IJCTA, 7(2) December 2014, pp. 201-214

An Efficient Probabilistic Random B-tree Indexing Technique for Fast Retrieval of Multidimensional Features in Uncertain Data

M. Sathish* and S. Sugumaran**

ABSTRACT

The ability to store and retrieve multidimensional features from uncertain data is highly required in the recent data mining applications. Different indexing techniques are used for mining multi-dimensional feature from uncertain data. Though optimal indexing structure deals with multi-dimensional data object in an efficient manner it requires more time for extracting specific feature of multi-dimensional data. Few other uncertain data mining algorithms worked on multi-dimensional data using standard deviation and density function. However dimensional feature has to be defined properly. Initially, Random B-tree Indexing is designed that is a fast indexing technique which supports update operations on cluster data objects generated from uncertain data. With this, the Random B-tree fits on processing with different cluster simultaneously. The updating process is carried out using two metrics, addition and removal of information on the cluster object search space. Secondly, probabilistic multi-dimensional feature extraction is formularized to 'achieve precise information from the random B-tree indexed clustered objects. Finally to process the user query, adaptive pruning method is extended in RBI-PMF to reduce the memory usage. RBI-PMF is evaluated with varied performance factors indexing speed, quality of multidimensional feature and I/O memory usage.

Keywords: Random B-Tree Indexing, Adaptive Pruning Method.

1. INTRODUCTION

Querying multi-dimensional data space has emerged a significant problem in data mining process. Indexing is an effective technique for querying multi-dimensional space. The indexing scheme answers to the user queries and produces better search result rate on uncertain data. The multi-dimensional database system which is uses storing and monitoring uncertain information. Due to limited resources, some of the existing indexing systems were not feasible at every point of time. Indexing a set of data points for answering range queries with minimal processing time is one of the biggest issues to be handled in the proposed work.

Fig. 1 shows indexing of user query to the stored multidimensional data points. The indexing procedure is clearly explained through climatic condition elated user queries. In particular, since climatic condition database information is a continuously changing entity, the system only receives and works with old samples of uncertain data. The situation is not being helped by the fact that data may not arrive at the system on time, and may even get lost during clustering due to mining network problems.

Dimensional space holds up by outsourced servers and consumes lower storage costs for large databases. However, care taken to safeguard the valuable data requirements besides the indexing based query access. You Jung Kim *et al.*, discussed K-nearest-neighbor query handle network spatial queries using general

* Assistant Professor, Department of Computer Applications, Sree Amman Arts Science College, *E-mail: msatnet@rediffmail.com*

** Associate Professor, Department of Computer Science, Erode Arts and Science College, Erode-638 009, Tamilnadu, India, *E-mail: prof_sukumar@yahoo.co*

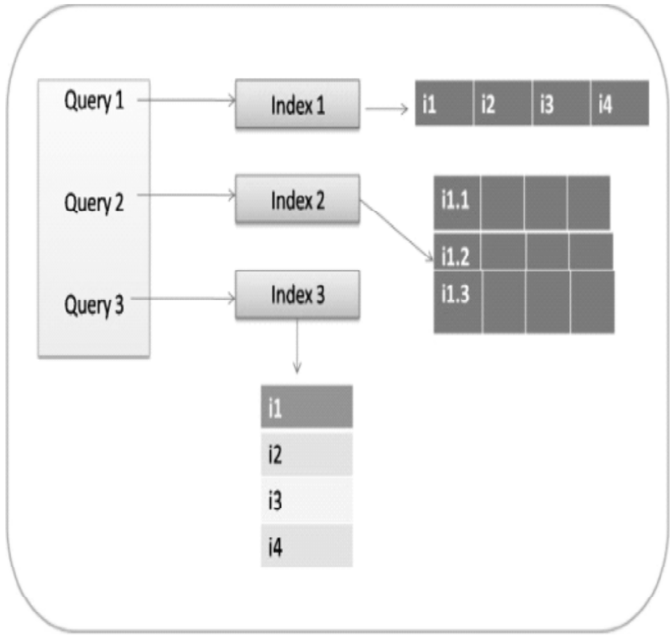

Figure 1: Query Indexing on Multi-Dimensional Data

framework but multiple data owner queries were not handled with higher probabilistic results [12]. Lei Zou., and Lei Chen, *et al.*, described as Pareto-Based Dominant Graph (DG) improved the probabilistic query efficiency results. DG built offline to communicate dominant relationship between records and top-k query. DG algorithm is openly related to cardinality of skyline points in record set but DG indexing is not effective on dominant relationship analysis [13].

Yuzhe Tang, *et al.*, discussed as LIGhtweight Hash Tree (LIGHT) is a very efficient dominant relationship analysis query with low-maintenance indexing scheme. LIGHT employs an effective identification mechanism and tree summarization strategy for polished distribution of its index structure. LIGHT during analysis hold up various complex queries with near optimal performance and also reveal generic DHT indexing schemes [14]. Luýs Leitao., *et al.*, described Existing XML duplicate detection using Bayesian structure determined the important increase result over un-optimized version of algorithm. Probabilistic Duplicate Detection algorithm for hierarchical data called XML Dup believed on equal terms the likeness of attribute contents and the relative importance but did not address multi-dimensional indexing [3].

Sharadh Ramaswamy, et al., described Multi-Cluster Feature Selection (MCFS) addressed the multi dimensional indexing to avoid unlawful access with L1-regularized models. MCFS offered an efficient spatial filtering with a moderately small preprocessing storage overhead [7]. MCFS need to tighten the work by optimizing clustering algorithm and cluster distance bound for balancing load. Marco Muselli., and Enrico Ferrari, *et al.*, discussed in Logical Analysis of Data (LAD) and Shadow Clustering (SC) were the techniques developed for retrieving more relevant logical products by balancing the load. The kernel followed the LAD consisting of a breadth-first enumeration for entire prime implicate based clustering. SC adopted a heuristic process for retrieving most capable resulting AND-OR expression but the computational complexity was not reduced [10].

