DOI: 10.21917/ijsc.2020.0311

IMPROVED SKYLINE QUERY RETRIEVAL USING PARTICLE SWARM OPTIMIZATION BASED SWEEP LINE OPERATOR OVER REAL TIME DATASETS

A. Sairam and Seenuvasan Arumugam
Department of Computer Science and Engineering, Thirumalai Engineering College. India

Abstract
In this paper initially clusters the search area's slopes, i.e. it is shaped into settings according to its behavior in the search area, both past and present. In this study, these points were identified using a PSO control unit that works in a multi-dimensional search space. A PSO controller is employed to find the points in the search area under the new framework suggested in the paper. It contains several pre-processing methods for clearing incomplete or uncertain data in the area of unsafe data. Various preprocessing operations include: sort, fusion, filter and intercluster predominance in dimensional sets. The dominance between local points occurs at the nearest distance from each other. The points are of a global rank and sent in order to provide a rank according to the specific query to the PID controller. Moreover, the system proposed removes the point non-skyline in the search area by means of 2 new algorithms: Dynamic Pivot Sweep Line (DPSL), which can reduce the reaction time for a particular query. DPSL provides an ideal mechanism for searching the search area with skyline points, providing the best comparison. The DPSL algorithm is combined with real-life and synthetic preprocessing and PSO data sets for reducing storage and removal of redundant data in multi-dimensional search spaces. The whole controlled processing uses the values of the past and the present lines to predict future instances. This results in the dynamic query operation of the skyline and gets the whole data according to the specific query. In addition, PSO controller reduces response time, which saves more time than standard methods.

Keywords:
Skyline, PSO, Sweep Line, Retrieval Engine, Real Time Data

1. INTRODUCTION

Recently, research into the uncertain database has attracted many researchers to improve data mining capacity and increase interest in the field of data remediation, data extraction, real and synthetic data sets, data integration etc. [1]-[4], and gain a high degree of popularity. In various applications such as sensor networks, social data and market analysis, uncertain data is considered important. Increased data accumulation in all these applications leads to an important analysis of the basic tools for data analysis in the case of uncertainty.

The use of several techniques to improve the decision support system and support process related to a rapid increase in multi-dimensional data sets is used to extract meaningful insights from the uncertain database. Through various methods like mathematical, analytical, statistical, and data mining, decision-making is combined with operational research.

For obtaining top skyline points, e.g. top-*k* query models, several methods such as rank-based scoring methods are used. Methods such as cumulative data score help reduce large-scale data sets. However, the instances relating to the query can never be found. This only works with separate scalar values.

Taking into account this method, the questions of skyline deflect ranking methods such as guidance and top-*k* query approaches. A preference for data attributes indicating the likes and dislikes of the user is found in every skyline query. This helps to remove objects not rated by any user from the operation. The positive and negative preferences for rating a product are nevertheless considered the same. This forms a small subset of the most interesting skyline points or user preference data items. This is known as an optimal or skyline set for Pareto.

In case of a large database, Skyline query has become an important problem, extracting interested items from multidimensional datasets. For many data mining apps with no cumulative function, a query process can be applied to find the best query based on the preference of users in the decision-making process. In the multiple-criteria datasets assessment tools are used to filter the points out from the skyline operator. This is interesting because during evaluation of criteria the issue is not dominated by other points. The popularity of the skyline depend on the simplicity of the paradigm and is mainly applicable to multi-criteria user preferences support system [17] – [19].

1.1 SKYLINE OPERATOR

The main aim is to return or find objects that are not dominated in the database by other objects. Therefore, the skyline operator (*fs*) is expressed by n-dimensional points in an uncertain database (*Db*),

$$f_s\left(Db\right)=\frac{Db}{\left\{o\middle|\exists p\in Db\left(\forall i\in\left\{1,2.,,,n\right\}:p\left(i\right)<o\left(i\right)\right)\right\}} \quad (1)$$

In many application sectors, unsafe data, personal identifying systems, or moving objects that have unsafe data, are generally referred to as unsafe data. The reference could therefore be that the skyline operator only applies to uncertain data in which data objects with a single vector are represented, i.e. gravity center (see Fig.1). This combination leads to loss of information.

Other than unsafe data, uncertain data are also available [5]. In case of the extraction of objects like satellite vessels from an image, it is very useful for the probability for each data object that refers to the confidence level of each object. Therefore, such probability values should be taken into consideration by the skyline operator. The object in the database is set as a skyline object if it is less than one probability. Here, for a number of different large uncertain vessels, the calculated skyline.

