A Survey on Probabilistic Query Methods of Uncertain Data

Alok Kumar Mishra , Yuan Chen , Chhavi Vishnoi, Priyanka Vyas
amishra2@kent.edu, ychen135@kent.edu , cvishnoi@kent.edu , pvyas2@kent.edu

Abstract

Probabilistic data is data that is based on behavioral events and whose value is determined on the basis of past conditions and probabilities. Probabilistic models incorporate randomness in their approach which leads the data to being uncertain. Data uncertainty leads to errors in the database and leads to incorrect output. This survey intends to provide a summary of various papers detailing the query methods and data models to cope with the issue of Uncertain data. Various query types like range queries, skyline queries, probabilistic threshold queries, etc, and how they are used to manage the uncertainty of the data have been explored in this survey in section 1. Probabilistic models and applications are defined in section 2. We also explored the challenges and optimization of uncertain data queries in this survey.

Introduction

Uncertainty is unavoidable and happens in all the events we encounter in the real world. A moving object has an uncertainty of its location at a particular time. It is unfeasible to store the location value of a moving object in all instances, so we store the values at intervals. This results in uncertainty of data for that interval. Similarly, in temperature sensor devices it is not possible to store all the values in the database. So the values are only stored only when there is a change in the sensor input temperature. This causes uncertainty in the query of temperature at time t.

According to the US National Research Council, *“uncertainty is a general concept that reflects our lack of sureness about something or someone, ranging from just short of complete sureness to an almost complete lack of conviction about an outcome* ***[41]****.”*

Though it is hard to eliminate the uncertainty there are many methods used to gain confidence in the uncertain data such as Probabilistic, fuzzy, and Probabilistic databases that constitute major ways to tackle uncertainty in data. A combination of traditional databases such as Query processing, Join processing, and estimation with traditional mining problems such as clustering, outlier detection, and classifications is the probabilistic database.

Categorized summary of query type of the probabilistic data			
Probabilistic Spatial query		Probabilistic Preference Query	
Range queries. PRQ, PNNQ	Probabilistic Spatial/Similarity Join Query	Skyline query	Top-k Query
KNN Query (K-NN)	Group Nearest Neighbors query		
Probabilistic threshold query (PTQ)	Aggregate NN Query (ANN)		
Probabilistic parking queries (PPQ)	Probabilistic reverse nearest neighbor (PRNN)		

Background

In recent years, with the progress of technology and the deepening of data acquisition and processing technology, uncertain data has been widely paid attention to. In many real applications, such as economic, military, logistics, financial, telecommunications and other fields, the uncertainty of data is widespread, uncertainty data plays the key role, the traditional data management technology can't effectively manage uncertainty, this raises the uncertainty of academia and industry to develop new data management technology.

Query analysis is the ultimate goal of dealing with uncertain data management. There are many types of queries, such as relational query operation, xml processing, streaming data query, Ranking query, Skyline query, OLAP analysis, data mining and so on. Although the number of possible world instances that can be targeted separately is much larger than the size of the uncertainty database, it is necessary to combine sorting, pruning and other techniques to optimize the processing to improve the efficiency of query analysis.

In this survey report, we summarize the main types of currently widely used uncertain data queries from more than 40 papers, and summarize the main characteristics of uncertain data queries in section 1. We also summarize some information about uncertain data mining, data pruning and management applications in section 2 and 3. In particular, we introduce and analyze several typical uncertain queries in detail, such as skyline query, top-*k*, Range queries et al.

Section1: Query Types and Indexing on Uncertain data types

1.1. Skyline query

Skyline queries are a popular and powerful paradigm for incorporating user preferences into relational queries and extracting interesting points from a set of points. When a data item is not dominated by any other data item, this data is said to be a member of the skyline. However, with the diversity of user query requirements, the traditional skyline query is not practical enough and even cannot meet users' requirements.

To address the problem that the number of uncertain skyline queries results is so numerous that cannot offer any practical insights effectively, X. Li, et al. **[1]** propose a dominance-capability-based parallel uncertain k-dominant skyline queries method named PKDS.

C. -C. Lai et al. **[2]**proposes an effective method, mid Indexing (MI), to filter out a large number of irrelevant data in uncertain data streams by sorting the data in order to improve the updating efficiency of the K-dominant skyline.

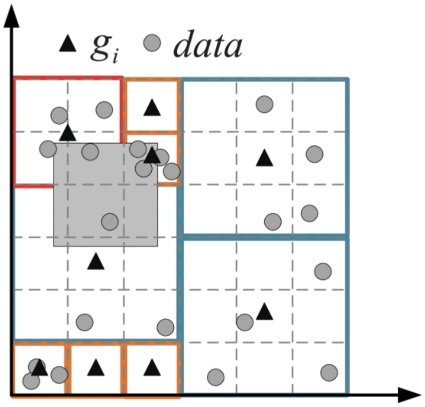
Apart from that, there was a problem of continuous probabilistic skyline queries for uncertain moving objects. To address that problem Shichang Fu et al. **[31]** proposed an algorithm named U-CPSQ (Continuous

Probabilistic Skyline Queries algorithm for Uncertain Moving Objects) to handle continuous probabilistic skyline queries and which proved to be efficient.

1.2. Range queries.

In range queries, the aim is to find all the objects in a given range. Since the objects are uncertain, their exact positions cannot be known, and hence their membership in the range also cannot be known deterministically. Therefore, a probability value is associated for each object to belong to a range. All objects whose probability of membership lies above a certain threshold are retained.

[3] C. Guo et al. mentioned that the pivot set was used to map the search area, and all the squares intersecting the search area were identified. The codes of the center points of these squares together constituted the code set of the search area (Fig [1]).



Fig[1]. Example of range query.

Probabilistic range queries are used to identify the moving objects in the road networks. Y. Shi et al. **[39]** discuss the construction of the processing framework of the probabilistic range query of moving objects in road networks. The probabilistic range query algorithm of moving objects in the road network with an uncertain trajectory caused by sampling frequency is designed and implemented. It can improve query efficiency and ensure query accuracy.

The estimation of link travel times is another issue in road networks as it is hard to determine a reliable path to travel and with less time. Mohammad Asghari et al. **[38]** determines the fastest route on the basis of expected travel times and probability density function(pdf) and according to it the travel time which they got through pdf is proved to be more efficient.

[35] Jian Pei et al. has done the systematic review of some representative studies on answering various queries on uncertain and probabilistic data. It talks about the probabilistic database model and uncertain object model. It also discusses the range search queries and ranking queries.

1.2.1 Calculating Range

***[11]** Wolfson, Ouri, et al. Talks about the calculation of range using this method.* A plane-o is constructed using the position of the attribute of object o. The position is modeled by two functions, First is called upper-o shown as u(t) shows the upper bond of distance from starting position (P.x.startposition, P.y.startposition). Fast deviation is denoted by

BF(t)=min(sqrt(2(V-v)C),(V-v)t) or min{2C/t ,(V-v)} , speed of object is denoted by P.speed=v, the u(t) =vt + BF(t) where t is the number of time unit after P.starttime. Since the object moves on a linear route, the position of x and y can be calculated using u(x, y, t) x, y is a route distance at u(t) from starting position (p.x.startposition, p.y.startposition) The other function to be calculated is lower bound l(t) from the start position (P.x.startposition, P.y.startposition). Slow-deviation denoted by BS(t) = min{sqrt(2(v)C), vt } or min{2C/t , vt }
Similarly p.speed=v we define l(t) = vt-BS(t) where t is the number of time units from p.starttime.
Fig 2 The uncertainty interval of object o at t greater than 0 is equivalent to the distance between the route l(t) and u(t).

