Computers and Electrical Engineering 71 (2018) 887–900

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Optimizined skyline queries over uncertain data using improved scalable framework☆

A. Sairam[a,*], C. Suresh Gnana Dhas[b]

[a] Research Scholar, Manonmaiam Sundaranar University, Tirunelveli, Tamilnadu 627012, India
[b] Professor, Department of Computer Science and Engineering, Vivekanandhan College of Engineering for Women, Namakkal District, Tamilnadu 637205, India

ARTICLE INFO

Article history:
Received 12 April 2017
Revised 22 August 2017
Accepted 23 August 2017
Available online 5 October 2017

Keywords:
Skyline operator
PID controller
Array manipulation
Dynamic pivot sweep line
Dynamic pivot reuse algorithm
Uncertain database

ABSTRACT

Skyline operator is an attractive tool for making decision while accessing skyline queries. Varied settings are allowed over this skyline operator, due to its easy use over skyline queries. Most skyline algorithms operates based on the principle of completeness, in making the value of data points to be known. However, in major case, the presence of incomplete values in the dataset allows algorithms to behave ineffective over skyline queries. Conventional Algorithms failed in redefining the dominance notion and this incomplete data can be handled using proposed approach. To eliminate the incomplete data redundancy, a novel framework model is tested with two algorithms: Dynamic Pivot Sweep Line and Dynamic Pivot Reuse Algorithm. A proportional–integral–derivative (PID) controller framework is tested with realand synthetic datasets proved effective and efficient than conventional ones.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Skyline querying mechanism is a multi-criteria support decision tool in many real time applications and over real time databases. The mechanism operates based on the behavior of point in a specified dimensional space and depending on the behavior of dominance of points, the best data is chosen. The identification of points and dominance behavior over other points is closely watched and top point is chosen to answer the user specified query. The flexibility and intuition helps the user to choose or formulate skyline querying rather than other querying techniques. The presence of incomplete or uncertain data over multi-dimensional datasets makes most techniques to operate on completeness or knowing all the points in the space. The presence of incomplete data makes futile the skyline operator rather pre-processing or cleaning of uncertain data followed by the application of skyline query proves much worth. Many techniques operate on the basis of dominance notion behavior that eliminates the minimum or incomplete or repeated data. The elimination helps in removing the redundant space in the database and provides effective comparisons.

For a specified dataset, a skyline query returns the objects/points, which are not dominated by others. Specifically, consider a given D-dimensional dataset P, if a point $p_1 \in P$ dominates the point $p_2 \in P$, it must hold that (1) $\forall i \in [1, D]$, $p_1[i] \leq p_2[i]$, and (2) $\exists j \in [1, D]$, $p_1[j] < p_2[j]$, where, p[i] represents ith dimensional point value and the smaller value is better. For example, consider a set $P = \{a,b,\ldots,h,i\}$ of hotels and the hotel price of a is cheaper than b and closer to beach than b. Hence,

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. V. Suma.

* Corresponding author.
E-mail addresses: asairammsu@rediffmail.com (A. Sairam), sureshc.me@gmail.com (C.S.G. Dhas).

http://dx.doi.org/10.1016/j.compeleceng.2017.08.024
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

the hotel *a* dominates the hotel *b* during query search. Further, the hotels *a, g, h*, and *i* are dominated by other hotel within skyline of P. It is seen from this, the skyline query is useful in multi-criterion decision making and user preference queries. The results of hotel could be returned for tourists, a small set of interesting hotels from a big pool, to save the tourists' time, when they need to find a cheap hotel yet close to the beach to stay [8].

Motivation: The main problem associated with obtaining skyline query points is its increased response time when query is generated at user end. Also, with increased size of datasets, the response time increases rapidly. The presence of empty datasets allows multiple pairwise comparisons between the skyline points leading to reduced scalability with increased response time in obtaining top skyline point. Such issue associated with response time is addressed in this paper with three different proposed approaches.

In this paper, the points in the search space are clustered (dimensional sets) based on the past behavior in the search space. These points are identified using PID controller in the multi-dimensional search space. A framework is proposed that uses PID controller and followed by some preliminary sorting, merging, filtering and dominance operation within a cluster (within dimensional sets). This operation is chosen to occur between the nearest neighborhood distance or local points. The global points are assigned with the feedback mechanism provided with PID controller for future operations. The non-skyline points are eliminated with this framework and the skyline query is tested with Dynamic Pivot Sweep Line (DPSL) and Dynamic Pivot Reuse Algorithm (DPRU) for best comparison of points in search space. The proposed algorithms combined with novel framework are tested in real-life and synthetic datasets for storage reduction and redundancy removal in multi-dimensional search space. The operation is made simple with the use of PID controller that assigns the past and future value of the point in dynamic skyline query environment. The framework effectively retrieves meaningful complete data from the datasets using pre-processing and PID approach. The presence of PID feedback controller reduces the response time associated with the given query.

The paper is summarized with following contributions:

- To increase the response speed of skyline query, PID controller is combined with several pre-processing technique that forms a framework.
- To eliminate multiple pairwise comparisons, Dynamic Pivot Sweep Line (DPSL) and Dynamic Pivot Reuse Algorithm (DPRU) are proposed.
- To improve the pairwise comparison, certain steps were included in the framework to eliminate the non-skyline points.
- The proposed framework and the algorithms are tested with conventional techniques over both real-life and synthetic datasets.

The reminder of the paper is organized as follows: Section 2 provides the related works on query skyline. Section 3 discusses the preliminaries and Section 4 formulates the framework with the use of PID controller. This section also relates the use of proposed algorithms over novel framework. Section 5 evaluates the performance comparison of proposed system with conventional techniques and reports the results acquired with the comparisons. Finally, Section 6 concludes the work of entire research.

