Poster Paper

Maximum Coverage Representative Skyline

Malene S. Søholm
Aarhus University, Denmark
soeholm@cs.au.dk

Sean Chester
NTNU, Norway
sean.chester@idi.ntnu.no

Ira Assent
Aarhus University, Denmark
ira@cs.au.dk

ABSTRACT

Skyline queries represent a dataset by the points on its pareto frontier, but can become very large. To alleviate this problem, representative skylines select exactly k skyline points. However, existing approaches are not *scale-invariant*, not *stable*, or must materialise the entire skyline.

We introduce the *maximum coverage representative skyline*, which returns the k points collectively dominating the largest area of the data space. It satisfies the above properties and reflects a critical property of the skyline itself.

1. INTRODUCTION

Grasping large datasets can be overwhelming. The skyline query [2] helps by summarizing a dataset with only those points that represent the pareto frontier of the data. A point p is on the pareto-frontier (and thus in the skyline) if it is not *dominated* by some other point q, i.e., p is better than all non-equal points in at least one attribute. However, even this is often not enough. The skyline may grow quite large: e.g., on high dimensional data, points have more opportunities (dimensions) on which to be better than other points.

In order to solve this, several approaches have been proposed. Given an integer k, a *ranking skyline* [3, 9, 10] returns the k points with the highest score according to a skyline-based utility function. However, the full skyline must be retrieved, which, even using highly parallel computation on a GPU, can still take several seconds [1].

Regret minimising sets [4, 7] return the k points for which a worst-case linear utility function evaluates to a score on the subset as closely as possible to the one on the skyline. Computing such a set also requires knowing the skyline.

Existing approaches for *representative skylines* [5, 6, 8] return the k skyline points best *representing* the full skyline, but require knowing the skyline to be calculated: the number of skyline points between all pairs of representative skyline points [6]; the maximum distance from any non-representative skyline point to its nearest representative [8]; or the k skyline points maximising the number of non-skyline points dominated [5]. Also, [8] is not *scale-invariant*, e.g., scaling miles to kilometres distorts the result, and [5] is not *stable*, i.e., "junk" non-skyline points can be added to manipulate the representative skyline. In this paper, we introduce the first representative skyline to avoid all of these pitfalls.

2. MAXIMIZING COVERAGE

The skyline is defined as the subset of non-dominated points [2]. The skyline also has various interesting properties, e.g. that it dominates the rest of the dataset. In [5], this property is emphasized and a representative skyline is developed, but it is not stable as mentioned above.

Another property is that no other subset of points dominates a larger area of the data space. I.e, the skyline captures *the contour of the data space occupied by the data.* This property is agnostic to non-skyline points, suggesting inherent stability. Thus we introduce the *maximum coverage representative skyline* (MCRS): the size-k set that dominates the largest area of the data space. It is the set of k points that best achieves this critical property of the skyline.

Note that the MCRS is necessarily a subset of the skyline (at least if k is smaller than the size of the skyline), since every non-skyline point dominates less area than the skyline points that dominate it. The MCRS is also both stable and scale-invariant, since neither adding/removing non-skyline points nor scaling the dataset in any dimension affects the relative size of the dominance area. Perhaps most appealingly, the MCRS is skyline-agnostic: there is no inherent dependence on knowing the skyline to compute the optimal MCRS: the size of the area collectively dominated by any given set of points is unrelated to knowing the full skyline.

Formally, if *dom-area*(p) denotes the area occupied by the (infinitely many) points $q \in \mathbb{R}^d$ dominated by p, then the MCRS of size k on dataset $\mathcal{D}$ is[1]:

$$\text{MCRS}(\mathcal{D}, k) = \underset{\mathcal{S} \subseteq \mathcal{D}, |\mathcal{S}|=k}{\text{argmax}} \left| \bigcup_{p \in \mathcal{S}} \textit{dom-area}(p) \right| \qquad (1)$$

Figure 1 gives an example: the MCRS of size 2 is $\{p_1, p_3\}$, since p_1 and p_3 cover an area of 42, whereas p_1 and p_2 only cover 40 and p_2 and p_3 only cover 34.

An algorithm for 2d In the following, we show that in two dimensions, the MCRS can be computed in time $\mathcal{O}(m^3k + nlogn)$ and space $\mathcal{O}(mk + n)$, where $n = |\mathcal{D}|$ and

[1] Note, importantly, that this set formulation avoids multiply counting area dominated by more than one point in a set $\mathcal{S}$.

EDBT '16 Bordeaux, France

Series ISSN: 2367-2005

10.5441/002/edbt.2016.95

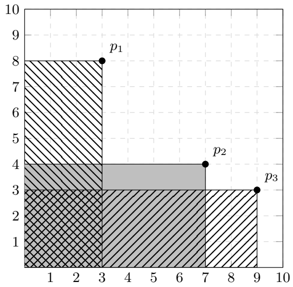
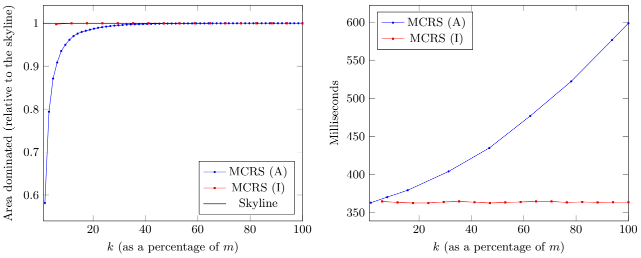


Figure 1: Three skyline points p_1, p_2, and p_3 and their corresponding dominance areas. The optimal size-2 MCRS is indicated by the striped area.