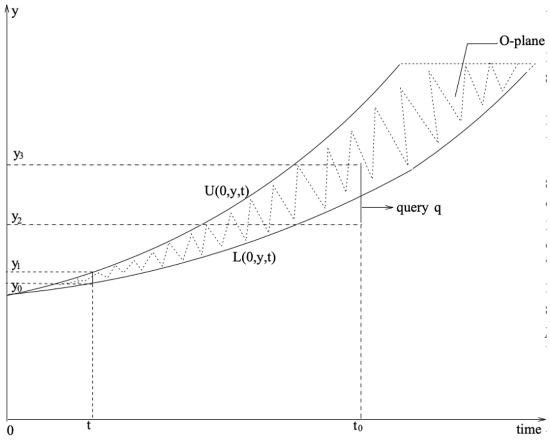


Fig 2: Graph showing uncertainty

1.2.2 PRQ and PNNQ

[17] In this paper the author experimented with probabilistic range and nearest neighbor query over uncertain data of moving objects. They applied both two common models: line-segment and free-moving uncertainty. Uses indexes for efficient execution of queries. In the experiment Probabilistic nearest neighbor outperformed nearest neighbor in free-moving uncertainty.

1.2.3 Querying with density function

Wolfson, Ouri, et al **[18]** in this paper author describe using a density function f(x) of random variable. For small interval f(x)dx lies between [x, x+dx] at time t.
RETRIEVE o FROM Moving-objects WHERE C.
C is a conditional part of the query that depends on the color and location attribute of data.
Similarly below query can be used to retrieve ambulance in region R.
RETRIEVE o FROM Moving-objects WHERE o.type =0 ambulance0 ∧ inside(o, R)

1.2.4 Probabilistic reverse nearest neighbor query

RNN (Reverse nearest neighbor) given an object q returns object o nearest to query object q. Whereas in PRNN it returns nearest uncertain object which has a query object q. ***[19]****Bernecker, Thomas, et al* propose a pruning method using distance to increase the efficiency of PRNN answers.
This paper talks about two methods PRkNN (k==1) and PRkNN(k>=1)

Using spatial pruning method derived Probabilistic pruning filter method. Using spatial and probabilistic pruning obtained a efficient PRNN algorithm which later extended to PRkNN queries.

1.3. Top-k Query

The efficient Top-k query returns k results that can best meet the user's query conditions, which has attracted more and more attention from researchers. Top-k processing in uncertain databases is semantically and computationally different from traditional top-k processing. The query ranking problem of uncertain data is to first determine a specific semantic ranking rule, then form the corresponding algorithm according to this semantic rule, and finally use this algorithm to conduct Top-k query ranking of uncertain data so as to get the corresponding Top-k ranking result. Attribute value and probability of tuples are two basic attributes of uncertain data, and also the basis for generating Top-k ranking semantic rules. The balance between score value and probability has always been the focus of Top-k query of uncertain data.

K. Yi et al.**[4]** introduce a new polynomial algorithm to deal with top-k queries in uncertain databases under the commonly used X-relation model. They take the X-relational model and add a score attribute to sort tuples according to the internal structure of the model. They also provide solutions for the U-Topk and u-krank queries, both of which are significantly faster and use much less space under the X-relational model.

Andrei Neculai et al **[22]** uses MSCOCO dataset and proposed a probabilistic model that uses a composite set of k queries in arbitrary modalities. The approach model each embedding as a multivariate Gaussian probability density function (PDF), with the aim of composing different Gaussian PDFs according to a parametric probabilistic rule, i.e. the sum of k Gaussian PDFs, to create different embedding compositions. By modality the author means testing the model for multimodal image retrieval on the basis of various configurations that combine several images and (or) text queries. In order to retrieve images, probabilistic embeddings are created for each image in the database. Then, cosine similarity is used to match composite queries with all images that meet the compositional probabilistic embeddings' mean c and mean t values. The top-k retrieved images are the images with the highest similarity scores in the database.

The study also compares the performance of non-probabilistic methods with probabilistic methods and it is noted that the performance for non probabilistic methods degrade to almost random guess when there is increase in number of queries whereas the probabilistic models are able to retain a relatively reasonable performance and able to encode richer semantics and interactions between different semantics concepts, which strengthen its capability of composing a flexible amount of queries for image retrieval.

1.4. Probabilistic threshold query (PTQ)

Probabilistic threshold query (PTQ) is a widely used query method in uncertain databases. It returns all objects that satisfy the query with a probability higher than the threshold.

Cheng et al. **[5]** proposed two structures to index the uncertain data of PTQ. The first index, called PTI, adds uncertain information to internal nodes so that more search paths can be pruned when accessing the index. This redesigned index structure makes full use of probabilistic uncertainty information during index search. This structure, called probabilistic threshold index (PTI), is based on a modification of a one-dimensional R-tree, where probabilistic information is augmented to its internal nodes to facilitate pruning. This index forms the basis for a broader scheme in which intervals with similar variance values are clustered together. Cheng et al. also showed that queries with a uniform pdf uncertainty interval can be answered at the optimal time given a fixed probability threshold, and laid a theoretical foundation for the problem.

Reynold Cheng et al. **[33]** talks about the probabilistic queries called Probabilistic Threshold Query, which requires answers to have probabilities larger than a certain threshold value. It is defined to manage the uncertainty in the sensor database as sensors are used to monitor changing entities like location of moving objects and temperature. And these readings are recorded in a centralized database system. But sometimes queries can produce incorrect results by using the old values from the database. So, a framework representing uncertainty of sensor data has been defined.

Shi, Yaqing, et al. **[42]** talk about probabilistic range query to determine data query and uncertain data query respectively.

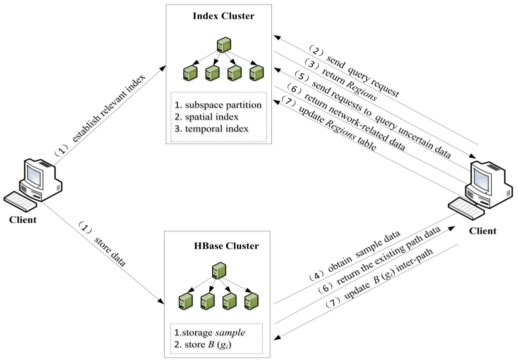
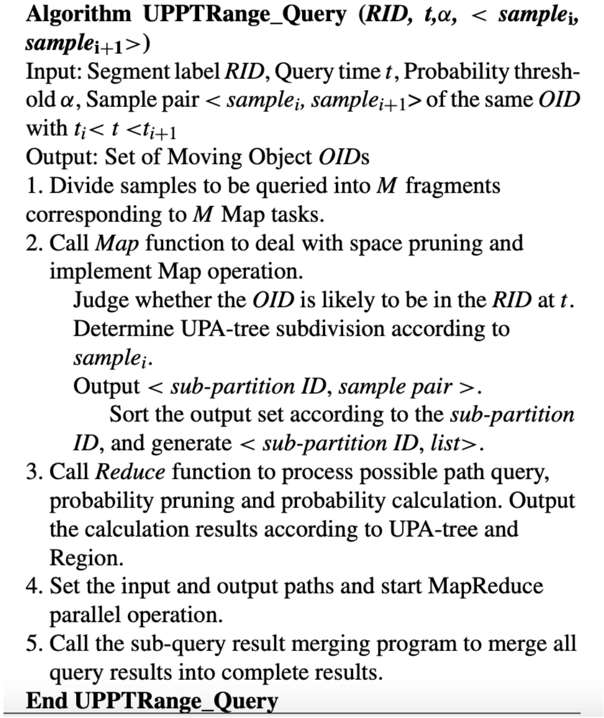


Fig 3: Spatio-temporal method

Fig (3)A traditional Spatio-temporal method is directly adopted to get the leaf node from a UPA tree.



```
Algorithm UPPTRange_Query (RID, t,α, < sample_i, sample_{i+1}>)
Input: Segment label RID, Query time t, Probability threshold α, Sample pair < sample_i, sample_{i+1}> of the same OID with t_i < t < t_{i+1}
Output: Set of Moving Object OIDs
1. Divide samples to be queried into M fragments corresponding to M Map tasks.
2. Call Map function to deal with space pruning and implement Map operation.
     Judge whether the OID is likely to be in the RID at t.
     Determine UPA-tree subdivision according to sample_i.
     Output < sub-partition ID, sample pair >.
        Sort the output set according to the sub-partition ID, and generate < sub-partition ID, list>.
3. Call Reduce function to process possible path query, probability pruning and probability calculation. Output the calculation results according to UPA-tree and Region.
4. Set the input and output paths and start MapReduce parallel operation.
5. Call the sub-query result merging program to merge all query results into complete results.
End UPPTRange_Query
```

1.5. Nearest Neighbor Query

1.5.1. Aggregate NN Query (ANN）

As a basic query operation, aggregated nearest neighbor (ANN) query is one of the most useful operators for analyzing network and graph data, such as social network, traffic network and biological network. The aggregate distance between the retrieved data entity and the given query data entity (such as sum and max) is smaller than the aggregate distance of other data entities in the database.