2. Literature review

The techniques used so far in uncertain database and skyline query is shown here:

2.1. Approaches in uncertain database

Since, the existence of uncertain data is increasing; various techniques are utilized in eliminating the uncertain data. Probabilistic least influenced set [1] technique retrieves uncertain objects with high probabilities in business planning to reduce the search space. Monochromatic and bichromatic versions are implemented over this technique with Regional, Candidate pruning and MINMAX (minimum-maximum) pruning model. This technique enables the use of the model over uncertain query object and uncertain moving objects. A compressed uncertain frequent-pattern tree technique [2] is designed to merge the same items over a tree branch for finding out the uncertain items in the database. A three layered framework [3] is designed to retrieve top-k tuples to find the uncertain data in distributed sites. This framework solved the problem of centralized processing of queries with reduced cost over distributed sites with efficient rank. Weighted Uncertain Interesting Pattern Mining technique [4] is used to correlate patterns in the form of a tree structure with new measures over uncertain database. Initially, the weight of an item is considered with correlation and then the method is improved with prefix proxy value, new metrics and pProxy to improve the mining pattern. Potential high-utility item set mining [5] technique identifies items with high probabilities and utilities using tuple uncertainty model. An upper bound algorithm with level wise search finds the utilities and repeated scanning is avoided using probability utility list mining is adopted over real-life and synthetic datasets. SUB probabilistic top-k dominating [6] query type approach is used as an effective tool over arbitrary subspaces for better pruning. This is applied over k uncertain objects that dominate dynamically to reduce the search space. Probabilistic reverse top-k queries over uncertain database [24] with probabilistic top-l influential query addresses the gap between the uncertain and certain database. A direct processing over uncertain database with such approach eliminates the gap. Advanced uncertainty belief functions theory skyline [14] is further applied over uncertain databases to avoid the issues associated with evidential data. This further improves the search pattern and that retrieves best evidential skyline objects.

2.2. Skyline approaches in uncertain database

User queries generate imprecise information in uncertain database but skyline queries generate relevant objects related to incomplete user preferences. However, skyline queries suffer from drawback associated with incomplete user preferences. This leads to irrelevant skyline size.

An interactive preference elicitation framework [7] minimizes the interaction of users and increases the reduced skyline size. Data partitioning indexing technique is used to enhance the reverse skyline query [8] for real and synthetic datasets or multidimensional datasets. This reverse skyline is retrieved using constrained reverse skyline query approach over a constrained region. A two tier streaming framework [9] that maintains and monitors the probable skyline changes with upper bounds is applied over distributed environment. The communication and computational cost is reduced effectively with this approach. Sliding windowing technique [10] is applied over probabilistic and continuous skyline queries using possible threshold. This technique mainly retrieves top-k data of skyline queries in high speed streaming real time environment. Probabilistic group subspace skyline [11] querying technique with uncertain data is analyzed on sensor data analysis. Pruning with indexing and pre-processing is added to reduce effectively the search space. Combinatorial skyline query [12] finds the best tuple combinations using past skyline query. Fast Skyline computation for multiple query algorithm [13] eliminates the non-member data from skyline query. Discriminant analysis is provided that finds the skyline objects in the memory for the user specific with good scalability. Mutual (monochromatic and bichromatic [1]) skyline query retrieves the data from dynamic skyline with reverse query skyline object. Here, reusing is done using data partition indexing with pruning over real and synthetic datasets. A k-skyband query processing [15] with expired and shadow skyline and thickness warehouse is used over multi-dimensional data items. Tackle constrained and group-by skyline algorithms improves the search performance of incomplete data in real and synthetic datasets. Leveraging uncertainty auxiliary indexes [16] algorithm is designed to eliminate incomplete data in privacy preserving and scientific Web data. A pivot point technique [17] is used to reduce the comparisons and skytree technique is used for point deletion and insertion using point based space partitioning. The pivot point technique is tested over partitioning and sorting algorithm and further greedy algorithm is tested over k-skytree representation. QPSkyline and QPUpdateSkyline [19] algorithm limits the uncertain data in real life datasets. Top-k combinatorial metric skyline query [20] uses location based approach with sorting, threshold, reusing and pruning heuristics are the combinations used in skyline query. Along with this, p-skyline is computed with ϵ - approximation technique. Grid partitioning system [22] with pre-processing and iteration method prunes incomplete skylines over grid environment. Global skyline is computed iteratively using optimal feedback mechanism. An σ-neighborhood skyline query [23] finds the similar points closest to the skyline points. Similarly, to find the query points based on scoring function and the query, top-k Manhattan spatial skyline query problem with respect to monotone scoring functions is used. Reverse k-skyband [25] query and most desirable skyline object manage and identify the size of skyline objects. Former technique is tested with five algorithms and the latter is based on weighted object search pattern. Most of the techniques use threshold value, which is difficult to set. The setting of higher value result in loosing of data and lower value with reduce the result quality. To solve this, probabilistic tuples are defined and returned, and without enumeration the tuples are found. A combined technique in [18] uses filtering, local, dominance skyline and bitmap indexes to reduce the search space using weighted and top-δ dominance algorithms. The above techniques discuss the conventional mechanism used in retrieving the queries related to user specific queries from real and synthetic datasets using skyline query mechanism.

Above all, the preliminaries discussed in [21] is used to remove the redundant information in the proposed system. However, technique in [21] suffers from increased response time based on given query and if multiple query is submitted, the response time will increase. Also, the proposed uses similar definitions but with modifications related to PID controller is discussed in the next section.

3. Preliminaries

Definitions related to skyline queries over uncertain database is initially explained. The proposed framework is developed over incomplete relational database, D_b. The D_b is represented by relation name, R with d dimensional set and m similarity.

Definition 1. The skyline query, S is retrieved by skyline technique, where S is not dominated by other data items in D_b.

Definition 2. The S_i (data set) dominates S_j (dataset) or $S_i \succ S_j$ iff $\forall d_k \in d$, $S_i.d_k \geq S_j.d_k \wedge \exists d_l \in d$, $S_i.d_l > S_j.d_l$.

Definition 3. Choose S_i iff $S_i \succ S_j$ $(i \neq j)$ in d or atleast at one d. The skyline data item sets are defined by S_{sl}, $S_{sl} = (S_i | \forall S_i, S_j \in d, S_i \succ S_j)$.

Definition 4. The database D_b $(r_1, r_2, \ldots, r_n)$ with relation $r_i(d_1, d_2, \ldots, d_m)$ is *incomplete* iff S_j has one or more missing values (m_v) in the given dimensions, d_k. Otherwise, the D_b is said *complete*.