In addition, all objects in the database can be interpreted as unsafe or local. Uncertain or data set queries for skyline could be the key tool for detecting the most valuable players. The vector dimension refers to the achieved points, the total assists and the complete rebounds in a given game. Each player is a vector set. Therefore, uncertain local data is shown as a vector set.

Existential uncertain data are, however, similar to the proportion of played games.

Processor, brand, operating system and memory are referred for domain knowledge. Some features or features can be accepted as faster processors. However, preferences for other features cannot be accepted as variations are purely based on exact preferences between customers. Furthermore, the relative weight of factors is also uncertain.

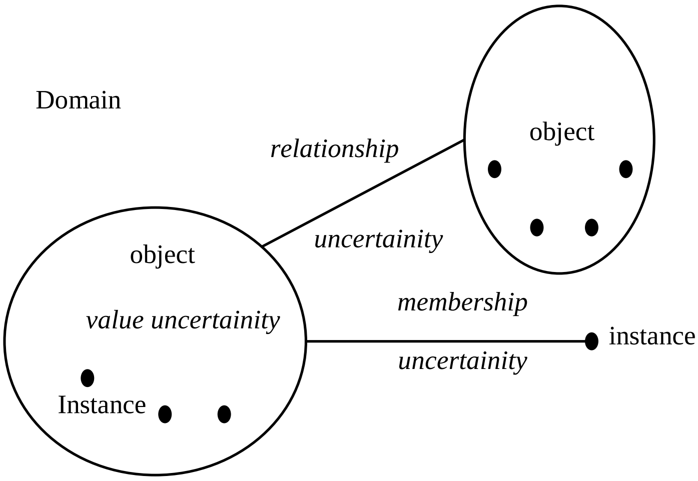


Fig.1. Relationship between various uncertainties

Three types of uncertainty: value, membership and relationship correspond to three uncertain levels of information: object, instance and field (refer to Fig.1). An uncertain dataset, for example, may be considered an uncertain set of objects with a number of events that possess several attributes. Uncertainty value represents non-unique objects whose representation is based on probabilities of instance. The uncertainty of membership checks if an instance is a precise object representation. Finally, uncertain knowledge of the domain is demonstrated with uncertain relationships that compare the insecurity of the objects.

In general, the issues associated with these 3 definitions can be solved with the aggregation of important knowledge data. Using an outlier detection to sum up the cases, an anomalous and noisy observation is well discovered. Here, the uncertainty of membership can be reduced by finding the noisy instances that help to find the true cases. The sequence of skyline values across each dimension is learned and the domains can therefore be summarized. This increases the domain knowledge acquisition and provides a strong representation of each dimensional domain value, in which the objects are uncertainly related.

The principal contribution of the work proposed involves: Increasing the response time in an uncertain database by using the Skyline method. Increased processing ability in uncertain databases to enhance system scalability. This is done by removing the missing or incomplete values from uncertain datasets. In case of uncertain data, to reduce the number of several comparisons in pairs between incomplete and complete skylines. In general, the issues associated with these 3 definitions can be solved with the aggregation of important knowledge data. Using an outlier detection to sum up the cases, an anomalous and noisy observation is well discovered. Here, the uncertainty of membership can be reduced by finding the noisy instances that help to find the true cases. The sequence of skyline values across each dimension is learned and the domains can therefore be summarized. This increases the domain knowledge acquisition and provides a strong representation of each dimensional domain value, in which the objects are uncertainly related.

2. LITERATURE SURVEY

This section discusses findings from the literature on the technique of finding points on an uncertain database. While mining uncertain data for user generated queries, imprecise information always exists. Such inaccuracy in detection is usually reduced using a skyline query technique that produces skyline points for querying at the end of the user. Even if the database is uncertain or incomplete, the skyline query technique is designed to provide better skyline points. If the technique of skyline query cannot load the skyline points, it leads to irrelevant results. This section, which is given below, discusses some of these techniques.

In [6] the authors help to find the right skyline points, which are based on the user preferences and the size of the retrieval process. In this section. In addition, custom skylines were ranked on the basis of the user preferences as a dynamic search for sub-spaces. The compressed structure for reducing storage space was submitted to an algorithm to study the construction of the interleaves for the particular scenario. In addition, two algorithms are suggested to support user preference based on preference for equivalence, and to improve model effectiveness.

In [7] the authors suggested an overhead mitigation algorithm to treat the tuples on completely and partially ordered domains. Because of massive data collections, the proposed method uses the FAST-SKY algorithms to solve the problem of high dimensionality overhead and poor dominance comparison. The assessment of gradual slope in its work guarantees a topological order of sorting and new index structures. Partly ordered domains were adopted using the stratification technique with index data and proposed two new index structures. This includes: stratified *R*-trees for low and high-dimensional data, and stratified MinMax heaps. So the quick prevailing comparison and reduction in dimensionality is achieved by reporting instead of dominance queries.