Eric Hsueh-Chan Lu, *et al.*, presented Cluster-based Temporal Mobile Sequential Pattern Mine (CTMSP-Mine) provide effectual scheme to reduce computational complexity on user's subsequent behaviors. Location Based Service Alignment (LBS-Alignment) offered the similarity measure on user behavior prediction. In CTMSP-Mine, user clusters are built up by Cluster-Object-based Smart Cluster Affinity Search Technique (CO-Smart-CAST), but failed to integrate with Hybrid Prediction Model. The failure of hybrid prediction model in CTMSP-Mine, improved the false positive ratio [15].

The structure of this paper is as follows. In Section 2, demonstrates the related works and Section 3, an overall view of the Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality Features is presented. Section 4 and 5 outline experiment results with parametric factors and present the result graph. Finally, Section 6 concludes the work with better result outcome.

2. RELATED WORK

Research works on uncertain data have observed an increasing in database community. Ying Zhang., Wenjie Zhang *et al.*, discussed in U-Quadtree systematizes uncertain objects in multi-dimensional space such that the range searching is rejoined proficiently by relating filtering techniques. Particularly, range search on uncertain data devised a cost model which considered range searching operation. U-Quadtree organizes the objects, but fails in tackling the correlation among multi-dimensional uncertain objects [1]. Zhisheng Li *et al.*, presented as Efficient Index for Geographic Document Search in enhancing IR-tree index with top-k document search algorithm [9].

Costas Panagiotakis, *et al.*, described an index-based global voting method permitting to represent trajectory with different smooth continuous descriptor function. Method for trajectory segmentation and sampling is based on representativeness of (sub) trajectories if MOD fails to hold each sub-trajectory clustering operation. Still, there is a need to work for considering each sample set with cluster representative [5]. An effective bi-clustering technique is briefed by S Roy et al., ,is capable of detecting positively co-regulated genes. Bi-clustering uses pattern based approach to compare between genes and identify all possible bi-clusters using non-greedy method with minimal polynomial time [6].

Gene selection recognizes optimal subset of relevant genes and performs clustering operation is the foremost step. Sari Kontunen-Soppela *et al.*, discussed K-means clustering and Sammon's mappings is used to detect parallel gene expression patterns within sampling times and treatments. Gene expression was caused by O3 with the combination of CO2. O3 induced as a suspicious reaction to oxidative pressure and previous leaf nodes were analyzed to decrease the exterior of photosynthesis and carbon fixation-related genes. Davies–Bould in validity index increased expression of senescence associated genes [11]. Hans Bindera, *et al.*, presented transcriptional regulation model was associated to the product of connecting network times. Genomic regulation estimate single peaked distribution methods of expression value improves clustering efficiency rate according to power laws [4].

In many practical applications, existing work with simplifying assumptions took uncertain data to perform operation. One such simplified assumption is that the occurrence and nonexistence of different tuples on uncertain data are probabilistically independent. Charu C Aggarwal, *et al.*, described as existing Uncertain Data Algorithms on multi-dimensional data used derived standard deviation of density function. Uncertain frequent pattern mining algorithm did not process queries with effective pruning tricks on multi-dimensional data [2].

Ali Shahbazi., *et al.*, proposed as Extended Sub Tree (EST) similarity function as described compared tree structured data by defining a set of mapping rules in sub trees rather than nodes. EST with new sub tree mapping preserved the structure of trees and enhanced edit base mappings. However the tree distance function was incapable of quality maintenance, which in turn increased processing time [8].

In the proposed work, focus is made on indexing multi-dimensional cluster data objects using Random B-Tree Indexing based on Probabilistic Multi-Dimensionality Features (RBI-PMF). Random B-Tree Indexing is supported in the proposed system, where the overall scanning is carried out in the single pass, thereby reducing the processing time. The indexing carried out in the B-tree structure improves speed of the user query result retrieval by processing different cluster simultaneously. The updating process in RBI-PMF performs addition and removal of information from clustered data objects. Then updated result is used on indexing user query using adaptive pruning method.

3. RANDOM B-TREE INDEXING BASED ON PROBABILISTIC MULTIDIMENSIONALITY FEATURES(RBI-PMF)

The uncertain data works with cluster data objects in RBI-PMF attain minimal processing time. The Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality (RBI-PMF) Features captures all elements and perform ordered list with minimal processing time. The tuple elements on cluster objects help capturing the properties of dynamic attributes for performing indexing operation. The Random B-tree indexing operation supports update operations using multi-dimensional clustered data objects as input vector. The indexing technique in RBI-PMF is represented in Fig. 2.