Definition 5. If the data item x_i and $x_j \in R$, iff S has no m_v then x_i is *comparable* with x_j, and viceversa (x_i is *not comparable* with x_j) [21].

Definition 6. If the dataset Scontaining m_v and x with past value is given as an input to PID controller. Then, future prediction of S without m_v can be *predicted* by PID controller.

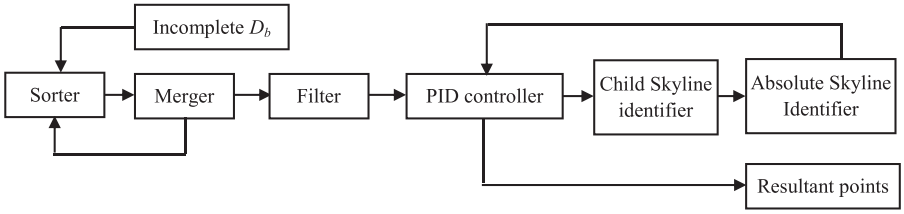


Fig. 1. Proposed framework with PID controller.

d	d_1	d_2	d_3	d_4	d_5	d_6	d_7
m_1	*	1	4	4	3	*	2
m_2	5	*	*	4	2	*	*
m_3	*	2	4	*	*	5	1
m_4	4	*	*	3	1	*	*
m_5	*	*	3	*	1	3	4
m_6	4	1	4	2	*	*	*
m_7	*	*	*	5	*	1	3
m_8	3	1	5	*	2	1	*

Fig. 2. Sample product rating in online site.

4. Novel PID framework

Here, in this section, incomplete data or dominated data are eliminated using the proposed framework components. This elimination is done before the skyline querying technique that mainly avoids cyclic redundant dominance, transitivity and repeated comparison with incomplete data. The proposed framework consists of five major components: Sorter, Merger, Filter, Dominance operation, PID controller and followed by this is the child and absolute skyline identifier that is shown in Fig. 1.

This proposed framework is explained using buying a product in online site, having database, D_b. There is an incomplete data set, $S\ (m_v)$ in D_b and that is shown in Fig. 2 (* symbol represent $S\ (m_v)$).

Data Sorter –

Initial D_b is sorted in ascending order w.r.t $d(m_v)$. An array groups are constructed that stores the sorted value, x in particular dimension d. The array is sorted within the values of $S_i(x_i)$ in particular dimension, d_i. Thus, total number of arrays equals the total dimension and the missing values. i.e. $arrays = d = m_v$. The Fig. 3(a) has 7 arrays and the values are manipulated in the form of array strings. The Fig. 3(b) shows that the data items are sorted out in ascending order and the m_v values are placed in top rows.

Consider the dimension d_1 has m_2, m_4, m_6 and m_8 strings with values 5, 4, 4 and 3. The values are arranged in ascending order 3, 4, 4 and 5 (m_8, m_6, m_4 and m_2) along with m_v values and so on. The same is the procedure for all the other dimensional forming sorted values in multi-dimensional arrays.

Data Merger –

This section aims in sequential scanning of data sets and finds the value of same data set over a single row. The same values in a single row belonging to various dimensions are merged into one data set holding same value. The leading dimension now has null or m_v value in its initial position. This additionally helps in pairwise comparison of data in multi-dimensional data sets. Thus the same values are grouped in to one forming a region and selection of trailing dimension holds the dataset. The Fig. 4(a) shows the evaluation of dataset and merging of same values during sequential scanning. Once this is done, the datasets are sorted in ascending order, as shown in Fig. 4(b).

The process repeats until the same values in the dataset is removed and this is shown in Fig. 5. After performing entire operation, the resultant sorted and merged data set is shown with much clarity. This perfectly removes all the redundant data sets to provide perfect processing for skyline queries.

Data Filter –

This component analyzes the sorted data and eliminates the dominated data in each dimension. The examining data is done in sequential pattern, ensuring all the data is analyzed atleast once. This results in providing the number of dominant dimensions of each data item and reduces the pairwise comparison. The occurrence of data item after each successful scan over sorted array is eliminated. The process starts with initial count value zero and with total number of similar items in the array dimensions, the count increases. The main idea is to place the dominated data set at the bottom of the list.

d_1	d_2	d_3	d_4	d_5	d_6	d_7
*	m_1	m_1	m_1	m_1	*	m_1
m_2	*	*	m_2	m_2	*	*
*	m_3	m_3	*	*	m_3	m_3
m_4	*	*	m_4	m_4	*	*
*	*	m_5	*	m_5	m_5	m_5
m_6	m_6	m_6	m_6	*	*	*
*	*	*	m_7	*	m_7	m_7
m_8	m_8	m_8	*	m_8	m_8	*

(a)

d_1	d_2	d_3	d_4	d_5	d_6	d_7
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	m_5	m_6	m_5	*	*
m_8	m_8	m_6	m_4	m_4	m_8	m_3
m_6	m_6	m_3	m_1	m_8	m_7	m_1
m_4	m_1	m_1	m_2	m_2	m_5	m_7
m_2	m_3	m_8	m_7	m_1	m_3	m_5

(b)

Fig. 3. Array manipulation (a). The replaced values with strings; sort (b). Sorted in ascending order.

d_1	d_2	d_3	d_4	d_5	d_6	d_7
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	m_5	m_6	m_5	*	*
m_8	m_8	m_6	m_4	*	*	*
m_6	m_6	*	*	m_8	m_7	*
m_4	m_1	*	*	m_2	*	*
m_2	m_3	*	*	m_1	*	m_5

(a)

d_1	d_2	d_3	d_4	d_5	d_6	d_7
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
m_8	m_8	*	*	m_5	*	*
m_6	m_6	*	*	m_8	*	*
m_4	m_1	m_5	m_6	m_2	*	*
m_2	m_3	m_6	m_4	m_1	m_7	m_5

(a)

Fig. 4. Merging (a). The dataset with same values are merged; Sorted (a). Filtering over merged dataset.

d_1	d_2	d_3	d_4	d_5	d_6	d_7
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
m_8	m_8	*	*	*	*	*
m_6	m_6	*	*	*	*	*
m_4	m_1	m_5	m_6	*	*	*
m_2	m_3	m_6	m_4	*	m_7	*

Fig. 5. Sorting after 2 iterations.

