{t \in U} \delta_{\mathsf{rel}}(t, Q) + \lambda \cdot \sum_{t, t' \in U} \delta_{\mathsf{dis}}(t, t').$$

To assess the diversity, $F_{\mathsf{MS}}(U)$ only requires to compute $\delta_{\mathsf{dis}}(t, t')$ for t and t' in a given k-element set U. In contrast, consider a (revised) mono-objective function given by [17]:

$$F_{\mathsf{mono}}(U) = \sum_{t \in U} \Big((1-\lambda) \cdot \delta_{\mathsf{rel}}(t, Q) + \frac{\lambda}{|Q(D)| - 1} \cdot \sum_{t' \in Q(D)} \delta_{\mathsf{dis}}(t, t') \Big).$$

It asks for $\delta_{\mathsf{dis}}(t, t')$ for each $t \in U$ and for *all* $t' \in Q(D)$, *i.e.,* the *average dissimilarity w.r.t.* all other results in $Q(D)$ [26]. The question is what *different impacts* $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{mono}}(\cdot)$ have on the complexity of diversification. □

To the best of our knowledge, no prior work has answered these questions. These issues require a full treatment for different query languages and objective functions, to find out *where the complexity of query result diversification arises.*

Contributions. We study several problems in connection with result diversification for relational queries, and establish their upper bounds and lower bounds, all matching, for a variety of query languages and objective functions.

Diversification problems. We identify three problems, denoted by QRD, DRP and RDC. Given a query Q, a database D, an objective function $F(\cdot)$, and a positive integer k,

(1) QRD is to determine whether there exists a k-element set $U \subseteq Q(D)$ such that $F(U) \geq B$ for a given bound B, *i.e.,* whether there exists a set U that satisfies the users' need at all; this is a decision problem fundamental to diversification;

(2) DRP is to decide the rank r of a given k-element set $U \subseteq Q(D)$, such that there exist no more than $r-1$ sets $S \subseteq Q(D)$ of k elements with $F(S) > F(U)$; as advocated in [20], a decision procedure for DRP helps us assess how well a given choice U satisfies the users' request, and helps vendors evaluate their products *w.r.t.* users' need; and

(3) RDC is to count the number of k-element sets $U \subseteq Q(D)$ such that $F(U) \geq B$ for a given bound B. It is a *counting problem* that helps us find out how many k-element sets can be extracted from $Q(D)$ and be suggested to the users, and as a result, provide a guidance for us to adjust our stock.

Complexity results. For all these problems we establish their combined complexity and data complexity (*i.e.,* when both data D and query Q may vary, and when Q is fixed while D may vary, respectively [1]). We parameterize these problems with various query languages, including conjunctive queries (CQ), unions of conjunctive queries (UCQ), positive existential FO queries ($\exists$FO$^+$) and first-order logic queries (FO). These languages have been used in query result diversification tools, *e.g.,* CQ [8], $\exists$FO$^+$ [35] and FO [9]. For each of the languages, we study these problems with each of the objective functions proposed by [17]: max-sum diversification, max-min diversification and mono-objective formulation.

We provide a comprehensive account of upper and lower bounds for these problems, all matching. We also study special cases of these problems, such as when either only diversity or only relevance is considered, when Q is an identity query, and when k is a predefined constant. We identify practical tractable cases. It should be remarked that all the previous results (NP-hardness) are established for a special case of QRD only, namely, when Q is an identity query.

Impact. These results tell us where the complexity arises.

(1) Query languages $\mathcal{L}_Q$. Query languages may dominate the combined complexity analysis. For objective functions defined in terms of max-sum or max-min diversification, QRD, DRP and RDC are NP-complete, coNP-complete and #·NP-complete, respectively, when $\mathcal{L}_Q$ is CQ. In contrast, when it comes to FO, these problems become PSPACE-complete, PSPACE-complete and #·PSPACE-complete, respectively. This said, the presence of disjunction in $\mathcal{L}_Q$ does not complicate the diversification analyses. Indeed, these problems remain NP-complete, coNP-complete and #·NP-complete, respectively, when $\mathcal{L}_Q$ is UCQ or $\exists$FO$^+$.

In contrast, different query languages have no impact on the data complexity of these problems. Indeed, for max-sum or max-min diversification, QRD, DRP and RDC are NP-complete, coNP-complete and #·NP-complete, respectively, and for mono-objective formulation, they are in PTIME (polynomial time), PTIME and #P-complete, respectively, no matter whether $\mathcal{L}_Q$ is CQ or FO. Intuitively, a naive algorithm for QRD works in two steps: first compute $Q(D)$, and then finds whether there exists a k-element set U from $Q(D)$ such that $F(U) \geq B$; similarly for DRP and RDC. When Q is fixed as in the setting of data complexity analysis, $Q(D)$ is in PTIME regardless of what query language $\mathcal{L}_Q$ we use to express Q. The data complexity of the problems arises from the second step, *i.e.,* the diversification computation.

(2) Objective functions $F(\cdot)$. When $F(\cdot)$ is defined for mono-objective formulation, however, the objective function *dominates* the complexity: QRD, DRP and RDC are PSPACE-complete, PSPACE-complete and #·PSPACE-complete, respectively, no matter whether $\mathcal{L}_Q$ is CQ or FO. Contrast these with their counterparts given above when $F(\cdot)$ is defined for max-sum or max-min diversification.

The impact of $F(\cdot)$ is even more evident on the data complexity. As remarked earlier, when $F(\cdot)$ is for max-sum or max-min diversification, these problems are NP-complete, coNP-complete and #·NP-complete, respectively, for data complexity, whereas they are in PTIME, PTIME and #P-complete, respectively, when $F(\cdot)$ is the mono objective.

(3) Diversity vs. relevance. The complexity is mostly introduced by the diversity requirement. This is consistent with the observation of [36], which studied a special case of QRD when $F(\cdot)$ is defined for max-sum diversification. Indeed, when the relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ is absent, the combined complexity and data complexity remain unchanged for these problems, when $F(\cdot)$ is any of the three objective functions. In contrast, when the distance function $\delta_{\mathsf{dis}}(\cdot,\cdot)$ is dropped, QRD and DRP become tractable when data complexity is considered. Moreover, for mono-objective $F(\cdot)$, when $\delta_{\mathsf{dis}}(\cdot,\cdot)$ is absent, the combined complexity bounds of QRD, DRP and RDC become NP-complete, coNP-complete and #·NP-complete, down from PSPACE-complete, PSPACE-complete and #·PSPACE-complete, respectively.

These results reveal the impacts of various factors on the complexity of query result diversification. In particular, the complexity of these problems for CQ, UCQ and $\exists\text{FO}^+$may be *inherent to* result diversification itself, rather than a consequence of the complexity of the query languages. Various techniques are used to prove these results, including a wide range of reductions and constructive proofs with algorithms.

These results are not only of theoretical interest, but may also help practitioners when developing diversification models and algorithms in practice. Indeed, to develop these, we may want to decide on the following. What query language should be supported? What diversification function should be adopted? Would a relevance function alone suffice in our applications so that we do not have to pay the price of the complexity introduced by distance functions? Would a fixed set of queries suffice for users to express their requests? What is the best one can hope for a given query language and objective function? These are not only useful in the analyses of diversification, but are also of interest to the study of recommender systems (see below).

Related work. This work is related to previous work on result diversification (for search and queries), recommender systems and top-k query answering, discussed as follows.

Result diversification. Diversification has been studied for Web search [3, 6, 7, 17, 36], recommender systems [38, 39, 40, 41], and structured databases [9, 16, 25, 35] possibly with user preferences [8, 31] (see [12, 26] for surveys). As remarked earlier, the previous work has mostly focused on metrics for assessing relevance and diversity, and algorithms and optimization techniques for computing diverse answers. The prior work often adopts specific objective functions based on the similarity of, *e.g.,* taxonomy [41], explanations [38], features [35] or locations [16]. A general model for result diversification was proposed in [17] based on an axiom system, along with the three objective functions mentioned earlier. A minor revision of the function for max-sum diversification of [17] was presented in [36]. This work extends the model of [17] by incorporating queries. Like in [6], we focus on the objective functions proposed in [17].

The complexity of diversification has been studied in [3, 17, 25, 31, 36], which differ from ours in the following.

(1) The previous work provided lower bounds (NP-hardness) but stopped short of giving a matching upper bound. In contrast, we provide a complete picture of matching upper and lower bounds, for both combined and data complexity.

(2) The prior work assumes that the search space $Q(D)$ is already computed, and is taken as input. As remarked earlier, this assumption is not very realistic in practice. In contrast, we treat Q and D as input instead of $Q(D)$, and investigate the impact of query languages on the complexity of diversification. As will be seen later, the complexity bounds of these problems when $Q(D)$ is not available is quite different from their counterparts when $Q(D)$ is assumed in place.

(3) The previous work focused on a special cases of QRD, when Q is an identity query (*i.e.,* $Q(D)$ is already given). It is one of the special cases studied in Section 5 of this paper. Note that the intractability of QRD for max-sum or max-min diversification given in the prior work [11, 17, 36] may be adapted to establish the data complexity of QRD in these settings. Nonetheless, the detailed proofs are not given in those papers. Furthermore, for mono-objective formulation, no previous work has studied the complexity of QRD for identity queries, which is shown in PTIME in this work.

