Beyond Skyline and Ranking Queries: Restricted Skylines (Extended Abstract)

Paolo Ciaccia[1] and Davide Martinenghi[2]

[1]Università di Bologna, Italy, email: paolo.ciaccia@unibo.it
[2]Politecnico di Milano, Italy, email: davide.martinenghi@polimi.it

Abstract. Traditionally, skyline and ranking queries have been treated separately as alternative ways of discovering interesting data in potentially large datasets. While ranking queries adopt a specific scoring function to rank tuples, skyline queries return the set of non-dominated tuples and are independent of attribute scales and scoring functions. Ranking queries are thus less general, but cheaper to compute and widely used. In this paper, we integrate these two approaches under the unifying framework of *restricted skylines* by applying the notion of dominance to a set of scoring functions of interest.

1 Introduction

When simultaneously optimizing different criteria (e.g., the different attributes in a dataset), a problem known as *multi-objective optimization*, three approaches are prevalent [4]: *(1) ranking queries*, which focus on a scoring function expressing the relative importance of the different attributes; *(2) lexicographical queries*, which establish a strict priority among the attributes; *(3) skyline queries*, which return all the non-dominated objects (t dominates s iff t is no worse than s on all the attributes, and strictly better on at least one). As argued in the literature [4], each of these methods has pros and cons (also refer to Table 1).

Table 1: Pros and cons of multi-objective optimization approaches.

Evaluation criteria ↓ Queries →	Ranking	Lexicographic	Skyline
Simplicity of formulation	No	Yes	Yes
Overall view of interesting results	No	No	Yes
Control of result cardinality	Yes	Yes	No
Trade-off among attributes	Yes	No	No
Relative importance of attributes	Yes	Yes	No

Ranking queries heavily depend on the particular choice of weights in the scoring function, and thus fail to offer an overall view of the dataset. Lexicographic queries enforce a linear priority between attributes, thus even the smallest difference in the most important attribute cannot be compensated by the other attributes (a problem inherited by newer approaches, such as *p-skylines* [5]). Skyline queries provide a good overview of potentially interesting tuples, but may contain too many objects.

SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy.

Restricted skyline (R-skyline) queries, introduced in [2], consider arbitrary families of scoring functions (induced, e.g., by constraints on the weights), thus allowing greater flexibility than skylines and ranking queries. The linchpin of R-skylines is the novel concept of $\mathcal{F}$-*dominance*: t $\mathcal{F}$-dominates s when t is always better than or equal to s according to *all* the scoring functions in the family $\mathcal{F}$ (and strictly better for at least one scoring function in $\mathcal{F}$).

In this paper, we present a synthesis of the results in [2], where we introduced two R-skyline operators: ND, characterizing the set of non-$\mathcal{F}$-dominated tuples; PO, referring to the tuples that are potentially optimal, i.e., best according to some function in $\mathcal{F}$. R-skylines capture in a single framework all the practically relevant approaches to multi-objective optimization, traditionally dealt with separately, and enable the study of other scenarios of practical interest. For example, decision makers may encounter objectives in which the model parameters are characterized by complex preferences, perhaps coming from preference elicitation from a crowd (see, e.g., [3] and references therein for strategies for collecting preferences between tuples).

2 Restricted Skylines

We consider a relational schema $R(A_1, \ldots, A_d)$, with $d \geq 1$, where, w.l.o.g., each attribute value ranges in $[0, 1]$; we refer to an instance r over R.

The *skyline* of r, denoted $\textsc{Sky}(r)$, is equivalently defined *(i)* as the set of all *non-dominated* tuples (Eq. (1)), or *(ii)* as the set of *potentially optimal* tuples, i.e., those that are better than all the others according to at least one monotone scoring function [1] (Eq. (2)).

$$\textsc{Sky}(r) = \{t \in r \mid \nexists s \in r.\ s \prec t\} \tag{1}$$

$$\textsc{Sky}(r) = \{t \in r \mid \exists f \in \mathcal{M}.\ \forall s \in r.\ s \neq t \rightarrow f(t) < f(s)\} \tag{2}$$

where $s \prec t$ means "s dominates t", and $\mathcal{M}$ is the set of monotone scoring functions. While the former view is typically adopted for skylines, the latter is commonly applied to "top-k" queries (here with $k = 1$), i.e., those queries whose goal is to return the k best tuples according to a given scoring function.