Data item	Count
m_1	1
m_2	1
m_3	1
m_5	1
m_7	1
m_4	2
m_8	2
m_6	4

Fig. 6. Filtering after Sequential scanning.

Initially, the count of m6 is zero, however sequential scanning increased the value to 4. Likewise, it calculated for other data items. This does not count the value, instead it count the value of the occurrence of the data items in the array. This is clearly shown in Fig. 6.

Skyline identifier –

In skyline identification process, the initial phase or child skyline identifier terminates all the low potential data set possessing values < 2. This eliminates the dominance of high data sets over the lower potential data set. The final phase or absolute child skyline identifier, checks for the m_v or incomplete skyline candidate in the database. Here, the pairwise comparison process starts by comparing the initial or child candidate skyline and absolute candidate skyline. This process is done to avoid dominance and the final skyline behaves independent. The comparison process takes place as follows:

- The first data item in the child skyline set is compared with other data items in the array.
- The data item that is not dominated by the opponent or comparing candidate, the dominating data item will be placed and the opposite one is eliminated.
- The iteration continues until the comparison is performed between the next successful candidates and removes the dominated data items.
- The data items that are compared conventionally against each other are not compared once again.
- If the count value remains the same, the elimination of candidate is chosen based on ascending order.

The final candidate skyline sets that is placed in the array set is the absolute candidate set. The Fig. 7(a) shows the initial candidate identifier and Fig. 7(b) shows the absolute candidate identifier.

4.1. PID controller

Definition 7. The skyline order O_1, O_2,...,O_n over D_d dimensional set point P gives the order index points $p \in P$, i iff p $\in$ s_i (subset of query points).

The skyline queries provide interesting points or non-dominated points. The interesting points should be obtained with the resultant feedback from skyline absolute identifier. The resultant filtered result from skyline absolute identifier is sent as a feedback to PID controller. PID controller stores the resultant set in the form of skyline order and uses it for future prediction of more skyline interesting points.

Data item	Count
m_4	2
m_8	2
m_6	4

(a)

Data item	Count
m_8	2
m_6	4

(b)

Fig. 7. (a). Candidate skyline identifier, (b). Absolute skyline identifier.

Here, the PID controller is tuned with the help of Gaussian variable, resulting in optimization based on Jacobian rule:

$$\frac{\partial J}{\partial K_t} = \frac{\partial J}{\partial u_t}\frac{\partial u_t}{\partial K_t}$$

where,

$$\frac{\partial u_t}{\partial K_t} = e_t$$

$$\frac{\partial J}{\partial u_t} = 2\left(\mu_{t+1} - y^r_{t+1}\right)\frac{\partial \mu_{t+1}}{\partial u_t} + \frac{\partial \sigma^2_{t+1}}{\partial u_t} + 2\lambda\Delta u_t$$

The process variable is represented here,

$$u_t = \widehat{u} + k_c\left[e_t + \frac{1}{\tau_i}\int_0^t e_t + \tau_D\frac{\mathrm{d}e_t}{dt}\right]$$

and the error variable is represent by,

$$e_t = y_t^{set} - y_t$$

k_c defines the proportional gain, τ_i represses integral time constant and τ_d represents derivative time constant.

Tuning of PID controller with its parameters determines the closed loop performance. The tuning of PID controller helps in improving the accuracy and provides better results with past available datasets. This provides good performance with the system possessing redundant or missing values. The presence of redundant data in the PID controller results in poor prediction and computation. The selection of datasets is based on predictive variance of the skyline points in the datasets. The uncertainty in predicting the skyline points indicates the predictive variance with better prediction capability than larger variance measurement. The larger variance indicates poor confidence and smaller variance refers better confidence. The predictive variance using Gaussian process selects the skyline points with high uncertainty.

Initially, a high uncertainty region is found and finally admissible range is calculated with this. The entire principle is based on predictive variance interpretation and the variance in predictive region implies nil data. The data has to be added in high variance region. The predictive variance value or threshold value is identified to find the high certainty region based on hyper parameter magnitude. This is taken as the process variance estimate and changes in pre-processing affects the threshold value of predictive variance. The unavailability of prior knowledge leads to auto threshold scaled valued or an assumed value. The predictive variance points that exceeds threshold value is considered to be the high certainty points or Top-Skyline points.

The PID model is a random variables set with joint distribution function with X set of inputs and Y set of outputs.

$$P(y|C, X) = \frac{1}{Z}\exp\left(-0.2(y-\mu)^T C^{-1}(y-\mu)\right)$$

The equation represents the correlation between various points in input datasets. X is not a manipulated variable and it is a Gaussian process variable. The resultant model contains past manipulated (m) and processed (l) variable. The dynamic modeling is referred to as:

$$x_t = [u_{t-1}...u_{t-m+1}\ y_{t-1}...y_{t-l+1}]^T$$

The covariance matrix (C) with parameterized function is represented by $C_{ij} = C(x_i, x_j)$ and Z refers to normalizing constant. The joint density function (P) is Gaussian, the data $D = (X, y)$ is obtained using conditional distribution function, which

is Gaussian:

$$P(y_{t+1}|y) = \frac{P(y_{t+1}, y)}{P(y)}$$

The posterior distribution is

$$P(y_{t+1}|y) = \frac{1}{Z} \exp\left[-\frac{(y_{t+1} - \mu_{t+1})^2}{2\sigma^2_{\mu_{t+1}}}\right]$$

and

$$\sigma^2_{t+1} = C(x_{t+1}, x_{t+1}) - g^T(x_{t+1})C^{-1}g(x_{t+1})$$
$$\mu_{t+1} = g^T(x_{t+1})C^{-1}y$$

g^T represents transpose of T and μ_{t+1} and $\sigma^2_{\mu_{t+1}}$ represents mean and standard deviation prediction of x. C^{-1} is the smoothening function that weights the training outputs to improve the new input prediction, i.e. x_{t+1}.

The PID controller helps in achieving better skyline points from independent points over user queries. The controller retains the order at the end of each skyline query in the form of subsets. This subset collection is used during the periodical skyline point retrieval from independent points after all pre-processing operations. The use of variance calculation using Euclidean distance measurement between the initial neighborhood independent points helps in attaining better variance. This provides more clarity in obtaining better skyline points than expected. In addition to this, the use of PID controller helps in reducing the distance between the process and control variable. The PID optimization involves the use of variance calculation that results in more robust system.

