CoSky: A Practical Method for Ranking Skylines in Databases

Hana Alouaoui, Lotfi Lakhal, Rosine Cicchetti and Alain Casali
Laboratoire d'Informatique et Système, CRNS UMR 7020, Aix Marseille Université, France

Keywords: Databases, IR, MCDA, Ranking, Skylines.

Abstract: Discovering Skylines in Databases have been actively studied to effectively identify optimal tuples/objects with respect to a set of designated preference attributes. Few approaches have been proposed for ranking the skylines to resolve the problem of the high cardinality of the result set. The most recent approach to rank skylines is the dp-idp (dominance power- inverse dominance power) which extensively uses the Pareto-dominance relation to determine the score of each skyline. The dp-idp method is in the very same spirit as tf-idf weighting scheme from Information Retrieval. In this paper, we firstly make an Enrichment of dp-idp with Dominance Hierarchy to facilitate the determination of Skyline scores, we propose then the CoSky method (Cosine Skylines) for fast ranking skylines in Databases without computing the Pareto-dominance relation. Cosky is a TOPSIS-like method (Technique for Order of Preference by Similarity to Ideal Solution) resulting from the cross-fertilization between the fields of Information Retrieval, Multiple Criteria Decision Analysis, and Databases. The innovative features of CoSky are principally: the automatic weighting of the normalized attributes based on Gini index, the score of each skyline using the Saltons cosine of the angle between each skyline object and the ideal object, and its direct implementation into any RDBMS without further structures. Finally, we propose the algorithm DeepSky, a Multilevel skyline algorithm based on CoSky method to find Top-k ranked Skylines.

1 INTRODUCTION

Skyline computation, previously known as Pareto sets and maximal vectors (Bentley et al., 1978), has received a great attention in the statistical and mathematical fields since many past decades. The skyline computation is crucial to many multi-criteria decision making applications. Therefore, skyline queries have attracted considerable attention in the context of databases, especially with the introduction of skyline operator by (Borzsonyi et al., 2001). These queries are simple and expressive. They do not need user-defined scoring functions. They only require the user preferences concerning the minimization or the maximization of attribute values. Suppose a customer who wishes to buy a car and he is seeking for a car with high power, low mileage and low price. Nevertheless, these criteria of selecting Cars are complementary since cars of higher power and lower mileage are more expensive. In order to find such cars, we must query the corresponding Cars database relation (Table 1). Let price, mileage (klm) and power be the attributes of Cars, the users prefer to minimize the price and the Mileage (klm) and maximize the power by selecting items that are better than others regarding these three attributes.

Table 1: Database Relation Cars.

idcar	price	klm	power
C_1	25	10	8
C_2	20	30	6
C_3	25	15	7
C_4	5	40	7
C_5	25	45	5
C_6	45	15	6
C_7	35	40	5
C_8	45	45	4

Here is an example of a skyline query (with skyline operator) on the relation Cars:

```
SELECT * FROM Cars SKYLINE OF price MIN, klm MIN, power MAX;
```

The associated SQL query without skyline operator is:

```
SELECT * FROM Cars Car1
WHERE NOT EXISTS (
 SELECT * FROM Cars Car2
 WHERE (
   Car2.price =< Car1.price
   AND Car2.klm =< Car1.klm
```

Alouaoui, H., Lakhal, L., Cicchetti, R. and Casali, A.
CoSky: A Practical Method for Ranking Skylines in Databases.
DOI: 10.5220/0008363005080515
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 508-515
ISBN: 978-989-758-382-7

```
  AND Car2.power >= Car1.power)
AND (Car2.price < Car1.price
  OR Car2.klm < Car1.klm
  OR Car2.power > Car1.power));
```

As a result, a set of good cars is returned (C_1, C_2, C_4). Good cars in our case is the set of cars which are as good or better in all dimensions (price, klm and power) and better in at least one dimension. This set of good points (cars) forms the Skylines. The Pareto-dominance specifies which data points belong to the skyline and it can be formalized as follows:

Let r be a relational table, with $A_1,\ldots,A_m$ attributes. A preference over A_j is an expression of one of two forms: Pref (A_i) = `min` or Pref (A_i) = `max`. Let p and q be two tuples of r. We say that p dominates q (denoted $p \prec_d$ q) if and only if $p.A1 \leq q.A1,\ldots,p.A_m \leq q.A_m$ and $\exists j \in [1..m] : p.A_j < q.A_j$

With:

$$(\preceq,\prec) = \begin{cases} (\leq,<) \text{ iff } pref(A_j) = min \\ (\geq,>) \text{ iff } pref(A_j) = max \end{cases} \tag{1}$$

In other words, an object (tuple) p dominates another object q if it is as good in all attributes, and is strictly better in at least one attribute. The Skyline is a set S of tuples which are not dominated by any other tuple. $S = \{t \in r | \text{t is not dominated}\}$

One of the major issues of the skyline operator is the high cardinality of the result set which does not offer any interesting insights. All objects are equally interesting and there is no significant discrimination between them. In order to face this obstacle, an efficient ranking of skyline objects has become a compelling need. This solution is efficient especially in the case of high dimensional or anti-correlated data and participates in reducing the huge size of the result set. Our contribution lies within this scope. In this paper, we introduce a novel approach that aims on one hand at enriching an IR- style ranking mechanism based on dp-idp scoring scheme. Our enrichment is founded on the integration of a Dominance Hierarchy (DH) in order to improve the calculation of skyline scores and their afterward ranking. And on the other hand, we propose our CoSky (Cosine Skylines) approach that handles with skyline objects ranking without favoring any dominance relationship. CoSky is a TOPSIS-like method (the Technique for Order of Preference by Similarity to Ideal Solution (Lai et al., 1994)) is a cross-fertilization between the fields of Information Retrieval, Multiple Criteria Decision Analysis, and Databases. CoSky innovations are summarized by the following steps: the attributes normalization using Gini index, the automatic weighting of the normalized attributes: they do not need user-defined weighting attributes as in TOPSIS method (Tscheikner-Gratl et al., 2017), the calculation of each skyline score using the Salton's cosine of the angle between each skyline object and the ideal object, and its direct implementation into SQL without further structures.

The rest of the paper is organized as follows: section 2 gives an overview of the methods proposed in the literature to extract and rank the skyline objects. In section 3, we explain the dp-idp method and we point out its weaknesses and the difficulties grasped while calculating the scores and defining the layers of minima. In section 4, we describe our proposal of enrichment and extension of dp-idp Ranking mechanism based on Dominance Hierarchy pre-computation. In section 5, we present our non-dominance based approach The Cosky method, we describe the main steps and we discuss the obtained results given by a running example. The algorithm DeepSky, a Multilevel skyline algorithm to find Top-k ranked Skylines that have k highest scores is described in section 6. Finally, we sum up the main conclusions of this paper as well as point out directions for future works.