Problem DRP was first formulated in [20]. However, we are not aware of any prior work on DRP and RDC studied here. These two problems are obviously important but unfortunately, have been overlooked by and large.

(4) The prior results were established for one of the variants of max-sum or max-min diversification [17, 11, 36] defined in terms of specific $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ functions. In contrast, we study all three objective functions defined with $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ that are only assumed PTIME computable.

(5) This work also considers several special cases of diversification (Section 5), to identify tractable cases and explore the impact of diversity and relevance requirements on the complexity of the diversification analyses.

Recommender problems. Recommender systems are to recommend information items or social elements that are likely to be of interest to users (see [2] for a survey). There has been a host of work on recommender systems [4, 10, 23, 21, 28, 37], studying item recommendation and package recommendation. Given a query Q, a database D of items and a utility (scoring) function $f(\cdot)$ defined on items, *item recommendation* is to find top-k items from $Q(D)$ ranked by $f(\cdot)$, for a given positive integer k. *Package recommendation* takes as additional input a set Σ of compatibility constraints, two functions $\mathsf{cost}(\cdot)$ and $\mathsf{val}(\cdot)$ defined on sets of items, and a bound C. It is to find top-k packages of items such that each package satisfies the constraints in Σ, its cost does not exceed C, and its val is among the k highest. Here a package is *a set of items* that has a *variable* size.

There is an intimate connection between recommendation and diversification: both aim to recommend top-k (sets of) items from the result $Q(D)$ of query Q in D. Among other things, diversification has been used in recommender systems to rectify the problem of retrieving too homogeneous results. However, there are subtle differences between them.

(1) Item recommendation is a single-criterion optimization problem based on a utility function $f(\cdot)$ defined on individual items. In contrast, query result diversification is a bi-criteria optimization problem based on a relevance function $\delta_{\sf rel}(\cdot, \cdot)$ and a distance function $\delta_{\sf dis}(\cdot, \cdot)$ defined on *sets of items*. In particular, the distance function $\delta_{\sf dis}(U)$ assess the diversity of elements in a set U, and is not expressible as a utility function of item recommendation.

(2) Package recommendation is to find top-k sets of items with *variable sizes*, which are ranked by $\mathsf{val}(\cdot)$, subject to compatibility constraints Σ and aggregate constraints defined in terms of $\mathsf{cost}(\cdot)$ and bound C, where $\mathsf{cost}(\cdot)$ and $\mathsf{val}(\cdot)$ are generic PTIME computable function [10]. In contrast, query result diversification is to find *a single set of k items*, based on a *particular* objective function $F(\cdot)$. When $F(\cdot)$ is defined by max-sum or max-min diversification, diversification can be viewed as a special case of package recommendation for finding a single set of a fixed size k, based on a particular $F(\cdot)$, and in the absence of compatibility constraints and aggregate constraints. As a consequence of the specific restrictions of $F(\cdot)$, the lower bounds developed for package recommendation *do not* carry over to its counterpart for diversification, and conversely, the upper bounds for diversification *may not be* tight for package recommendation. When $F(\cdot)$ is defined by mono-objective formulation, $F(U)$ is *not* even expressible in the model of recommendation, since it assess the diversity of elements in a set U with *all* tuples in $Q(D)$, and is not in PTIME in $|U|$.

There has been work on the complexity of recommendation analyses [23, 21, 28, 37], including our own work [10]. In addition to different settings of the two as remarked earlier, this work differs from the prior work in the following.

(3) Problems QRD and DRP studied in this paper have not been considered in the previous work for recommendation, including [10]. This said, the results of this work on these problems may be of interest to the study of recommendation.

(4) Problem RDC considered here is similar to a counting problem studied in [10] for recommendation. However, given the different settings remarked earlier, RDC differs from that counting problem from complexity bounds to proofs. Indeed, the counting problem for recommendation is #·coNP-complete when $\mathcal{L}_Q$ is CQ, UCQ or $\exists\mathrm{FO}^+$ [10]; in contrast, as will be seen in Section 4, for the same query languages, (a) RDC is #·NP-complete when $F(\cdot)$ is defined by max-sum or max-min diversification, while #·coNP = #·NP iff P = NP; and (b) RDC is #·PSPACE-complete when $F(\cdot)$ is defined by mono-objective formulation, substantially more intriguing than the problem studied in [10]. Furthermore, the proofs of this paper have to be tailored to specific objective functions, as opposed to the proofs of [10]. Indeed, the proofs for $F(\cdot)$ defined by max-sum or max-min diversification are quite different from their counterparts for $F(\cdot)$ defined by mono-objective formulation, as indicated by the different combined complexity bounds in these settings.

Top-k query answering. Top-k query answering aims to retrieve top-k tuples from a query result, ranked by a scoring function [19]. It typically assumes that the attributes of tuples are already sorted, and studies how to combine different ratings of the attributes for the same tuple based on a (monotonic) scoring function, possibly by incorporating user preference [32]. A number of top-k query evaluation algorithms have been developed (*e.g.,* [14, 20, 24, 30]; see [19] for a survey), focusing on how to achieve early termination and reduce random access. This work differs from the prior work in the following. (a) A scoring function for top-k query answering is defined on individual items, as opposed to the distance function $\delta_{\sf dis}(\cdot)$ and the objective function $F(\cdot)$ that are defined on sets of items and are more involved. (b) We focus on the complexity of diversification problems rather than the efficiency or optimization of query evaluation.

Organization. We present a general model for query result diversification in Section 2, by extending the model of [17]. Problems QRD, DRP and RDC are formulated in Section 3, and their combined and data complexity are established in Section 4. Special cases of these problems are studied in Section 5, followed by directions for future work in Section 6.

2. Diversification and Objective Functions

In this section we first present a model for query result diversification. We then review the three objective functions proposed by [17], which are used to define diversification.

2.1 Query Result Diversification

Query result diversification is to improve user satisfaction when computing answers to a query Q in a database D. We specify D with a relational schema $\mathcal{R} = (R_1, \ldots, R_n)$.

We consider query Q expressed in a query language $\mathcal{L}_Q$.

Diversification. Given Q, D, a positive integer k and an objective function $F(\cdot)$, *query result diversification* aims to find a set $U \subseteq Q(D)$ such that (a) $|U| = k$, and (b) $F(U)$ is maximum, *i.e.,* for all other sets $U' \subseteq Q(D)$, if $|U'| = k$ then $F(U) \geq F(U')$. Here $F(\cdot)$ is an objective function defined on *sets* of tuples of R_Q, where R_Q denotes the schema of query result $Q(D)$, such that given any set U of tuples of R_Q, $F(U)$ returns a non-negative real number.

Intuitively, diversification is to retrieve a set U of k answers to Q in D such that the tuples in U are as relevant as possible to Q and meanwhile, as diverse as possible. It extends the notion of result diversification given in [17] by taking Q and D as input, *rather than* assuming that $Q(D)$ is already computed and available. In fact the notion of [17] is a special case of query result diversification, when Q is an identity query, *i.e.,* when $Q(D) = D$ is given as input.

Query result diversification is a bi-criteria optimization problem characterized by objective function $F(\cdot)$, which is defined in terms of a relevance function $\delta_{\sf rel}(\cdot, \cdot)$ and a distance function $\delta_{\sf dis}(\cdot, \cdot)$, presented as follows.

Relevance functions and distance functions. A *relevance function* $\delta_{\sf rel}(\cdot, \cdot)$ is defined on tuples of schema R_Q and queries in $\mathcal{L}_Q$. It specifies the relevance of a tuple t of R_Q to a query $Q \in \mathcal{L}_Q$. More specifically, $\delta_{\sf rel}(t, Q)$ is a non-negative real number such that the larger $\delta_{\sf rel}(t, Q)$ is, the more relevant the answer t is to query Q.

A *distance function* $\delta_{\sf dis}(\cdot, \cdot)$ is a binary function defined on tuples of schema R_Q. It specifies the diversity between two tuples $t_1, t_2 \in Q(D)$: $\delta_{\sf dis}(t_1, t_2)$ is a non-negative real number such that the larger $\delta_{\sf dis}(t_1, t_2)$ is, the more diverse (dissimilar) the two tuples t_1 and t_2 are to each other. We assume that $\delta_{\sf dis}(\cdot, \cdot)$ is symmetric, *i.e.,* $\delta_{\sf dis}(t_1, t_2) = \delta_{\sf dis}(t_2, t_1)$ for all tuples t_1, t_2 of R_Q. We also assume *w.l.o.g.* that $\delta_{\sf dis}(t, t) = 0$, *i.e.,* the distance between a tuple and itself is 0.

We simply assume that $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ are PTIME computable functions, as commonly found in practice.

Example 2: Recall the request of Peter for shopping a gift for Grace described in Example 1. It can be expressed as a query Q_0 in FO (written in relational calculus) as follows:

$$Q_0(n) = \exists\, t, p, s\ \big(\mathsf{catalog}(n,t,p,s) \wedge p \leq 30 \wedge p \geq 20 \wedge \\ \forall n', b, r, g,\ a,\ x, e,\ y\ \neg(\mathsf{history}(n', b, r, g, a, x, e, y) \wedge \\ b = \mathrm{id}_P \wedge r = \text{“Grace”} \wedge n = n')\big),$$

where id_P denotes Peter's buyer id. The query selects such gifts in the price range [\$20, \$30] that have not been purchased by Peter for Grace earlier.