For real and synthetic datasets, the following algorithms are proposed to analyze the correctness in the datasets. The current research is included with two case studies:

4.2. Novel framework for identifying the skyline points

The framework is aligned with certain procedures using the proposed algorithms. The details of which is shown in following sections:

4.2.1. Dynamic pivot sweep line algorithm

The DPSL algorithm identifies skyline points using dominating and preference scores from existing data available in PID controller. The entire operation of PID framework is processed here but the selection of points during query is varied here. The technique of selecting points is done using sweep line algorithm along with the concept of dynamic pivot. The dynamic pivot selects the chosen points from the trained data and applies it as an initial test samples over the PID framework and thus the process goes on until the skyline absolute identifier sends the feedback to PID to obtain the output. This algorithm is tested over both synthetic and real-life datasets.

Algorithm 1. Dynamic Pivot Sweep Line Algorithm.

Input: The dataset and returned skyline object
Output: Desired skyline dataset

Initialize the set, $S(=\theta)$ that accepts skyline points (S, μ,τ)
Perform the sort operation in PID Framework with min-Heap sorting, H, ($H = H' = \theta$)
Insert all the sorted entries of root node of the tree, R into H and H'
Compute the skyline points using PID framework with H'.
Obtain the *Top–Data sets* using PID framework
Refine Top-Data sets: Form a vertical axis to sweep iteratively the skyline points along its coordinate axis (from left to right).
Add dynamic pivot model [23] over the obtained points to reduce the difference between the test and training data (existing data with PID controller). This is calculated using class distribution function.

$$CF\left(\frac{pr}{nr}\right) = \frac{\sum_{S-M/2}^{S+M/2}\left(\frac{pr}{nr}\right)_i}{M}$$

pr/nr represent the trained class distribution data.
S represent Pivot model with M models.
Count of the object (C_o) is evaluated with PID framework
Update the Sweep value to the PID controller. (PID controller reduces the difference between the values of past inputs of skyline point with the sweep variable)
Updated PID senses future prediction points accurately.

The R node calculates the preference and dominance score and proposed PID framework calculates the skyline of the dataset. The main aim is to calculate the scores in a single traversal over organized dataset. The algorithm is written in such form that during initialization the skyline datasets are accepted as inputs. After initialization, min-heap sorting is done in PID controller and the datasets gets sorted. The sorted data items are stored in root nodes of the heap tree and finds S, μ and τ until$H = \theta$. Then skyline computation is carried out over sorted H' using PID controller. The framework obtains top-data sets and the resultant sets are iteratively swept from left to right co-ordinate axis through sweep line. The evaluated data

is a skyline data sets, it is then tested with the trained data (Past value of the node visited and dominance score of each skyline point) from PID controller using dynamic pivot model. Finally, the object count is calculated and the sweep value is updated at regular intervals to the PID controller to choose best skyline individuals. This algorithm with PID controller helps in predicting the accurate skyline points in future queries.

4.2.2. Dynamic pivot reuse algorithm

Reuse algorithm is applied over dynamic pivot technique and the resultant algorithm is again tested in both real-life and synthetic datasets. The query processing is further improved with reuse technique and the use of aR-Tree [12] improves the prediction and dominance score. The aR-tree stores minimum bounding rectangle (MBR) for each data set value and it possess the count value (of each dataset in array) of PID frame work. The I/O overhead is further eliminated using aggregation query that access the intermediate nodes rather than entire nodes. The aR-Tree maintains MBR information and records the total data objects in the sub-tree.

Algorithm 1. Dynamic Pivot Sweep Line Algorithm.

Input: A count of dataset, total returned skyline points
Output: Desired skyline dataset
Initialize the set, $S(=\theta)$ that accepts skyline points (S, μ,τ)
Perform the sort operation in PID Framework with two min-Heap sorting, H, ($H = H' = \theta$)
Insert all the sorted entries of root node of the tree, R into H'
Compute the skyline points using PID framework with H'.
Maintain the dataset and MBR node dominated by skyline data into H
Obtain the *Top–Data sets* using PID framework based on the count of datasets, C_o with PID framework.
Refine Top-Data sets: The *Top–Data sets* is calculated and the best set is chosen based on preference and dominance score.
Add dynamic pivot model over the obtained points to reduce the difference between the test and training data (dominance and prediction score in PID controller). This is calculated using class distribution function.

$$CF\left(\frac{pr}{nr}\right) = \frac{\sum_{S-M/2}^{S+M/2}\left(\frac{pr}{nr}\right)_i}{M}$$

pr/nr represent the trained class distribution data.
S represent Pivot model with M models.
Update the Sweep value to the PID controller. (PID controller reduces the difference between the values of past inputs of skyline point with the sweep variable)
Updated PID senses future prediction points accurately.

The main aim is to calculate skyline point or object using PID framework but without elimination, the dataset and the dominated MBR node is stored in PID. Dominance (μ) and preference (τ) score is calculated for each skyline point. This is calculated based on the past value of the visited node, which is stored in PID controller. The initial step accepts the object in the form of skyline object. Sorting operation is performed based on two-min heaps with minimum L_1-norm distance from the dataset origin. Then the PID framework is used to compute the skyline point by aR-tree traversal and stores visiting node in PID. The dataset and MBR skyline dominated node is maintained with H and this helps in computing the scores for future prediction. During non-skyline point is accessed, the object that has higher dominance and prediction score than its predecessor is increased with a single count. Depending on the increased count, the top-skyline point is calculated and updated in PID controller. The other steps are similar to the sweep algorithm and this is computed for both real-life and synthetic datasets.

5. Performance evaluation

In this section, the proposed technique is evaluated to provide its effective in skyline query retrieval. Here, the framework is initially tested, followed by this is the application of both algorithms over real-life and synthetic datasets. Various experiments are conducted over different datasets to evaluate the performance of the proposed system. The proposed approach is compared with conventional techniques (SkyQUD, QPSkyline, Mapreduce, quadtree and top-δ dominance). The experiments are carried out with high computation system with better processing capability. The performance metric is evaluated in pairwise comparison between the skyline points. This metric evaluates the query processing capability of skyline approach, hence this is chosen as a metric for evaluation. The metric represents the dataset size, number of m_v dimensions and total dimension sets. The main intention is to obtain the maximum skyline points in over a specified array instead of each dimension. The system involves three datasets: one real-life cancer dataset and two synthetic correlated and independent dataset. This is shown in table. 1.