Z,Liu et al. **[6]** proposed Aggregate Nearest neighbor query (UG-ANN) for uncertain graphs, and on this basis proposed two pruning methods, namely structure pruning and instance pruning. Structure pruning uses UG-ANN candidate filtering in the pruning process to derive upper and lower bounds of the set distance by using the monotonicity of the set distance to reduce the size of the graph. Instance pruning reduces the number of instances to check in the search tree.

1.5.2. KNN Query (K-NN）

Konstantinos A. Tsintotas et al **[29]** proposed an approach to identify loop closing image pairs using nearest neighbor voting schemes. The probabilistic score produced by pdf is used to recognize the proper location or feature detection. The study models the approach in two steps (1) Building the BoTW database and (2) querying the database. The first part contains keypoint extraction, point tracking, guided feature detection, Tracking Words (TW) generation, and BoTW (bag of tracking works), while the second part contains voting method, probabilistic belief generator, and geometrical check for final decision and validation. To build the BoTW Database first, point tracking across consecutive images is achieved by a set of SURF points from a previous image fed into a KLT point tracker alongwith currently perceived camera measurement along with description vector matched with local points for accurate tracking. Descriptor matching is performed across various frames and to generate distinct TWs for map representation. To avoid the tendency of Tracked Points (TP = {tp1 , tp2 ,... tpv })to drift along the trajectory, a guided-feature-detection mechanism is used and a K-NN search is performed on the points coordinate space between A k-NN (k = 1) search is performed on the points' coordinate space between the TPt elements and the ones extracted by SURF (SPt). For each tracked point tpt, the nearest spN N t (spN N t ∈ SPt) is detected and evaluated by measuring the 2 distance between its descriptor dN N t and the one (dt−1) corresponding to spt−1 in the previous image It−1. A point is accepted and its descriptor is considered to be a good match, providing that the following conditions are satisfied (a) The Euclidean distance between tpt and its corresponding spN N t is lower than α. l_2 (tpt,spN N t) < α and (b) the descriptors' absolute difference is lower than β: l_1 (dt−1 − dt) < β.. The merging of the descriptors to create the TWs is the last step of the BoTW database method. When a particular point is no longer tracked, its total length τ (measured in subsequent frames) indicates if a new word needs to be established (τ>ρ). Finally a binomial pdf method is used to examine the rareness of the event. Any identification that takes place within a small circle surrounding the query location is regarded as a correct match, whereas false-positive detections occur outside of this circle. False-negative detections are the places that the approach should have picked up on but didn't. A distance that enables the precise estimation of the fundamental matrix using RANSAC corresponds to the tolerance utilized for the evaluation, which is set to 10 neighboring locations.Obtaining the maximum recall score for perfect precision is the objective of a loop closure algorithm.

The paper Hybrid Potential based Probabilistic Roadmap (HPPRM): path planning algorithm proposed by Ankit A Ravankar et al **[24]** addresses three main problems that are experienced widely during path planning. I.e. Feasibility, Optimality and Speed. The study combines local and global navigation methods for robot navigation in a static and dynamic environment. The HPPRM works a multi query sampling algorithm in two phases i.e. construction and query phase.

The construction phase of the roadmap is constructed in free configuration space where R=(N,E) is an undirected graph of sets of nodes N and a set of edges E. Here an edge corresponds to a simple, feasible path connected by a line segment between two nodes. The potential map is created for a given map based on the obstacle information, decomposing the potential map into equal grids in x, y directions. Each grid area is classified into two areas, and the total number of nodes to be placed in each region is decided, define the median value of the total potential as a global threshold and finally find out the nodes distributed in each region. The study explains when the number of nodes are the same, HPPRM's success rate improves in every scenario. The fact that nodes are dispersed equally throughout both maps using the map segmentation method may be the cause of HPPRM's increased success rate. The path planning failed because the start and goal configurations in PRM could not be connected by edges since not enough nodes were dispersed in the narrow regions. Also, when the nodes are the same, HPPRM's mean calculation time(s) is slightly longer than PRM's. However, HPPRM requires a longer average calculation time to achieve the success rate. Due to the increased cost during the query phase, HPPRM's mean calculation time (s) is longer than PRM's. In line with HPPRM model on path planning Felipe Felix Arias et al **[27]** proposes a concept to leverage a neural network to generate Avoidance critical Probabilistic Roadmap (ACPRM) that contain motion structures which enables efficient obstacle avoidance, reduce search and planning space and increase a roadmaps reusability and coverage. The study uses a PRM scheme that uniformly samples configurations in the environment and connects each sample to its kp nearest neighbors which may or may not be part of the learning model. The neural network is trained to identify regions important to dynamic environment navigation from local environment features. ACPRM uses the trained neural network and works on multi query settings to construct sparse probabilistic roadmaps with sufficient structure to support multi-agent motion planning in environments with narrow passages.

1.5.3. Group Nearest Neighbors query

Group nearest neighbor query has been used in many fields like clustering, device management, outlier detection etc. Probabilistic nearest neighbor query on uncertain data becomes popular and important in many applications.

[37] Weige Wang et al. stated an example saying, a forest which is on fire so to determine the exact area of fire is not possible and not precise because of the changing of wind direction or moving speed of the fire, and it can be modeled as an uncertain object. Several firefighters located at different places want to efficiently put out fires together. In this case, firefighters may issue a PGNN query to find a site that minimizes their total distance or minimum the time for them to reach the site.

[37] Weige Wang et al. proposes the Voronoi diagram and how it is used to solve the problem of probabilistic group nearest neighbor query on uncertain data. It is stated that by constructing the Voronoi diagram of uncertain objects and the convex hull of the set of the query points, a candidate set of target objects is obtained. Then, the space pruning algorithm reduces the candidate set. And then the probability value of each target object in the candidate set has been calculated. Finally, the uncertain objects whose probabilities are greater than or equal to the user-specified threshold are put into the result set.

1.6 Probabilistic Spatial/Similarity Join Query

The similarity join is a crucial database basis for frequently used feature databases. It creates a new set from two datasets that are combined based on a similarity predicate and contains objects from both of the original sets in pairs. Distances between objects must be calculated based on unclear and imprecise data in many different application domains, such as sensor databases, location-based services, or face recognition systems.

Kento Sugiura et al [25] in the paper proposes an efficient calculation method by using a determination finite automation (DFA) and to calculate occurrence probability efficiently the approach divides a window into chunks and reuse the previous calculation results. The study explains the probability of time series events within a window by using the possible world semantics. For study, first a probabilistic event stream is defined in terms of an infinite sequence of probabilistic events and then Query pattern and matches. The approach is compared using Naive method and DFA based method. The DFA based method slices a window into small chunks with chunk length l, and then retains the product of transition matrices in each chunk.

For experiment, the dataset used comprises the indoor location event streams gathered as part of the RFID ecosystem study. The locations of tracked objects are recorded in this collection as nodes in a graph, and they are approximated using RFID sensing data every second i.e. the location is represented as probabilistic distribution at each time step.