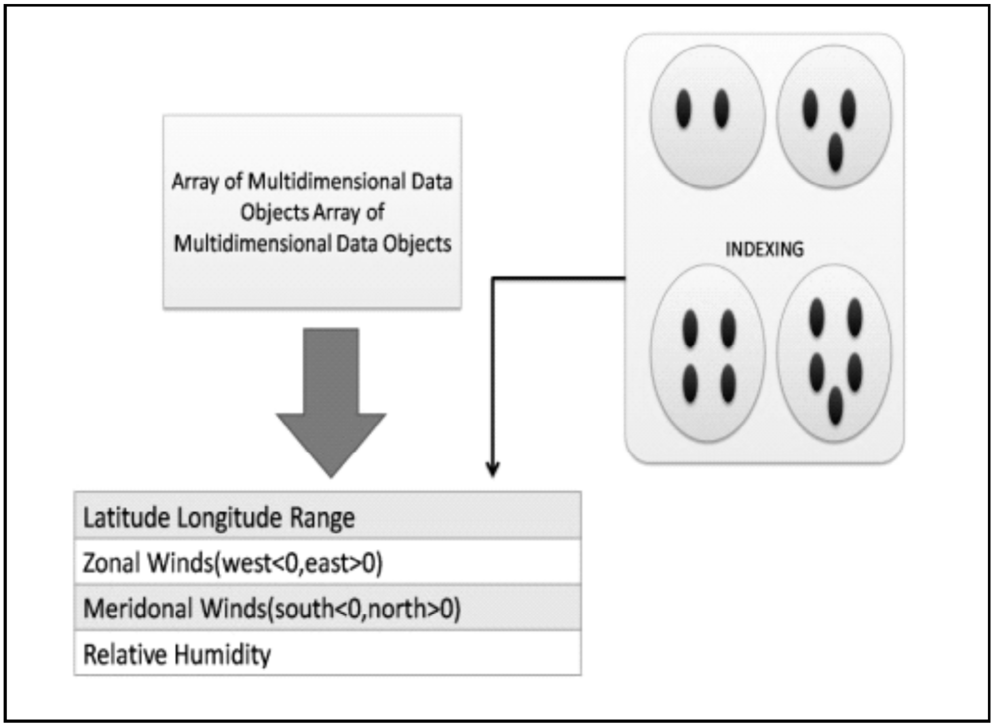

Figure 2: Random B-tree Indexing Technique

The multidimensional clustered data object contains attributes to perform an effective indexing technique. For instance, fig. 2 is briefly explained with El Nino Data Set metrological readings to perform operation using Random B-tree indexing technique. El Nino Data Set takes all the attributes and performs clustering of the objects. Similar type of data objects is clustered for effective processing and simultaneous processing achieves faster retrieval rate. The store clustered is used on processing user query and produce improved speed of data object retrieval.

Fig. 3 describes the pruning procedure using El Nino Data Set. The adaptive pruning includes a means for adaptively filtering uncertain data objects to speed up indexing process. The RBI-PMF process is adaptive, where small rejection is carried out based on specified threshold points. The adaptive pruning removes rejects the unwanted multi-dimensional data objects and fetches the result of user query based on indexing. The Architecture Diagram of RBI-PMF is illustrated in

Fig. 4 multi-dimensional data objects are used for performing effective indexing operation using RBI-PMF. The primary focus of RBI-PMF is to process the user query on multi-dimensional data and to capture the accurate result using minimal input and output memory usage. Random B-tree Indexing uses the clustered data objects generated from uncertain data for performing indexing. Random B-tree Indexing also performs updating operation using add and remove operation. The adding and removing of data objects is carried out using random B-tree indexing and produces precise information.

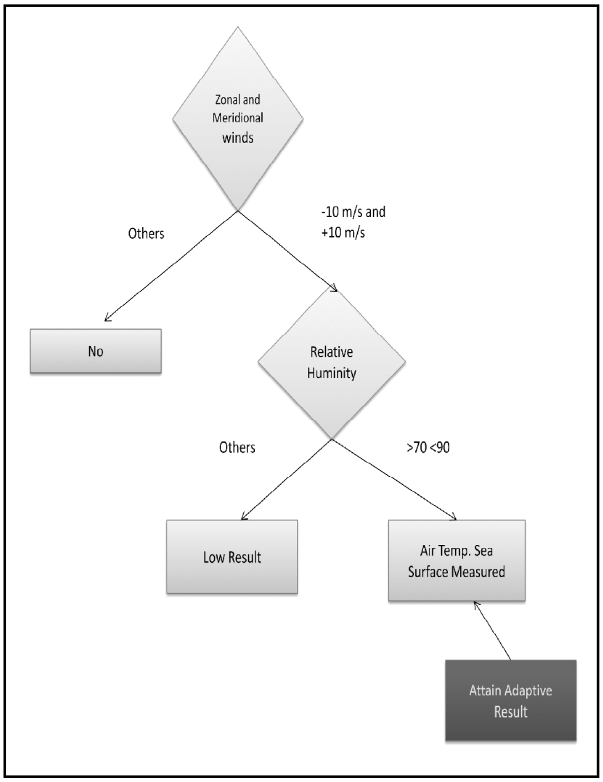

Figure 3: Adaptive Pruning Procedure

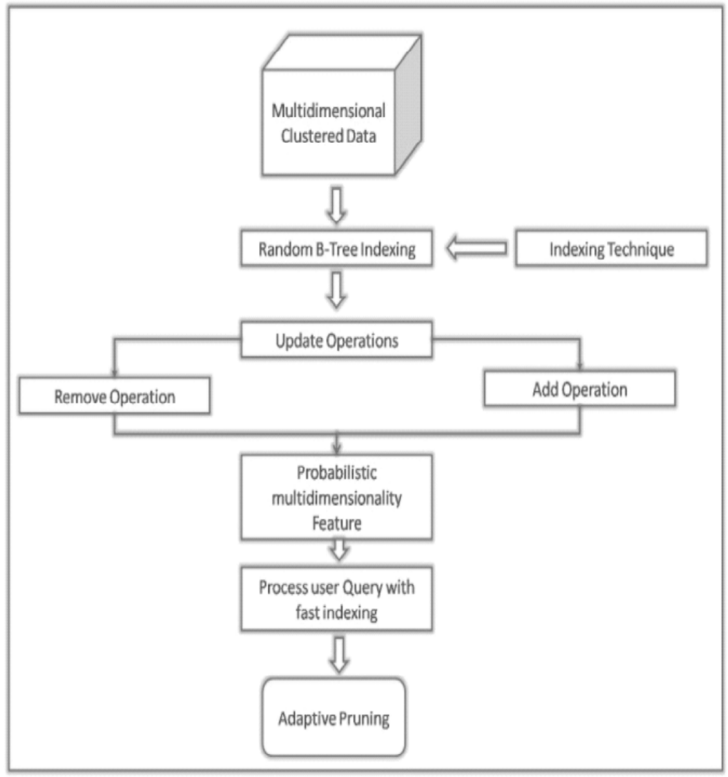

Figure 4: Architecture Diagram of RBI-PMF

The probabilistic multi-dimensionality features perform processing of user query to fetch effective information from clustered data objects. The user query extends RBI-PMF with adaptive pruning method. The adaptive pruning selectively index the random B-tree and selects according to user query model from root points dynamically. The section given below briefly explains RBI-PMF processing in each section. Section 2.1 described about random B-tree indexing, section 2.2 explains about probabilistic multi-dimensionality feature on cluster objects. Final section 2.3 describes about adaptive pruning method.