We now introduce R-skylines, whose behavior is similar to Sky, but applied to a limited set of monotone scoring functions $\mathcal{F} \subseteq \mathcal{M}$. To this end, we say that t $\mathcal{F}$-*dominates* s, $s \neq t$, denoted by $t \prec_{\mathcal{F}} s$, iff $\forall f \in \mathcal{F}.\ f(t) \leq f(s)$.

Example 1. Consider the tuples $t = \langle 0.5, 0.5 \rangle$, $s = \langle 0, 1 \rangle$, the monotone scoring functions $f_1(x, y) = x + y$ and $f_2(x, y) = x + 2y$, and the set $\mathcal{F} = \{f_1, f_2\}$. We have $t \prec_{\mathcal{F}} s$, since $f_1(t) = f_1(s) = 1$ and $f_2(t) = 1.5 < f_2(s) = 2$. However, $t \not\prec_{\mathcal{M}} s$, since $\mathcal{M}$ includes $f_3(x, y) = 2x + y$, for which $f_3(t) = 1.5 > f_3(s) = 1$.

The *non-dominated restricted skyline* of r wrt. $\mathcal{F}$, denoted by $\textsc{nd}(r; \mathcal{F})$, is defined as the set of non-$\mathcal{F}$-dominated tuples:

$$\textsc{nd}(r; \mathcal{F}) = \{t \in r \mid \nexists s \in r.\ s \prec_{\mathcal{F}} t\}. \tag{3}$$

Here, as a convention, we consider lower values to be better than higher ones.

Note that the right-hand side of Eq. (3) is similar to that of Eq. (1), where $\prec$ has been replaced by $\prec_{\mathcal{F}}$. Observe that, clearly, $\prec_{\mathcal{M}}$ coincides with $\prec$.

The *potentially optimal restricted skyline* of r wrt. $\mathcal{F}$, denoted by $\textsc{po}(r;\mathcal{F})$, returns the tuples that are best (top 1) according to some scoring function in $\mathcal{F}$.

$$\textsc{po}(r;\mathcal{F}) = \{t \in r \mid \exists f \in \mathcal{F}.\ \forall s \in r.\ s \neq t \rightarrow f(t) < f(s)\}. \tag{4}$$

The right-hand side of Eq. (4) is as in Eq. (2), with $\mathcal{F}$ instead of $\mathcal{M}$.

The following containment relationships between R-skylines and Sky hold:

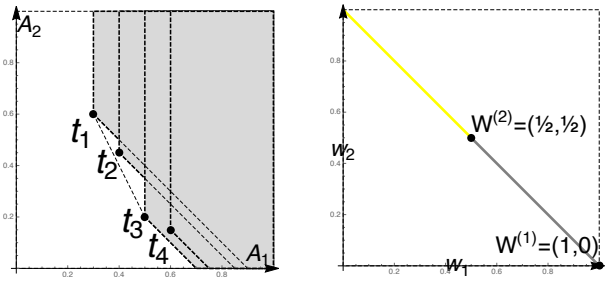
$$\textsc{po}(r;\mathcal{F}) \subseteq \textsc{nd}(r;\mathcal{F}) \subseteq \textsc{Sky}(r). \tag{5}$$

$$\textsc{po}(r;\mathcal{M}) = \textsc{nd}(r;\mathcal{M}) = \textsc{Sky}(r). \tag{6}$$

The *$\mathcal{F}$-dominance region* $DR(t;\mathcal{F})$ of a tuple t under $\mathcal{F}$ is the set of all points in $[0,1]^d$ that are $\mathcal{F}$-dominated by t:

$$DR(t;\mathcal{F}) = \{s \in [0,1]^d \mid t \prec_{\mathcal{F}} s\}. \tag{7}$$

Such a region grows larger for smaller sets: $DR(t;\mathcal{F}_1) \supseteq DR(t;\mathcal{F}_2)$, if $\mathcal{F}_1 \subseteq \mathcal{F}_2$.



(a) Tuples from Example 2 in $[0,1]^d$, $d = 2$. $\mathcal{F}$-dominance regions in gray.

(b) Region of normalized weights such that $w_1 \geq w_2$ (in gray).