(a) Representativeness

(b) Execution time

Figure 2: Evaluation of the MCRS

$m = |\text{skyline}(\mathcal{D})|$. This is a nice asymptotic result given that the naive search space is $\mathcal{O}(n^k)$. The algorithm proceeds in three steps: First, sort $\mathcal{D} \bigcup \{\langle 1, 0\rangle\}$ so that $p_0 \leq, ..., \leq p_n$; then, discard points dominated by their predecessors and relabel points $p_0, \ldots, p_m$; the MCRS will now be the value of $\text{MCRS}(m, m, k+1) \setminus \{p_m\}$ in the following recursion:

$$\text{MCRS}(x, y, 0) = \{p_y\}$$
$$\text{MCRS}(x, y, \kappa), y \geq \kappa, = \{p_0, \ldots, p_{\kappa-1}\}$$
$$\text{MCRS}(x, y, \kappa) = \underset{s \in \left\{\text{MCRS}(x, \hat{y}, \kappa) \bigcup \{p_y\}, \text{MCRS}(x-1, y, \kappa)\right\}}{\text{argmax}} |\textit{dom-area}(s)|,$$

where:

$$\hat{y} = \underset{0 \leq y' < y}{\text{argmax}}\ \textit{area}\left(\langle p_x.x, p_{y'}.y\rangle, \langle 1, p_y.y\rangle\right) + \left|\textit{dom-area}\left(\text{MCRS}(x, y', \kappa - 1)\right)\right|.$$

The intuition behind the recursion is to sweep through all pairs of skyline points, calculating for each pair the best solution that dominates all the space that it dominates. Because dominance is transitive, the result for each pair of points is very similar to those for nearby pairs of points. One can see this as traversing column-by-column the intersection points of the grid-partitioning induced by the skyline. (In Figure 1 the sequence $[(9,0), (9,3), (7,0), (7,3), (7,4), (3,0), (3,3), (3,4), (3,8), (0,0), (0,3), (0,4), (0,8)]$.)

By adding the sentinel $p_m = \langle 1, 0\rangle$ with $|\textit{dom-area}(p_m)| = 0$ to the end of the list, the last column aggregates the best solution from the entire grid. Using dynamic programming to solve the recursion leads to the asymptotic results.

This non-indexing algorithm first computes the skyline to improve efficiency. However, that does not imply computing the skyline first is always more efficient; an index may permit prioritising promising regions of the data space. Also, this algorithm computes the *optimal* solution, not a greedy approximation, thereby allowing us to study the proposed model itself (rather than just the algorithm's efficiency).

3. REPRESENTATIVENESS OF AN MCRS

In this section, we evaluate the MCRS concept and algorithm. We generate independent (I) and anti-correlated (A) datasets of 1 million 2-dimensional points as per [2], which have 17 and 64 skyline points each.[2] We implement the algorithm in `C++` and execute it on a machine with an Intel Core i7-4770K 3.50GHz CPU and 16GB of memory.

Figure 2a shows the dominance area of the MCRS relative to the skyline and Figure 2b shows execution time, both as a function of k. (I): The MCRS almost exactly represents the skyline, even at $k = 1$, with stable execution time. (A): It quickly approaches the skyline, dominating > 90 % with fewer than 10 % of the points. Computing representations with ≤ 60 % of the skyline takes less than a half-second.

4. CONCLUSION AND FUTURE WORK

We introduced the *maximum coverage representative skyline* (MCRS), a scale-invariant, stable, skyline-agnostic representative skyline, achieving what the skyline achieves. We gave an efficient algorithm to compute an optimal $2d$ MCRS with which we illustrated that the MCRS covers much of the data space as the full skyline, even for small k.

We will extend this work with algorithms for $> 2d$ and multi-dimensional indexes, e.g. R-tree extensions, that can exploit the independence of the MCRS from the skyline.

5. REFERENCES

[1] K. S. Bøgh, S. Chester, and I. Assent. Work-efficient parallel skyline computation for the GPU. *PVLDB*, 8(9):962–973, 2015.

[2] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.

[3] C.-Y. Chan, H. Jagadish, K.-L. Tan, A. Tung, and Z. Zhang. On high dimensional skylines. In *EDBT*, pages 478–495. Springer, 2006.

[4] S. Chester, A. Thomo, S. Venkatesh, and S. Whitesides. Computing k-regret minimizing sets. *PVLDB*, 7(5):389–400, 2014.

[5] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. Selecting stars: The k most representative skyline operator. In *ICDE*, pages 86–95, 2007.

[6] M. Magnani, I. Assent, and M. L. Mortensen. Taking the big picture: representative skylines based on significance and diversity. *VLDB J*, 23(5):795–815, 2014.

[7] D. Nanongkai, A. D. Sarma, A. Lall, R. J. Lipton, and J. Xu. Regret-minimizing representative databases. *PVLDB*, 3(1–2):1114–1124, 2010.

[8] Y. Tao, L. Ding, X. Lin, and J. Pei. Distance-based representative skyline. In *ICDE*, pages 892–903, 2009.

[9] G. Valkanas, A. N. Papadopoulos, and D. Gunopulos. Skyline ranking à la IR. In *EDBT/ICDT Workshops*, pages 182–187, 2014.

[10] A. Vlachou and M. Vazirgiannis. Link-based ranking of skyline result sets. In *M-Pref Workshop*, 2007.

[2] Correlated $2d$ skylines are already sufficiently small.