3.1. Random B-tree Indexing

Random B-tree contains tree structure to sort the attributes and permit effective searching operation. The searching process is carried out using self-balancing the attribute values in RBI-PMF. Random B-tree is optimized where clustered data objects are indexed to n-dimensional blocks. RBI-PMF significantly improves indexing operation on primary set, where by allows the users to manipulate indexes and gain higher flexibility rate from uncertain data. The main advantage of Random B-tree indexing is that the nodes on the overall tree are linked, so that the overall cluster object is scanned with linear pass. The scanning through linear pass reduces the processing time performance.

Random based B-tree index frequently access random nodes that lie closer to the root for performing indexing. Random based index reduces execution time and perform indexing more quickly on cluster data objects generated from uncertain data. The random B-tree indexing technique builds root, internal and leaf points. The degree of the elements is analyzed and performs ordered list in RBI-PMF. The ordered list performs addition and removal operation, where addition is performed by initially searching down the tree of multi-dimensional points. The removal operation is performed when data objects are unrelated to indexing structure. The adding up and removal procedure on clustered data objects generated from uncertain data is described as,

// Adding operation
Begin
Step 1: If the dimensional space is not full, then add the data objects
Step 2: Allocate new leaf structure, and then add the newly clustered data objects
$Adding\ operation = MD[add\ new(d1, d2, d3..)$
Step 3: If the root point splits, then the root node has one key and two intermediate nodes
Step 4 :Repeat step until found no iteration
// Removal operation
Step 5 :Root node on B-tree where it belongs
Step 6 :Remove the unrelated leaf structure as
$Removal\ operation = MD[remove\ old(d1, d2, d3..dn)$
Step 7: If the leaf node removed, then the leaf node has two intermediate nodes
Step 8: Repeat step until found no iteration
End

If the clustered data objects are not placed on dimensional space, then objects are added into the relevant field. Similarly, the unrelated data objects are removed from dimensional space using removal operations. RBI-PMF transfers the block after searching of user information and attain precise results.

3.2. Probabilistic Multi-Dimensionality Features

Formally, a clustered data object is associated with 'i' probabilistic Multi-dimensionality (PMF) performing query processing to achieve precise result. Probabilistic Multi-dimensionality Features [Multi pdf(i)] in RBI-PMF consists of 'i' in the clustered data objects on multi-dimensional data set. The RBI-PMF also holds probabilistic range query to effectively answer every query user's request. The probabilistic appearance on multi-dimensionality features is formularized as,

$$MP(i,q) = \int_{multi\ pdf\ (i=1)}^{n} i.pdf(x).dx$$

Where, *multi pdf* denotes the intersection of user query result value with multi-dimensional space value. The intersected result value helps on improving feature extraction from random B-tree indexed clustered objects. The 'MP' is multi-dimensionality features computed on 'i' clustered data objects.

3.3. Adaptive Pruning Method

Let us assume a set of user range query 'Q' to search the result from multi-dimensional data space. The query series of Random B-tree node is analyzed and performs processing operation in RBI-PMF. Starting from the lower bound of the multi-dimensional space, RBI-PMF follows adjacent links to perform pruning process. The pruning process still removes unwanted objects and extracts only the user requested query result in RBI-PMF. The searching process is carried out as,

Begin
Step 1: N dimensional Clustered Data objects generated from uncertain data
Step 2: Perform search based on use query on 'N'
Step 3: While N=N.right and Nlow<i do
Step 4: Produce result to the searched user query
End

The first optimized result using Random B-tree indexing creates search with lower bound of the tree and extract the features accordingly. The result is uniformly distributed among the users, thereby reducing the input output memory usage. Random B-tree indexing achieves relative adaptivity on pruning, thereby improving robustness result on extracting the features. Random B-tree indexing achieves the key idea by producing user query search result. The unwanted values on multi-dimensional cluster data space are removed, thereby reducing the memory capacity rate on storing input and output information.

4. EXPERIMENTAL EVALUATION

Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality Features is implemented in JAVA. The dataset from UCI repository taken for experiment evaluation are given below.

- Diabetes Dataset
- El Nino Dataset

Diabetes dataset contain the patient records obtained from different source points. Diabetes Dataset contain 20 attributes in the categorical and integer form. The regular device had an internal clock to timestamp events, whereas records contain the reasonable time slots such as breakfast, lunch, dinner, bedtime. Diabetes files consist of four fields per record such as MM-DD-YYYY format, Time format, Code and value.

El Nino Data Set contains oceanographic and surface meteorological readings taken from a series of uncertain buoys positioned throughout the equatorial pacific with 178080 instances. El Nino Data Set contains 12 attribute set including date, latitude, longitude, zonal winds (west<0, east>0), meridional winds (south<0, north>0), relative humidity, air temperature, sea surface temperature and subsurface temperatures down to a depth of 500 meters. The multidimensional data are taken from various rainfall millimeter, solar radiation value, current levels, and subsurface temperatures.

Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality (RBI-PMF) Feature is compared against the existing novel U-Quadtree indexing mechanism and uncertain frequent pattern mining algorithm. The experiment is conducted on factors such as CPU processing time, indexing speed, quality of multidimensional feature, I/O memory usage, precision rate, and false query probabilistic rate.

Indexing Speed

The indexing level defines rate (speed) at which indexing work is carried out in multi-dimensional clustered data space. The indexing level is computed to identify and improve percentage level of efficiency.

The indexing speed is used as a major factor on measuring efficiency rate. Different set of features are taken for indexing speed measurement. The memory usage defines the amount of memory consumed to perform indexing operation. The input, processing and output results are stored in memory for result providence on user query.

Precision rate

Precision rate is defined as the amount of percentage which clearly measures correct answer indexed to the user query. Precision rate is always measured in terms of fractional percentage count. The larger the count of objects produces averagely the higher precision result rate. The accurate indexing of result improves the quality of multi-dimensional feature. The quality of multi-dimensional feature level is based on indexing speed. The higher rate of indexing produces increased quality level on RBI-PMF. The probability of false occurrence (i.e.,) wrong query result to users is termed as false positive probabilistic rate. In addition I/O memory usage, quality of multi-dimensional feature and query probabilistic rate metrics are taken for experimental evaluation.