As remarked in Example 1, for each gift $t \in Q_0(D_0)$, the relevance $\delta_{\mathsf{rel}}(t, Q_0)$ of t to Q_0 can be assessed in terms of the rating of t if t appears in the $\mathsf{history}$ relation. For instance, $\delta_{\mathsf{rel}}(t, Q_0)$ is high if t was presented as a gift for a girl of age [12, 16] by a relative for a holiday, and was rated high. If t is not in relation $\mathsf{history}$, $\delta_{\mathsf{rel}}(t, Q_0)$ takes a default value.

For tuples $t_1, t_2 \in Q_0(D_0)$, $\delta_{\mathsf{dis}}(t_1, t_2)$ can be defined in terms of the difference between their types, *e.g.,* $\delta_{\mathsf{dis}}(t_1, t_2) = 2$ if t_1 is in the “artsy” category and t_2 is in “educational”, and $\delta_{\mathsf{dis}}(t_1, t_2) = 1$ if t_1 is of type “jewelry” and t_2 is of “fashion” [15]. The types can be classified into various categories and brands, and $\delta_{\mathsf{dis}}(t_1, t_2)$ is defined accordingly. □

2.2 Objective Functions

An objective function $F(\cdot)$ is defined in terms of relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and distance function $\delta_{\mathsf{dis}}(\cdot,\cdot)$. Like in [6], we focus on the objective functions proposed in [17].

Consider $\delta_{\mathsf{rel}}(\cdot,\cdot)$, $\delta_{\mathsf{dis}}(\cdot,\cdot)$, a parameter λ to balance relevance and diversity ($0 \leq \lambda \leq 1$), a query Q, a database D and a positive integer k. Let $U \subseteq Q(D)$ be a set of tuples with $|U| = k$. A minor revision of max-sum diversification of [17] was given in [36] by associating $(1-\lambda)$ with the relevance component, which allows us to study two extreme cases: diversity only (*i.e.,* when $\lambda = 1$), and relevance only (*i.e.,* when $\lambda = 0$). Along the same line as [36], we consider minor variations of the max-sum and max-min diversification, as well as mono-objective functions introduced in [17].

Max-sum diversification. The first objective is to maximize the sum of the relevance and dissimilarity of the selected set U, computed by objective function $F_{\mathsf{MS}}(\cdot)$ [17, 36]:

$$F_{\mathsf{MS}}(U) = (k-1)(1-\lambda) \cdot \sum_{t \in U} \delta_{\mathsf{rel}}(t, Q) + \lambda \cdot \sum_{t, t' \in U} \delta_{\mathsf{dis}}(t, t').$$

It measures the sum of both the relevance of the tuples in U to query Q, and the diversity among the k tuples in U. Following [17], we scale up the two components $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ by using $k-1$ since the relevance sum ranges over k numbers while the diversity sum is over $k(k-1)$ numbers.

As observed by [17, 36], when the objective function is $F_{\mathsf{MS}}(\cdot)$ for max-sum diversification, result diversification can be recast in terms of the Maxsum Dispersion Problem studied in operations research [29], when Q is an identity query.

Max-min diversification. The second objective is to maximize the minimum relevance and dissimilarity of the selected set, computed by objective function $F_{\mathsf{MM}}(\cdot)$:

$$F_{\mathsf{MM}}(U) = (1-\lambda) \cdot \min_{t \in U} \delta_{\mathsf{rel}}(t, Q) + \lambda \cdot \min_{t, t' \in U, t \neq t'} \delta_{\mathsf{dis}}(t, t').$$

It is computed in terms of both the minimum relevance of the k tuple in U to query Q, and the minimum distance between any pair of the tuples in U. As shown in [17], diversification by $F_{\mathsf{MM}}(\cdot)$ can be expressed as the Maxmin Dispersion Problem studied in [29] when Q is an identity query.

Mono-objective formulation. The third objective aims to combine the relevance and diversity values into a single value for each tuple in $Q(D)$, computed by $F_{\mathsf{mono}}(\cdot)$ [17]:

$$F_{\mathsf{mono}}(U) = \sum_{t \in U} \big((1-\lambda) \cdot \delta_{\mathsf{rel}}(t, Q) + \frac{\lambda}{|Q(D)| - 1} \cdot \sum_{t' \in Q(D)} \delta_{\mathsf{dis}}(t, t')\big).$$

As opposed to $F_{\mathsf{MS}}(U)$ and $F_{\mathsf{MM}}(U)$ that compute intro-list diversity, $F_{\mathsf{mono}}(U)$ measures the “global” diversity of a tuple $t \in U$ by taking the mean of its distance to *each* tuple in the entire set $Q(D)$, rather than its distances to the tuples in U [17]. While mono-objective objective is not yet as popular as $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, it represents an objective that does not reduce to facility dispersion. It may also prove useful in practical applications since it computes the average dissimilarity of tuples in U regarding all other results in $Q(D)$ [26], to assess *the novelty and coverage* of the results in U. Hence we also study it in this work to give a complete picture for various diversification objective functions.

Example 3: Consider the query Q_0, database D_0, and the relevance and distance functions $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ described in Example 2. Assume that $k = 10$. Then

(1) for max-sum diversification, query result diversification with the objective function $F_{\mathsf{MS}}(\cdot)$ aims to find a set U_1 of 10 gifts from $Q_0(D_0)$ such that the weighted sum of the relevance values of the selected gifts in U_1 to Q_0 and the dissimilarity values among the gifts in U is maximum.

(2) For max-min diversification, function $F_{\mathsf{MM}}(\cdot)$ is to find a set U_2 of 10 gifts from $Q_0(D_0)$ such that the weighted sum of the minimum relevance of the gifts in U_2 to Q_0 and the minimum distance between pairs of gifts in U_2 is maximum.

(3) For mono-objective, function $F_{\mathsf{mono}}(\cdot)$ is to find a set U_3 of 10 gifts from $Q_0(D_0)$ such that the weighted sum of the relevance values of the gifts in U_3 to Q_0 and the mean of the distances between the selected gifts in U_3 and *all candidate* gifts in the entire set $Q_0(D_0)$ is maximized. In particular, here the diversity criterion is to assess the coverage of various gifts in the entire set $Q_0(D_0)$ by the set U chosen. □

Remarks. Observe the following.

(1) Objective functions $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ and $F_{\mathsf{mono}}(\cdot)$ are defined in terms of relevance ($\delta_{\mathsf{rel}}(\cdot,\cdot)$) and diversity ($\delta_{\mathsf{dis}}(\cdot,\cdot)$). The larger λ is, the more weight we place on the diversity of the results selected. When $\lambda = 0$ (resp. 1), $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ and $F_{\mathsf{mono}}(\cdot)$ measure the relevance (resp. diversity) only.

(2) For a given set $U \subseteq Q(D)$, $F_{\mathsf{MS}}(U)$ and $F_{\mathsf{MM}}(U)$ are PTIME computable as long as $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$ are PTIME computable. In contrast, $F_{\mathsf{mono}}(U)$ may *not* be PTIME computable when Q and D are given as input but $Q(D)$ is not assumed available. Indeed, for each tuple $t \in U$, $F_{\mathsf{mono}}(U)$ has to compute $\delta_{\mathsf{dis}}(t, t')$ *for all* tuples $t' \in Q(D)$.

3. The Analyses of Result Diversification

In this section we identify three problems in connection with query result diversification. In the next two sections we will provide the complexity of these problems.

Consider a database D, a query Q in a language $\mathcal{L}_Q$, a positive integer k, and an objective function $F(\cdot)$ defined with relevance and distance functions $\delta_{\mathsf{rel}}(\cdot,\cdot)$ and $\delta_{\mathsf{dis}}(\cdot,\cdot)$.

The query result diversification problem. We start with a decision problem, referred to as *the query result diversification problem* and denoted by $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$. To formulate this problem, we need the following notations.

We call a set $U \subseteq Q(D)$ a *candidate set for* (Q, D, k) if $|U| = k$. Given a bound B, we refer to a candidate set U as *a valid set for* (Q, D, k, F, B) if $F(U) \geq B$. That is, the $F(\cdot)$ value of U is large enough to meet the objective B.

Given these, $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is stated as follows.

$\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$:	*The query result diversification problem.*
INPUT:	A database D, a query $Q \in \mathcal{L}_Q$, an objective function $F(\cdot)$, a real number B and a positive integer $k \geq 1$.
QUESTION:	Does there exist a valid set for $(Q, D, k,\ F, B)$?

Intuitively, $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is to decide whether there exists a candidate set $U \subseteq Q(D)$ that meets the objective B. This is the decision version of the function problem for computing a top-ranked set U based on $F(\cdot)$, and is fundamental to understanding the complexity of query result diversification. As remarked earlier, we simply consider generic PTIME functions $\delta_{\mathsf{rel}}(\cdot, \cdot)$ and $\delta_{\mathsf{dis}}(\cdot, \cdot)$ when defining $F(\cdot)$.

The diversity ranking problem. In practice, given a candidate set U picked by users or produced by a system, we want to assess how well U meets a diversification objective and hence, satisfies the users' need. This suggests that we study another decision problem, referred to as *the diversity ranking problem* and denoted by $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$, to determine the rank of a given candidate set based on $F(\cdot)$. To state this problem, we use the following notion of ranks.