In [8], the authors conducted an investigation into restricted, unstructured distribution queries in uncertain datasets. Geographically, the data is broken up and partition algorithms are proposed to divide the data sets into equal size groups. This enables parallel groups to be paralleled without changing the results. For parallel processing of skyline between the partitioned group datasets, a PaDSkyline algorithm is proposed. It further utilizes the group optimization process and uses a multi-filter model to improve skyline query processing across each group. In order to recognize the unqualified points on the data set, local skylines with inquiries are also sent through filter points. This reduces network redundancy and enhances the response time. In order to find the filtering points in the dataset and heuristic suggestions for guide the filtering points on a super set, the author has a cost effective model.

In [9], the authors addressed issues associated with subspace skyline calculations in distributed settings. In order to reduce overhead, this method uses a preprocessed skyline called skyline views. Furthermore, it was proposed that the distributed subspace skyline method reduces the total cost by leveraging the skyline's views.

In [10] the authors suggested the finding of non-spatial attributes in the Skyline Space using two index based methods. A dominance and an enlarged R-tree are included in the two index methods. The first approach is a solution based on which the objects are linked to non-dominance, without the control of other query points. The enlarged R-tree uses aggregated non-spatial attributes to locate the dominance control in index nodes during the index transverse process. The skyline queries also allow to use the dominant diagram to compare query and non-dominance points independently and in parallel. This increases search performance in the search area.

In [11], the authors proposed to separate the query space into grids the grid-summary distributed probabilistic skyline method. The distribution of the data process is tedious due to ineffective query aggregation in multi-criteria decision making on uncertain data. Therefore, with unknown global data values, the complexity of the system increases and creates a challenge to find a skyline over an uncertain database. To find the effective queries using the iterative feedback mechanism via a grid summary the problem of distributed probabilistic skyline query will be improved. Furthermore, local and server cutting and multiple selection are used to optimize requests to achieve optimal points.

In [12] the authors proposed the process of SkyQUD in order to calculate the probability of skyline in an unsure database. This approach aims to find out which skyline points are applicable in full and in certain databases in all dimensions. The aim of the proposed study is also to improve skyline calculation in all dimensions with better determination of dominance relations. Furthermore, the theory of the dominance relationship is used to improve the relation between the skyline points in the data set, and offers good support in finding the leading skyline points according to user preferences.

In [13] the authors examined the problems when the skyline points were identified in a large spatial dataset in the area of Hadoop, where spatial operation is effectively carried out. This study is considered interesting as it uses mobile application generated spatial big data, and the skyline operator is further used as a decision maker for enterprise intelligence system. A sketch query framework was therefore designed to find solutions to questions at the top of spatial Hadoop. The filtering method was parameterized, which skyline the candidate points and fuses with the local skylines. Two different algorithms were also used to improve query processing with the above framework. This process uses an efficient filtering and fusion optimization model to enhance the recovery efficiency in large scales.

In [14] authors looked through the skyline join query at the multiple relationship between the skyline queries. This helps to locate the lines from several sources. The authors have examined problems associated with skyline join query and the existing algorithms operate on two relationships and dismiss the common event between the two relationships. In this study the common event between two or more relationships was found through Skyjog or skyline joint algorithm. The results of joining skyline points can be identified with easier calculation and the group breakdown approach is used to reduce the intermediate results and to eliminate redundant or complicated computations.

In [15], the authors proposed two algorithms in order to process the skyline points and eliminate the central issue associated with locally based global points. This algorithm contributes to the reduction of dominance testing and to parallel dominance testing. The old algorithm divides the space of the skyline into segments with an angle grid partition system. The Dominance test is reduced by two rules set for two partition methods. This can help to locate the local skyline points. The second algorithm filters non-skyline and incomplete items with grid segments to reduce the domination tests in reducers and mappers. The connection between the mapper and the reducer.

In [16], the authors investigated two problems in uncertain data, namely veracity issues and the amount of data in the uncertain database. In order to resolve this issue, a resource description framework was proposed. For veracity management, an insecurity or possibility theory has been used to represent and manage data in a possible description of resources. The large data relating to the resource description framework were filtered by a skyline operator. This has been used to find the small resource set to meet user needs and to extract it from the resource description framework, which is not dominated by another resource as per the Pareto principle.