Fig. 1: Example 2 – tuples and weights in $[0,1]^d$, $d = 2$, $\mathcal{C} = \{w_1 \geq w_2\}$, $\mathcal{F} = \mathcal{L}_1^{\mathcal{C}}$, where $\mathcal{L}_1$ is the set of monotone scoring functions that are weighted sums of attribute values.

Example 2. Let $\mathcal{F}$ be the set of linear scoring functions $f(x,y) = w_1 x + w_2 y$ with $w_1 \geq w_2$ and let r consist of $t_1 = \langle 0.3, 0.6\rangle$, $t_2 = \langle 0.4, 0.45\rangle$, $t_3 = \langle 0.5, 0.2\rangle$, $t_4 = \langle 0.6, 0.15\rangle$ (Figure 1a). We have $\textsc{po}(r;\mathcal{F}) = \{t_1, t_3\} \subseteq \textsc{nd}(r;\mathcal{F}) = \{t_1, t_2, t_3\} \subseteq \textsc{Sky}(r) = r$. Indeed, no tuple in r dominates any other tuple in r, and so $\textsc{Sky}(r) = r$. However, $t_3 \prec_{\mathcal{F}} t_4$, since $f(t_3) \leq f(t_4)$ holds for any $f \in \mathcal{F}$, as $w_1(0.5 - 0.6) \leq w_2(0.15 - 0.2)$ always holds when $w_1 \geq w_2$. Therefore $t_4 \notin \textsc{nd}(r;\mathcal{F})$. Figure 1a shows in gray the region of $[0,1]^d$ whose points (including tuple t_4) are $\mathcal{F}$-dominated by some tuple in r, i.e., $\cup_{t \in r} DR(t;\mathcal{F})$; Figure 1b shows in gray the region of normalized weights such that $w_1 \geq w_2$.

Finally, with linear scoring functions, as is well known [7], top-1 tuples can only lie in the boundary of the convex hull of the $\mathcal{F}$-dominated region, thus $t_2 \notin \text{PO}(r;\mathcal{F})$, since there is no function $f \in \mathcal{F}$ for which both $f(t_2) < f(t_1)$ and $f(t_2) < f(t_3)$, as there are no w_1, w_2 such that $w_1(0.4-0.3) < w_2(0.6-0.45)$, $w_1(0.4-0.5) < w_2(0.2-0.45)$, and $w_1 \geq w_2$ all hold.

3 Restricted Skylines and L_p Norms

A practically relevant case to consider is that of the *weighted* L_p norms, defined as follows, where $W = (w_1, \ldots, w_d) \in \mathcal{W}$ is a normalized weight vector, i.e., $\mathcal{W} \subseteq [0,1]^d$ and, for each $W = (w_1, \ldots, w_d) \in \mathcal{W}$, we have $\sum_{i=1}^{d} w_i = 1$:

$$L_p^W(t) = \left(\sum_{i=1}^{d} w_i t[A_i]^p \right)^{1/p}, \quad p \in \mathbb{N}. \tag{8}$$

We therefore turn our attention to the case in which the set of monotone scoring functions coincides with the family $\mathcal{L}_p$ of weighted L_p norms:

$$\mathcal{L}_p = \{L_p^W \mid W \in \mathcal{W}\}, \quad p \in \mathbb{N}. \tag{9}$$

The behaviors of ND and PO are very different under $\mathcal{L}_p$.

Theorem 1. *For every value of p and every r,* $\text{ND}(r;\mathcal{L}_p) = \text{SKY}(r)$.

Thus, any $\mathcal{L}_p$ family is "powerful enough" to reveal all skyline points with ND. However, this does not hold for PO, as indicated in the following theorems.

Theorem 2. *Let $p < p'$, with $p, p' \in \mathbb{N}$. Then, for every r,* $\text{PO}(r;\mathcal{L}_p) \subseteq \text{PO}(r;\mathcal{L}_{p'})$.

Theorem 3. *For each $p \in \mathbb{N}$, there exists r such that* $\text{PO}(r;\mathcal{L}_p) \subset \text{SKY}(r)$.

The results of Theorems 1, 2 and 3 suggest that, by imposing some constraints on the weights, one can use any $\mathcal{L}_p$ family to smoothly move from the skyline (no constraints) to top-1 queries (a single weight vector satisfies the constraints).

