International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 5, October 2015, pp. 1188~1193
ISSN: 2088-8708

Computing Subspace Skylines without Dominance Tests using Set Interaction Approaches

T. Vijaya Saradhi*, Dr. K. Subrahmanyam, Dr. Ch. V. Phani Krishna*****
*Department of Computer science and Engineering, K.L University,
**Department of Computer science and Engineering, K.L University
***Department of Computer science and Engineering, K.L University

Article Info

Article history:

Received Feb 21, 2015
Revised Jul 4, 2015
Accepted Jul 20, 2015

Keyword:

High dimensional data
Skyline computation
Skyline cube

ABSTRACT

Now a day's preference answering plays major role in all crucial applications. If user wants to find top k–objects from a set of high dimensional data based on any monotonic function requires huge computation. One of the promising methods to compute preference set is *Skyline Technology*. Sky line computation returns the set objects that are not overruled by any other objects in n a multi dimensional space. If data is high dimensional, different users requests sky line set based on different dimensions. It requires subspace skyline computation. If objects are d-dimensional we need to compute skyline sets in 2^d different subspaces, called as SKYLINE CUBE computation, which incurs lot of computation cost. In this paper we address the problem of finding subspace skyline computation with minimum effort by using simple set interaction methods. By that we can decrease the number of subspace skylines need to be searched to find full sky cube. In this paper we developed one algorithm which uses Boolean algebra rules to reduce dominance test for preparing sub space skylines.

Corresponding Author:

T. Vijaya Saradhi
Department of Computer science and Engineering,
K.L. University, Vaddeswaram.
e-mail: saradhi1440@kluniversity.in

1. INTRODUCTION

Now day's people are expecting accurate predictions from available applications based on enormous previous data and current data [1]. So for exact prediction, people need to consider each and every piece of data. If data is high dimensional data we need to consider each and every dimension [2]. So there is need of multi criteria decision making system. In selection, Player A dominates player B if and only if player A is better than player B in minimum one dimension and player A is not worst than player B in all dimension[3]. For instance assume franchise wants to select players. Franchise X wants to consider only fielding_economy and bowling economy of players for selection. So A.fielding_economy<=B. fielding_economy and A.bowling_economy <=B.bowling_economy.one of the efficient computation to find the required subset of object not dominated my any other remaining object is skyline computation [3]. In traditional skyline computation they are considering fixed dimensions of objects [4]. In selection Different franchise require results based on different criteria. Other franchise may consider catch_drops and duck_outs. So recently sub space skyline became more crucial in skyline computation. If player is having D dimensions to answer all user queries we need to consider 2^D -1 sub space dimensions. Finding skyline of problem space by finding all sub space skylines is called skyline cube [5]. Even though existing methods seems to be overcome the problem of unnecessary computation of dominance tests like pruning through spatial relations, sky cube computation has still its unique challenges [5, 1]. In sky cube we will compute skyline sets in all different sub spaces. In existing studies to build Sky cube they have to find and search sky lines in all available subspaces, which may lead to pitiable performance in high dimensional data sets. **Stellar,** a sky cube computation method prevents computing every subspace skyline [3, 5]. It starts finding seed skylines from that it builds full space skyline with the help of decisive subspaces concepts .In this, problem is determine number of seed skyline groups and we need to compare every object of seed skyline group with objects not belongs to full space skyline. This will be the source of deprived performance [5]. Stellar algorithm won't consider the relationships between properties among skyline points yet to be derived on different subspaces [6, 1]. We can take advantage of these relationships to derive skylines of other subspaces for fast computation of sky cube.

this work consists of two important aspects first one is reducing dominance test to compute sky cube in a subspace by identifying the sub spaces from which we can derive skylines without dominance tests. Second one is we can reduce effort by result sharing in sky cube computation. We can divide the subspaces as 2 groups. First group is known as CATEGORY- I subspace. Skyline of these types of spaces can be found with simple deduction rules. Other group, CATEGORY- II, subspace skylines can be found using special derivation formulas or possibly by performing dominance tests

This paper is structured as follows: Section 2 is related to preliminaries and the previous work related to this work. Section 3 states the problem in formal way and it will be executed with small example Section 4 contain the proposed algorithm for subspace skyline computation. In Section 5 paper will be concluded with the directions of future work