Consider a candidate set U and a positive integer r. We say that the *rank* of U is r, denoted by $\mathsf{rank}(U) = r$, if there exists a collection $\mathcal{S}$ of $r - 1$ distinct candidate sets for (Q, D, k) such that (a) for all $S \in \mathcal{S}$, $F(S) > F(U)$; and (b) for any candidate set S' for (Q, D, k), if $S' \notin \mathcal{S}$, then $F(U) \geq F(S')$. That is, there exist at most $r-1$ candidates sets for (Q, D, k) that are ranked above U based on $F(\cdot)$. Intuitively, the less $\mathsf{rank}(U)$ is, the higher U is ranked.

Given this, problem $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ is stated as follows. Assume a positive integer r that is a constant.

$\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$:	*The diversity ranking problem.*
INPUT:	D, Q, $F(\cdot)$ and k as in $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, and a candidate set U for (Q, D, k).
QUESTION:	Does $\mathsf{rank}(U) \leq r$?

This problem was first advocated in [20]. The need for studying this is evident: on one hand, users may employ a decision procedure for DRP to assess how good their selections are; on the other hand, vendors may also leverage the procedure to evaluate how popular their products are and hence, adjust their sale and promotion, among other things.

The result diversity counting problem. Given an objective B, one often wants to know how many valid sets are out there and hence, can be selected. This suggests that we study the counting problem below, referred to as *the result diversity counting problem* and denoted by $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$.

$\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$:	*The result diversity counting problem.*
INPUT:	D, Q, $F(\cdot)$, k and B as in problem $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$.
QUESTION:	How many valid sets are there for (Q, D, k, F, B)?

That is, $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is to count the number of candidate sets in D that satisfy the users' request. An effective counting procedure obviously finds applications in practice [10]. For instance, it helps the manager of a recommender system find out how many items carried by the system meet the users' need, and adjust the stock accordingly.

Parameters of the problems. We study these problems for (a) objective functions $F(\cdot)$ ranging over $F_{\mathsf{MS}}(\cdot)$ for max-sum diversification, $F_{\mathsf{MM}}(\cdot)$ for max-min diversification, and $F_{\mathsf{mono}}(\cdot)$ for mono-objective formulation (Section 2), and for (b) query languages $\mathcal{L}_Q$ ranging over the following (see [1] for details of these languages):

(1) conjunctive queries (CQ), built up from atomic formulas with constants and variables, *i.e.,* relation atoms in database schema $\mathcal{R}$ and built-in predicates $(=, \neq, <, \leq, >, \geq)$, by closing under conjunction $\wedge$ and existential quantification $\exists$;

(2) union of conjunctive queries (UCQ) of the form $Q_1 \cup \cdots \cup Q_r$, where for each $i \in [1, r]$, Q_i is in CQ;

(3) positive existential FO queries ($\exists$FO$^+$), built from atomic formulas by closing under $\wedge$, *disjunction* $\vee$ and $\exists$; and

(4) first-order logic queries (FO) built from atomic formulas using $\wedge$, $\vee$, *negation* $\neg$, $\exists$ and *universal quantification* $\forall$.

To the best of our knowledge, no prior work has studied $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ for diversification. When it comes to $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, only a special case was studied, when $\mathcal{L}_Q$ consists of identity queries and $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$. No previous work has considered $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ when $\mathcal{L}_Q$ is CQ or beyond, or when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$.

4. Complexity Results

In this section we establish the complexity of problems $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$. We explore the impact of query languages $\mathcal{L}_Q$ and objective functions $F(\cdot)$ on the complexity of these problems, for $\mathcal{L}_Q$ ranging over query languages CQ, UCQ, $\exists$FO$^+$and FO, and when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ (max-sum diversification), $F_{\mathsf{MM}}(\cdot)$ (max-min diversification), or $F_{\mathsf{mono}}(\cdot)$ (mono-objective formulation).

For each of these problems, we establish (a) its *combined complexity*, when *both* query Q and database D may vary, and (b) its *data complexity*, when query Q is predefined and fixed, but database D may vary, *i.e.,* the complexity of evaluating a *fixed* query for variable database inputs (see [1] for details). The latter is also of practical interest since in many real-life applications, one often uses a fixed set of queries, while the database may be frequently updated.

4.1 Combined Complexity

We first study the combined complexity of these problems.

The query result diversification problem. We start with the decision problem $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$. Given a query Q in $\mathcal{L}_Q$, a database D, an objective function $F(\cdot)$, a positive integer k, and an objective bound B, it is to decide whether there exists a set $U \subseteq Q(D)$ such that $|U| = k$ and $F(U) \geq B$, *i.e.,* whether there exists a valid set for (Q, D, k, F, B).

Recall the naive algorithm for QRD: first compute $Q(D)$, and then rank k-element sets U of $Q(D)$ based on $F(U)$. One might think that the complexity of QRD would equal the higher complexity of the two steps. The result below tells us that this is the case when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$. However, when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the story is quite different.

(1) When the objective is for max-sum or max-min diversification, query language $\mathcal{L}_Q$ has impact on the combined complexity of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$: it is NP-complete when $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$, but becomes PSPACE-complete when

$\mathcal{L}_Q$ is FO. That is, while the presence of disjunction (UCQ and $\exists$FO$^+$) does not make our lives harder than CQ, the presence of negation (FO) does complicate the analysis.

(2) When it comes to mono-objective formulation, the problem becomes harder when $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$: it is already PSPACE-complete, the same as its complexity for FO. Note that *the membership problem* for CQ is NP-complete, which is to determine, given a CQ query Q, a database D and a tuple t, whether $t \in Q(D)$. In contrast, QRD(CQ, $F_{\mathsf{mono}}(\cdot)$) is PSPACE-complete. This is because mono-objective requires to aggregate distances between elements in U and all tuples in $Q(D)$, and is more intriguing to compute than max-sum and max-min diversification, as remarked in Section 2. Hence in this case, the complexity is inherent to query result diversification, and is not equal to the higher complexity of the two steps given above.

Theorem 1. *For* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$*, when objective function* $F(\cdot)$ *is* $F_{\mathsf{MS}}(\cdot)$ *or* $F_{\mathsf{MM}}(\cdot)$*, the combined complexity is*

- NP*-complete when* $\mathcal{L}_Q$ *is CQ, UCQ or* $\exists$*FO*$^+$*, and*
- PSPACE*-complete when* $\mathcal{L}_Q$ *is FO.*

Nevertheless, the combined complexity is

- PSPACE*-complete when* $\mathcal{L}_Q$ *is CQ, UCQ,* $\exists$*FO*$^+$ *or FO, when* $F(\cdot)$ *is* $F_{\mathsf{mono}}(\cdot)$*.*

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is NP-hard when $\mathcal{L}_Q$ is CQ, by reductions from the 3SAT problem, which is known to be NP-complete (cf. [27]). We also show that the problem is in NP when $\mathcal{L}_Q$ is $\exists$FO$^+$, by providing an NP algorithm that checks whether there exists a valid set for given (Q, D, k, F, B).

(2) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that the problem is PSPACE-hard by reductions from the membership problem for FO (see the statement of the problem above), which is known to be PSPACE-complete [34]. We also develop a PSPACE algorithm to check the existence of a valid set.

(3) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the proof is more involved. We show that $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is already PSPACE-hard when $\mathcal{L}_Q$ is CQ, by reduction from Q3SAT. Q3SAT is to decide whether a quantified sentence $\varphi = P_1x_1, \ldots, P_mx_m\psi$ is true, where P_i is either $\exists$ or $\forall$; it is known to be PSPACE-complete (cf. [27]). The reduction is defined in terms of (a) a CQ query Q and a database D, such that $Q(D)$ encodes all the 2^m truth assignments for variables $x_1, \ldots, x_m$; (b) $B = 1$ and $k = 1$; and (c) $F_{\mathsf{mono}}(\cdot)$ with a distance function $\delta_{\mathsf{dis}}(\cdot,\cdot)$ and $\lambda = 1$ as follows. Consider a pair of tuples $\vec{u} = (u_1, \ldots, u_m)$ and $\vec{v} = (v_1, \ldots, v_m)$ such that $u_i = v_i$ for $i \in [1, l]$, and $u_{l+1} \neq v_{l+1}$, *i.e.,* $\vec{u}$ and $\vec{v}$ agree on the first l attribute values but have different values for the $(l+1)$th attribute, where $m - 1 \geq l \geq 0$. Denote by $\vec{u}_l$ the prefix of $\vec{u}$ of length l. We define $\delta_{\mathsf{dis}}(\cdot,\cdot)$ to ensure the following.

Given a truth assignment μ_X^l for variables $x_1, \ldots, x_l$, $P_{l+1}x_{l+1}, \ldots, P_mx_m\ \psi$ is true with μ_X^l iff there exist $\vec{u} = (u_1, \ldots, u_m)$ and $\vec{v} = (v_1, \ldots, v_m)$ such that $\delta_{\mathsf{dis}}(\vec{u}, \vec{v}) = 1$, where $\vec{u}_l = \vec{v}_l = \mu_X^l$ but $u_{l+1} \neq v_{l+1}$ ($\vec{u}_l$ and $\vec{v}_l$ encode μ_X^l).

Leveraging this property, we show that there exists a valid set U with $|U| = k = 1$ and $F_{\mathsf{mono}}(\cdot) \geq B = 1$ iff ψ is true. This is verified by a nontrivial counting argument. Intuitively, to find such a U with a single tuple t_0, we need to inspect 2^m truth assignments t generated by $Q(D)$ and evaluate $\delta_{\mathsf{dis}}(t_0, t)$, which is equivalent to verifying that ψ is true.