Checking $\mathcal{F}$-dominance for $\mathcal{L}_p$ norms with linear constraints on the weights can be done in PTIME. Let $\mathcal{L}_p^{\mathcal{C}}$ indicate the set $\mathcal{L}_p$ with linear constraints $\mathcal{C} = \{C_1, \ldots, C_c\}$ on weights, where $C_j = \sum_{i=1}^{d} a_{ji} w_i \leq k_j$ (for $j \in \{1, \ldots, c\}$).

Theorem 4 ($\mathcal{F}$-dominance test). *$t \prec_{\mathcal{L}_p^{\mathcal{C}}} s$ iff the following linear programming problem in the unknowns $W = (w_1, \ldots, w_d)$ has a non-negative solution:*

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{d} w_i (s[A_i]^p - t[A_i]^p) & \\ \text{subject to} & w_i \in [0,1] & i \in \{1, \ldots, d\} \\ & \sum_{i=1}^{d} w_i = 1 & \\ & \sum_{i=1}^{d} a_{ji} w_i \leq k_j & j \in \{1, \ldots, c\}. \end{array} \tag{10}$$

Computing $\text{ND}(r;\mathcal{F})$ using Theorem 4 is likely to be time-consuming, since a different linear programming (LP) problem needs to be solved for each $\mathcal{F}$-dominance test. An alternative approach is to explicitly compute the $\mathcal{F}$-dominance regions of tuples, and then discard those tuples that belong to at least one of such regions. The advantage of this approach is that the computation of the $\mathcal{F}$-dominance region of a tuple t can be performed just once, thus independently of how many $\mathcal{F}$-dominance tests involve t.

In order to compute $DR(t;\mathcal{L}_p^{\mathcal{C}})$, we observe that the subset of $\mathcal{W}$ that satisfies $\mathcal{C}$, denoted $\mathcal{W}(\mathcal{C})$, is a convex polytope contained in the unit $(d-1)$-simplex.

Theorem 5 ($\mathcal{F}$-dominance region). *Let $W^{(1)},\ldots,W^{(q)}$ be $\mathcal{W}(\mathcal{C})$'s vertices, with $W^{(\ell)}=(w_1^{(\ell)},\ldots,w_d^{(\ell)})$ for $\ell\in\{1,\ldots,q\}$. The dominance region $DR(t;\mathcal{L}_p^{\mathcal{C}})$ of a tuple t under $\mathcal{L}_p^{\mathcal{C}}$ is the locus of points s defined by the q inequalities:*

$$\sum_{i=1}^{d} w_i^{(\ell)} s[A_i]^p \geq \sum_{i=1}^{d} w_i^{(\ell)} t[A_i]^p, \quad \ell\in\{1,\ldots,q\}. \tag{11}$$

Example 3. Let $d=2$, $p=1$, and consider tuples $t_1=\langle 0.3,0.6\rangle$, $t_3=\langle 0.5,0.2\rangle$, $t_4=\langle 0.6,0.15\rangle$ from Example 2. For $\mathcal{C}=\{w_1\geq w_2\}$ and considering that $w_1+w_2=1$ and $0\leq w_1,w_2\leq 1$, the vertices of $\mathcal{W}(\mathcal{C})$ are $W^{(1)}=(1,0)$ and $W^{(2)}=(0.5,0.5)$. Figure 1a shows the tuples along with their $\mathcal{L}_1^{\mathcal{C}}$-dominance regions, while Figure 1b shows $\mathcal{W}(\mathcal{C})$. By Theorem 5, $DR(t_3;\mathcal{L}_1^{\mathcal{C}})$ is characterized by the system of inequalities:

$$\{s[A_1]\geq 0.5,\quad s[A_1]+s[A_2]\geq 0.7\}. \tag{12}$$

Tuple t_4 satisfies (12) and thus $t_3\prec_{\mathcal{L}_1^{\mathcal{C}}} t_4$. For tuple t_1, the system becomes:

$$\{s[A_1]\geq 0.3,\quad s[A_1]+s[A_2]\geq 0.9\}. \tag{13}$$

Here, t_4 does not satisfy (13) and therefore $t_1\not\prec_{\mathcal{L}_1^{\mathcal{C}}} t_4$.