2. RELATED WORK

2.1 Preliminaries

Skyline Computation: we start defining skyline computation with the assumption that smaller values are better. Skyline computation retrieves subset of elements from the given set of elements, that subset is called skyline set [7]. Formally assume that d- dimensional record set with cardinality n is available. Skyline computation retrieves m data points which are not dominated by any other points [8, 9]. To specify that a point X dominates another point Y can be represented as $X \Delta Y$ iff $X[i] <= Y[i]$ $\forall 1<=i<=d$ and $\exists$ k such that $X[k] < Y[k]$ [10]. Here i^{th} dimension of the X object is denoted as X[i]. We can represent the facts that X not dominating Y using $X \nabla Y$. $X \approx Y$ to indicate that X and Y are incomparable (means $X \Delta Y$ and $Y \nabla X$) and $x \subseteq y$ to indicate that either $x \subset y$ or $x = y$ holds. If D is the data set then skyline set of D is defined as

$$\{ \ X \in D \mid Y \nabla X, \ \forall y \in D \ \}$$

Initial sky line algorithms are BNL (Block Nested loop) which compares each point with all other points and qualifies only when it is not dominated by all others [10]. SFS (Sort Filter Skyline) is same as BNL but it sorts the data by this it will be advantageous than BNL [9]. DC (Divide and Conquer) divides the given space into regions and finds the skyline in every region by that it produces the final skyline [3]. LESS (Linear Elimination Sort skyline) is having attractive worst case performance [7]. In all above algorithms we must read the database at least once. But Index based methods need to visit only a portion of data base [8].

3. PROPOSED FRAMEWORK

We will take n-dimensional space D = {d1, d2, d3, ... dn}. Assume a set of points S in space D.If you consider any point P in set S, we can refer $P(d_i)$ as d^{th} dimension value of point P. Total possible numbers of subspaces (M) of full space D are 2^D. i.e. $M \subseteq 2^D$. Consider a U-dimensional subspace of the full space D.If we want to declare M as maximal subspace in M there should not exist, $V \in M$ such that $U \subset V$ [3]. In a subspace, $U \subseteq D$. Let p, q are points. If p is said to be **dominate another** point q, denoted by $p \Delta U q$, if $\forall di \in U$, $p(di) \leq q(di)$ and $\exists dj \in U$, $p(dj) < q(dj)$[10].we use the notation $p \approx Uq$ to represent that p and q are **unique**. It mean p is not dominating q and q is not dominating p. if all respective dimension values of any two points of any sub space skyline set are equal Then those two points are known as **indistinct skyline points.** We can represent indistinct points p, q of subspace U as $p = Uq$ if $\forall x_i \in U$, $p(x_i)=q(x_i)$. Point p can be indicated as indisstinct point using notation #p.

__Definition 1:__ Point 'O' of certain set X wants to be qualified as skyline point in certain subspace S of full space D iff $\forall q \in S$, $O \Delta S q$. Sub space skyline set consists all skyline points in that subspace. If you consider full space with D dimensions then sky cube of S can be considered as multi set of all sub space skyline sets $\{SKY(S) \mid S \in 2^D\}$

Collection of all sub space skyline sets of a full space is known as Skycube.This concept is similar as data cube concept in data warehouse. If full space is having d dimensions then we can form sky cube by finding skyline sets in 2^d subspaces. With sky cube concept subspace skyline queries can be answered effectively.We can find skyline sets in two approaches namely BUS (bottom up skyline) and TDS (top down skyline).we are following BUS [3].

Definition 2 (Indistinct/unique skyline)
If we want to qualify one point p ∈ SKL(U) as indistinct skyline in a specified subspace U there exist another point q∈SKL(U) with properties p≠q and p=Uq.if we want to denote a subset, X as **indistinct skyline Group** in subspace U if and only if it satisfies the below 3 conditions. (1) Set size of X ≥ 2. (2) All dimensional values of any two skyline points of X should be same with respect to subspace U. (3) Take any skyline point p ∈ X, q′ ∈ SKL (U) − X, p ≈Uq′ .We can declare a skyline point ‘p’ of any subspace U if it it is not member of any of the indistinct skyline group. It means P is unique to the other Skyline group members of the subspace U. We denote p by ≈ p.
In this work two important concepts are indistinct skyline groups and unique skyline groups. With the help of indistinct skyline groups we used to eliminate unnecessary information from subspace. Both important and non redundant information will be characterized by Indistinct and unique skyline groups.