For the upper bound, we provide a PSPACE algorithm to determine whether there exists a valid set for given (Q, D, k, F, B), when $\mathcal{L}_Q$ is FO and $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$. □

The diversity ranking problem. We next study the decision problem $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$. It is to assess the rank of a candidate set U for (Q, D, k), *i.e.,* whether $\mathsf{rank}(U) \leq r$ for a given r. From the result below we can see the following.

(1) Like $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, different query languages $\mathcal{L}_Q$ lead to different combined complexity of $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$. In contrast, when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the complexity is inherent to diversification.

(2) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ is coNP-complete for CQ, UCQ and $\exists$FO$^+$, as opposed to NP-complete for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ for the same languages.

Theorem 2. *For* $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$*, when* $F(\cdot)$ *is* $F_{\mathsf{MS}}(\cdot)$ *or* $F_{\mathsf{MM}}(\cdot)$*, the combined complexity is*

- coNP*-complete when* $\mathcal{L}_Q$ *is CQ, UCQ or* $\exists$*FO*$^+$*, and*
- PSPACE*-complete when* $\mathcal{L}_Q$ *is FO.*

However, the combined complexity is

- PSPACE*-complete when* $\mathcal{L}_Q$ *is CQ, UCQ,* $\exists$*FO*$^+$*or FO, when* $F(\cdot)$ *is* $F_{\mathsf{mono}}(\cdot)$*.*

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, and when $\mathcal{L}_Q$ is CQ, we show that $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ is coNP-hard by reduction from the complement of the 3SAT problem. For its upper bound, we show that $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ is in coNP when $\mathcal{L}_Q$ is $\exists$FO$^+$by developing an NP algorithm to check whether $\mathsf{rank}(U) > r$ for a given set U and a rank r.

(2) The proofs of the lower and upper bounds for the following are variations of their counterparts for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$: (a) when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, and $\mathcal{L}_Q$ is FO, and (b) when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$ and $\mathcal{L}_Q$ is CQ, UCQ, $\exists$FO$^+$or FO. □

The result diversity counting problem. We now study the counting problem $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$. Given a bound B, it is to count the number of valid sets for (Q, D, k, F, B), *i.e.,* those sets $U \subseteq Q(D)$ such that $|U| = k$ and $F(U) \geq B$. Here we use the framework of predicate-based counting classes introduced in [18]. For a complexity class C of decision problems, $\#{\cdot}\mathsf{C}$ is the class of all counting problems (*i.e.,* functions) f associated with a C-computable predicate R_L and a polynomial q, such that for every string x, f is to compute the cardinality of the set $\{y \mid R_L(x, y), |y| \leq p(|x|)\}$.

The following result tells us that when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, the problem becomes harder for FO than for CQ, UCQ and $\exists$FO$^+$. In contrast, $F_{\mathsf{mono}}(\cdot)$ has greater impact on the complexity than $\mathcal{L}_Q$. This is consistent with its counterparts for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$. The results are verified by parsimonious reductions. A *parsimonious reduction* from a counting problem $\#A$ to a counting problem $\#B$ is a PTIME function $\sigma(\cdot)$ such that for all x, $|\{y \mid (x, y) \in A\}| = |\{z \mid (\sigma(x), z) \in B\}|$, *i.e.,* $\sigma(\cdot)$ is a bijection [13].

Theorem 3. *For* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$*, when* $F(\cdot)$ *is* $F_{\mathsf{MS}}(\cdot)$ *or* $F_{\mathsf{MM}}(\cdot)$*, the combined complexity is*

- $\#{\cdot}$NP*-complete when* $\mathcal{L}_Q$ *is CQ, UCQ or* $\exists$*FO*$^+$*, and*
- $\#{\cdot}$PSPACE*-complete when* $\mathcal{L}_Q$ *is FO.*

However, when $F(\cdot)$ *is* $F_{\mathsf{mono}}(\cdot)$*, the combined complexity is*

- $\#{\cdot}$PSPACE*-complete for CQ, UCQ,* $\exists$*FO*$^+$ *and FO.*

All the results hold under parsimonious reductions.

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{RDC}(\text{CQ}, F(\cdot))$ is $\#\cdot\mathsf{NP}$-hard by *parsimonious* (one-to-one and surjective) reduction from the $\#\Sigma_1$SAT problem, which is $\#\cdot\mathsf{NP}$-complete [13]. An instance of $\#\Sigma_1$SAT consists of an existentially quantified Boolean formula of the form $\varphi(X, Y) = \exists X \psi(X, Y)$, where ψ is an instance of 3SAT on variables in $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$. It is to count the number of truth assignments of Y that satisfy φ. For the upper bound, it suffices to show that it is in NP to verify whether a given set U is valid for (Q, D, k, F, B).

(2) For $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{RDC}(\text{FO}, F(\cdot))$ is $\#\cdot\mathsf{PSPACE}$-hard by parsimonious reductions from #QBF, which is $\#\cdot\mathsf{PSPACE}$-complete [22]. An instance of #QBF is a formula $\varphi{=}\exists X\ \forall y_1\ P_2 y_2\ \cdots\ P_n y_n\, \psi$, where $P_i \in \{\exists, \forall\}$ for $i \in [2, n]$, and ψ is a quantifier-free Boolean formula over the variables in $\{x_1, \ldots, x_m\}$ and $\{y_1, \ldots, y_n\}$. It is to count the number of truth assignments of X variables that satisfy φ. The upper bound is verified by showing that the problem for deciding whether a given set U is valid is in PSPACE.

(3) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, we show that $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is $\#\cdot\mathsf{PSPACE}$-hard when $\mathcal{L}_Q$ is CQ, and that the problem is in $\#\cdot\mathsf{PSPACE}$ when $\mathcal{L}_Q$ is FO. The lower bound is verified also by parsimonious reduction from #QBF, and by using a counting argument. The upper bound is proven along the same lines as its counterpart for (2) given above. □

4.2 Data Complexity

We next re-investigate these problems for data complexity. As remarked in Section 1, when it comes to data complexity, objective functions dominate the data complexity.

The query result diversification problem. When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, fixing query Q does not simplify the analysis of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ for CQ, UCQ and $\exists\text{FO}^+$: the problem remains NP-complete, the same as its combined complexity. In contrast, fixed queries make our lives easier when

(1) $\mathcal{L}_Q$ is FO, and when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$: $\mathsf{QRD}(\text{FO}, F(\cdot))$ becomes NP-complete; or

(2) when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$: the problem becomes tractable.

Contrast these with the PSPACE-completeness of their combined complexity (Theorem 1). These demonstrate that when Q is fixed, query languages have no impact.

THEOREM 4. *For* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$*, the data complexity is*

- NP*-complete when* $F(\cdot)$ *is* $F_{\mathsf{MS}}(\cdot)$ *or* $F_{\mathsf{MM}}(\cdot)$*, and*
- *in* PTIME *when* $F(\cdot)$ *is* $F_{\mathsf{mono}}(\cdot)$*,*

when $\mathcal{L}_Q$ *is CQ, UCQ,* $\exists\text{FO}^+$ *or FO.*

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{QRD}(\text{CQ}, F(\cdot))$ is NP-hard for fixed *identity queries*, which are a special case of CQ queries (see Section 5 for details), by reduction from the 3SAT problem. We also show that $\mathsf{QRD}(\text{FO}, F(\cdot))$ is in NP for fixed FO queries, by giving an NP algorithm to check the existence of a valid set.

(2) For $F_{\mathsf{mono}}(\cdot)$, we develop a PTIME algorithm to compute a valid set, if it exists. This is possible since when Q is fixed, we can compute $Q(D)$ in PTIME, and hence, for each $t \in Q(D)$, its relevance to Q and its distances to all tuples in $Q(D)$ can also be computed in PTIME. We can simply pick the set U consisting of k tuples with the highest relevance and distances, if there exist k tuples in $Q(D)$, in PTIME. □

The diversity ranking problem. Like $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ (Theorem 4), fixing Q makes $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ simpler only (1) when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, or (2) when $\mathcal{L}_Q$ is FO, and $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$. Its data complexity remains the same as its combined complexity when $\mathcal{L}_Q$ is CQ, UCQ or $\exists\text{FO}^+$, and when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$ (see Theorem 2).

THEOREM 5. *For* $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$*, the data complexity is*

- coNP*-complete when* $F(\cdot)$ *is* $F_{\mathsf{MS}}(\cdot)$ *or* $F_{\mathsf{MM}}(\cdot)$*, and*
- *in* PTIME *when* $F(\cdot)$ *is* $F_{\mathsf{mono}}(\cdot)$*,*

when $\mathcal{L}_Q$ *is CQ, UCQ,* $\exists\text{FO}^+$*or FO.*

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ for CQ is coNP-hard by reduction from the complement of 3SAT by using fixed identity queries. We also show that the problem is in coNP for fixed FO queries, by giving an NP algorithm to check whether $\mathsf{rank}(U) > r$.