As Example 3, Figure 1a and Inequalities (11) suggest, the "shape" of $DR(t;\mathcal{L}_p^{\mathcal{C}})$ (modulo cropping in the $[0,1]^d$ hypercube) is independent of t, since the left-hand sides are the same and the right-hand sides are, for any given t, a constant. The only significant overhead introduced by this approach is the enumeration of the vertices of $\mathcal{W}(\mathcal{C})$. However, this has to be done just once for *all* tuples.

For any set $\mathcal{F}$, $\text{PO}(r;\mathcal{F})$ can be computed from $\text{ND}(r;\mathcal{F})$ by retaining only the tuples that are not $\mathcal{F}$-dominated by any "virtual" tuple obtained by combining other tuples in $\text{ND}(r;\mathcal{F})$. When $\mathcal{F}=\mathcal{L}_p^{\mathcal{C}}$, this is done by solving an LP problem.

Theorem 6 (PO test). *Let $\text{ND}(r;\mathcal{L}_p^{\mathcal{C}})=\{t_1,t_2,\ldots,t_\sigma,t\}$. Then, $t\in\text{PO}(r;\mathcal{L}_p^{\mathcal{C}})$ iff there is no convex combination s of $t_1,\ldots,t_\sigma$ such that $s\prec_{\mathcal{L}_p^{\mathcal{C}}} t$, i.e., iff the following linear system in the unknowns $\alpha=(\alpha_1,\ldots,\alpha_\sigma)$ is unsatisfiable:*

$$\begin{aligned} \textstyle\sum_{i=1}^{d} w_i^{(\ell)}\left(\sum_{j=1}^{\sigma}\alpha_j t_j[A_i]^p\right) &\leq \textstyle\sum_{i=1}^{d} w_i^{(\ell)} t[A_i]^p \quad \ell\in\{1,\ldots,q\} \\ \alpha_j &\in [0,1] \quad j\in\{1,\ldots,\sigma\} \\ \textstyle\sum_{j=1}^{\sigma}\alpha_j &= 1. \end{aligned} \tag{14}$$

Algorithm 1: `SVE1F` for ND.

Input: *relation* r, *constraints* $\mathcal{C}$, *family* $\mathcal{F} = \mathcal{L}_p^{\mathcal{C}}$. Output: $\text{ND}(r; \mathcal{F})$.

1. **let** `ND` $:= \emptyset$; **let** $W^{(1)}, \ldots, W^{(q)}$ be the vertices of $\mathcal{W}(\mathcal{C})$
2. sort r using the coordinates of the centroid of $\mathcal{W}(\mathcal{C})$ as weights
3. **for each** s **in** r *// candidate $\mathcal{F}$-dominated tuple*
4. compute left-hand sides of Inequalities (11)
5. **for each** t **in** `ND` *// candidate $\mathcal{F}$-dominant tuple*
6. **if** $t \prec s \vee t \prec_{\mathcal{F}} s$ **then continue** to line 3
7. **let** `ND` $:=$ `ND` $\cup \{s\}$
8. **return** `ND`

Besides the $\mathcal{L}_p^{\mathcal{C}}$ family, Theorems 4, 5, and 6 also hold for *any* set $\mathcal{F}$ whose functions are weighted sums of monotone functions of single attributes and monotonic transforms thereof.

4 Algorithms and Experiments

We considered several algorithmic opportunities for the computation of ND.

- Appropriately sorting the dataset beforehand produces a topological sort with respect to the $\mathcal{F}$-dominance relation. With that, if tuple t precedes tuple s in the sorted relation, then $s \not\prec_{\mathcal{F}} t$.
- The $\mathcal{F}$-dominance test can be executed by either *i)* solving an LP problem as in Theorem 4 or *ii)* checking whether $t \in DR(s; \mathcal{F})$ as in Theorem 5 through vertex enumeration of the polytope $\mathcal{W}(\mathcal{C})$.
- We can either *i)* first compute Sky and then remove $\mathcal{F}$-dominated tuples (two phases), or *ii)* integrate dominance and $\mathcal{F}$-dominance tests (one phase).