2 RELATED WORK

In addition to the skyline queries, a panoply of algorithms have been proposed in order to meet skyline constraints which vary with the studied computational domain. In (Borzsonyi et al., 2001), the Block Nested Loop (BNL) algorithm is proposed in database context; its principle is based on a window (memory block with limited space) of size w. This window stores the first points that are undominated in each pass. Passes are made over the data until obtaining all the skyline points and each dominated point is eliminated to not be read in the future. In case the window is full, a temporary disk file is used to hold the candidate objects. The BNL algorithm is provided of a timestamping mechanism allowing it to find out when a point is in the skyline and when all points it dominates were eliminated. In (Spyratos et al., 2012), authors propose an approach to compute the skyline of a relational table taking into account preferences expressed over one or more attributes. This approach does not consider the table structure or the tuples indexing. It is based on query lattice concept presented and explained in the paper. An algorithm is developed to construct the skyline as the union of answers to a subset of queries from that lattice without directly accessing the table R. To rank the query, they consider the maximum path from the root query to another query q. The higher the rank of a query the less the tuples in its answer are preferred. Two index-based algorithms namely; Bitmap and Index are introduced in (Tan et al., 2001). Bitmap uses the bitmap encoding to determine the dominating points. The Index approach is based on the partitioning of different objects into sorted lists. The sorting parameter is the minimum coordinate. The lists are then indexed by a B-tree. These algorithms return skyline points in a fixed order which cannot be adapted to the users preferences. In (Papadias et al., 2005), the Branch and Bound Skyline (BBS) algorithm is proposed, this algorithm is based on nearest-neighbor search and only nodes containing skyline points are accessed. BBS is simple to implement due to its progressiveness and I/O optimality. The SaLSa algorithm (Bartolini et al., 2007) is a natural extension of SFS (Chomicki et al., 2003) algorithm, whose originality is the ability of computing the result without applying the computation of the Pareto-dominance relation to all the objects. This is achieved by pre-sorting the data using a monotone (as SFS) limiting function, and then checking that unread data are all dominated by a so-called stop point (object). A Randomized Skyline algorithm (RAND) is presented in (Das Sarma et al., 2009). RAND is a multi-pass streaming algorithm which takes into account randomized I/Os. A comparison between RAND and other known algorithms is given and shows that it performs in the case of minor and simple variations in the input (*eg.* Perturbations of the data orders) while the other algorithms do not return significant results with such variations. As we mentioned above, the huge cardinality of a skyline set is a main obstacle that a decision maker faces. In order to avoid it, an efficient selection of skyline objects has to be performed. Numerous works have been devoted to study the ranking of skylines. In (Chan et al., 2006), a metric called skyline frequency is proposed in order to rank the skyline by retaining the interesting points with high skyline frequency. This method scales well with dimensionality unlike other ranking methods. Experiments show that the proposed algorithm runs faster than other algorithms and returns correct results even when considering a huge number of dimensions

An alternative to skyline queries is presented in (Yiu and Mamoulis, 2007); the top-*k* dominating queries. These queries are proposed as a ranking method not based on a scoring function. The top-k dominating queries are evaluated in a multi-dimensional data context.

In (Bartolini et al., 2007), a ranking approach applied on an image Database is introduced. The technique is based on the shaping of what the user is looking for by specifying user defined regions that dominate all other regions. Authors show that the obtained results are as good as those based on scoring functions and that the proposed approach provides a perfect running time.

(Vlachou and Vazirgiannis, 2010) propose a ranking approach of skyline objects for a SKYCUBE (Lakhal et al., 2017) in order to focus on the most informative objects. A Skyline graph is built up and captures the dominance relationships between skyline objects belonging to different subspaces. In (Chen et al., 2015), a new operator is introduced in order to find the most desirable skyline object (MDSO). A ranking criterion is formalized, it considers the number of the non-skyline objects dominated by a skyline object s. The higher this criterion is, the more interesting the skyline object is. To process MDSO queries, three algorithms are proposed namely; Cell Based algorithm (CB), Sweep Based algorithm (SB), and Reuse Based algorithm (RB). They return the most desirable k skyline objects.

3 SKYLINE RANKING BASED ON DP-IDP

The dp-idp (dominance power- inverse dominance power), is inspired by the tf-idf weighting scheme from Information Retrieval which assigns to a term t a weight in a document d. The idea is not to determine the number of occurrences of each query term t in d, but instead the tf-idf weight of each term in d. The aim is to find important keywords in a document corpus. In the skyline context, dominated points impact skyline point differently. Consequently, these points have not the same importance. Their contribution depends on some local (per skyline point) and some global characteristics (the entire skyline), similarly to tf-idf.

The explored idea in (Valkanas et al., 2014) is that a point's importance has to be inversely proportional to the number of skyline points that dominate it. The dp-idp scheme takes into account the relative positions of the dominated points in order to differentiate between them. It is centered on points that are not dominated by many others: *e.g.* if $sp \prec_d p_1, sp \prec_d p_2$, and p_1 and p_2 do not dominate each other, they contribute equally to sp. Otherwise, if $p_1 \prec_d p_2$, then $score(p_1) > score(p_2)$, consequently the contribution of p_1 is more considerable. The idp of a point $p \in D \backslash S)$ is the number of skyline points which dominate p. A point p is important if it does not appear frequently in a skylines point dominated set :

$$idp(p) = log \frac{|S|}{|\{sp \in S : sp \prec_d p\}|} \tag{2}$$

To measure the dp of a dominated point p, its relative position to the skyline point sp is very important and has to be taken into account. Hence, the same dominated point may contribute differently to different skyline points. For this reason, we have to find the layer of $minimalm(p,sp)$ where the dominated point p is located with respect to sp. The dominance power of p is then given by the inverse of the layer where it lies : The score (sp) which measures the importance of a skyline point sp is given by the following formula:

$$Score(sp) = \sum_{p:sp \prec_d p} \frac{1}{lm(p,sp)} log \frac{|S|}{|\{sp' \in S : sp' \prec_d p\}|} \qquad (3)$$