5. RESULT ANALYSIS

To better perceive the efficacy of the proposed Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality (RBI-PMF) Feature, substantial experimental results are tabulated. The RBI-PMF is compared against the existing novel U-Quadtree indexing mechanism [1] and uncertain frequent pattern mining algorithm [2].

Table 1
Tabulation of Indexing Speed

Feature Count	*Indexing Speed (Kbps)*		
	U-Quadtree Indexing Mechanism	*FPM Algorithm*	*RBI-PMF*
5	148	162	185
10	290	350	391
15	455	503	542
20	584	655	714
25	734	807	962
30	870	950	1063
35	1026	1155	1263

Java is used to experiment the factors and analyze the measures of the result percentage with the help of table and graph values. Results are presented to different number of features considering the oceanographic and surface meteorological readings. The results reported here confirm that with the increase in the number of features, the quality of the multi-dimensional features also increases.

Table 1 and Fig. 5 illustrate the indexing speed in U-Quadtree Indexing Mechanism [1], FPM Algorithm [2] and proposed RBI-PMF.

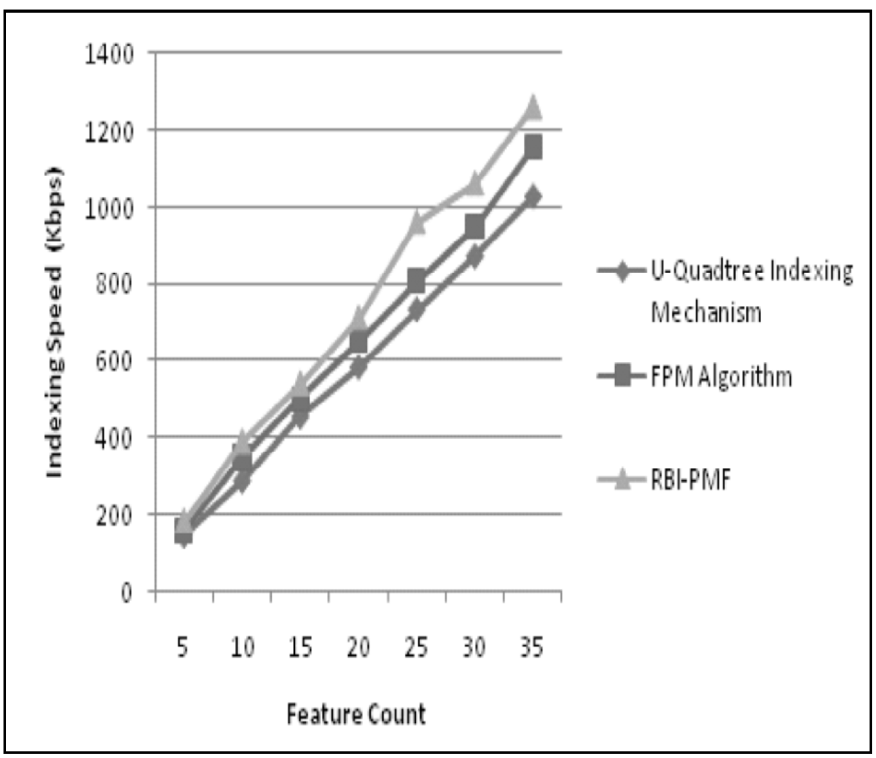

Figure 5: Measure of Indexing Speed

The tuple elements on the cluster objects help creating the properties of dynamic attributes for performing indexing operation. The store clustered is used on processing the user query and to produce the improved speed of indexing. The indexing speed is increased to 14 – 35% in RBI-PMF when compared with the U-Quadtree Indexing Mechanism [1].

Feature count for experimental work ranges from 5 to 35 ranges. The multi-dimensional feature is used on computing over the indexing speed range. Similar type of data objects is clustered for effective processing and simultaneous processing achieves faster indexing. The indexing rate is increased to 7 – 19 % in RBI-PMF when compared with FPM Algorithm.

Table 2
Tabulation of I/O Memory Usage

I/O Access	*I/O Memory Usage (MB)*		
	U-Quadtree Indexing Mechanism	*FPM Algorithm*	*RBI-PMF*
50	93	88	81
100	115	109	98
150	139	121	111
200	137	128	115
250	153	146	135
300	175	161	151
350	222	203	189