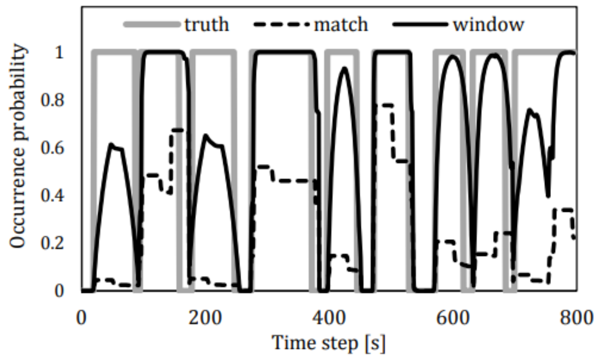


Fig 4: Estimated occurrence probabilities with p_1 and window size =30

Fig 4. A gray line represents truth probabilities; when the line reaches the top, a time series event takes place in a window. The outcome of the current techniques is a dotted line. The time series occurrences, such the initial room entry/exit event around [20: 60] are most likely overlooked, because the odds of matching are so low.The approach proposed in this study (i.e., a black solid line) may recognize time series events with high probabilities as opposed to match-based detection.

Another approach proposed by Jianmin Li el al **[28]** is a study on diversified routing (DR) queries to discover the most convenient routes in a dynamic road network. For a given a dynamic transportation network, a source, a destination, a journey time threshold (τt) a threshold for the probability of congestion (τp), and a threshold for the total cost of travel (τg), The DR query determines: 1) the route with the shortest travel time, whose global travel cost does not exceed τg; 2) the route with the shortest probability of congestion, whose global travel cost does not exceed τg; and 3) the route with the shortest global travel cost, whose travel time does not exceed τt, and whose probability of congestion does not exceed τp. Here the first two queries are known as "global-travel-cost threshold DR queries" (GTCT-DR), while the third query can be referred to as "time-and-probability threshold DR queries" (TPTDR).

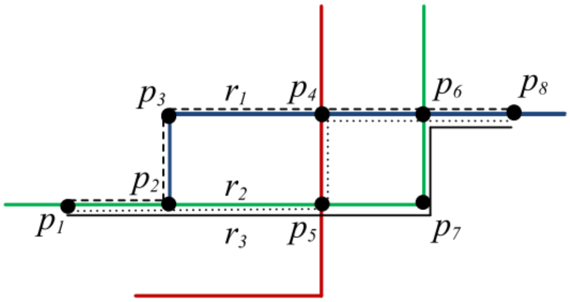
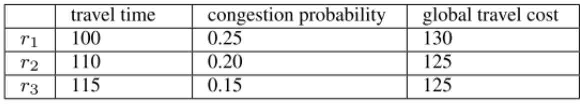


Fig 5: DR queries approach



	travel time	congestion probability	global travel cost
r_1	100	0.25	130
r_2	110	0.20	125
r_3	115	0.15	125

Table 1: Attributes of three routes

In fig (5) where p_1 is the source and p_8 is the destination, and routes r_1 =(p_1, p_2, p_3, p_4, p_6, p_8),, r_2 = (p_1, p_2, p_5, p_4, p,p_8), and r_3 =(p_1, p_2, p_5, p_7, p_6, p_8) . Given a global travel cost threshold of 128, the GTCT-DR queries return route r_2 because it has the shortest trip time and r_3 because it has the lowest probability of congestion. Table 1, Route r_3 is returned for the TPTDR query with a travel time threshold of 118 and a congestion probability of 0.18

because it has the lowest overall travel cost. To reduce the size of the search space and increase the effectiveness of the queries, several optimization approaches are developed. A smaller value of the global travel cost threshold, τ.g, results in a stronger pruning effect and a smaller search space, which is good for narrowing down the search space.

1.7 Probabilistic parking queries (PPQ)

The navigation systems compute the cost-optimal way to a given destination. But the systems only give the estimated time to the destination and sometimes it is possible that the drivers may not find the parking spaces at the destination after reaching there. It is often not possible to directly park the car at the destination of a route. It is defined that finding a free parking spot is difficult for the drivers. The navigation system can only tell the efficient way to reach the destination but not about if the parking is available or not at that location. However, recent advances in car sensoring techniques and vehicle ad-hoc networks allow the construction of real-time maps of currently unoccupied parking spots.

[40] Gregor Jossé et al. talks about the problem of guiding a driver to an unoccupied parking spot. And for that purpose, probabilistic parking queries (PPQ) which receive the current position of a driver, his destination and a set of currently unoccupied parking spots has been introduced. PPQ returns a route along the set of parking spots, maximizing the likelihood of finding a parking spot in the vicinity of the target location. A PPQ operates on the setting of a driver reaching the vicinity of his destination and wanting to find a parking spot within a given walking distance.

1.8 Spatio-range queries

[16]*Trajcevski, Goce et al* talk about evaluating the probability values of the answer to spatio-range queries, when the trajectory has the uncertainity and it can be processed using ORDBMS. the 2D motion of the data management concept can be used for 3D motions as well. Fig 6, The operator for querying the MOD (Moving objects database), the operator returns the quantitative probabilistic values**.**

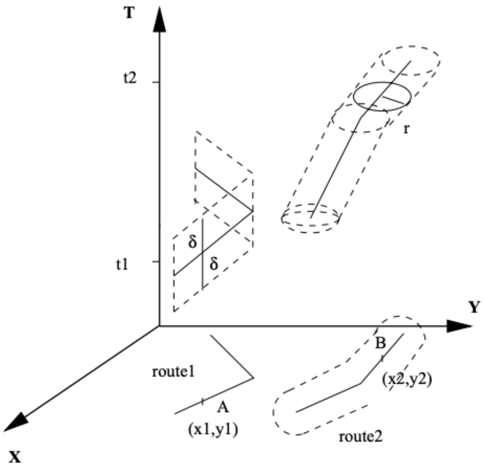


Fig 6: Routing model

Section 2: Probabilistic Model and Applications

2. 1. Probabilistic XML Model

In many applications, including data integration and web information extraction, data uncertainty naturally exists. One of the concepts given forth to model and handle various types of uncertain data is probabilistic XML. A probabilistic XML document is basically a compact representation of a probability distribution over conventional XML documents. In the information retrieval field Jaekwang Kim et al [21] proposes an approach in the paper to rearrange search results in accordance with a query's context when the query suddenly attracts interest from the public. A search engine retrieves a document set in response to a given query. The query is then checked to see if it is in a burst state or not. If so, context words are retrieved from microblogs and used to reevaluate the documents that were retrieved. The document is ranked based on the probabilistic relationship between the query and the document. As given in fig(7) the approach for the application if a query is given, the proposed method first assesses if it is now in a popular state, or a burst state, by looking at the search volume of the query. If so, the system extracts context-relevant terms for the query from microblogs and then the retrieved result is ranked and re-ranked based on context terms. The sample queries of Google and Twitter are randomly chosen and this whole moving pattern of searching volumes and microblogging volumes is compared with the temporal pattern of the volumes of postings in Twitter and searching volume in Google to calculate the correlation coefficient for similarity measurement. The correlation coefficient ranged from -1 to 1, showing an exact positive linear relationship when the value equals 1 and an exact negative relationship when the value equals -1.