The Baseline algorithm (Valkanas et al., 2014) is proposed to rank the skyline on the basis of dp-idp scheme. The main steps of this algorithm are:

1. Extracting the minimal layers of each skyline point sp;
2. Considering each point (p) in each layer of minima lm (p, sp), the number of skyline points dominating it has to be found. The score sp is then updated;
3. Sorting the skyline on the basis of the returned scores.

This algorithm is time consuming and has lots of shortcomings as mentioned in (Valkanas et al., 2014). But the most important weakness according to us, is the difficulty of the calculation of the layer of minima which may lead to wrong results. For this reason, we propose an enrichment of this approach in order to make the Baseline algorithm faster .

At this state, our objective is to ameliorate the dominance based approach (dp-idp) and to improve the skyline ranking (Section 4). Further, in this paper, we will introduce our non-dominance based approach and we will discuss its performance in ranking skyline objects without referring to any dominance relation calculation (Section 5 and Section 6).

4 ENRICHMENT OF DP-IDP WITH DOMINANCE HIERARCHY

The dominance relation can be seen as an hierarchical sorting, i.e. A skyline point has necessarily a hierarchical position superior to its dominated points. This assumption motivated us to map the dominance relationship, studied above, to a graph that we call Dominance Hierarchy. The integration of DH into the dp-idp skyline r anking method permits a better computation of layers of minima (first step in the baseline algorithm) and consequently leads to better ranking results. Our proposed graph is a kind of Directed Acyclic Graph (DAG) used to give a presentation of a partially ordered set (poset) by drawing its coverage graph (i.e, graph which represents the same reachability relation of the main graph but with the fewest possible edges). The DAG has a topological ordering which may give an excellent presentation of the Hierarchy evolution from a skyline point sp to its dominated points.

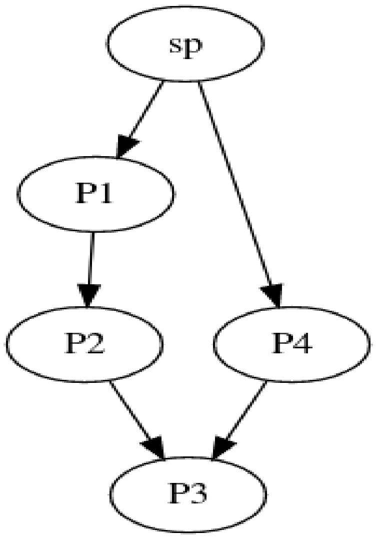

Figure 1: An example of a DH graph.

Given a set of objects (D) and an order of dominance $\prec_d$, DH is the coverage graph of the ordered set $(D, \prec_d)$.

Table 2: Scores Calculation.

sp	Dominated points p	Lm (p, sp)	Score (sp)
C_1	C_3	$Lm(C_3,C_1) = 2$	0.297
	C_5	$Lm(C_5,C_1) = 3$	
	C_6	$Lm(C_6,C_1) = 3$	
	C_7	$Lm(C_7,C_1) = 3$	
	C_8	$Lm(C_8,C_1) = 4$	
C_2	C_5	$Lm(C_5,C_2) = 2$	0
	C_7	$Lm(C_7,C_2) = 2$	
	C_8	$Lm(C_8,C_2) = 3$	
C_4	C_5	$Lm(C_5,C_4) = 2$	0
	C_7	$Lm(C_7,C_4) = 2$	
	C_8	$Lm(C_8,C_4) = 3$	

The DH is a direct acyclic graph (cf. Figure 1) contains as vertex the skyline point (sp). The order of Dominance is illustrated by the edges between sp and