3. PROPOSED SKYLINE IDENTIFIER

A skyline object or point is considered as an object, which is not dominated by other skyline objects. Consider a skyline object (p) dominates another skyline object (q), when the skyline object p is better than q in one dimension and in remaining dimension, skyline object p is equal or better than object q. It is supposed to be arranged orderly in the domains of each dimensions. In the uncertain database, reasoning and forecasting are considered to be key areas of artificial intelligence. In terms of preferences or grading, the unquantifiable attributes or objects in a database are compared. However, a user's elicitation or annotation is used to reach an object's pair presence rather than a full attribute value.

The skyline query mechanism is considered as a multi-criterion tool for improving support decisions in real time database applications. The skyline search mechanism works in a given dimensional space on the behavior of skyline points. The entire mechanism depends on the dominance of the point, which makes the best skyline point to choose from. The dominance and identification of other skyline points is closely linked. Based on the available top skyline points, the user-specific question is answered.

Skyline queries are intuitive and flexible to assist the user in developing his behavior rather than choosing other query techniques. The access of unsafe or incomplete data in multi-dimensional data sets tends to make skyline query technique more comprehensive and related to all skyline points in the field of data. In addition, the incomplete or uncertain data available makes it futile to use the skyline operator to identify skyline points as a whole. The cleaning of unsure data with the correct preprocessing operation does, however, valuable in a multi-dimensional dataset for skyline querying. Various methods are used to eliminate repeatedly, incomplete or minimum data in the large space of data on the features of the dominance concept behaviour. Such removal of data prevents redundancies in the space of sloping data which provide useful comparisons between the slopes to achieve highest ranking.

Take a certain data set with skyline points, when skyline query returns the objects or points. The points are not dominated in skyline space by other points. Specifically, consider a given D-dimensional dataset P, if a point $p_1 \in P$, which dominates the point $p_2 \in P$. It must hold: (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 i^{th} dimensional point value and the smaller value is better. For instance, a set $P = \{a,b,\ldots,h,i\}$ is considered as a graph of hotels and the price for a hotel a is cheaper than the price of hotel b, and hotel a is nearer to the beach than hotel b. Therefore, the hotel a tires to dominates over the hotel b at the time of searching for the given query. In the same way, different hotels within skyline P and so on dominate the skyline points of other hotels. The results show that the skyline inquiry method provides solutions for user-defined queries and helps with better decision making for multi-criteria. Finally, the results of hotel queries are sent to users or tourists as results. The tourist will have the opportunity to view the nearest hotels at a low fare to the beach.

Such a skyline querying method faces some problems in the way the skyline points can be obtained according to the query. This includes an increase in the response time based on the user's query. With high dimension data sets and increasing data size, the response time is further increased. The availability of null or empty datasets compares the points in the skyline space in several times. This reduces the response time and the skyline querying system's scalability. This section deals with skyline query response time using a novel PSO approach with two different algorithms.

3.1 PSO FRAMEWORK

This section is originally divided into sizes according to past and present behavior in search space. The sizes of the search spaces are clustered in this section. In this study, these items are found using a PSO that operates specifically in a multidimensional search environment. A PSO is used to find skyline points in the search field in the new framework in the paper suggested. It includes several pre-processing methods in an area of uncertainty for clearing incomplete or uncertain data.

Different pre-processing operations include: sorting, merge, filtering and inter-cluster dominance in the dimensional sets. The dominance was done between local areas at the nearest distance between each other. The operation was carried out. The points received are given a global ranking and sent to the PSO in response to the query.

In addition, with two novel algorithms, the proposal eliminates the non-skyline points in the search area for reducing the answer time for a particular query, namely: the DPSL-PSO. Both algorithms offer a perfect search mechanism for the point in your search area and make the best comparison for the points in skyline. For reducing storage space and removing redundant data in multi-dimensional search area, the DPSL algorithm is coupled with pre- and PSO operations through real-life and synthetic data sets. The whole processing takes place in a controlled way, using the values of past and present points to determine the instances of the future. This results in a dynamic query process and recovers complete data according to the given query.

The main steps of proposed method is given below:

Step 1: Database with entire data is contacted to generate local skyline points after the query given by user. It generates local skyline points is with incomplete datasets and non-dominance relations between them.

Step 2: Reduction Phase or PSO phase (PSO reduces the error rate between the dominance points)

Step 3: Data is sorted based on price, distance and rating

Step 4: Merge the results of price, distance and rating from other partitions

Step 5: Filter out the dominance points from price, distance and rating

Step 6: Save the retrieved local reduced skyline points with dominating scores and preference scores for each query in PSO

Step 7: Step 3.Generate the Global skyline points using DPSL algorithm

Step 8: Merge the global skyline with local skyline points using DPSL. DPSL compares the non-dominated local skyline points to ensure that the return data items are the skyline of the entire database.