The best choice is to use sorting, vertex enumeration and a single phase. The corresponding pseudocode is shown in Algorithm 1, called `SVE1F`. The main idea is to scan the tuples sortedly and to populate a current window `ND` of non-$\mathcal{F}$-dominated tuples. No tuple will ever be removed from `ND`, as no tuple can be $\mathcal{F}$-dominated by a tuple found later in the sorted relation. The vertices of the polytope $\mathcal{W}(\mathcal{C})$ are computed just once (line 1). Algorithm 1 interleaves dominance and $\mathcal{F}$-dominance tests, thus performing, for each new tuple s, a single pass over `ND`. Every candidate $\mathcal{F}$-dominated tuple s (line 3) is compared against every candidate $\mathcal{F}$-dominant tuple t (line 6) to decide whether s should be added to `ND`. The $\mathcal{F}$-dominance test of line 6 is done via Theorem 5 and, thus, it is useful to precompute the left-hand sides of Inequality (11) already at line 4.

For the computation of $\text{PO}(r; \mathcal{F})$, we start from the tuples in $\text{ND}(r; \mathcal{F})$ and, by Theorem 6, we discard any tuple t that is $\mathcal{F}$-dominated by a convex combination of tuples in $\text{ND}(r; \mathcal{F}) \setminus \{t\}$. However, directly checking $\mathcal{F}$-dominance via (14) may be prohibitively time consuming when $\sigma = |\text{ND}(r; \mathcal{F})| - 1$ is large. We therefore try, in Algorithm 2 (`POND`, i.e. PO via ND), to reduce as early as possible the set of candidate potentially optimal tuples (`PO`) by adopting the following heuristics: *i)* we start with a convex combination of only $\tilde{\sigma} = 2$ tuples (line 2), which will give rise to smaller, faster-to-solve systems (14) for testing $\mathcal{F}$-dominance; as long

Algorithm 2: POND for PO.

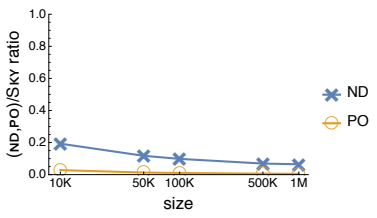
Input: *relation* r, *constraints* $\mathcal{C}$, *family* $\mathcal{F} = \mathcal{L}_p^{\mathcal{C}}$. Output: $\text{PO}(r; \mathcal{F})$.

1. **let** PO := $\text{ND}(r; \mathcal{F})$ // *via* SVE1F
2. **let** $\tilde{\sigma}$:= 2; **let** *lastRound* := **false**
3. **while**($\neg$*lastRound*)
4. **if** $\tilde{\sigma} \geq |\text{PO}| - 1$ **then** *lastRound* := **true**
5. **for each** t **in** PO in reverse order // *candidate* $\mathcal{F}$*-dominated tuple*
6. **if** $\exists s.\, s \prec_{\mathcal{F}} t$, s is convex comb. of the first $\min(\tilde{\sigma}, |\text{PO}|-1)$ tuples in PO$\backslash\{t\}$ **then let** PO:=PO$\backslash\{t\}$
7. **let** $\tilde{\sigma}$:= $\tilde{\sigma} \cdot 2$
8. **return** PO

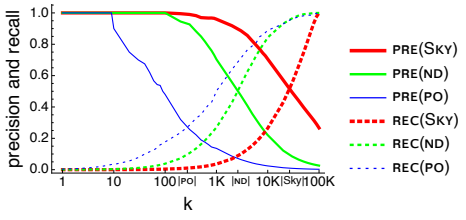
as $\tilde{\sigma} < |\text{PO}| - 1$, this condition is only sufficient for pruning, but not necessary; after each round, we double $\tilde{\sigma}$ (line 7); *ii)* we sortedly enumerate candidate $\mathcal{F}$-dominated tuples from PO in reverse order (line 5), as the worst tuples wrt. the ordering are the most likely to be $\mathcal{F}$-dominated; *iii)* using linear system (14), we check the existence of a convex combination of the *first* $\tilde{\sigma}$ tuples in PO (line 6), as they are the best wrt. the ordering and thus more likely to $\mathcal{F}$-dominate other tuples. In the last round (enabled by line 4) all the remaining tuples are checked against a convex combination of all the other tuples still in PO, which is now a necessary and sufficient condition for pruning, as in Theorem 6.

In [2], efficiency and effectiveness of the proposed algorithms have been measured on both real and synthetic scenarios with varying *i)* data distribution, *ii)* dataset size, *iii)* number of dimensions, and *iv)* number of constraints. With default settings (anticorrelated data, $N = 100K$ tuples, $d = 6$ dimensions, $\mathcal{F} = \mathcal{L}_1^{\mathcal{C}}$ family with $|\mathcal{C}| = 3$ constraints), our algorithms incur sub-second execution times. Such times increase as N increases, d increases and $|\mathcal{C}|$ decreases.