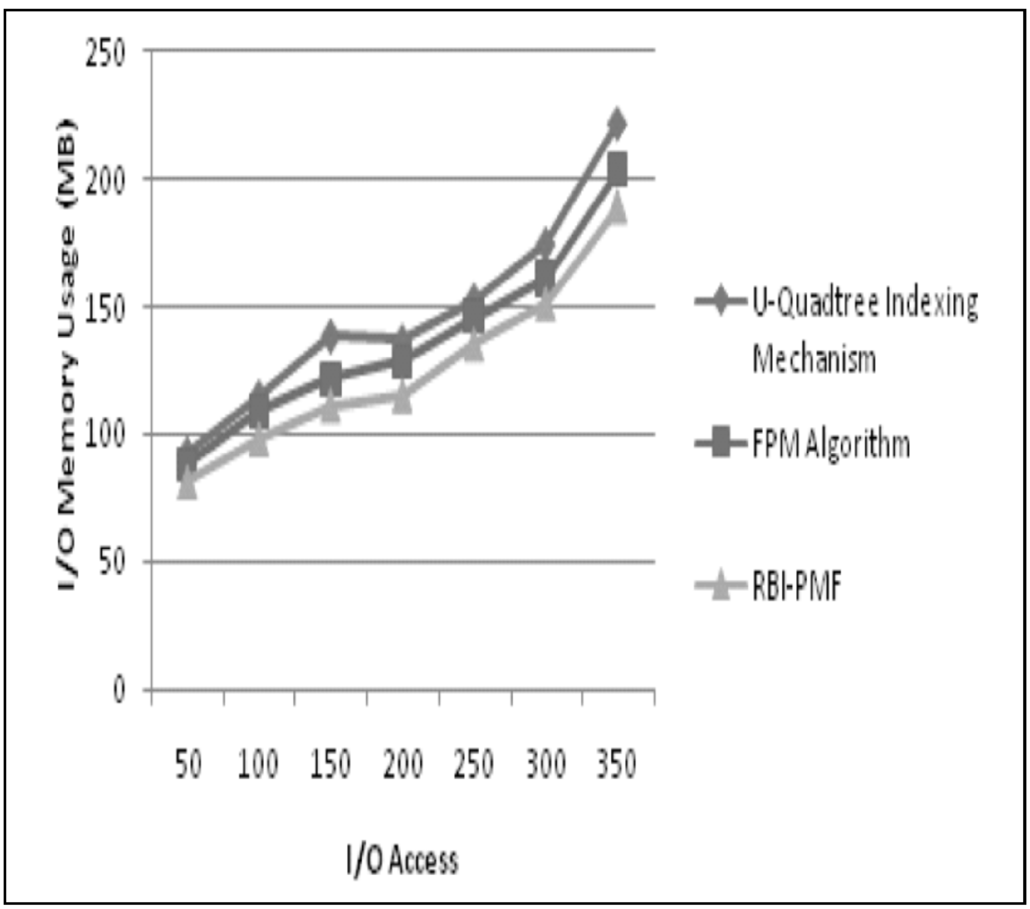

Figure 6: I/O Memory Usage Measure

I/O memory usage measures storage space used for indexing operation as tabulated in table 2 and Fig 6. The I/O access ranges are plotted against memory usage measure. The adaptive pruning technique is significantly used in RBI-PMF to reduce the memory usage rate. The pruning includes filtering of irrelevant data, thereby reducing memory usage by 11 – 20% in RBI-PMF when compared with U-Quadtree Indexing Mechanism [1]. For instance, on taking the 200 I/O instances, U-Quadtree Indexing Mechanism consumes 137 MB, FPM Algorithm consumes 128, whereas the RBI-PMF consumes only 115 MB. Thereby the RBI-PMF reduces the I/O memory usage. RBI-PMF significantly improves the indexing operation on the primary set, where by allows the users to manipulate the indexes. The indexing result gains higher flexibility rate with 6 – 10% minimal I/O memory usage when compared with the FPM Algorithm [2].

Table 3
Tabulation on Precision Rate

Object Count	*Precision Rate (%)*		
	U-Quadtree Indexing Mechanism	*FPM Algorithm*	*RBI-PMF*
100	0.73	0.62	0.86
200	0.75	0.64	0.89
300	0.74	0.67	0.90
400	0.78	0.69	0.92
500	0.82	0.70	0.93
600	0.83	0.72	0.93
700	0.85	0.73	0.95
800	0.86	0.75	0.97

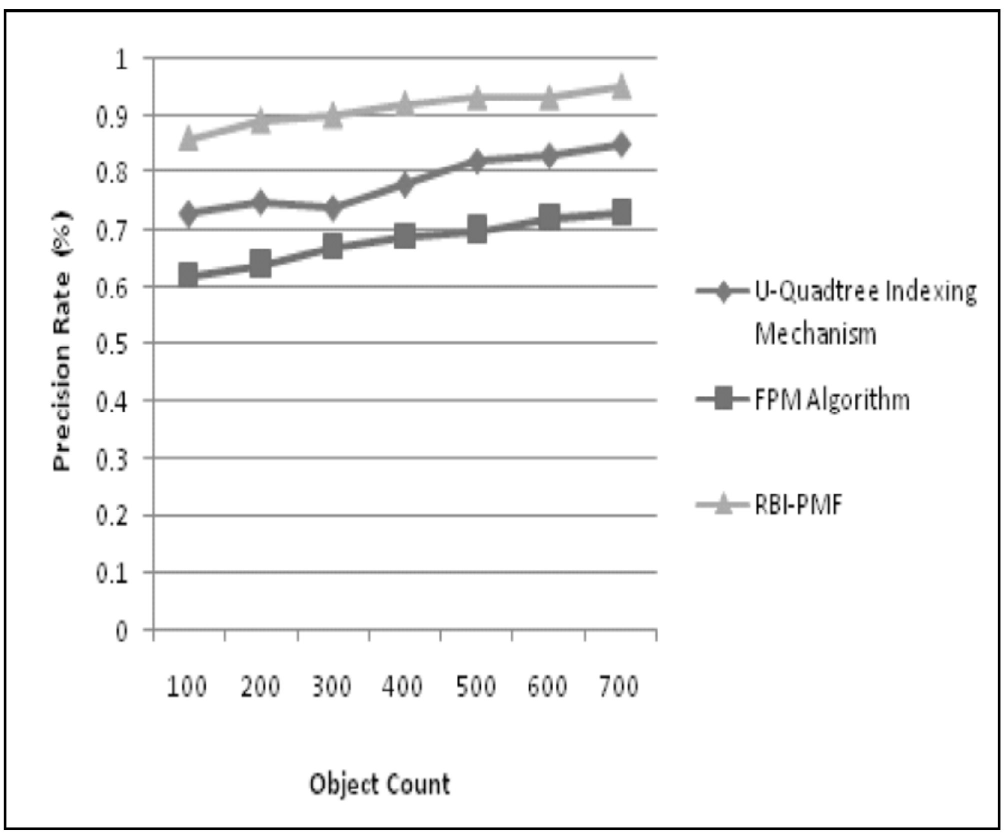

Figure 7: Precision Rate Measure

Table 3 and Fig. 7 illustrate precision rate based on the object count. The precision rate measures the accuracy level of indexing on the proposed RBI-PMF with the existing U-Quadtree Indexing Mechanism [1], FPM Algorithm [2]. The adding and removing of data objects is carried out using random B-tree indexing, so that it achieves the higher precise rate when compared with existing U-Quadtree Indexing Mechanism. The precision rate is improved by 11-21% when compared with U-Quadtree Indexing Mechanism, because the clustered data objects are used for the experimental evaluation. The RBI-PMF also improved by 29 – 39% in RBI-PMF when compared with existing FPM Algorithm [2].