Step 9: Evaluate the proposed method

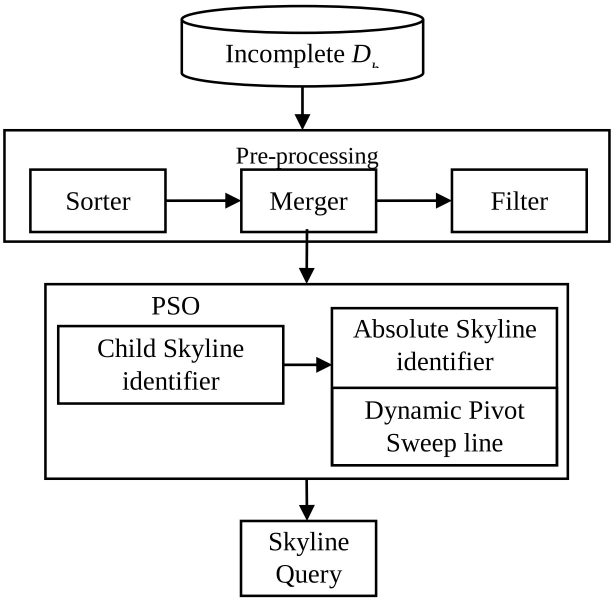


Fig.2. Proposed PSO Framework

This section addresses the removal by the suggested framework of the dominance data or the incomplete data in a data set. Usually in the skyline query method the removal process is performed to remove dominance resulting from cyclic discharge, repeated comparison and transitivity with incomplete data. Incomplete data, including type, merge and filters, the dominance and Skyline Identification with the PSO, and child-skyline and absolute skyline identifiers, can be removed from the PSO framework in five components. This is illustrated in Fig.2 and the work below:

Step 1: Preprocessing

a. Presort the data and Make sure that no point can be dominated by the ones comes after it

b. After sorting on the data set, the one with the maximum value will be certainly the skyline point and low values are filtered out.

c. Since attribute domains are totally ordered, it is possible to partition them. This idea is at the heart of divide and conquer approaches, which have been pioneered in the computational geometry field. The merging is developed for dealing with large instances that do not fit in main memory.

Step 2: The main idea is to merge the partitions, thus reducing the number of queries that have to be performed. Partitions that are contained in other ones can be eliminated in the process.

Step 3: Interesting or non-dominance points are created by skyline queries. The skyline absolute identifier finds the interesting points using resultant feedback. The filtered results consisting of interesting points is given as an input to the PSO. Finally, the resultant set is stored as skyline order through PSO, which predicts the future local or global skyline points.

Step 4: Local Skyline Computation will include the originator to send the query to only one of its neighbor which will compute its local skyline and send the query to one of its own neighbors that have not received the query yet. When there is no other neighbor to propagate the query or the query has been completed, the results are returned through the same path and merged along the peers. Finally when the results have reported to the originator it will be responsible to compute its local skyline, remove the duplicate tuples that received and compute the final skyline.

Step 5: The global skyline point computed will not be globally dominated by another point. From the local skyline points, non-dominance point is selected as global skyline point.

3.2 PSO CONTROLLER

The performance is based on the PSO tuning with its parameters. The tuning helps to improve the accuracy of the skyline points and helps to predict future situations. Therefore, with reduced redundancy or incomplete data sets, system performance is increasing. If the dataset remains redundant, or is still incomplete, PSO performs poorly and with the worst prediction. The dataset is selected using the value of predictive skyline points.

The prediction of uncertain skyline values shows that the variance values predict that contribute to increasing the prediction ability rather than to measuring greater variance. The presence of a greater difference indicates a poor level of confidence and the other way round. By the use of the Gaussian predictive variance the skyline points are selected with great uncertainty.

This process detects the high level of uncertainty and estimates the final range. The interpretation of predictive variance in the proposed procedure improves predictive accuracy and suggests that the difference value in the predictive region is empty. The predicted data is added in the region of high variances or the threshold value determination by means of the parameter hyper magnitude discovers the area of high certainty.

This process is called the process variance estimator and is affected by changes in the preprocessing procedure by the threshold predictive variance. The lack of past data leads to assumed values or scaled thresholds. Finally, when the points exceed the threshold level, the highest points of certainty are collected from prediction points. The top-Skyline point has a higher certainty.