(2) For $F_{\mathsf{mono}}(\cdot)$, we give a constructive proof: we provide a PTIME algorithm to check whether $\mathsf{rank}(U) \leq r$ for a given set U and a rank r. The algorithm first finds a collection $\mathcal{S}$ of top-r candidate sets for (Q, D, k), and then checks whether either $U \in \mathcal{S}$ or there exists $S \in \mathcal{S}$ such that $F(U) = F(S)$. It returns true iff one of the conditions holds. When Q is fixed, we can sort $Q(D)$ in PTIME based on $F_{\mathsf{mono}}(\cdot)$, and identify $\mathcal{S}$ in PTIME. In contrast, for $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, we cannot sort $Q(D)$ or find $\mathcal{S}$ in PTIME, since in those cases, the diversity has to be computed for each subset of $Q(D)$. Hence $F_{\mathsf{mono}}(\cdot)$ yields a lower data complexity. □

The result diversity counting problem. When it comes to $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$, fixing queries makes our lives easier:

(1) for $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, RDC becomes $\#\mathsf{P}$-complete under parsimonious reductions, down from $\#\cdot\mathsf{NP}$-complete (for CQ, UCQ and $\exists\text{FO}^+$) and $\#\cdot\mathsf{PSPACE}$-complete (for FO); and

(2) for $F_{\mathsf{mono}}(\cdot)$, RDC is $\#\mathsf{P}$-complete under polynomial Turing reductions, instead of $\#\cdot\mathsf{PSPACE}$-complete (Theorem 3).

Here $\#\mathsf{P}$ is the class of functions that count the number of accepting paths of nondeterministic PTIME Turing machines, in the machine-based framework of [33]. It is known that $\#\mathsf{P} = \#\cdot\mathsf{P}$ [13], where $\#\cdot\mathsf{P}$ is the predicate-based counting class defined with a PTIME predicate (Section 4.1).

A counting problem $\#A$ is *polynomial Turing reducible* to $\#B$ if there exists a PTIME function $\sigma(\cdot)$ such that for all x, $|\{y \mid (x, y) \in A\}|$ is PTIME computable by making multiple calls to an oracle that computes $|\{z \mid (\sigma(x), z) \in B\}|$. Parsimonious reductions (Section 4.1) are stronger than polynomial Turing reductions, *i.e.,* a parsimonious reduction from $\#A$ to $\#B$ is also a polynomial Turing reduction from $\#A$ to $\#B$, but not necessarily vice versa.

Note that a parsimonious reduction from $\#A$ to $\#B$ is also a PTIME reduction from its decision problem A to the decision problem B of $\#B$. Hence if B is in P, $\#B$ *cannot* be $\#\mathsf{P}$-complete under parsimonious reductions since for many NP-complete problems, *e.g.,* 3SAT, their counting problems are in $\#\mathsf{P}$. This is precisely the case for $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$ and $\mathcal{L}_Q$ is CQ, UCQ, $\exists\text{FO}^+$or FO (data complexity). Indeed, its decision problem $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is in P (Theorem 4). Thus $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is $\#\mathsf{P}$-complete under polynomial Turing reductions, but not under parsimonious reductions. In contrast, for $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is $\#\mathsf{P}$-complete under parsimonious reductions.

Theorem 6. *For $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$, the data complexity is*

- *$\#\mathsf{P}$-complete under parsimonious reductions, when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$; and*
- *$\#\mathsf{P}$-complete under polynomial Turing reductions, when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$,*

when $\mathcal{L}_Q$ is CQ, UCQ, $\exists$FO$^+$or FO.

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, we show that $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ for fixed identity queries is already $\#\mathsf{P}$-hard by parsimonious reduction from #SAT. An instance of #SAT is an instance $\varphi(X)$ of 3SAT over a set X of variables. It is to count the number of truth assignments of X that satisfy φ, and is known to be $\#\mathsf{P}$-complete under parsimonious reductions (cf. [27]). For the upper bound, we show that it is in PTIME to verify whether a given set U is a valid set for fixed Q in FO, by the definition of $\#\mathsf{P}$ (*i.e.,* $\#\cdot\mathsf{P}$).

(2) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the proof is more involved. We show that the problem for fixed identity queries is $\#\mathsf{P}$-hard by polynomial Turing reduction from #SSP, the #subset sum problem, which is known to be $\#\mathsf{P}$-complete [5]. Given a finite set W, a function $\pi : W \to \mathbb{N}$ and a natural number $s \in \mathbb{N}$, $\#\mathsf{SSP}$ is to count the number of subsets $T \subseteq W$ such that $\sum_{w\in T} \pi(w) = s$. To do it, we first show that $\#\mathsf{SSPk}$ is $\#\mathsf{P}$-complete by parsimonious reduction from $\#\mathsf{SSP}$, where $\#\mathsf{SSPk}$ is to count the number of sets $T \subseteq W$ such that $|T| = l$ and $\sum_{w\in T} \pi(t) = s$, for a given natural number l. We then show that $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is $\#\mathsf{P}$-hard for fixed identity queries by polynomial Turing reduction from $\#\mathsf{SSPk}$. For the upper bound, we show that it is also in PTIME to verify whether a given set U is valid for (Q, D, k, F, B), for fixed FO queries and for objective given by $F_{\mathsf{mono}}(\cdot)$. □

Summary. From the results above we find the following.

(1) Both query languages and objective functions have impact on the combined complexity of these problems. More specifically, (a) when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, these problems for FO have a higher combined complexity than their counterparts for CQ, UCQ and $\exists$FO$^+$; and (b) for mono-objective formulation, the objective function dominates the combined complexity; moreover, for CQ, UCQ or $\exists$FO$^+$, $F_{\mathsf{mono}}(\cdot)$ makes our lives harder than $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$.

(2) The data complexity is *inherent to result diversification itself*, rather than a consequence of the complexity of query languages. Indeed, the data complexity bounds of these problems remain unchanged when $\mathcal{L}_Q$ is CQ, UCQ, $\exists$FO$^+$or FO, for a given objective function $F(\cdot)$. Moreover, when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ become tractable, but it is not the case for $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$.

5. Special Cases of Result Diversification

In this section we identify and investigate special cases of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$. The reason for studying this is twofold. (1) The results of Section 4 tell us that these problems have rather high complexity. This suggests that we find their special yet practical cases that are tractable. (2) We want to further understand the impact of various parameters of these problems on their complexity, including query languages with low complexity, relevance functions $\delta_{\mathsf{rel}}(\cdot,\cdot)$, distance functions $\delta_{\mathsf{dis}}(\cdot,\cdot)$, and the bound k for selecting query answers.

Identity queries. We first consider $\mathcal{L}_Q$ consisting of identity queries only, *i.e.,* when Q is a CQ query of the form:

$$Q(\vec{x}) = R(\vec{x}),$$

where R is a relation atom, and $|\vec{x}|$ is the arity of R. Note that $D = Q(D)$ for any instance D of schema R. As remarked early, in this setting $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ was shown NP-hard by [17] for max-sum and max-min diversification. No prior work has studied $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ for mono-objective, or $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ for any $F(\cdot)$.

We show that identity queries indeed simplify the analyses of query result diversification, to an extent.

(1) $\mathsf{QRD}(\mathcal{L}_Q, F_{\mathsf{mono}}(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F_{\mathsf{mono}}(\cdot))$ are in PTIME, as opposed to the NP-hardness of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ for $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$ [17], and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ becomes $\#\cdot\mathsf{P}$-complete. In contrast, these problems are PSPACE-complete, PSPACE-complete, and $\#\cdot\mathsf{PSPACE}$-complete (Theorems 1, 2 and 3), respectively, when $\mathcal{L}_Q$ is CQ. This further verifies that query languages have impact on the complexity of diversification.

(2) In contrast, for $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, the combined complexity and data complexity of these problems are the same as their counterparts for CQ. In other words, in this setting, query languages with a low complexity (for its membership problem) do not simplify the diversification analyses.

The result below also tells us that the combined complexity and data complexity of these problems for identity queries coincide with their data complexity counterparts for CQ (Theorems 4, 5 and 6). Intuitively, this is because $Q(D)$ can be computed in PTIME when Q is either an identity query or is fixed. Nonetheless, there are subtle differences between the proofs in these two settings.

Corollary 1. *For identity queries, when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, both the combined and data complexity are*

- *NP-complete for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$,*
- *coNP-complete for $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$, and*
- *$\#\mathsf{P}$-complete for $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ under parsimonious reductions.*

For $F_{\mathsf{mono}}(\cdot)$, both the combined and data complexity are

- *in PTIME for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$,*
- *in PTIME for $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$, and*
- *$\#\mathsf{P}$-complete for $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ under polynomial Turing reductions.*

These are the same as their data complexity given in Theorems 4, 5 and 6, respectively.

Proof sketch: (1) For $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, the lower bounds follow from their counterparts of Theorems 4, 5 and 6, which uses fixed identity queries in the reductions. Moreover, the upper bound proofs given there have shown that if $Q(D)$ is PTIME computable (*e.g.,* when Q is an identity query), these problems are in NP, coNP and $\#\mathsf{P}$, respectively.

(2) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the proofs of Theorems 4, 5 and 6 for mono-objective remain intact for identity queries Q, even when Q is not fixed, since $Q(D)$ and $F_{\mathsf{mono}}(U)$ are both PTIME computable for any set $U \subseteq Q(D)$. Moreover, the lower bound proof of Theorem 6 for $F_{\mathsf{mono}}(\cdot)$ uses a fixed identity query only. Hence those results carry over here. □

When λ = 0. We next study the impact of the relevance and diversity requirements on the complexity of diversification analyses. We show that diversity has bigger impact than relevance, which is consistent with the observation of [36]. We first consider the case when $\lambda = 0$, *i.e.,* when only the relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ is used in $F(\cdot)$ (recall the definitions of $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ and $F_{\mathsf{mono}}(\cdot)$ from Section 2).