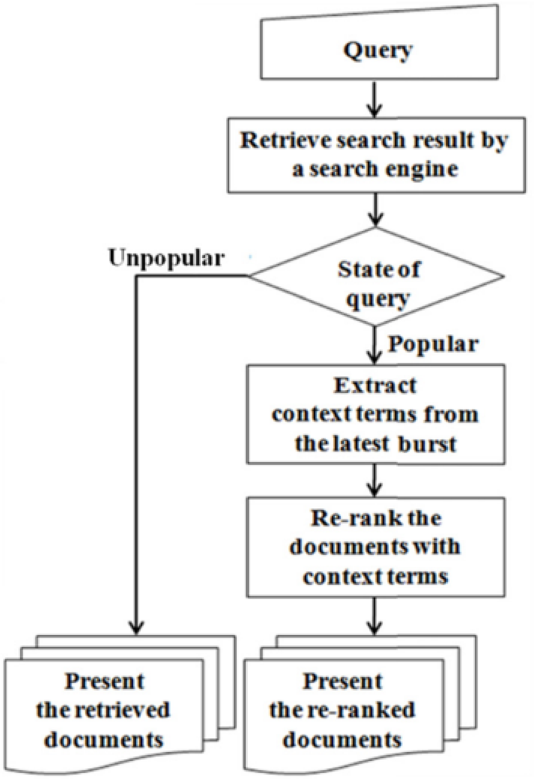

Query
Retrieve search result by a search engine
State of query
Unpopular
Popular
Extract context terms from the latest burst
Re-rank the documents with context terms
Present the retrieved documents
Present the re-ranked documents

Fig7: approach for information retrieval

For analysis the author has defined burst and volume to set the required threshold for assessment. Queries are selected at various threshold levels in order to compare the coefficient value and see how bursts in search volumes affect the relationship.

With the increase in threshold of burst there is an increase in proportion of queries that have strong correlation. The documents are then converted into a term vector to find out how much content of each document in the search result is close to the context vector. The experiment is further performed with three types of similarity measures i.e. Jaccard, Cosine and Weighted Cosine to estimate the degree of similarity. If the document has high degree of similarity to the context vector it means the document has many words in the context vector and their f among total term frequencies are similar to context vector, hence the document becomes the highest rank in terms of query related context.

2.2. Probabilistic Graph Model

When it comes to encoding joint (multivariate) distributions over a large number of random variables that interact with one another, probabilistic graphical models (PGMs) provide a comprehensive framework. These representations draw on ideas from machine learning, graph algorithms, probability theory, and other fields as they lie at the convergence of statistics and computer science. Alastair R. Ruddle et al [26] in the paper uses a probabilistic graph model for eliminating the likelihood of electromagnetic disturbances causing the system malfunctions with various degrees of severity. The analysis is carried out using Bayesian Network which is a graphical structure made up of a network of nodes that represent model variables and edges that show the causal connections between connected pairs of nodes.The probability of impacts at a specific degree of severity for the system function can be determined by taking into account the functional relationship between components. A functional dependency between data transmission components C2 and C1 and its associated conditional probability (CPD). The process used to find the values of CPD table P (C2 | C1) for node C2 is shown using an event tree **Fig (8)**. It is noted that only the specific event P(C2 = low | C1 = high) is never seen out of the 6263 simulated noise samples that were initially taken into account for calculating the CPD of EMI impact P(C2 | C1).

[36] Sujoy Chatterjee et al. proposed a probabilistic graphical model for crowd group decision making problems. It states that the group decision considers a feedback set comprising a range of continuous values so this problem has been addressed here with a probabilistic approach taking into account the annotator accuracy, annotator bias and question difficulty.

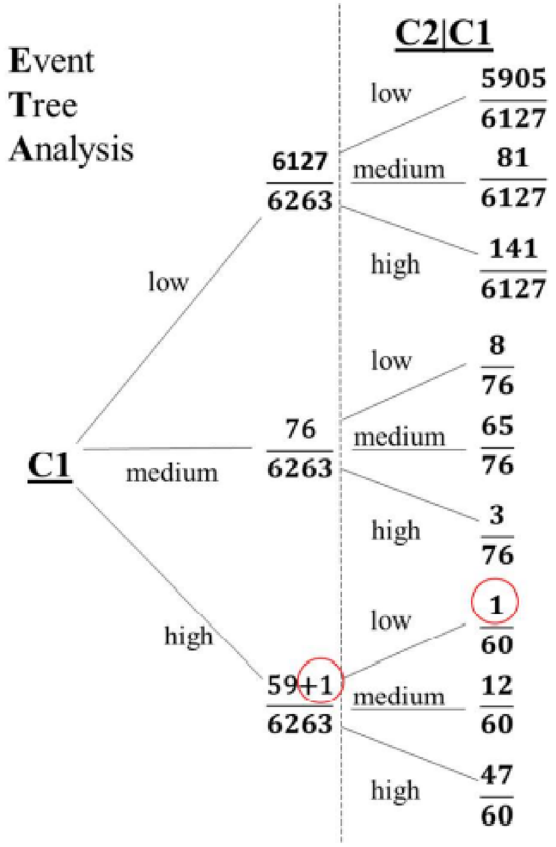


Fig (8) event tree to illustrate the condition probability values

Due to the effect it would have on associated probability values, assigning zero probability to a rare but probable event only because it wasn't seen in a sample research would be inappropriate. The sample size needed to observe this event cannot be calculated in advance, and its probability can only be expressed as <1/6262. It is assumed that the event would be detected if noise sample 6263 were to be generated (additional sample indicated in the red circles in **Fig. (8)**, in order to avoid the uncertainty of attributing a zero-probability value to a potential occurrence. Due to the unidirectional nature of fault trees and event trees, it would be necessary to use different tree structures for a single system to analyze common cause failures with bidirectional functional relationships across components. Hence by using PGM bidirectional functional interdependence can be modeled in a single system model. Also PGM makes the more efficient and less time consuming.

Graph probabilistic dependencies (GPDs) were introduced by Muhammad Sadiq et al. [23] to solve the problem of uncertainty over large-scale graphs. Real-world graphs have missing, inconsistent, and default values in addition to being noisy. As a result, it is now very uncertain if certain entities in these graphs even exist. By identifying both negative and positive patterns, these graph patterns identify redundant information and discrepancies. The underlying uncertainty in graphs can be effectively handled by the probabilistic techniques. In order to specify the integrity and semantics of the semi-structured, schema-free graphs, graph dependencies set formal constraints on the topological structure of the graphs. To specify data quality and consistency, these dependencies must be found before matching over the graph. Graph dependencies operate in three stages: (i) specifying the types of patterns over the topological structure of the graph; (ii) finding these patterns on the labeled graph; and (iii) matching these patterns in the test graph to detect inconsistencies and errors.

A graph is defined in this study as a group of vertices and edges. This graph's vertices can be divided up into c separate sets, allowing for labeled edges to exist between the vertices of various sets. By minimizing duplications and inconsistencies, this study proposes to integrate multiple graphs and enhance data quality over the integrated graph. The proposed research defines the following graph pattern to account for the uncertainty in GDs:
$(Q[u], X\alpha\ \phi \longrightarrow P\ Y\ \beta\ \phi\ )$

It is difficult to calculate the probabilities for graph patterns over vast graphs. It necessitates the NP-complete problem of mining negative and positive graph patterns across numerous graphs. To solve this problem, we combine several graphs by locating and merging duplicate entities, and we enhance the quality of the data on the combined graph through effective GPD discovery. Three graphs with various numbers of vertices and edges are taken into consideration for integration in this experiment. Over these graphs, a MusicBrainz dataset is constructed to determine the likelihood of each pattern. Seventy-five percent (75%) of the dataset's instances have missing values. The dataset has a significant percentage of missing values, making it a noisy dataset.

Except for the entity id vertex, each graph has five other sorts of vertices: title, year, artist, album, and name. These five vertices represent items such as albums and artists on two of them, while labels are represented on the other three. Disambiguating the entities in these graphs is necessary for their integration. In the study, several patterns are extracted, their probabilities are computed across the individual graphs, and then these probabilities are aggregated over many graphs. According to a Pattern γL1 in the study, albums created by the same artist correspond to the same actual thing (such an album), hence they are all duplicates. The probability is computed over each unique graph first, and then the final probability is aggregated across the many graphs, in order to verify the accuracy of this graph pattern. By determining the satisfiability and violation of this graph pattern over each individual graph, the probability of this pattern is calculated. The dataset is initially converted into a multi-typed graph by treating each instance of the data as a musical entity and each specific characteristic within that instance as the musical entity's direct neighbor vertex. Vertices from several data sources that have the same values are combined to generate a single vertex.