Running Example:
In this paper, we will consider data set of 6 players, each with 4 dimensions as running example. The possible subspaces are 2^4-1.In our example to represent sky cube we need to find the skyline sets of 15 subspaces listed like Table 2. There are 2 indistinct skyline points, p$_1$(BOE) and p$_2$(BOE) with value 3 in the subspace {A}.we will use two different symbols to identify unique and indistinct skyline points. With the help of those symbols we can specify the SKL ({A, D}) as multi set: {{≈o$_3$},{≈o$_4$},{#o$_{1,}$#o$_2$}}

Table 1. Sample data set

OBJECTS	BO_(A)	FL_E(B)	ICCBAT_R(C)	ICCBOL_R(D)
P1	3	5	8	10
P2	3	5	7	10
P3	4	6	7	9
P4	6	6	6	8
P5	5	7	11	9
P6	7	10	9	9

Table 2. Subspaces and Skyline Sets

SUB SPACE	SKYLINE SETS	SUB SPACE	SKYLINE SETS
{A}	{#P$_{1,}$#$_,$P$_2$}	{B,D}	{ #P$_{1,}$ #P$_2$, ≈P$_4$}
{B}	{#P$_{1,}$ #P$_2$}	{C,D}	{≈ P$_4$}
{C}	{≈P$_4$}	{A,B,C}	{≈ P$_{2,}$ ≈P$_4$ }
{D}	{≈P$_4$}	{A,C,D}	{≈P$_{2,}$ ≈ P$_{3,}$ ≈ P$_4$ }
{A,B}	{#P$_{1,}$ #P$_2$}	{B,C,D}	{ ≈P$_{2,}$ ≈P$_4$ }
{A,C}	{≈P$_{2,}$ ≈P$_4$}	{A,B,D}	{#P$_{1,}$#P$_2$ ≈P$_{3\ ,}$ P$_4$}
{A,D}	{#P$_{1,}$#P$_{2,}$≈P$_{3,}$≈P$_4$}	{A,B,C,D}	{ ≈P$_{2,}$≈ P$_{3,}$ ≈ P$_4$ }
{B,C}	{#P$_{1,}$ #P$_2$}		

Theorem 1: (**Skyline union derivation**). We can apply union rule to find skyline sets and objects in skyline set. Take any two sub spaces U and V of full space D. If any point p belongs to the skyline sets of both subspaces then p will also belongs to skyline set of union set .i.e if p∈ SKL (X) and p ∈ SKL (Y), then p ∈ SKL (X ∪ Y)

Corollary 1. ∀ X, Y subspaces, SKL (X) ∩ SKL (Y) ⊆ SKL (X ∪ Y)

In our above example (Table 1), P$_1$ is a skyline point in both sub spaces {A} and {B}. By theorem 1, we can say that P1 is a skyline in subspace {A, B}. Moreover, point P$_2$ is a skyline point in both subspace of {A, C} and {B}. Using Corollary-1, we have can derive that point {P$_2$} ⊂ SKY ({A, B, C}).By the above Derivation we can conclude that it is possible to derive some subspace skyline points using simple set

operations without performing any dominance tests.
Our aim is deriving complete skyline set without dominance tests. To achieve this we should investigate and define more rules.

Definition 4: (**Categories of subspaces**).
If all points in a subspace skyline are indistinct with each other then we can recognize that subspace as CAT-1 subspace. i.e $\forall p; q \in SKL\ (U), p=Uq$.If sub space is not CAT-1 then it will be CAT-2 sub space.

Theorem 2: Take any two subspaces of CAT-1 if there is any common skyline point between subspace skylines then we can conclude that subspace skyline intersection is equivalent to subspace skyline union[4]. $\forall$ X, Y$\in$CAT-1 subspace if $SKL(X) \cap SKL(Y) \neq \phi$, then $SKL(X) \cap SKL\ (Y) = SKL(X \cup Y)$.
Using theorem-2 we can derive skyline set with simple set operation in a CAT-I subspaces effectively. Obviously, every subspace with single dimension is a CAT-I Sub space. We will fallow bottom up fashion to derive skylines with single scan in the single dimensional subspace without any dominance test. We can derive skylines in CAT-II sub spaces using unique skylines.