(1) These problems have lower data complexity: when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ become tractable for fixed Q. Moreover, $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is in FP for max-min diversification, where FP is the class of all functions that can be computed in PTIME (cf. [27]).

(2) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the combined complexity analyses of these problems become simpler, for CQ, UCQ or $\exists\text{FO}^+$.

Corollary 2. *For $\lambda = 0$, when $F(\cdot)$ is either $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, the combined complexity bounds of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ remain the same as their counterparts given in Theorems 1, 2 and 3, respectively; but their data complexity bounds for CQ, UCQ, $\exists$FO$^+$ and FO are*

- *in* PTIME *for* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$,
- *in* PTIME *for* $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$, *and*
- #P*-complete under polynomial Turing reductions for* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ *when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, but it is in* FP *when $F(\cdot)$ is $F_{\mathsf{MM}}(\cdot)$.*

When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the combined complexity becomes

- NP*-complete for* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ *when $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$, and* PSPACE*-complete when $\mathcal{L}_Q$ is FO;*
- coNP*-complete for* $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ *when $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$, and* PSPACE*-complete for FO; and*
- #·NP*-complete for* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ *when $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$, and* #·PSPACE*-complete for FO.*

Their data complexity bounds remain the same as their counterparts given in Theorems 4, 5 and 6, respectively.

Proof sketch: (1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, the lower bounds of Theorems 1, 2 and 3 are established by taking $\lambda = 0$. Furthermore, the upper bound proofs of those theorems obviously remain intact in the special case when $\lambda = 0$.

(2) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, for the data complexity we show the following. (a) For fixed FO queries, $\mathsf{QRD}(\mathcal{L}_Q, F_{\mathsf{MS}}(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F_{\mathsf{MS}}(\cdot))$ are in PTIME, by presenting corresponding PTIME algorithms. These are possible because in the absence of distance function $\delta_{\mathsf{dis}}(\cdot,\cdot)$, $Q(D)$ can be computed in PTIME (for fixed Q) and can be sorted based on $F(\cdot)$ values in PTIME. (b) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $\mathsf{RDC}(\text{CQ}, F(\cdot)$ is #P-hard by polynomial Turing reduction from #SSPk, which is shown #P-complete in the proof of Theorem 6. Moreover, $\mathsf{RDC}(\text{FO}, F_{\mathsf{MS}}(\cdot))$ is in #P since it is in PTIME to verify whether a set is valid. (c) When $F(\cdot)$ is $F_{\mathsf{MM}}(\cdot)$, we show that the number of valid sets can be computed in PTIME for fixed FO queries by giving such an algorithm, and thus $\mathsf{RDC}(\text{FO}, F(\cdot))$ is in FP. The algorithm leverages certain properties of $F_{\mathsf{MM}}(\cdot)$.

(3) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, for the combined complexity we show that when only relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ is used to define $F(\cdot)$, $F_{\mathsf{mono}}(\cdot)$ and $F_{\mathsf{MS}}(\cdot)$ are equivalent when $k = 2$. Moreover, the lower bounds proofs of Theorems 1, 2 and 3 for $F_{\mathsf{MS}}(\cdot)$ use $\lambda = 0$ and $k = 2$. Therefore, the combined complexity bounds of these problems for $F_{\mathsf{MS}}(\cdot)$ carry over to their counterparts for $F_{\mathsf{mono}}(\cdot)$. In addition, the algorithms given in the upper bound proofs of those theorems also work for $F_{\mathsf{mono}}(\cdot)$ when $\lambda = 0$. Hence the statement follows.

For the data complexity, the PTIME algorithms given for $F_{\mathsf{mono}}(\cdot)$ in the proofs of Theorems 4 and 5 carry over to the special case when $\lambda = 0$. Moreover, the #P lower bound proof of Theorem 6 uses $\lambda = 0$ for $F_{\mathsf{mono}}(\cdot)$, and its corresponding upper bound obviously holds for $\lambda = 0$. □

When $\lambda = 1$. In contrast to Corollary 2, dropping the relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ from $F(\cdot)$ does not make our lives easier. Indeed, in this setting, both the combined complexity and data complexity of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ remain the same as their counterparts when both relevance and diversity are taken into account. This further verifies that the diversity requirement $\delta_{\mathsf{dis}}(\cdot,\cdot)$ dominates the complexity of these problems.

Theorem 7. *When $\lambda = 1$, the combined complexity of Theorems 1, 2 and 3 and the data complexity of Theorems 4, 5 and 6 remain unchanged for* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ *and* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$, *respectively, when the objective function $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ or $F_{\mathsf{mono}}(\cdot)$, and when the query language $\mathcal{L}_Q$ is CQ, UCQ, $\exists$FO$^+$ or FO.*

Proof sketch: The proofs of the results for $\lambda = 1$ are more involved than their counterparts for $\lambda = 0$. While the upper bounds of Theorems 1–6 carry over to this special case, the lower bounds require new proofs when λ is fixed to be 1.

(1) When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$, Theorems 4, 5 and 6 remain intact for the data complexity. Indeed, those lower bounds are verified when λ is set to be 1.

We prove the following lower bounds for the combined complexity when $\lambda = 1$, *i.e.,* only $\delta_{\mathsf{dis}}(\cdot,\cdot)$ is used in $F(\cdot)$, with different reductions when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$.

(1.1) We show that $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ is NP-hard by reduction from 3SAT, when $\mathcal{L}_Q$ is CQ. The proof is different from its counterpart given in the proof of Theorem 1, which uses the relevance function $\delta_{\mathsf{rel}}(\cdot,\cdot)$ in the reduction (when $\lambda = 0$).

(1.2) We verify that $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ is coNP-hard by reduction from the complement of 3SAT, when $\mathcal{L}_Q$ is CQ.

(1.3) We prove that $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ is #·NP-hard by parsimonious reduction from #Σ_1SAT, when $\mathcal{L}_Q$ is CQ.

(1.4) When $\mathcal{L}_Q$ is FO, we show that both $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ are PSPACE-hard by reduction from the membership problem for FO, using $\delta_{\mathsf{dis}}(\cdot,\cdot)$ only.

(1.5) When $\mathcal{L}_Q$ is FO, we verify that $\mathsf{RDC}(\text{CQ}, F_{\mathsf{MS}}(\cdot))$ is #·PSPACE-hard by parsimonious reduction from #QBF.

(2) When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the combined complexity bounds of Theorems 1, 2 and 3 remain intact here, since their lower bounds are established when λ is set to 1. In addition, the PTIME algorithms given in the proofs of Theorems 4 and 5 for mono-objective functions still work when $\lambda = 1$. Moreover, the #P upper bound of Theorem 6 remains the same when $\lambda = 1$. In light of these, we only need to prove:

(2.1) $\mathsf{RDC}(\mathcal{L}_Q, F_{\mathsf{mono}}(\cdot))$ is #P-hard for CQ, by polynomial Turing reduction from #SSPk (see Theorem 6). □

When k is a predefined constant. Finally, we study the impact of the cardinality $|U|$ of the selected sets U of query answers. When $|U|$ is required to be a predefined *constant* k, the result below tells us the following.

(1) When Q is also fixed, $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ are all tractable. That is, fixing the size of U simplifies their data complexity analyses.

(2) In contrast, fixing k does not make our lives easier when it comes to combined complexity.

COROLLARY 3. *For a predefined constant k,*

- *the combined complexity bounds given in Theorems 1, 2 and 3 remain unchanged for* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ *and* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$*, respectively; and*
- *the data complexity is in*
  - PTIME *for* $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$,
  - PTIME *for* $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$*, and*
  - FP *for* $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$,

no matter whether $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ or $F_{\mathsf{mono}}(\cdot)$, and when $\mathcal{L}_Q$ ranges over CQ, UCQ, $\exists$FO$^+$and FO.

Proof sketch: (1) For the combined complexity, the lower bounds of Theorems 1, 2 and 3 are established by using $k = 2$ when $F(\cdot) = F_{\mathsf{MS}}(\cdot)$, and $k = 1$ when $F(\cdot)$ is $F_{\mathsf{MM}}(\cdot)$ or $F_{\mathsf{mono}}(\cdot)$. Hence these lower bounds remain intact when k is a constant. Moreover, the upper bounds of these problems obviously hold in the special case when k is a constant.

(2) When it comes to the data complexity, observe the following. (a) When query Q is fixed, $Q(D)$ is PTIME computable. (b) When k is a constant, there are only polynomially many subsets of $Q(D)$ that consist of k elements. Putting these together, we can develop PTIME algorithms for $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$. Hence these problems are in PTIME, PTIME and FP, respectively, when Q is fixed and k is a constant. □

Summary. From these results we can see the following.

(1) The impact of $\mathcal{L}_Q$. When $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, problems $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ have higher combined complexity for FO than their counterparts for CQ, UCQ and $\exists$FO$^+$(Theorems 1, 2 and 3). In contrast, the complexity bounds remain intact when $\mathcal{L}_Q$ is CQ or the class of identity queries (Corollary 1).

When $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$, the combined complexity bounds of these problems remain unchanged when $\mathcal{L}_Q$ subsumes CQ (Theorems 1, 2 and 3). In contrast, for the class of identity queries, all these problems become simpler (Corollary 1).