Table 4
Quality of Multi-Dimensional Feature

Cluster Counts	*Quality of Multi-Dimensional Feature (%)*		
	U-Quadtree Indexing Mechanism	*FPM Algorithm*	*RBI-PMF*
5	73	82	86
25	74	84	87
50	74	87	91
75	78	89	93
100	79	88	93
125	82	90	92
150	83	91	93

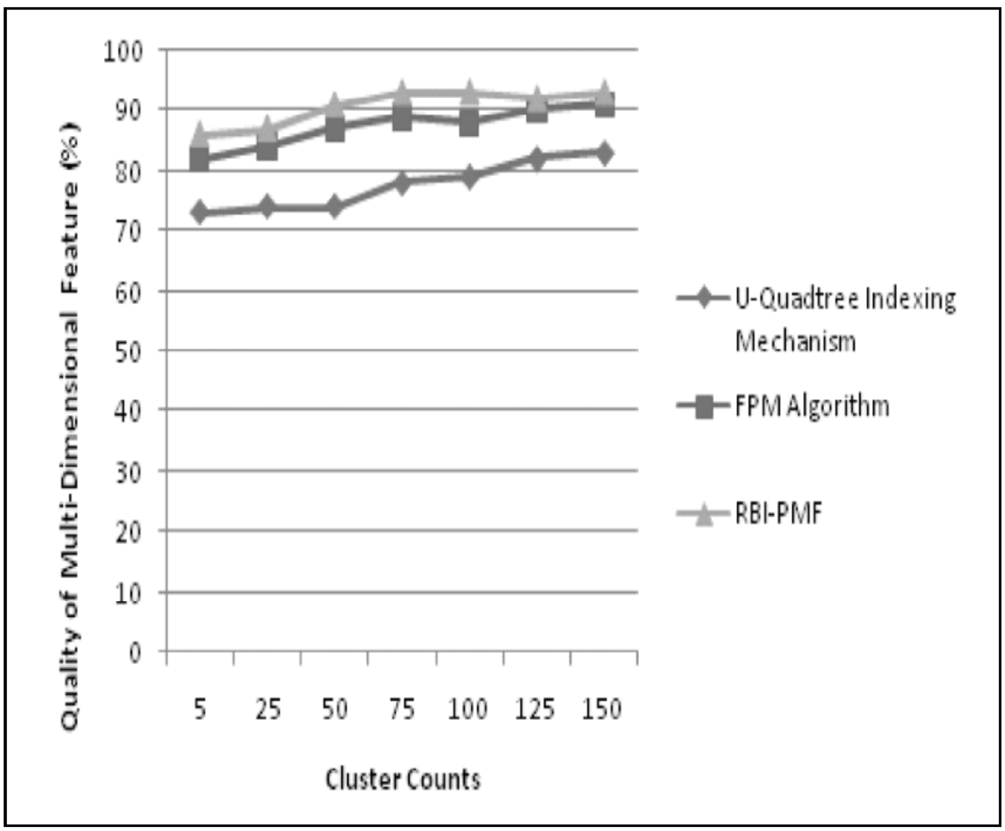

Figure 8: Quality of Multi-Dimensional Feature Measure

Table 4 and Fig. 8 demonstrate the quality of multi-dimensional feature based on cluster counts. The cluster counts on Random B-tree indexing operation supports update operations on using multi-dimensional clustered data objects as input vector. The RBI-PMF process is adaptive, where small rejection is carried out based on specified threshold points.

The threshold points help improving the quality of multi-dimensional feature by 12 – 22% when compared with U-Quadtree Indexing Mechanism [1]. The adaptive pruning removes (i.e..,) rejected the unwanted multi dimensional data features and improves the quality level by 2 – 5% in RBI-PMF when compared with FPM Algorithm [2].

Table 5
False Query Probabilistic Rate Tabulation

Query Probability Range	*False Query Probabilistic Rate (False %)*		
	U-Quadtree Indexing Mechanism	*FPM Algorithm*	*RBI-PMF*
0.10	0.07	0.05	0.04
0.20	0.09	0.08	0.06
0.30	0.10	0.08	0.06
0.40	0.15	0.13	0.11
0.50	0.18	0.16	0.13
0.60	0.22	0.19	0.17
0.70	0.25	0.22	0.19
0.80	0.27	0.24	0.20
0.90	0.28	0.26	0.22
1.00	0.30	0.27	0.24