Theorem 3: For any 2 subspaces U, V and $U \subseteq V$, if p is a unique skyline point in U then $p \in SKL\ (V)$ also.
Theorem 3 can be considered as a derivation of Theorem-1. With the help of Theorem-3 we can directly derive skyline points in CAT-II subspaces.

In our existing example P2, P4 belongs to SKL {A, C} with the help of theorem-3 we can conclude that P2, P4 also belongs to the skyline sets of supersets of {A, C}. We specified categories and subspace skyline points in Table 2. Apart from derived skylines we need to find only a small number of skylines using techniques other than the two theorems to form full skyline set.

Finding Skylines by Domination Tests

Indistinct skylines

When we try to derive skyline points from subspace skylines indistinct skylines (i.e.$\exists p,q \in SKL(U)\ p=Uq$) in subspaces prevents the applicability of unique rule. If that is the case we can apply following rule

Theorem 4. For any subspaces U, W and $U \subset W$,if we want to say a point $p \in SKY(U)$ is a skyline in subspace W if and only if there should not exist q such that $q =_U p$, $q \Delta_{W-U}\ p$.To derive skyline using dominance test using therorem-4 it requires an approach like BNL (Block Nested Loop). In this approach we need to compare the indistinct skylines of U in the subspace W-U. This rule reduces the effort [4].

New unique Skylines:

Up to now we used to derive skyline of a space from its subspaces. From examples we can conclude that all skylines of a space not necessarily the members of its subspace skylines. In our current example all skyline points in subspace {A, D} are not skylines in either {A} or {D}.It means a point can be a member of a skyline space even though it is not a subspace skyline. To derive these types of skylines we need to do more exercise. We will do it with the help of encompass candidate rule.

Definition 5 (Encompass candidate). If we want to declare a point p as encompass point in any subspace X, every dimension value of that point p should lies between the maximum and minimum values of the skyline set points with respect to that dimension. $\forall d_i \in X, \min_{SKL(X)}(d_i) \leq p(d_i) \leq \max_{SKL(X)}(d_i)$.

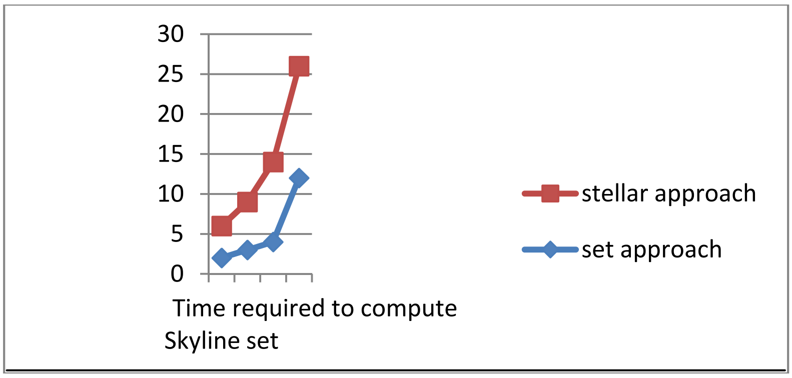
Theorem 5. If a point p is not derived by skyline derivation rule or indistinct rule or it's not skyline by encompass candidate rule then definitely it is not a skyline.
With help of theorem -5 we can find the full skyline set in the subspace. This theorem -5 will be considered as pruning technique [6].