Query languages $\mathcal{L}_Q$ have no impact on the data complexity of these problems (Theorems 4, 5, 6 and Corollary 1).

(2) The impact of $\delta_{\mathsf{rel}}(\cdot, \cdot)$ and $\delta_{\mathsf{dis}}(\cdot, \cdot)$. The complexity of diversification also arises from the diversity requirement. The absence of the distance function $\delta_{\mathsf{dis}}(\cdot, \cdot)$ simplifies (a) the data complexity analyses of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ when $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$ or $F_{\mathsf{MM}}(\cdot)$, (b) the data complexity of $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ for $F_{\mathsf{MM}}(\cdot)$, and (c) the combined complexity analyses of all these problems when $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$ (Corollary 2). In contrast, the absence of $\delta_{\mathsf{rel}}(\cdot, \cdot)$ does not make our lives easier (Theorem 7).

(3) The impact of k. When k is a fixed constant, the data complexity analyses of $\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$, $\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ and $\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ become tractable, no matter whether $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ or $F_{\mathsf{mono}}(\cdot)$ (Corollary 3).

6. Conclusions

We have extended the result diversification model of [17] by incorporating query languages, without assuming the entire set $Q(D)$ of query answers as input. We have identified three decision and counting problems in connection with query result diversification. We have established the upper and lower bounds of these problems, *all matching*, for both combined complexity and data complexity, when the query language $\mathcal{L}_Q$ is CQ, UCQ, $\exists$FO$^+$or FO, and when $F(\cdot)$ ranges over the three objective functions $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ and $F_{\mathsf{mono}}(\cdot)$ given in [17]. We have also studied special cases of these problems, and identified tractable cases.

The main complexity results are summarized in Table 1 annotated with their corresponding theorems. The complexity bounds of special cases are shown in Table 2, when they differ from their unrestricted counterparts. These results provide a comprehensive picture of the complexity of the analyses of query result diversification. The tables also show the impact of various factors on the complexity of diversification analyses, such as query languages $\mathcal{L}_Q$, objective functions $F(\cdot)$, relevance functions $\delta_{\mathsf{rel}}(\cdot, \cdot)$, distance functions $\delta_{\mathsf{dis}}(\cdot, \cdot)$, and bound k on the number of answers. As remarked earlier, these results may help practitioners decide what query language, objective function and query set should be supported when developing diversification models and recommender systems, and help vendors evaluate their products and adjust their stock of items.

Several extensions are targeted for future work. First, query result diversification analyses are mostly intractable. We need to identify more special cases that are practical and tractable. Second, we also need to determine whether these problems are approximation hard, and develop approximate algorithms when it is possible. Finally, in practice one may want to incorporate user preferences [8, 31] into the diversification model. While we may encode certain preferences in, *e.g.,* the relevance and distance functions $\delta_{\mathsf{rel}}(\cdot, \cdot)$ and $\delta_{\mathsf{dis}}(\cdot, \cdot)$, this issue deserves a full treatment.

Acknowledgments. Deng is supported in part by the 863 2012AA011203, NSFC 91118008, 61103031 and MOST project 2012BAH46B04 of China. Fan is supported in part by EPSRC EP/J015377/1 of the UK, the 973 Program 2012CB316200 and NSFC 61133002 of China.

7. References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Leong. Diversifying search results. In *WSDM*, 2009.

[4] S. Amer-Yahia. Recommendation projects at Yahoo! *IEEE Data Eng. Bull.*, 34(2):69–77, 2011.

[5] G. Berbeglia and G. Hahn. Counting feasible solutions of the traveling salesman problem with pickups and deliveries is #P-complete. *Discrete Applied Mathematics*, 157(11):2541–2547, 2010.

[6] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166, 2012.

[7] G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri. Efficient diversification of Web search results. In *VLDB*, pages 451–459, 2011.

[8] Z. Chen and T. Li. Addressing diverse user preferences in SQL-query-resul navigation. In *SIGMOD*, 2007.

[9] E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl. DivQ: Diversification for keyword search over struc-

Table 1: Combined complexity and data complexity

Objective functions	Languages	Problems		
		$\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$ (Th. 1, 4)	$\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$ (Th. 2, 5)	$\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$ (Th. 3, 6)
		Combined complexity		
$F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$	CQ, UCQ, $\exists$FO$^+$	NP-complete	coNP-complete	#·NP-complete
	FO	PSPACE-complete	PSPACE-complete	#·PSPACE-complete
$F_{\mathsf{mono}}(\cdot)$	CQ, UCQ, $\exists$FO$^+$, FO	PSPACE-complete	PSPACE-complete	#·PSPACE-complete
		Data complexity		
$F_{\mathsf{MS}}(\cdot)$ and $F_{\mathsf{MM}}(\cdot)$	CQ,UCQ,$\exists$FO$^+$,FO	NP-complete	coNP-complete	#P-complete (parsimonious)
$F_{\mathsf{mono}}(\cdot)$	CQ,UCQ,$\exists$FO$^+$,FO	PTIME	PTIME	#P-complete (Turing)

Table 2: Special cases

Conditions	Complexity	Problems		
		$\mathsf{QRD}(\mathcal{L}_Q, F(\cdot))$	$\mathsf{DRP}(\mathcal{L}_Q, F(\cdot))$	$\mathsf{RDC}(\mathcal{L}_Q, F(\cdot))$
identity queries; $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$	combined (Cor. 1)	PTIME	PTIME	#P-complete (Turing)
$\lambda = 0$; $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$	data (Cor. 2)	PTIME	PTIME	#P-complete (Turing)
$\lambda = 0$; $F(\cdot)$ is $F_{\mathsf{MM}}(\cdot)$	data (Cor. 2)	PTIME	PTIME	FP
$\lambda = 0$; $F(\cdot)$ is $F_{\mathsf{mono}}(\cdot)$; $\mathcal{L}_Q$ is CQ, UCQ or $\exists$FO$^+$	combined (Cor. 2)	NP-complete	coNP-complete	#·NP-complete
k is a constant; $F(\cdot)$ is $F_{\mathsf{MS}}(\cdot)$, $F_{\mathsf{MM}}(\cdot)$ or $F_{\mathsf{mono}}(\cdot)$	data (Cor. 3)	PTIME	PTIME	FP

tured databases. In *SIGIR*, pages 331–338, 2010.

[10] T. Deng, W. Fan, and F. Geerts. On the complexity of package recommendation problems. In *PODS*, 2012.

[11] M. Drosou and E. Pitoura. Diversity over continuous data. *IEEE Data Engineering Bulletin*, 32(4), 2009.

[12] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010.

[13] A. Durand, M. Hermann, and P. G. Kolaitis. Subtractive reductions and complete problems for counting complexity classes. *TCS*, 340(3):496–513, 2005.

[14] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *JCSS*, 66(4):614–656, 2003.

[15] FindGift. http://www.findgift.com.

[16] P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k bounded diversification. In *SIGMOD*, 2012.

[17] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[18] L. A. Hemaspaandra and H. Vollmer. The satanic notations: Counting classes beyond #P and other definitional adventures. *SIGACT News*, 26(1):2–13, 1995.

[19] I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4):11:1–11:58, 2008.

[20] W. Jin and J. M. Patel. Efficient and generic evaluation of ranked queries. In *SIGMOD*, pages 601–612, 2011.

[21] G. Koutrika, B. Bercovitz, and H. Garcia-Molina. FlexRecs: expressing and combining flexible recommendations. In *SIGMOD*, pages 745–758, 2009.

[22] R. E. Ladner. Polynomial space counting problems. *SIAM J. Comput.*, 18(6):1087–1097, 1989.

[23] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD*, pages 467–476, 2009.

[24] C. Li, M. A. Soliman, K. C.-C. Chang, and I. F. Ilyas. RankSQL: Supporting ranking queries in relational database management systems. In *VLDB*, 2005.

[25] Z. Liu, P. Sun, and Y. Chen. Structured search result differentiation. In *VLDB*, 2009.

[26] E. Minack, G. Demartini, and W. Nejdl. Current approaches to search result diversification. In *ISWC*, 2009.

[27] C. H. Papadimitriou. *Computational Complexity*. AW, 1994.

[28] A. G. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *TOIS*, 29(4), 2011.

[29] O. A. Prokopyev, N. Kong, and D. L. Martinez-Torres. The equitable dispersion problem. *European Journal of Operational Research*, 197(1):59–67, 2009.

[30] K. Schnaitter and N. Polyzotis. Evaluating rank joins with optimal cost. In *PODS*, pages 43–52, 2008.

[31] K. Stefanidis, M. Drosou, and E. Pitoura. Perk: Personalized keyword search in relational databases through preferences. In *EDBT*, pages 585–596, 2010.

[32] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *TODS*, 36(3), 2011.

[33] L. Valiant. The complexity of computing the permanent. *TCS*, 8(2):189 – 201, 1979.

[34] M. Y. Vardi. The complexity of relational query languages. In *STOC*, pages 137–146, 1982.

[35] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *ICDE*, pages 228–236, 2008.

[36] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. On query result diversification. In *ICDE*, 2011.

[37] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. Composite recommendations: From items to packages. *FCS*, 6(3):264–277, 2012.

[38] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: Diversification in recommender systems. In *EDBT*, pages 368–378, 2009.

[39] C. Yu, L. V. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *ICDE*, 2009.

[40] M. Zhang and N. Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *RecSys*, 2008.

[41] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.