3.3 DYNAMIC PIVOT SWEEP LINE ALGORITHM

The proposed pre-processing framework and optimization algorithms improve the response speed of skyline queries. The algorithms that were proposed for precision analysis and robustness improvement in so real and synthesized datasets. These two algorithms are part of the proposed method to improve the work shown below:

In order to identify the skyline points in the dataset through preferential dominance in the PSO feedback, the proposed DPSL algorithm is employed. Here the PSO operation takes place. However, the DPSL algorithm is used for the selection of skyline points during queries. The PSO is used to input the sweeping line algorithm. The dynamic pivot concept makes the skyline dots from the trained PSO data sets a good selection and is considered as the first tests. Finally, all points in the skyline data set are performed in accordance with the user query. This is done with the feedback received from the skyline points to PSO and the output. In addition, a single heap sorting search is used to test the proposed dynamic pivot line algorithm on real and synthetic datasets.

The node calculates the value and value of dominance or preference. The PSO method estimates skyline points in the dataset based on the resulting score obtained from the PSO following the pre-processing operation. The results on the organized dataset are computed in a single transition. During the initialization process, the algorithm accepts the Skyline data sets. Once this has been completed, min-heap sorting is carried out on the basis of the PSO output to find the skyline points. The data sets are finally sorted from the unorganized datasets. In the root node of the min-heap sort tree, the sorted skyline points in the data set are saved.

Skylines with the highest range of non-dominance are obtained using the proposed framework and the final sets are iteratively swung by a sweeping line from the left coordinate axis to the right coordinate axis. The estimated points depart from the trained data for the lines. The estimate is entirely based on the node value previously visited and the dominancy of skyline points obtained from PSO using a dynamic pivotal algorithm. The skyline number is calculated and the sweep value is periodically updated in PSO to get the highest skyline point. Finally, the PSO DPSL algorithm helps to predict exactly the best points of skyline for future questions.

Algorithm 1: Dynamic Pivot Sweep Line Algorithm

Input: The dataset and returned skyline object

Output: Desired skyline dataset

Step 1: Initialize the set, S (= θ) that accepts skyline points (S, μ, τ)

Step 2: Perform the sort operation in PSO Framework with min-Heap sorting, H, ($H = H' = \theta$)

Step 3: Insert all the sorted entries of root node of the tree, R into H and H'

Step 4: Compute the skyline points using PSO framework with H'.

Step 5: Obtain the Top–*Data sets* using PSO framework

Step 6: *Refine Top-Data sets*: Form a vertical axis to sweep iteratively the skyline points along its coordinate axis (from left to right).

Step 7: Add dynamic pivot model over the obtained points to reduce the difference between the test and training data (existing data with PSO). This is calculated using class distribution function.

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

where, pr/nr represent the trained class distribution data and S represent Pivot model with M models.

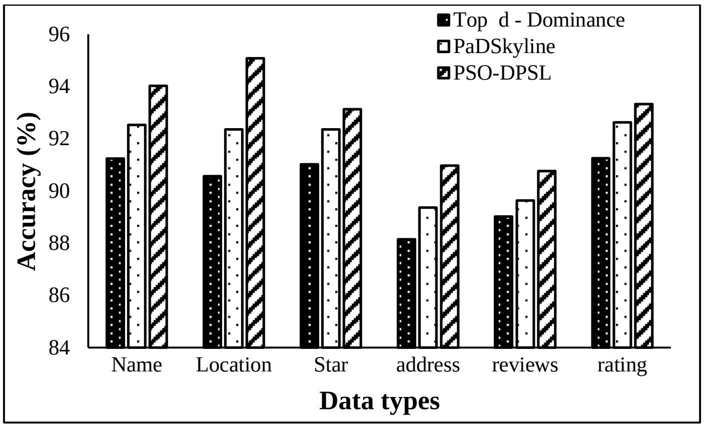
Step 8: Count of the object (C_o) is evaluated with PSO framework

Step 9: Update the Sweep value to the PSO. (PSO reduces the difference between the values of past inputs of skyline point with the sweep variable)

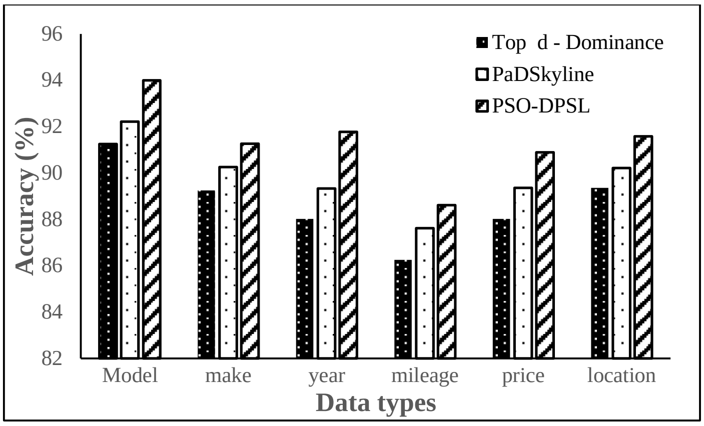
Step 10: Updated PSO senses future prediction points accurately.