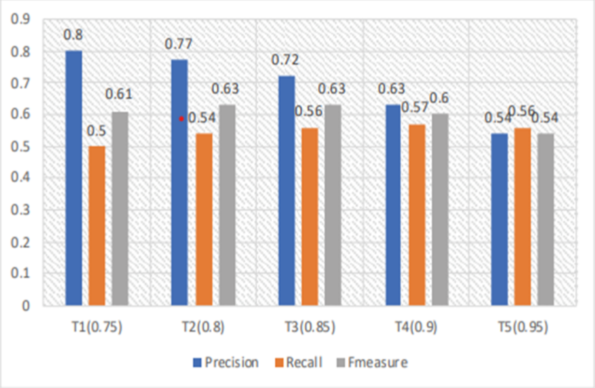


Fig 9: Shows the precision recall, and f-measure values achieved by proposed approach with varying thresholds for detecting duplicates over the multiple data sources for integration

After the data has been transformed into a multi-typed graph, the identified GPDs γL3, γA2 and for the album and artist vertices, respectively, are utilized to retrieve the missing values. Finally, the

GPD γ4 is used to identify duplicate listings of entities on the graph. Fig (9)With different criteria of 0.75, 0.85, 0.90, and 0.95, the precision, recall, and F-measure of the found GPDs are calculated for each sample. For instance, it obtained an average of 80% precision, 50% recall, and 61% F-measure on threshold 0.75. Overall, compared to all other thresholds, the performance of GPDs for duplication identification on threshold 75% was the highest.

Andreas Ritter et al [30] proposes a graph construction algorithm for vehicular application that runs in real time. The proposed approach comprises a pose estimator, which is realized as an extended Kalman filter, and three interchangeable, scenario-specific modes for creating graphs. The study suggests making predictions using (2) Gaussian process regression and (1) frequentist prediction intervals.

The graph's vertices are spaced about equidistant, at a distance of five meters, to allow for a reasonable resolution of the predictions. The created algorithm makes sure that each value recording of the predictive signals is saved in the appropriate vertex that corresponds to the location of the vehicle. These values are then properly merged on subsequent travels along the same route so that the predictions accurately reflect the individual recordings. The stream of pose estimates produced by a Kalman filter are first transformed into a series of equidistant samples to produce a graph with evenly spaced vertices. Each sample is then individually added to the graph via the map construction.

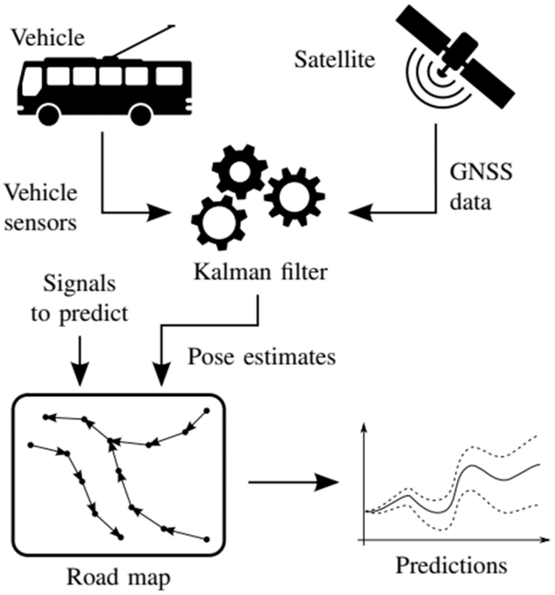


Fig 10: Proposed Learning approach

The merging technique is based on perpendicular projections and alters the map vertex locations similarly to the segment alignment described in the study, so long as the location estimations clearly show that the vehicle is traveling on a known road. If the location estimates are uncertain, the decision is delayed and the location estimates are buffered. The vertex sequence candidates are searched after the estimations are close enough to a known path, and the merging approach is based on dynamic temporal warping. Fig. (10) depicts the concept behind the suggested unsupervised learning approach to get probabilistic predictions of the upcoming driving scenario. It is especially helpful for the map creation method if the individual pose estimates of the vehicle follow smooth trajectories. In actuality, the suggested method struggles with jittery signals that do not accurately represent natural motions, despite being able to accommodate lateral deviations of the trajectories. So, using a discrete-time extended Kalman filter, we combine GPS data and odometry measurements (EKF).

The proposed method makes use of the motor shaft, four-wheel speed measurements, and GPS sensor location estimates. The GPS signals are often noisy and inaccurate at low speeds, and the wheel speed sensors only produce useful signals beyond 2 km/h, so the filter only takes into account measurement data over that speed threshold. Below this speed, only dead-reckoning using the shaft speed data is used to update the state estimate.

Fig 11, The two corresponding modes of the proposed construction algorithm are tracking for driving on a mapped road and discovering for driving on an unknown road. In the latter mode, the method ties the newly added equidistant pose estimates to the previously visited vertices in the graph by inserting them one at a time as new vertices. The approach combines the vertices of the current graph with the resampled posture estimates if the mode is tracking.

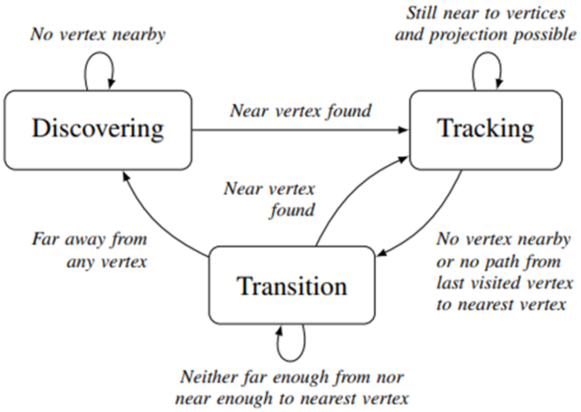

No vertex nearby
Still near to vertices and projection possible
Discovering
Near vertex found
Tracking
Near vertex found
Far away from any vertex
No vertex nearby or no path from last visited vertex to nearest vertex
Transition
Neither far enough from nor near enough to nearest vertex

Fig 11: Approach for Algorithm

However, in order to strengthen the robustness of our algorithm, the study includes a third mode that detects circumstances in which the vehicle seems to have deviated from a mapped route but the data is insufficient. Only the tracking mode can be used to access this transition mode. Until the distance to any known route is either short enough to go back to the tracking mode or long enough to switch to the discovering mode, the pose estimates are buffered. Fig. shows the three modes and potential transitions.

When a new prediction is needed, the directed graph is traversed till the predetermined horizon, either starting in tracking mode from the most recently visited vertex or in transition mode from the vertex closest to the current pose. The current implementation ensures that no loops are produced by following the branch that was last visited if a vertex has many successors. The output of this query includes sequences defining the stop probability, mean velocity, and variance of the velocity, as well as a distance vector with a monotonically increasing value representing the total Euclidean distance between the traversed vertices.The study proposes prediction strategies based on spatially dependent point estimates of the sample mean and variance of the data gathered, which enable significantly less data to be collected while preserving the necessary information. Additionally, both approaches take into account how many times the vehicle has traveled a particular route, providing a logical explanation for the width of the prediction interval.

2.3. Database Model

Wolfson, Ouri, et al. [13] talk about a model based on a traditional DBMS system.

The DOMINO (Database for moving object) approach. In this approach, an envelope is created on top of the DBMS layer.

Dynamic attributes:- When moving objects it's current it not just updates its current position but also updates its future position. For example, if DBMS has a heuristic of speed and route of the object it can calculate the future locations. These papers propose a novel model called MOST (Moving objects Spatial-temporal model). The dynamic attributes in moving objects change less frequently than the locations of objects. Using the attributes we database is updated with future locations for e.g we can know the highway and speed (north 361, 55mil/hr).The mechanism is to add dynamic attributes to the existing data model and add capabilities to the existing data model to deal with dynamic attributes.