Both ND and PO reduce the size of the result wrt. SKY (Fig. 2a). For default values, their effectiveness is remarkable (9.8% of the SKY points are in ND and only 1% in PO). Somewhat surprisingly, computing ND via SVE1F may require much less time than computing SKY, because $\mathcal{F}$-dominance tests, albeit more costly than dominance tests, are more effective in pruning redundant tuples.



(a) $|\text{ND}|/|\text{SKY}|$ and $|\text{PO}|/|\text{SKY}|$ cardinality ratios as dataset size N varies.



(b) Precision and recall of top-k queries wrt. SKY, ND, and PO. Centroid of the $\mathcal{W}(\mathcal{C})$ as weight vector; $N = 100K$ tuples in the dataset.

Fig. 2: R-skylines vs. skyline and ranking queries (anticorrelated data with $d = 6$, $\mathcal{F} = \mathcal{L}_1^{\mathcal{C}}$, $|\mathcal{C}| = 3$).

Let $\mathcal{T}_k(r; f)$ indicate the set of top-k tuples in relation r wrt. a scoring function f. The *precision* of $\mathcal{T}_k(r; f)$ wrt. a set $\mathcal{S}$ is defined as $\text{PRE}(\mathcal{S}) =$

$|\mathcal{S} \cap \mathcal{T}_k(r; f)|/k$, whereas the *recall* is $\text{REC}(\mathcal{S}) = |\mathcal{S} \cap \mathcal{T}_k(r; f)|/|\mathcal{S}|$. Figure 2b shows precision and recall when f is the most representative weighted sum in $\mathcal{F}$ (i.e., f's weight vector is the centroid of $\mathcal{W}(\mathcal{C})$). Recall grows with k, while precision decreases. With standard parameter values, when $k = |\mathcal{S}|$ (and thus precision and recall are equal), REC(ND) is 50%, while REC(PO) is 38%. Retrieving the entire ND with a top-k query may however require to scan almost the entire dataset. As expected, top-k queries incur a small fraction of the execution time of `SVE1F` (below 7%).

5 Discussion

In this paper, we have presented a framework aiming to unify skyline and ranking queries. We have done so by introducing two R-skyline operators implementing the notions of non-dominated (ND) and potentially optimal (PO) tuples with respect to a set of scoring functions $\mathcal{F}$. The greater flexibility of these operators captures not only standard skyline and top-1 queries, but also constraints on the weights in the scoring functions, which are highly relevant in practice.

It is well known that choosing the "right" weights for a scoring function is a difficult task for users, since it is usually hard to predict the effects on ranking of changing one or more parameters. Replacing precise values with constraints on weights, as R-skylines do, is therefore a viable way to alleviate the problem.

We have shown that both ND and PO are very effective in focusing on tuples of interest, even in very large datasets. In practical cases, computing ND requires no more than a few seconds. For the more complex problem of computing PO, we have developed heuristics to reduce the number of LP problems to be solved.

Natural extensions of this framework include the notions of top-k query and k-skyband, with $k > 1$ [6].

References

1. J. Chomicki, P. Ciaccia, and N. Meneghetti. Skyline queries, front and back. *SIGMOD Record*, 42(3):6–18, 2013.
2. P. Ciaccia and D. Martinenghi. Reconciling skyline and ranking queries. *PVLDB*, 10(11):1454–1465, 2017.
3. E. Ciceri et al. Crowdsourcing for top-k query processing over uncertain data. *TKDE*, 28(1):41–53, 2016.
4. A. A. Freitas. A critical review of multi-objective optimization in data mining: a position paper. *SIGKDD Explorations*, 6(2):77–86, 2004.
5. D. Mindolin and J. Chomicki. Preference elicitation in prioritized skyline queries. *VLDB J.*, 20(2):157–182, 2011.
6. D. Papadias et al. Progressive skyline computation in database systems. *TODS*, 30(1):41–82, 2005.
7. M. A. Soliman et al. Ranking with uncertain scoring functions: semantics and sensitivity measures. In *SIGMOD*, pages 805–816, 2011.