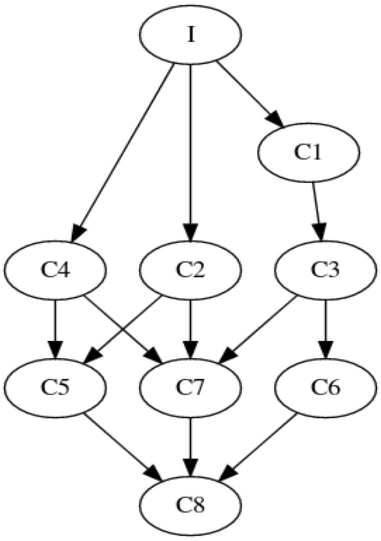

Figure 2: An example of a DH graph.

its dominated points (p_1, p_2, p_3 and p_4).

We define the layer of $minima(p, sp)$ as the number of vertices of the minimal path from sp to p in DH. To compute the layer of $minima(p3, sp)$, there are two paths from sp to p_3 as shown in the graph ($cf.$ Figure 1):

1. First path: $sp \rightarrow p_1 \rightarrow p_2 \rightarrow p_3$ ‘
2. Second path: $sp \rightarrow p_4 \rightarrow p_3$

The first path contains 4 vertices and the second one contains 3 vertices, then the minimal path from sp to p_3 is the second one and $lm(p_3, sp) = 3$. Lets consider another example, the DH graph in Figure 2 illustrates the dominance relations between the skyline points and the dominated points of the Cars database relation. I is the ideal point dominating all skylines.

To compute the score, we use formula 3. Thus, the obtained rank is : C_1, C_2, C_4 or C_1, C_4, C_2.

On the basis of the obtained results ($cf.$ Table 2), we conclude that the dp-idp method is unable to distinguish two skylines dominating objects which are dominated by all skylines as it is the case here of the skylines C_2 and C_4.

5 THE COSKY METHOD

In order to solve the ranking problem, we propose the CoSky method for ranking skylines in Databases. CoSky (COsine Skylines) is a multi-steps approach and it is not based on dominance relation calculation. CoSky is the first TOPSIS-like method applied to ranking the skylines. TOPSIS (Lai et al., 1994), (Behzadian et al., 2012) is based on a vectorial normalisation, a user-weight calculation of each attribute, and the score of each object uses a geometric calculation of the distances between each alternative (object) and the ideal/anti-ideal solutions. In CoSky method, the normalisation of the attributes is based on the sum, an automatic weighting of the normalized attributes based on Gini index, and the score uses the Salton's cosine of the angle between each skyline object and the ideal object. More calculation details are given later in this section. In the rest of the paper, we consider that $i \in [1..n]$ and $j \in [1..m]$ (where n is the number of tuples and m is the number of attributes).

5.1 Step 1: Attributes Normalization based on the Sum

The Skylines are normalized in the range between 0 and 1 to eliminate anomalies with different measurement units and scales. This process transforms different scales and units among various attributes (or criterias) into common measurable units to make these attributes comparables.

Let us consider the tuple $p_i = <v_{i1}, v_{i2}, .., v_{im}> \in$ rSkyNorm (Normalized skylines), then we have :

$$v_{ij} = \frac{t_i[A_j]}{\sum_{i=1}^{n} t_i[A_j]}, \forall t_i \in rSky(Skylines) \qquad (4)$$

5.2 Step 2: Automatic Weighting of Normalized Attributes based on Gini Index

Ranking the skylines aims principally to give an expressive discrimination between the selected objects. In order to reach such finality, it is crucial to fix a discriminative measure. In the literature, several measures were proposed. The entropy concept was used in various Multi-Attribute Decision Making problems. In (Huang, 2008) and (Hosseinzadeh Lotfi and Fallahnejad, 2010), an entropy based method was proposed. This method fits well with our computation context where we aim at differentiating attribute values in order to attempt a better decision making related to the skyline ranking. However, it has many limits especially related to the entropy calculation. Entropy requires logarithmic function computation what presents a computational time issue. Then, it is generally intended for attributes that occur in classes.