2.4. Probabilistic duplicate detection system

The most prominent problems in data quality today is the existence of duplicate records which leads to confusion and storage wastage in the database. The data cleaning systems available currently usually produce one cleaning instance of the input data by choosing the parameters of duplicate detection algorithms.

G. Beskales et al. [32] defines a system named ProbClean which is used to detect the duplicate records in the database. Fig 12 ProbClean supports relational queries and allows new types of queries against a set of possible repairs efficiently. This system is a probabilistic ETL tool that focuses on a specific data transformation task, which is detecting and eliminating duplicate records. Duplicate records are defined as records in the unclean database that refer to the same real world entity. Below is the chart for the Probabilistic ETL approach used in the ProbClean system to eliminate duplicate records from the database.

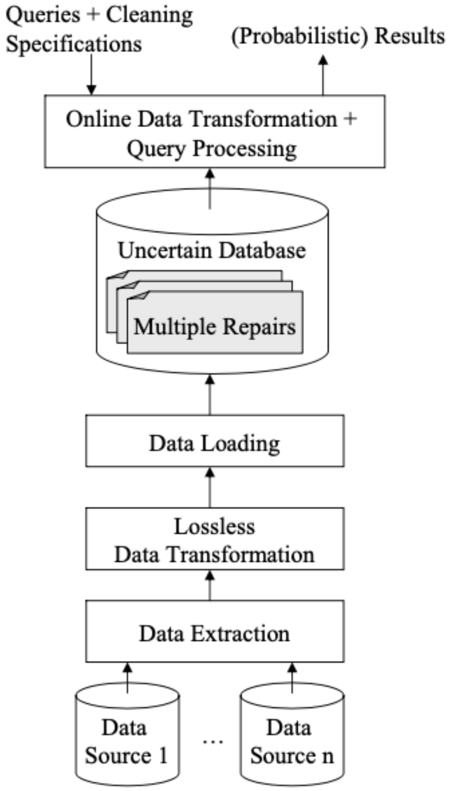


Fig 12: Data Transformation Query processing approach

Section 3: Challenge and optimization of uncertain data query

3.1 Challenge

Due to the existence of uncertainty, a fundamental challenge is how to deal with the probability of uncertain data and probabilistic data. For example, for some data, a range query returns points that fall within a given range. On uncertain and probabilistic data, a tuple or an uncertain object may fall into a given range with a probability.

In some queries on uncertain and probabilistic data, we are only concerned with the uncertainty within a single object or tuple involved in a single-generation rule. We say this kind of query involves local uncertainty. On the other hand, when answering some queries on uncertain and probabilistic data, we may want to consider global uncertainty, that is, the uncertainty of object/tuple combinations as answers in possible worlds.

Due to the large number of query tasks brought by large uncertain databases, we need to explore the tradeoff between accuracy and computational cost in order to solve the computational challenge of querying uncertain and probabilistic data. Sampling-based methods and randomization algorithms are particularly interesting because they can provide quality assurance of good approximate answers for expensive queries while, at the same time, remaining polynomials in computational cost.

3.2 Optimization

T Chen et al. [7] proposed an optimal dynamic programming scheme and a more efficient (in terms of time and space complexity) greedy algorithm to calculate the execution plan for executing queries to save the computational cost between them. R Cheng et al. [8] mainly studied an important class of join, namely probabilistic threshold join, which avoids semantic complexity when dealing with uncertain data. For this type of join, R Cheng et al. developed three sets of optimization techniques: item-level, page-level, and index-level pruning. These techniques facilitate pruning with little space and time overhead and are easily adapted to most join algorithms.

X. Ding et al. [9] introduced two effective pruning methods: spatial pruning and probabilistic pruning to speed up the query process by reducing the search space. Spatial pruning is based on the boundary region of the KTH nearest neighbor, and probabilistic pruning is based on the upper and lower probability boundaries of each k-NN candidate after spatial pruning, which successfully improves the query efficiency and effectiveness. Lee et al. [10] proposed and developed an iterative algorithm to solve the optimal solution under the condition that the theoretical probability bound of the optimal solution was true optimal. And the robust optimization is achieved by focusing on the "worst case" of uncertain data. Because worst-case realizations can often be obtained by clever reformulations that do not consider all possible cases, the resulting optimization problems are often easier to solve than the stochastic ones. The most critical assumption of robust optimization is that uncertainty can be represented as an "uncertainty set" that includes all possible realizations of the uncertain data.

Conclusion

This survey discusses an extensive literature review on probabilistic query types, probabilistic query models and probabilistic database models. We have discussed the types of query processing techniques and methods proposed by researchers for different data types and databases. In this study we have discussed the models that focus on query processing, query answering, implementation level and support ranking functions. We discussed several approaches and algorithms to illustrate the data management problem, challenges and optimizations methods of uncertain data query. Through probabilistic data model study, we discusssed in probabilistic databases, a large amount of data defines a simple probabilistic space, but a complex model which is generated by the query, makes the probabilistic inference difficult. In statistical relational models, the probabilistic model is typically stated using a much shorter first-order relational representation and these models are typically a huge graphical model, like a Markov network or a Bayesian network.

The key areas that are learned through this survey study are the techniques of dealing with data uncertainty in both data and queries is a challenging task, the concept of rank awareness needs to be leveraged as it provides the clarity for model cost, learning more about ways for ranking functions implementation and data privacy and restriction for privacy preserving before ideating the probabilistic model.

References:

[1] *X. Li, J. Liu, K. Ren, X. Li, X. Ren and K. Deng, "Parallel k-Dominant Skyline Queries over Uncertain Data Streams with Capability Index," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019,* *pp. 1556-1563, doi: 10.1109/HPCC/SmartCity/DSS.2019.00214.*

***[2]** C. -C. Lai, H. -Y. Lin and C. -M. Liu, "Highly Efficient Indexing Scheme for k-Dominant Skyline Processing over Uncertain Data Streams," 2021 30th Wireless and Optical Communications Conference (WOCC), 2021, pp. 97-101, doi: 10.1109/WOCC53213.2021.9603141.*

***[3]**C. Guo, S. Su, K. -K. R. Choo, P. Tian and X. Tang, "A Provably Secure and Efficient Range Query Scheme for Outsourced Encrypted Uncertain Data From Cloud-Based Internet of Things Systems," in IEEE Internet of Things Journal, vol. 9, no. 3, pp. 1848-1860, 1 Feb.1, 2022, doi: 10.1109/JIOT.2021.3088296.*

***[4]** K. Yi, F. Li, G. Kollios and D. Srivastava, "Efficient Processing of Top-k Queries in Uncertain Databases with x-Relations," in IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 12, pp. 1669-1682, Dec. 2008, doi: 10.1109/TKDE.2008.90.*

***[5]** Reynold Cheng, Yuni Xia, Sunil Prabhakar, Rahul Shah, and Jeffrey Scott Vitter. 2004. Efficient indexing methods for probabilistic threshold queries over uncertain data. In Proceedings of the Thirtieth international conference on Very large data bases - Volume 30 (VLDB '04). VLDB Endowment, 876–887.*

***[6]** Liu, Z., Wang, C. & Wang, J. Aggregate nearest neighbor queries in uncertain graphs. World Wide Web 17, 161–188 (2014). https://doi.org/10.1007/s11280-012-0200-6*

***[7]** T. Chen, L. Chen, M. T. Özsu and N. Xiao, "Optimizing Multi-Top-k Queries over Uncertain Data Streams," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp. 1814-1829, Aug. 2013, doi: 10.1109/TKDE.2012.126.*

***[8]** R Cheng, Sarvjeet Singh, Sunil Prabhakar, Rahul Shah, Jeffrey Scott Vitter, and Yuni Xia. 2006. Efficient join processing over uncertain data. In Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06). Association for Computing Machinery, New York, NY, USA, 738–747. https://doi.org/10.1145/1183614.1183719*