4. PERFORMANCE EVALUATION

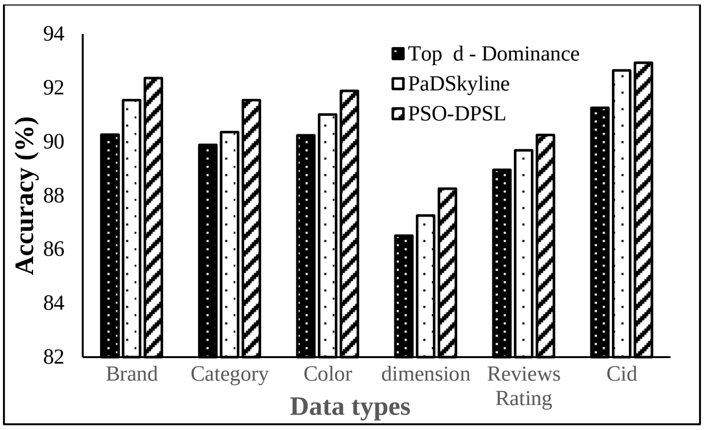
This section covers the assessment of the system proposed to test its efficiency in the search. In real-life and synthetic datasets, the proposed framework is tested to be effective against uncertain datasets. In order to test efficiency against faster response time, the proposed method was experimented on various datasets. The experiments on the proposed method have calculated with an improved computational system. The accuracy of the results collected against incomplete datasets is evaluated. This shows how highly skyline points have been returned for a certain query to the query site. The purpose of the system proposed is to find the maximum number of skyline points over each dimension in multi-dimensional array.



(a) Hotel Datasets



(b) Car Datasets



(c) Online shopping Datasets

Fig.3. Accuracy testing over Real-Life Datasets

The DPSL heap type improves the skyline query, and PSO also increases the skyline query by two-minute heap sorting. With aR-tree calculation, the recovery efficiency increased over skyline queries, which reduced the response time compared to the PSO framework for the DPSL algorithm. DPSL's two-minute heap sorting caused the response time at the initial stage to further reduce the number of missing components. The response time for the skyline points was reduced. The result is that in smaller datasets and with increasing dataset dimensions, the proposed method worked better with reduced scalability. Therefore, the conclusion is that DPSL can be used for a larger dataset and can handle the skyline queries because it more than other traditional methods lowers the response time.

PSO is used in this section to process the multi-dimensional dataset for skyline queries. Furthermore, two new algorithms are added to increase skyline query system response speed as an excellent method. The added benefit of these DPSL algorithms is the improvement of the proposed framework and the accuracy of the query set. By using the dynamic pivot concept, the best way to recognize the skyline is further improved. The response time for processing the query is further reduced by the PSO frame, increasing the answering speed. In Skyline datasets, the PSO framework provides a variance and predicts results of the query.

The PSO removes larger variance and prevents further failure to track and reduces the response time. The sublime PSO algorithm monitors the response time and thus the frame for identifying high-ranking queries is considered efficient. Thus, it helps to delete the missing values and to find the skyline points related to the questions in the database. An extensive experiment to verify the efficiency of the proposed system via a multi-dimensional data set was carried out. The analysis was performed to identify the processing time and response time of skyline objects across the synthetic and real-life datasets. With faster response time and object processing time the experimental results showed that proposed method. The proposed method has been shown to be effective as it increases operational scalability compared to other methods in which the response time of existing methods is higher. The versatility of DPSL algorithm PSO has demonstrated its effectiveness over a skyline database against conventional methods with reduced data sets in order to enhance query search.

5. CONCLUSIONS

The problems related to the skyline points have been analyzed in this study in the case of effective skyline queries over evidence data. This study shapes a way to control the operation in a sequence of steps to remove incomplete data sets. Finding the future skyline points, it handles the operation through past and current data. This algorithm was developed and combined with two case studies of the tree base to test the effectiveness of the algorithm. The experimental assessment shows interest in finding the skyline points of the proposed algorithm and demonstrates the improved scalability of the algorithm compared with other probabilistic methods. The problems related to the skyline points have been analyzed in this study in the case of effective skyline queries over evidence data. This study shapes a way to control the operation in a sequence of steps to remove incomplete data sets. Finding the future skyline points, it handles the operation through past and current data. This algorithm was developed and combined with two case studies of the tree base to test the effectiveness of the algorithm. The experimental assessment shows interest in finding the skyline points of the proposed algorithm and demonstrates the improved scalability of the algorithm compared with other probabilistic methods. Such constraints are eventually prevented by using the proposed optimization method, which allows users to flexibly select objects from the database according to the user's preference. This study allows the accurate extraction of skyline data points in terms of an incomplete and uncertain database compared to any other methodology.