We rather propose another measure 'the Gini index' which is faster than entropy and does not use logs to insure the automatic weighting of attributes. Moreover, it is intended for continuous attributes. The Gini index is employed to derive the weights of the evaluation criteria (attributes) in our proposed method.

Hence, it determines the degree of divergence of attribute values. The *Gini* index of A_j is calculated by the following equation:

$$Gini(A_j) = 1 - \sum_{i=1}^{n} t_i[A_j]^2 \tag{5}$$

The corresponding weight is given by the following formula:

$$W(A_j) = \frac{Gini(A_j)}{\sum_{j=1}^{m} Gini(A_j)} \tag{6}$$

Let us consider the tuple $p_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\} \in$ rSkyPond (rsky after attributes weighting), then we have :

$$a_{ij} = W(A_j) \times t_i[A_j], \forall t_i \in rSkyNorm \tag{7}$$

5.3 Step 3: Determination of the Ideal Skyline

The ideal, denoted I^+, is an object that corresponds optimally to the user preferences. For example, if we consider the cars table, then searching an ideal car combines conditions on the price which has to be the smallest possible, the number of kilometers; the smallest possible, and the power; the greatest possible.

Thus, if we consider $I^+ =< v_1, v_2, .., v_m >$, then we have:

$$v_j = \begin{cases} max(A_j) \text{ iff } pref(A_j) = max \\ min(A_j) \text{ iff } pref(A_j) = min \end{cases} \tag{8}$$

5.4 Step 4: Calculation of Skyline Scores on the Basis of the Salton's Cosine

This step aims to determine the score of a skyline object on the basis of the Salton's Cosine (or Similarity Cosine). We calculate the cosine of the angle between the ideal and the skyline. The more the angle is little (high cosine), the more the skyline object is important. The Salton cosine ranges between 0 and 1 and is given in the formula below. Let us consider the tuple $p_i =< x_{i1}, x_{i2}, .., x_{im} >\in$ rSkyPond and $I^+ =< y_1, y_2, .., y_m >$ the ideal object, then we have :

$$t_i[Score] = Cosine(p_i, I^+) = \frac{p_i.I^+}{||p_i||.||I^+||} \tag{9}$$

$$t_i[Score] = \frac{\sum_{j=1}^{m} x_{ij}.y_j}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}.\sqrt{\sum_{j=1}^{m} y_j^2}}, \forall t_i \in \text{rSkyScore} \tag{10}$$

As a consequence of equation 10, $t_i[score] = 1$ if and only if the skyline is the best object, and $t_i[score] = 0$ if and only if the skyline is the worst object.

Remark: We can use the similarity principle of TOPSIS to calculate the score of each skyline object as following: Let us consider $t_i[scoreIdeal] = Cosine(p_i, I^+)$ and $ti[scoreAideal] = Cosine(p_i, I^-)$, $p_i \in rSkyPond$, I^+ the ideal and I^- the anti-ideal object./ Thus, if we consider $I^- =< v_1, v_2, .., v_m >$, then we have:

$$v_j = \begin{cases} max(A_j) \text{ iff } pref(A_j) = min \\ min(A_j) \text{ iff } pref(A_j) = max \end{cases} \tag{11}$$

5.5 Step 5 : Ranking the Result by the Score

This is the final step in the CoSky process, it aims to sort the skyline objects on the basis of the calculated scores. The CoSky steps applied to the car database relation can be calculated using a SQL queries ($cf.$ Appendix). The obtained results are given in Table 3. The obtained rank is C_1, C_4, C_2.

Table 3: Skylines ranking with CoSky method.

idcar	price	klm	power	score
C_1	25	10	8	0.814
C_4	5	40	7	0.803
C_2	20	30	6	0.769

Unlike dp-idp, the CoSky method distinguishes clearly the skyline scores. The skyline objects C_4 and C_2 have explicit scores (score $\neq 0$) while using dp-idp their score is 0. Thus we can rank them.