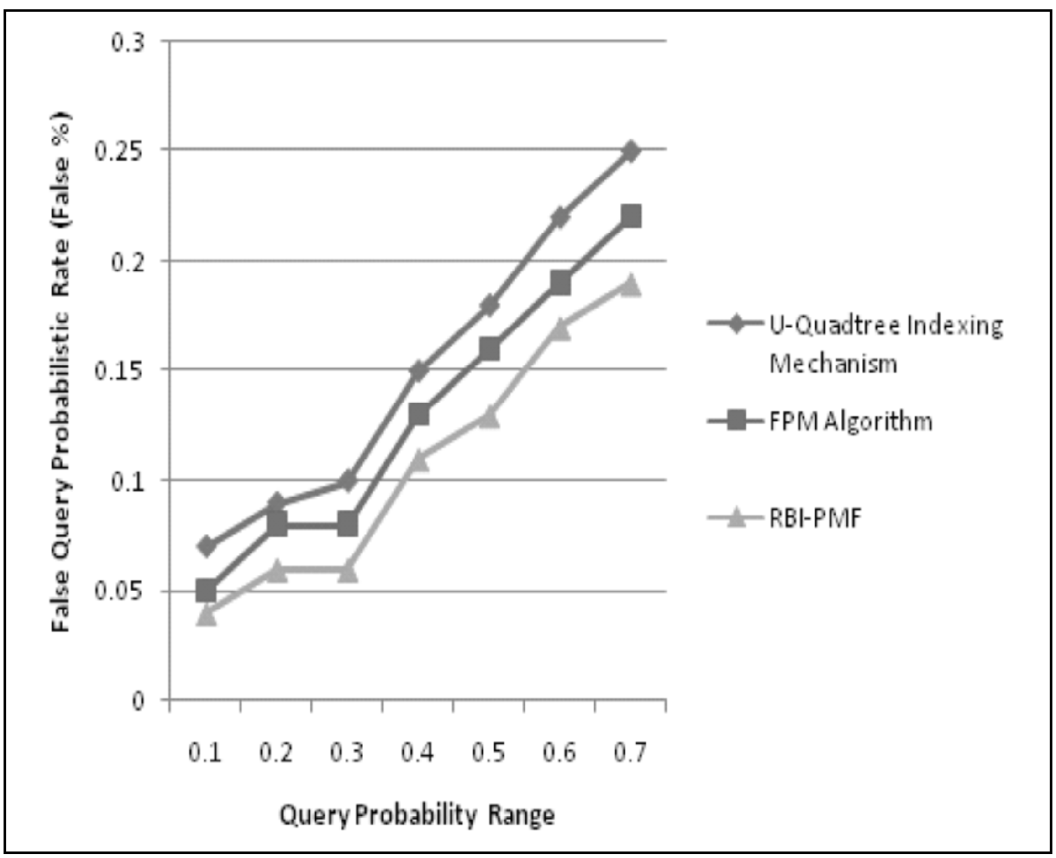

Figure 9: False Query Probabilistic Rate Measure

Table 5 and Fig. 9 presented the false query probabilistic rate based on query probability range. The Random B-tree node query series is analyzed and performs processing operations in RBI-PMF. RBI-PMF follows adjacent links and reduces false query probabilistic rate by 20 – 42% when compared with U-Quadtree Indexing Mechanism [1]. The pruning process still removes the unwanted objects and reduces false query probabilistic rate by 10 – 25% when compared with FPM Algorithm [2]. For instance, on the query probabilistic range of about 0.50 reduces the probabilistic rate to 0.13% when compared with FPM Algorithm [2]. The adjacent links reduces the probabilistic false rate in RBI-PMF when compared with the existing state of arts [1, 2].

Random B-Tree Indexing based on Probabilistic Multi-Dimensional Features supports update operations on cluster data objects generated from uncertain data. The supported technique for feature extraction provides precise information using random B-tree indexed clustered objects. Adaptive pruning method is also used to reduce the memory usage.

6. CONCLUSION

To minimize the processing time on indexing structure of multi-dimensional data, Random B-Tree Indexing based on the Probabilistic Multi-Dimensionality Features retrieve the user query result. The retrieval of result to user query is achieved with minimal processing time. Random B-Tree Index always achieves lesser memory and processing time on different cluster objects generated from uncertain data. The updating work is carried in Random B-Tree Indexing and achieves the precise information extraction. The probabilistic features used for matching the intersection of user query result value with multi-dimensional space value improve the extraction efficiency range. The processing of the user query is effectively done in RBI-PMF using adaptive pruning method. Theoretical analysis and experimental result shows that the RBI-PMF achieves 3%-4% effective indexing quality maintenance level, 7%-8% lesser processing time and memory rate. RBI-PMF responds to measure the false query probability rate and quality of the multi-dimensional feature.

REFERENCE

[1] Ying Zhang, Wenjie Zhang, Qianlu Lin and Xuemin Lin, "Effectively Indexing the Multi-Dimensional Uncertain Objects for Range Searching", *ACM Journal*, Vol. 2, No. 5, March 2012, Pp. 504-515.

[2] Charu C Aggarwal and Philip S Yu, "A Survey of Uncertain Data Algorithms and Applications," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 5, May 2009, Pp. 619-629.

[3] Luýs Leitao, Pavel Calado and Melanie Herschel, "Efficient and Effective Duplicate Detection in Hierarchical Data,"IEEE Transactions on Knowledge and Data Engineering", Vol. 25, No. 5, May 2013, Pp. 1028-1040.

[4] Hans Bindera, Henry Wirtha C and Jorg Gallea, "Gene Expression Density Profiles Characterize Modes of Genomic Regulation: Theory and Experiment", *Journal of Biotechnology*, *Elsevier Journal*, Vol. 149, No. 3, September 2010, Pages 98–114.

[5] Costas Panagiotakis and Nikos Pelekis et al., "Segmentation and Sampling of Moving Object Trajectories Based on Representativeness", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 7, July 2012, Pp.1328-1343.

[6] S Roy, D K Bhattacharyya and J K Kalita, "Deterministic Approach for Bi-clustering of Co-Regulated Genes from Gene Expression Data", *Advances in Knowledge-Based and Intelligent Information and Engineering Systems.*, 2012, Pp. 504-515.

[7] Sharadh Ramaswamy and Kenneth Rose, "Adaptive Cluster Distance Bounding for High-Dimensional Indexing," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 6, June 2011, Pp. 504-515.

[8] Ali Shahbazi and James Miller "Extended Sub tree: A New Similarity Function for Tree Structured Data," *IEEE Transactions on Knowledge and Data Engineering*, March 2013, Pp. 490-499.

[9] Zhisheng Li and Ken C.K. Lee, Baihua Zheng., Wang-Chien Lee., Dik Lun Lee., and Xufa Wang., "IR-Tree: An Efficient Index for Geographic Document Search," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 4, April 2011, Pp. 815-830.

[10] Marco Muselli, and Enrico Ferrari, "Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 1, January 2011, Pp. 37-50.

[11] Sari Kontunen-Soppela *et al.*, "Gene expression responses of paper birch (Betula papyrifera) to elevated CO2 and O3 during leaf maturation and senescence," Environmental Pollution., Science Direct., Vol. 158, No.4, April 2010, Pp. 959–968.

[12] You Jung Kim, and Jignesh M. Patel., "Performance Comparison of the R-Tree and the Quad tree for KNN and Distance Join Queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 7, July 2010, Pp. 1014-1027.

[13] Lei Zou and Lei Chen, "Pareto-Based Dominant Graph: An Efficient Indexing Structure to Answer Top-K Queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 5, May 2011. Pp. 727-741.

[14] Yuzhe Tang, Shuigeng Zhou and Jianliang Xu, "LIGHT: A Query-Efficient Yet Low-Maintenance Indexing Scheme over DHTs", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 1, 2010, Pp. 59-75.

[15] Eric Hsueh-Chan Lu, Vincent S. Tseng., and Philip S. Yu., "Mining Cluster-Based Temporal Mobile Sequential Patterns in Location-Based Service Environments," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 6, June 2011, Pp. 815-830.