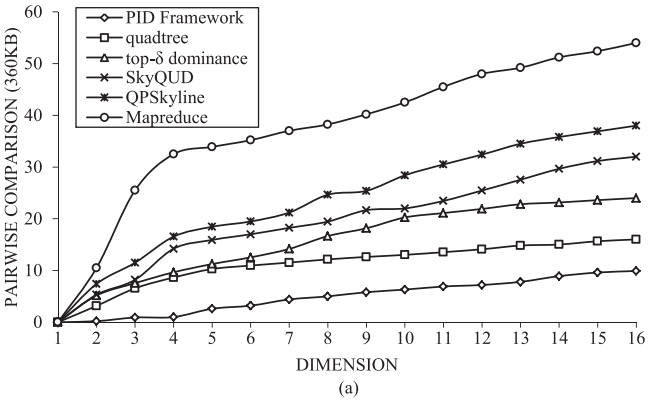
5.1. Effectiveness of PID framework

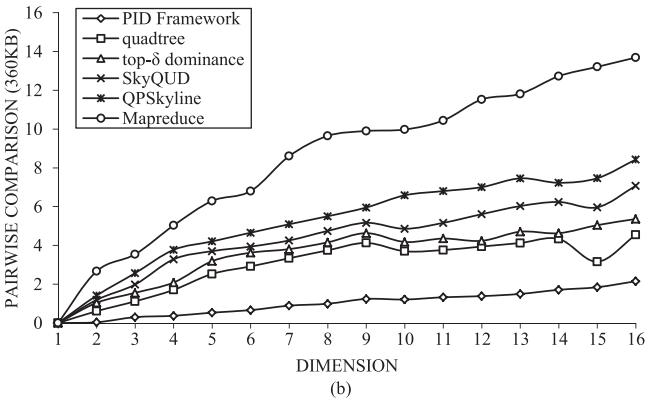
Total number of dimensional set in database and its size has high influence over skyline querying process. Initially, a set of experiments is conducted and that impacts the total dimensions in deriving skylines during pairwise comparison process.

The pairwise comparison for both real-life and synthetic datasets is shown in Fig. 8. It is seen that the Fig. 8(a), (b) clarifies that with correlated dataset having varied dimension a fixed dataset size of 360 kb, the results are measured. Likewise, Fig. 8(c) shows the real-life or cancer dataset with total size of 180 kb and the dimensions are varied between 6 and 18.

Table 1
Parameter setting of datasets.

Dataset name	Parameter settings		
	Total dimension	Dimensions with m_v	Dataset size (kb)
Correlated	8–16	4–12	60, 120, 180, 240, 300, 360
Independent	8–16	4–12	60, 120, 180, 240, 300, 360
Cancer	6–18	6–14	30, 60, 90, 120,150,180





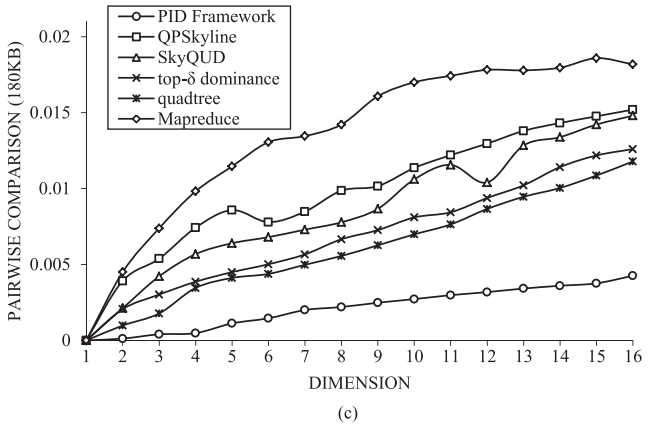


Fig. 8. (a) Correlated dataset to prove the dimensionality effect, (b) Independent dataset to prove the dimensionality effect, (c) Cancer dataset to prove the dimensionality effect.