6 FINDING TOP-K RANKED SKYLINES

The notion of Multilevel skylines for finding Top-*k* Skylines (not ranked) is introduced in (Preisinger and Endres, 2015). A Top-*k* Skyline query Qk on a Database relation *r* computes the Top-*k* tuples with respect to the skyline preferences. Let us consider $S_0(r)$ the classical skyline set and $Card(r) > k$, then we have:

1. If $Card(S_0(r)) > k$, then Qk returns only *k* tuples from $S_0(r)$;
2. If $Card(S_0(r)) = k$, then Qk returns the skylines (*i.e.* the set $S_0(r)$);

3. If $Card(S_0(r))k$, then the elements of $S_0(r)$ are not enough numerous for an correct answer. A Multilevel skylines approach has to be applied. That means, not only all elements of the $S_1(r)$, the first level, are returned from $(r \backslash S_0(r))$, but also some of the $S_2(r)$, the set of skylines result from $(r \backslash (S_0(r) \cup S_1(r))$ and if the number of result tuples is still less than k, then we have to build $S_3(r)$, and so on . . .

The following algorithm DeepSky uses this multilevel principle with our ranking method to find Top-*k* ranked Skylines. It returns the Multilevel *k* skylines that have the *k* highest scores computed by the CoSky procedure.

```
Input: The database relation $r$, Preferences
       on attributes, and $k$
Output: the Top-$k$ tuples/objects with best
        scores : Topk
FS := 0;
rlayer := $r$;
while $FS \geq k$ or $rlayer = \emptyset$ do
    rsky := CoSky(rLayer);
    FS := FS + card (rsky);
    if $FS$ ¡ $k$ then
        Topk := Topk $\cup$ rsky;
        rlayer := rlayer $\backslash$ rsky;
    end
    else if $FS \geq k$ then
        Topk := the first k skylines of rsky;
        return Topk;
    end
end
return Topk;
```

Algorithm 1: Algorithm DeepSky for finding the best Top-k skylines.

Example: If we consider $k = 4$, the algorithm DeepSky returns C_1, C_4, C_2, the ranked skylines from level 0 and C_3 the only skyline from level 1.

7 CONCLUSIONS

In this paper, we proposed novel techniques for ranking skyline objects. Three contributions are described: The first is an enrichment of dp-idp method by dominance hierarchy to fast scoring skylines. The second is the CoSky method based on the renowned TOPSIS schema from Multiple Criteria Decision Analysis and Salton's cosine similarity from Information retrieval. An example of an SQL implementation of the proposed method was also described. Finally, we presented an algorithm for finding the Top-*k* ranked skylines that have *k* highest scores using the principle of Multilevel skylines and the CoSky method. As a short term future work, we will implement our approach in online sales applications in a Big Data context.

REFERENCES

Bartolini, I., Ciaccia, P., Oria, V., and Ozsu, T. (2007). Flexible integration of multimedia sub-queries with qualitative preferences. *journal of Multimed Tools Appl*, 33:275—300.

Behzadian, M., Otaghsara, S. K., Yazdani, M., and Ignatius, J. (2012). A Review on state of the art survey of TOPSIS applications. *Expert Systems with Applications*, 39:13051—-13069.

Bentley, J. L., Kung, H. T., mellon Umversuy Putsburgh, C., Schkolnick, M., and Thompson, C. D. (1978). On the average number of maxima in a set of vectors and applications. *Journal of the ACM*.

Borzsonyi, S., Kossmann, D., and Stocker, K. (2001). The skyline Operator. In *Proceedings of the ICDE Conference*, page 421–430.

Chan, C. Y., Jagadish, H. V., Tan, K., Tung, A., and Zhang, Z. (2006). On high dimensional skylines. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 478–495.

Chen, L., Gang, C., and Li, Q. (2015). Efficient algorithms for finding the most desirable skyline objects. *Knowledge Based Systems*, 89:250–264.

Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. (2003). Skyline with presorting. pages 717– 719.

Das Sarma, A., Lall, A., Nanongkai, D., and Xu, J. (2009). Randomized multi-pass streaming skyline algorithms. *PVLDB*, 2:85–96.

Hosseinzadeh Lotfi, F. and Fallahnejad, R. (2010). Imprecise shannon's entropy and multi attribute decision making. *Entropy*, 12.

Huang, J. (2008). Combining entropy weight and TOPSIS method for information system selection. In *Proceedings of International Conference on Automation and Logistics*, pages 1184–1281.