REFERENCES

[1] L. Antova, C. Koch and D. Olteanu, "MayBMS: Managing Incomplete Information with Probabilistic World-Set Decompositions", *Proceedings of IEEE International Conference on Data Engineering*, 1479-1480, 2007.

[2] D. Burdick, P.M. Deshpande, T.S. Jayram, R. Ramakrishnan and S. Vaithyanathan, "OLAP over Uncertain and Imprecise Data", *International Journal on Very Large Data Bases*, Vol. 16, No. 1, pp. 123-144, 2007.

[3] T.J. Green and V. Tannen, "Models for Incomplete and Probabilistic Information", *Proceedings of International Conference on Extending Database Technology*, pp. 278-296, 2006.

[4] P. Sen and A. Deshpande, "Representing and Querying Correlated Tuples in Probabilistic Databases", *Proceedings of IEEE International Conference on Data Engineering*, pp. 596-605, 2007.

[5] X. Dai, M.L. Yiu, N. Mamoulis, Y. Tao and M. Vaitis, "Probabilistic Spatial Queries on Existentially Uncertain Data", *Proceedings of International Symposium on Spatial and Temporal Databases*, pp. 400-417, 2005.

[6] X. Lin, Y. Zhang, W. Zhang and M.A. Cheema, "Stochastic Skyline Operator", *Proceedings of IEEE International Conference on Data Engineering*, pp. 721-732, 2011.

[7] K. Benouaret, D. Benslimane and A. Hadjali, "Selecting Skyline Web Services from Uncertain QoS", *Proceedings of IEEE International Conference on Services Computing*, pp. 523-530, 2012.

[8] A. Hadjali, S. Kaci and H. Prade, "Database Preferences Queries–A Possibilistic Logic Approach with Symbolic Priorities", *Proceedings of International Symposium on Foundations of Information and Knowledge Systems*, pp. 291-310, 2008.

[9] J. Lee, G.W. You and S.W. Hwang, "Personalized Top-K Skyline Queries in High-Dimensional Space", *Information Systems*, Vol. 34, No. 1, pp. 45-61, 2009.

[10] H. Jung, H. Han, H.Y. Yeom and S. Kang, "A Fast and Progressive Algorithm for Skyline Queries with Totally and Partially-Ordered Domains", *Journal of Systems and Software*, Vol. 83, No. 3, pp. 429-445, 2010.

[11] L. Chen, B. Cui and H. Lu, "Constrained Skyline Query Processing Against Distributed Data Sites", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 2, pp. 204-217, 2011.

[12] J. Lee, J. Kim and S.W. Hwang, "Supporting Efficient Distributed Skyline Computation using Skyline Views", *Information Sciences*, Vol. 194, pp. 24-37, 2012.

[13] Y.W. Lee, K.Y. Lee and M.H. Kim, "Efficient Processing of Multiple Continuous Skyline Queries Over a Data Stream", *Information Sciences*, Vol. 221, pp. 316-337, 2013.

[14] X. Li, Y. Wang, X. Li and J. Yu, "GDPS: An Efficient Approach for Skyline Queries over Distributed Uncertain Data", *Big Data Research*, Vol. 1, No. 2, pp. 23-36, 2014.

[15] N.H.M. Saad, H. Ibrahim, A.A. Alwan, F. Sidi and R. Yaakob, "A Framework for Evaluating Skyline Query over Uncertain Autonomous Databases", *Procedia Computer Science*, Vol. 29, pp. 1546-1556, 2014.

[16] D. Pertesis and C. Doulkeridis, "Efficient Skyline Query Processing in Spatial Hadoop", *Information Systems*, Vol. 54, pp. 325-335, 2015.

[17] J. Zhang, Z. Lin, B. Li, W. Wang and D. Meng, "Efficient Skyline Query over Multiple Relations", *Procedia Computer Science*, Vol. 80, pp. 2211-2215, 2016.

[18] J.L. Koh, C.C. Chen, C.Y. Chan and A.L. Chen, "MapReduce Skyline Query Processing with Partitioning and Distributed Dominance Tests", *Information Sciences*, Vol. 375, pp. 114-137, 2017.

[19] A. Abidi, S. Elmi, M.A.B. Tobji, A. HadjAli and B.B. Yaghlane, "Skyline Queries over Possibilistic RDF Data", *International Journal of Approximate Reasoning*, Vol. 93, pp. 277-289, 2018.