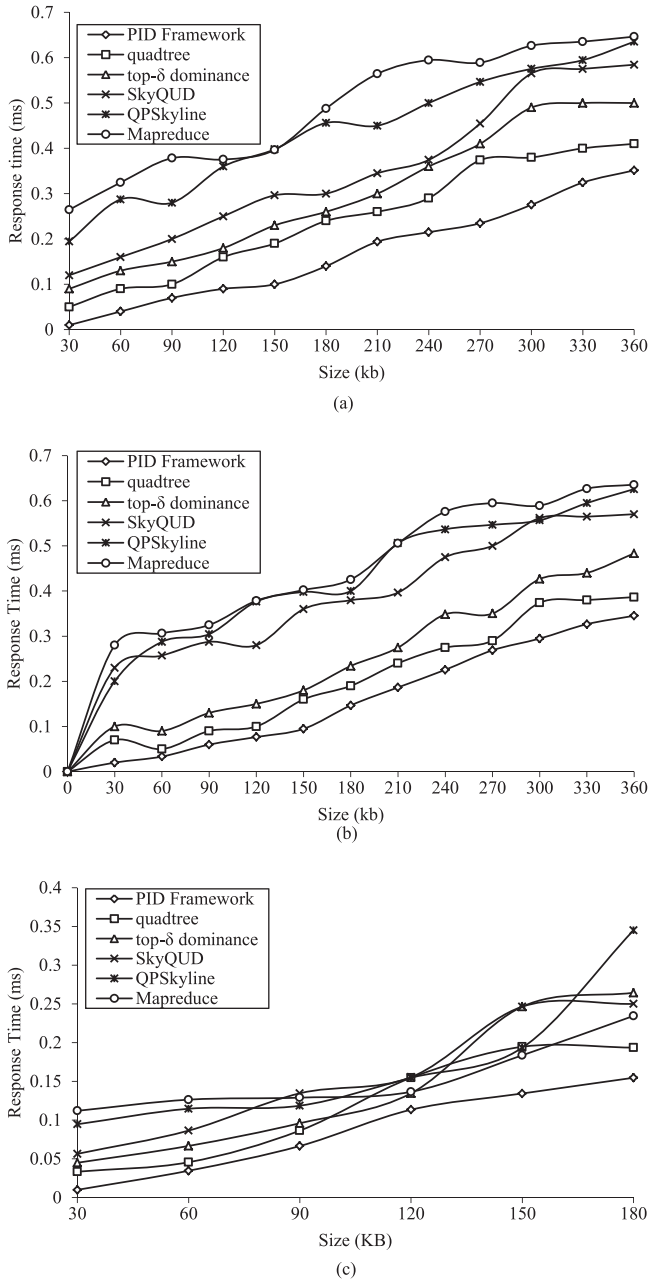


Fig. 9. (a) Correlated dataset to prove the response time behavior, (b) Independent dataset to prove the response time behavior, (c) Cancer dataset to prove the response time behavior.

It is proved that the proposed framework (PID framework) outperforms well when compared with conventional technique. It could be concluded that the pairwise comparison of PID framework is reduced to a better extent due to effective pre-processing elimination procedure. The PID framework has high scalability with increasing dimension. It process well the missing values in the given dataset and eliminates without leaving room for pairwise comparison with the missing values. The process or the application of array manipulation, sorting, filtering and other pre-processing steps worked well in eliminating these missing and prunes most dominated values in the dataset. This provides better efficient process in obtaining the top-skyline point in all the three datasets.

The effect of response time behavior w.r.t database size is shown in Fig. 9 for all the three datasets. The experiments are carried out with fixed dimensions and varied data size. Fig. 9(a–c) shows the significant impact on response time by proposed PID controller with the metrics. This evaluates well the skyline queries with lesser response time, where the dimensions in each dataset and size of dataset are set maximum. Both, real-life and synthetic dataset proved that the size of dataset has no impact, since the pre-processing is eliminates the non-dominated sets before skyline querying. The pairwise

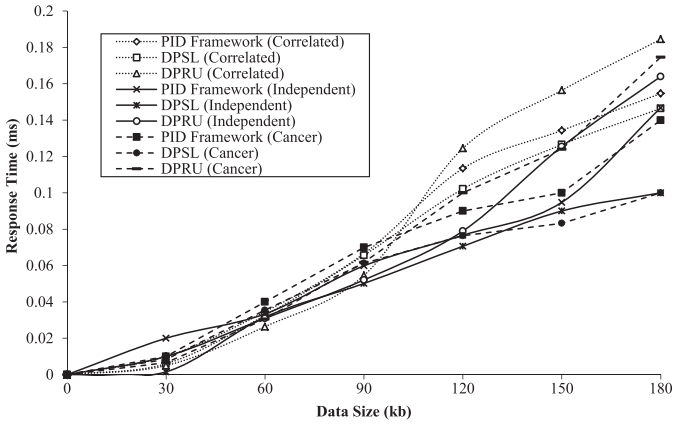


Fig. 10. Evaluation of case study.

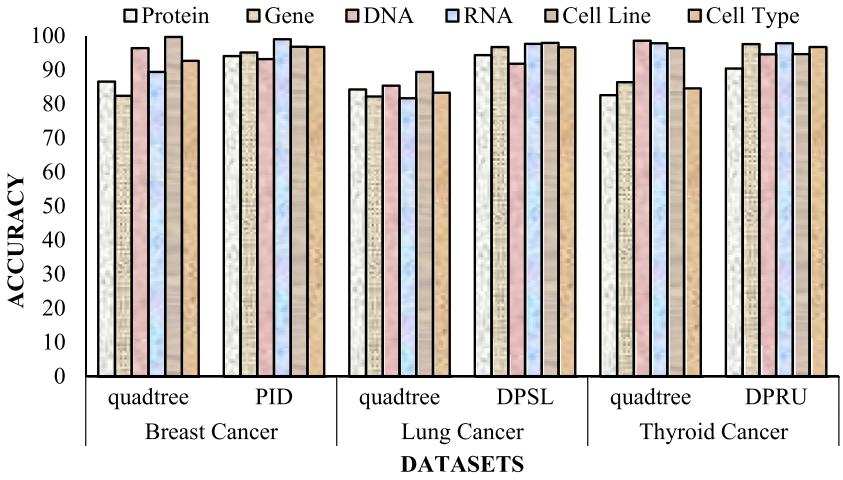


Fig. 11. Accuracy of proposed framework with other real-life datasets.

comparison with PID framework is thus reduced in all the three datasets. It could be concluded that the proposed approach outperforms well with reduced response time.

From the results of Fig. 10, it is proved that the response time of the graph increases with increased size of queries over large real and synthetic datasets. The results are tested with different datasets that includes: Correlated, Independent and cancer datasets. The tested algorithms include PID framework, DPSL, DPRU Framework over all three datasets. The response time of the proposed PID with DPSL and DPRU is compared and it is found that the response time is not much deviating and the results of the response time for all the three datasets is found more or less equal with slight deviations in it. However, as the query size increases, the response time increases for all the three testing algorithms.

The algorithms proposed over PID framework is tested with all the three datasets (size of 180 kb) in terms of its response time behavior considering the effect of dimension, effect of datasets size and missing values (Fig. 10). Initially, the system is tested than PID framework with all the three metrics and then with DPSL and DPRU. The use of heap sort in DPSL improved the proposed approach with PID framework. Use of aR-tree in the proposed method increases the effectiveness of retrieving the skyline queries with lesser response time than DPSL and the PID framework. The reduced response time is due to the application of two-min heap sort in the initial stage that cleans all the missing datasets and reduces the work of other stages with the use of PID controller. The proposed method works well with smaller datasets, as the size of dimensions increases, the scalability of DPRU reduces than DPSL and PID framework. So, for larger applications, DPSL can be used in handling skyline queries with lesser response time than the other techniques. Thus the system is well tested with all the three datasets and proved the effectiveness in terms of varied metrics.

5.2. Test case

The system is finally tested for accuracy with real-life datasets like Breast, lung and thyroid cancer with PID framework, DPSL and DPRU. The results are shown in Fig. 11 and it proves that for various real-life datasets the proposed techniques has better accuracy that is more than 80%. This proves the efficiency of the proposed methods in terms of any type of datasets.