Example 5: In a above example skyline set for subspace W={A,D} can be derived as follows. $\{o_4\} \in SKL(\{D\})$ so it belongs to {A,D}.$o_{1,}o_2 \in \{A\}$ so $o_{1,}o_2 \in W$.From Encompass candidate rule $\{o_{3,}o_5\} \in W$ because $3 \leq p(A) \leq 6$ and $8 \leq p(D) \leq 10$.but $o_3 \prec_U o_5$ so final $SKL(W)=\{\#o_1, \#o_{2,} \approx o_3, \approx o_4\}$.

```
  SubSky(S: Data Set, D: Dimensional Space)
   {
       While(X<-Genaratesubspace (D) ≠ϕ)
   {
          For each subspace W ∈X do
         W.parent is CAT-1 Subspace then selects any
         Equal 2 subspaces U, V⊂W

      If( SKL (U)∩ SKL (V) ≠ϕ) then
        {
           SKL (W) = SKL (U)∩ SKL (V)
            W is a CAT-I Subspace.
        }
           Else
        {
          W is a CAT-II Subspace.
         If (W is a CAT-II Subspace) then
           For each subspace U ∈W
          Do
            {
         Apply theorem-3 (unique skyline rule) and
         Copy all unique skylines from U.Process
         the indistinct skylines and use encompass rule .
        Using BNL to find full space skyline.
            }
      }

  Return (SKL (W));
  }
```



4. CONCLUSION

The paper work studies skyline computation of high dimensional data. We show that dominance tests can be highly reduced by finding unique and indistinct skyline groups than BNL and Index based algorithms. By using this method different skyline queries can be easily answered by finding sky cube effectively. We can also decrease subspaces searches. We have exercised above techniques with number of example data sets to prove effective skyline computation. In our future work, developing new algorithms to find compact sky cube based on aforementioned strategies.

REFERENCES

[1] J. Pei, A. W.C. Fu, X. Lin, and H. Wang. “Computing compressed multidimensional skyline cubes efficiently”. In *ICDE*, pages 96–105. IEEE, 2007.

[2] A. Vlachou, C. Doulkeridis, Y. Kotidis, and M. Vazirgiannis, “Skypeer: Efficient Subspace Skyline Computation over Distrib-uted Data”, *Proc. IEEE 23rd Int’l Conf. Data Eng.* (ICDE ’07), pp. 416-425, 2007.

[3] Y. Yuan, X. Lin, Q. Liu, W. Wang, J.X. Yu, and Q. Zhang. “Efficient computation of the skyline cube”. In *VLDB* 2005.

[4] C. Ra¨ıssi, J. Pei, and T. Kister. “Computing closed skycubes”. *PVLDB,* 3(1):838–847, 2010.

[5] J. Pei, W. Jin, M. Ester, and Y. Tao. “Catching the best views of skyline: A semantic approach based on decisive subspaces”. In *VLDB* 2005

[6] X. Lian and L. Chen, “Probabilistic Ranked Queries in Uncertain Databases”, Proc. *Int’l Conf. Extending Database Technology (EDBT ’08*), pp. 511-522, 2008

[7] P. Godfrey, R. Shipley, and J. Gryz, “Maximal Vector Computation in Large Data Sets”, *Proc. Int’l Conf. Very Large Data Bases (VLDB*), pp. 229-240, 2005.

[8] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, “Selecting Stars: The K Most Representative Skyline Operator”, Proc. *IEEE 23rd Int’l Conf. Data Eng. (ICDE ’07*), pp. 86-95, 2007.

[9] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An optimal and Progressive Algorithm for Skyline Queries”, Proc. *ACM SIGMOD Int’l Conf. Management of Data*, pp. 467-478, 2003.

[10] S. Borzsonyi, D. Kossmann, and K. Stocker, “The Skyline Operator”, Proc. *17th Int’l Conf. Data Eng.* (ICDE ’01), pp.421-430, 2001.

BIOGRAPHIES OF AUTHORS

T. Vijaya Saradhi working as Assistant professor in the department of CSE, KL university. Pursuing Ph.D. He is having more than 10 years experience in both academics and industry. His area of interest is Data mining.

Dr. K. Subrahmanyam working as Assoc Dean R&D, professor in the department of CSE, KL university. He is having more than 20 years experience in both academics and industry. He published and coauthored more than 50 research publicatiobns.His research interests include Software Engineering, Data Mining, and Cloud Computing. He is a senior member of the CSI.

Dr. Ch.V. Phani Krishna, working as Assoc professor in the department of CSE, KL university.He is having more than 10 years experience in both academics. He published and coauthored more than 20 research publicatiobns. His research interests include Software Engineering, Data Mining. He is a senior member of the CSI.