Lai, Y., Liu, T., and Hwang, C. (1994). TOPSIS for MODM. *European Journal of Operational Research*, 76:486–500.

Lakhal, L., Nedjar, S., and Cicchetti, R. (2017). Multidimensional skyline analysis based on agree concept lattices. *Intelligent Data Analysis*, 21:1245—-1265.

Papadias, D., Tao, Y., Fu, G., and Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30:41–82.

Preisinger, T. and Endres, M. (2015). Looking for the Best, but not too Many of Them: Multi-Level and Top-k Skylines. *International Journal on Advances in Software*, 8:467–480.

Spyratos, N., Sugibuchi, T., Simonenko, E., and Meghini, C. (2012). Computing the skyline of a relational table based on a query lattice. *CEUR Workshop Proceedings*, 876:145–160.

Tan, K.-L., Eng, P.-K., and Chin Ooi, B. (2001). Efficient progressive skyline computation. pages 301–310.

Tscheikner-Gratl, F., Egger, P., Rauch, W., and Kleidorfer, M. (2017). Comparison of multi-criteria decision support methods for integrated rehabilitation prioritization. *Water*, 9(2).

Valkanas, G., Papadopoulos, A., and Gunopulos, D. (2014). Skyline ranking à la ir. *CEUR Workshop Proceedings*, 1133:182–187.

Vlachou, A. and Vazirgiannis, M. (2010). Ranking the sky: Discovering the importance of SKYLINE points through subspace dominance relationships. *Data and Knowledge Engineering*, 69:943–964.

Yiu, M. and Mamoulis, N. (2007). Efficient Processing of Top-k Dominating Queries on Multidimensional Data. In *Proceedings of the VLDB conference*, pages 483–494.

APPENDIX

```
WITH rSky AS (SELECT * FROM Cars C1
 WHERE NOT EXISTS
   (SELECT * FROM  Cars C2
    WHERE (C2 .price <= C1.price
      AND  C2.klm <= C1.klm
      AND  C2.power >= C1.power)
      AND  (C2.price < C1.price
        OR C2.klm < C1.klm
        OR C2.power > C1.power))
 ),
 rSkyNorm AS (SELECT idcar,
   price/Sprice AS priceNorm,
   klm/Sklm AS klmNorm,
   power/Spower AS powerNorm
   FROM rSky,
     (SELECT SUM (price) AS Sprice,
      SUM (klm) AS Sklm,
      SUM (power) AS Spower FROM rSky)
 ),
 rSkyGini AS (SELECT
   1- (SUM (priceNorm * priceNorm)) AS pricegini,
   1-(SUM (klmNorm *    klmNorm)) AS klmgini,
   1- (SUM (powerNorm * powerNorm)) AS powergini
 FROM     rSkyNorm
 ),
 rSkyW AS (SELECT
   pricegini/ (pricegini + klmgini + powergini) AS pricew,
   klmgini/(pricegini + klmgini+ powergini) AS klmw,
   powergini/(pricegini +  klmgini + powergini)AS powerw
   FROM rSkyGini
 ),
 rSkyPond AS (SELECT
   idcar,
   pricew * priceNorm AS pricepond,
   klmw * klmNorm AS klmpond,
   powerw * powerNorm AS powerpond
   FROM rSkyNorm, rSkyW
 ),
 ideal AS (SELECT
   MIN (pricepond) AS IDLprice,
   MIN (klmpond) AS IDLklm,
   MAX (powerpond) AS IDLpower
   FROM rskyPond
 ),
 rSkyScore AS (SELECT
   idcar,
   (IDLprice * pricepond + IDLklm * klmpond +
      IDLpower * powerpond) /
   (sqrt (pricepond * pricepond + klmpond * klmpond +
      powerpond * powerpond)) *
   (sqrt (IDLprice * IDLprice + IDLklm *IDLklm +
      IDLpower * IDLpower)))
   AS score
   FROM ideal, rSkyPond
 )
SELECT ca.idcar, price, klm, power, score FROM
rSky ca, rSkyScore rs
WHERE ca.idcar = rs.idcar
ORDER BY score desc;
```