6. Conclusions

This paper proposed new PID framework that process multiple skyline queries, where two algorithms from conventional work is modified and proposed as sublime techniques. The algorithms provide added advantage to the proposed framework and helps in query effective skyline queries. The use of dynamic pivot concept in the algorithms is to improve the existing option of identifying best skyline points. Also, the PID framework overcomes the problem of more response time in processing the queries. The framework takes into account the variance information of the skyline datasets and prediction of results. The PID avoids larger variance value and avoids non-tracking, reducing the response time. The addition of algorithms over PID framework further reduced the response time, thus making the system efficient. The identification of lower values in skyline sample set using PID avoids missing data values in the database.

The extensive experiments are carried out between the proposed framework and algorithms, and conventional ones. The results are tested in terms of single object processing time and response time in both real and synthetic datasets. The results proved that the proposed framework is fast in terms of object processing and response time. The addition of algorithms over the framework proved much more efficient in terms of such parameters. The PID framework with algorithms is highly scalable than conventional ones with increased queries and object arrival. The application of PID over skyline database with two algorithms demonstrated the versatility of the proposed design in improving the query search from redundant datasets.

References

[1] Lian X, Chen L, Wang G. Finding the least influenced set in uncertain databases. Inf Syst 2011;36(2):359–85.
[2] Lin CW, Hong TP. A new mining approach for uncertain databases using CUFP trees. Expert Syst Appl 2012;39(4):4084–93.
[3] AbdulAzeem YM, ElDesouky AI, Ali HA. A framework for ranking uncertain distributed database. Data Knowl Eng 2014;92:1–19.
[4] Ahmed AU, Ahmed CF, Samiullah M, Adnan N, Leung CKS. Mining interesting patterns from uncertain databases. Inf Sci 2016;354:60–85.
[5] Lin JCW, Gan W, Fournier-Viger P, Hong TP, Tseng VS. Efficient algorithms for mining high-utility itemsets in uncertain databases. Knowl-Based Syst 2016;96:171–87.
[6] Lian X, Chen L. Probabilistic top-k dominating queries in uncertain databases. Inf Sci 2013;226:23–46.
[7] Lee J, You GW, Hwang SW, Selke J, Balke WT. Interactive skyline queries. Inf Sci 2012;211:18–35.
[8] Gao Y, Liu Q, Zheng B, Chen G. On efficient reverse skyline query processing. Expert Syst Appl 2014;41(7):3237–49.
[9] Lu H, Zhou Y, Haustad J. Efficient and scalable continuous skyline monitoring in two-tier streaming settings. Inf Syst 2013;38(1):68–81.
[10] Zhang W, Lin X, Zhang Y, Wang W, Zhu G, Yu JX. Probabilistic skyline operator over sliding windows. Inf Syst 2013;38(8):1212–33.
[11] Lian X, Chen L. Efficient processing of probabilistic group subspace skyline queries in uncertain databases. Inf Syst 2013;38(3):265–85.
[12] Papadias D, Kalnis P, Zhang J, Tao Y. Efficient OLAP operations in spatial data warehouses. In: International symposium on spatial and temporal databases. Berlin Heidelberg: Springer; 2001. p. 443–59.
[13] Lee YW, Lee KY, Kim MH. Efficient processing of multiple continuous skyline queries over a data stream. Inf Sci 2013;221:316–37.
[14] Elmi S, Tobji MAB, Hadjali A, Yaghlane BB. Selecting skyline stars over uncertain databases: Semantics and refining methods in the evidence theory setting. Appl Soft Comput 2017;57:88–101.
[15] Gao Y, Miao X, Cui H, Chen G, Li Q. Processing k-skyband, constrained skyline, and group-by skyline queries on incomplete data. Expert Syst Appl 2014;41(10):4959–74.
[16] Yong H, Lee J, Kim J, Hwang SW. Skyline ranking for uncertain databases. Inf Sci 2014;273:247–62.
[17] Lee J, Hwang SW. Scalable skyline computation using a balanced pivot selection technique. Inf Syst 2014;39:1–21.
[18] Miao X, Gao Y, Chen G, Zhang T. k-dominant skyline queries on incomplete data. Inf Sci 2016;367:990–1011.
[19] Kulkarni RD, Momin BF. Skyline computation for frequent queries in update intensive environment. J King Saud Univ-Comput Inf Sci 2015.
[20] Jiang T, Zhang B, Lin D, Gao Y, Li Q. Incremental evaluation of top-k combinatorial metric skyline query. Knowl-Based Syst 2015;74:89–105.
[21] Alwan AA, Ibrahim H, Udzir NI. A model for processing skyline queries over a database with missing data. J Adv Comput Sci Technol Res (JACSTR) 2015;5(3):71–82.
[22] Li X, Wang Y, Li X, Wang X, Yu J. Gdps: an efficient approach for skyline queries over distributed uncertain data. Big Data Res 2014;1:23–36.
[23] Bustos B, Pedreira O, Brisaboa N. A dynamic pivot selection technique for similarity search. In: Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. IEEE; 2008. p. 394–401.
[24] Xiao G, Li K, Li K. Reporting l most influential objects in uncertain databases based on probabilistic reverse top-k queries. Inf Sci 2017;405:207–26.
[25] Gao Y, Liu Q, Zheng B, Mou L, Chen G, Li Q. On processing reverse k-skyband and ranked reverse skyline queries. Inf Sci 2015;293:11–34.

A.Sairam, has more than 12 years of academic and 5 years of research and writing experience in the field of Data retrieval, Cloud storage, Data mining, Mobile Networks, Dynamic Clustering, database theory, data mining, Knowledge Engineering, big data analysis. He has published his research work in various interested fields.

Dr. C.Suresh Gnana Dhas has more than 28 years of experience in academic and research works. He is strong in the area of Artificial Intelligence, Data Storage and Retrieval, MANET, Image processing, Cryptography and Network security. Suresh has published papers with professional cycling teams and various research bodies. He has delivered speeches in many reputed institutions and Universities. He is the organizing chair of many International conferences. He has completed B.E, M.E and Ph.D. in Computer Science and Engineering.