***[9]** X. Ding, S. Sheng, J. Liu and P. Zhou, "Efficient Probabilistic K-NN Computation in Uncertain Sensor Networks," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 3, pp. 2575-2587, 1 July-Sept. 2021, doi: 10.1109/TNSE.2021.3099864.*

***[10]** Lee, Chungmok, 2022. "A robust optimization approach with probe-able uncertainty," European Journal of Operational Research, Elsevier, vol. 296(1), pages 218-239.*

***[11]** Wolfson, Ouri, et al. "Cost and imprecision in modeling the position of moving objects." Proceedings 14th International Conference on Data Engineering. IEEE, 1998.*

***[12]** Shi, Yaqing, et al. "A probabilistic range query of moving objects in road network." IEEE Access 7 (2019): 40165-40174.*

***[13]** Wolfson, Ouri, et al. "DOMINO: Databases for moving objects tracking." ACM SIGMOD Record 28.2 (1999): 547-549.*

***[14]**Cheng, Reynold, et al. "Efficient join processing over uncertain data." Proceedings of the 15th ACM international conference on*

Information and knowledge management. 2006.

***[15]**Agarwal, Pankaj K., et al. "Nearest-neighbor searching under uncertainty II." ACM Transactions on Algorithms (TALG) 13.1 (2016): 1-25.*

***[16]**Trajcevski, Goce. "Probabilistic range queries in moving objects databases with uncertainty." Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access. 2003.*

***[17]**Cheng, Reynold, Dmitri V. Kalashnikov, and Sunil Prabhakar. "Querying imprecise data in moving object environments." IEEE Transactions on Knowledge and Data Engineering 16.9 (2004): 1112-1127.*

***[18]**Wolfson, Ouri, et al. "Updating and querying databases that track mobile units." Distributed and parallel databases 7.3 (1999): 257-387.*

***[19]**Bernecker, Thomas, et al. "Efficient probabilistic reverse nearest neighbor query processing on uncertain data." Proceedings of the VLDB Endowment 4.10 (2011): 669-680.*

***[20]**Li, Yiping, Jianwen Chen, and Ling Feng. "Dealing with uncertainty: A survey of theories and practices." IEEE Transactions on Knowledge and Data Engineering 25.11 (2012): 2463-2482.*

***[21]**J. Kim, "A Document Ranking Method With Query-Related Web Context," in IEEE Access, vol. 7, pp. 150168-150174, 2019, doi: 10.1109/ACCESS.2019.2947166.*

***[22]** A. Neculai, Y. Chen and Z. Akata, "Probabilistic Compositional Embeddings for Multimodal Image Retrieval," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 4546-4556, doi: 10.1109/CVPRW56347.2022.00501.*

***[23]**M. S. H. Zada, B. Yuan, A. Anjum, M. A. Azad, W. A. Khan and S. Reiff-Marganiec, "Large-scale Data Integration Using Graph Probabilistic Dependencies (GPDs)," 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), 2020, pp. 27-36, doi: 10.1109/BDCAT50828.2020.00028.*

***[24]**A. A. Ravankar, A. Ravankar, T. Emaru and Y. Kobayashi, "HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots," in IEEE Access, vol. 8, pp. 221743-221766, 2020, doi: 10.1109/ACCESS.2020.3043333.*

***[25]**K. Sugiura and Y. Ishikawa, "Regular Expression Pattern Matching with Sliding Windows cover Probabilistic Event Streams," 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 2019, pp. 1-8, doi: 10.1109/BIGCOMP.2019.8679331.*

***[26]**L. Devaraj, A. R. Ruddle and A. P. Duffy, "EMI Risk Estimation for System-Level Functions Using Probabilistic Graphical Models," 2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium, 2021, pp. 851-856, doi: 10.1109/EMC/SI/PI/EMCEurope52599.2021.9559291.*

***[27]**F. F. Arias, B. Ichter, A. Faust and N. M. Amato, "Avoidance Critical Probabilistic Roadmaps for Motion Planning in Dynamic Environments," 2021 IEEE International Conference on Robotics and Automation*

(ICRA), 2021, pp. 10264-10270, doi: 10.1109/ICRA48506.2021.9560974.

[28]*J. Li, Y. Zhong and S. Zhu, "Diversified Routing Queries in Dynamic Road Networks," in IEEE Access, vol. 7, pp. 25452-25458, 2019, doi: 10.1109/ACCESS.2019.2893411.*

[29]*K. A. Tsintotas, L. Bampis and A. Gasteratos, "Probabilistic Appearance-Based Place Recognition Through Bag of Tracked Words," in IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1737-1744, April 2019, doi: 10.1109/LRA.2019.2897151.*

[30]*A. Ritter, F. Widmer, J. W. Niam, P. Elbert and C. Onder, "Real-Time Graph Construction Algorithm for Probabilistic Predictions in Vehicular Applications," in IEEE Transactions on Vehicular Technology, vol. 70, no. 6, pp. 5483-5498, June 2021, doi: 10.1109/TVT.2021.3077063.*

[31] *Shichang Fu, Yihong Dong and Maoshun He, "Continuous probabilistic skyline queries for uncertain moving objects," 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), 2010, pp. 396-399, doi: 10.1109/CAR.2010.5456816.*

[32] *G. Beskales, M. A. Soliman, I. F. Ilyas, S. Ben-David and Y. Kim, "ProbClean: A probabilistic duplicate detection system," 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), 2010, pp. 1193-1196, doi: 10.1109/ICDE.2010.5447744.*

[33] *Reynold Cheng and Sunil Prabhakar. 2003. Managing uncertainty in the sensor database. SIGMOD Rec. 32, 4 (December 2003), 41–46. https://doi.org/10.1145/959060.959068*

[34] *Xiaofeng Zhou, Yang Chen, and Daisy Zhe Wang. 2016. ArchimedesOne: query processing over probabilistic knowledge bases. Proc. VLDB Endow. 9, 13 (September 2016), 1461–1464. https://doi.org/10.14778/3007263.3007284*

[35] *Jian Pei, Ming Hua, Yufei Tao, and Xuemin Lin. 2008. Query answering techniques on uncertain and probabilistic data. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08). Association for Computing Machinery, New York, NY, USA, 1357–1364. https://doi.org/10.1145/1376616.1376774*

[36] *Sujoy Chatterjee and Malay Bhattacharyya. 2017. A Probabilistic Approach to Group Decision Making. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17). Association for Computing Machinery, New York, NY, USA, 2445–2451. https://doi.org/10.1145/3027063.3053226*

[37] *Weige Wang, Jian Xu, Ming Xu, Ning Zheng, and Enquan Ge. 2015. Probabilistic Group Nearest Neighbors query based on Voronoi diagram. In Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics (UrbanGIS'15). Association for Computing Machinery, New York, NY, USA, 36–41. https://doi.org/10.1145/2835022.2835029*

[38] *Mohammad Asghari, Tobias Emrich, Ugur Demiryurek, Cyrus Shahabi 2015, Probabilistic estimation of link travel times in dynamic road networks ACM SIGSPATIAL November 03-06, 2015, Bellevue, WA, USA*

[39] *Y. Shi, S. Huang, J. Feng and J. Lu, "A Probabilistic Range Query of Moving Objects in Road Network," in IEEE Access, vol. 7, pp. 40165-40174, 2019, doi: 10.1109/ACCESS.2019.2907108.*

[40] *Gregor Jossé, Matthias Schubert, and Hans-Peter Kriegel. 2013. Probabilistic parking queries using aging functions. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13). Association for Computing Machinery, New York, NY, USA, 452–455. https://doi.org/10.1145/2525314.2525458*

[41]*Nat'l Research Council, Risk Analysis and Uncertainty in Flood Damage Reduction Studies. Nat'l Academy Press, 2000.*