Engineering Sciences International Research Journal
ISSN (Print) : 2330 – 4338 ; Volume 1 Issue 1 || February. 2013 ||

An Integrated Approach to Model Uncertain Spatial Data with UK-Medoids Clustering, MBR Pruning and R*-Tree Indexing

Ramachandra Rao Kurada[1], Aruna Kumari M[2], Durga Sri B[3]

Abstract: The uncertainties in spatial data mining is ubiquitous and may exist in spatial data, theories and techniques, mining process, knowledge characteristics, knowledge representation and knowledge interpretation. Uncertainty in spatial data set is due to various causes like imprecision, inconsistency and inaccuracy in the information acquired through data collection and amalgamation from the instruments and infrastructures on Geo-Informatics. Such uncertain data is usually represented in terms of uncertain regions over which the Probability Density Function (PDF) is defined.

In the present paper, the problem of clustering uncertain data has been addressed by proposing UK-Medoids algorithm which is designed to overcome the issues like accuracy and efficiency that are raised in UK-Mean when computing the expected distance between uncertain objects and cluster centroids. The proposed algorithm employs a similarity function DIE deviation to find the distance between uncertain objects. Experiments have shown that UK-Medoids outperforms existing algorithm from an accuracy view point while achieving reasonably good efficiency, but results in quadratic complexity. To surmount this spin-off the Min-Max bounding box pruning technique is incorporated in UK-Medoids algorithm. It reduces the complexity and improves the accuracy of clustering by speed up the computations, but this results pruning overheads. An optimized framework to model uncertainty and triumph over the previous blemish R*-Tree indexing was incorporated in UK-Medoids. This incarnation engender quality clustering, effective pruning and consistent indexing in modeling the spatial data.

Keywords: Information Entropy, R*-Tree, UK-Medoids clustering, Uncertain Spatial Data.

1. INTRODUCTION

Handling uncertainly [9] in data management is of paramount importance in a wide range of application contexts. Indeed, data uncertainty [1], [6] naturally arises from implicit randomness in a process of data generation/ acquisition, imprecision in physical measurements, and loss of data freshness.

The uncertainty-based spatial data mining [5] is to extract knowledge from the vast repositories of practical spatial data under the umbrella of uncertainties with the given perspectives and parameters. With different granularities, scales, mining-angles, and uncertain parameters, it discovers the collective attribute distribution of spatial entities by perceiving various variations of spatial data and their combinations in the data space. The concept is in the integrated context of both uncertainty [7] and spatial data mining. It is an uncertain process for spatial data mining to discover the little-amount refined knowledge from the large-amount coarse data.

Various notations of uncertainty have been defined depending on the application domain. Attribute – level uncertainty [4], [1] has been considered in this paper, to model according to a probability model. The uncertain object is usually represented by means of probability density functions PDFs, which describe the likelihood that the object appears at each position in a multidimensional space rather than by a traditional vector form of deterministic values.

Clustering is useful in exploratory data analysis. Cluster analysis organizes data by grouping individuals into a population in order to discover structure or clusters. Many partitioned algorithms [8] have been proposed, some based on k-centroid, some based on K-Medoid, some based on fuzzy analysis, etc. K-Medoid clustering is similar to k-Centroid clustering. Both of them attempt to partition the data by assigning each object to a representative and then optimizing a statistical homogeneity criterion - namely, the total expected squared dissimilarity. However, K-Medoid clustering only allows objects to be chosen as representatives. In comparison with K-Centroid, the use of Medoids for clustering has several advantages. Firstly, this method has been shown to be robust to the existence of noise or outliers and generally produces clusters of high quality. Secondly, this method can be used not only on points or vectors for which the mean is defined but also on any objects for which a similarity measure between two objects is given.

The present paper focuses on the introduction of a new set of pruning technique for the UK-Medoid algorithm that is based on Minimum Bounding Rectangle [12]. With a highly effective pruning method, Decrease in Information Entropy deviations (DIE) is computed and it occupies largest fraction of the execution time. The overheads incurred in realizing the pruning strategy become relatively significant. To further reduce execution time, a performance boosting technique based on R-trees index [2] was developed. Instead of treating the uncertain objects as an unorganized set of objects, they are indexed using a bulk-loaded R-tree. Each node in the R-tree represents a rectangular region in space that encloses a group of uncertain objects. The idea is to apply Minimum Bounding Rectangle based pruning techniques [13] called Min Max Bounding Box (mbb-umm) in a batch on an R-tree node rather than as an individual object.

R*-Tree [2], [3] is one of the variants of data structure called R-Tree. R*-Tree is based on a heuristic optimization and uses combination of several optimization criteria to arrange objects in to group objects and build a balanced tree over them. The data structure is fully dynamic. Insertions, deletions, updates and queries can be mixed and no periodic global reorganization is required.

Each leaf node represents one object within a given bounding box (rectangle overlapping the entire object). Internal nodes contain information about the smallest common bounding boxes of all their children. This arrangement allows traversing the tree from root to leaves while ignoring large unimportant plane areas (sub trees).

This algorithm can be easily extended to consider the DIE deviations of the objects to the center point instead of using only the bounding boxes, the next level to the hierarchy to be added. Every leaf node reached in the main cycle is reinserted back to set S with its exact object distance. The set S is implemented by the heap data structure. Minimum in the heap can be found and removed in $O(\log n)$ times as well as a new item can be inserted. This paper proposes an integrated approach to model uncertain spatial data using pruning and indexing techniques. The remained sections will be organized as follows. Section 2 is the Introduction, Section 3 presents the Literature survey. The accessible methods are presented in Section 4, Section 5 gives the experimental results. Conclusion is drawn finally in Section 6.

2. LITERATURE SURVEY

Uncertain data [6] is ubiquitous in real world applications due to various causes. In recent years, clustering uncertain data has been paid more attention by the research community and the classical clustering algorithm based on partition, density and hierarchy have been extended to handle the uncertain data.

Clustering is known as very useful tool in many fields for data mining. The structure of data sets is found through the clustering methods [10]. Now, the more the ability of computers increase, the more works for uncertainties have been studied. In the past, data handled by the computers was approximately represented as one point or value because of poor ability of the computers. However, the ability is now enough to handle uncertain data. Therefore, a lot of researchers have tried to handle original data from the viewpoint that the data should been represented as not one point approximately but some range exactly in a data space. Uncertainty may have an influence on the confidential level, supportable level, and interesting level of spatial data mining [11].

The inherent uncertainties that have their own characteristics play an important role in spatial data mining. Spatial uncertainties [3], [11] further include positional uncertainty; attribute uncertainty, topological uncertainty, inaccuracy, imprecision/inexactitude, inconsistency, incompleteness, repetition, vagueness, noisy, omittance, misinterpretation, misclassification, abnormalities and knowledge uncertainty. To control and reduce uncertainty in an acceptable degree, one is data acquisition that highlights the information acquired from the process of data collection and data amalgamation, the other is data cognition that emphasizes the knowledge discovered from data extraction process and information generalization. Third, are the usable techniques and methods that may possibly cope with the uncertainties. K-Medoids [8] is a clustering algorithm that is very much like K-means. The main difference between the two algorithms is the cluster center being used. K-means uses the average of all instances in a cluster, while K-Medoids uses the instance that is the closest to the mean, i.e. the most 'central' point of the cluster. Using an actual point of the data set to cluster makes the K-Medoids algorithm more robust to outliers than the K-means algorithm.

Spatial indexing methods [7] are used to process magnanimous spatial database for fast and effective results. These indexing methods directly affect the memory efficiency of spatial data as well as the spatial retrieval performance. To reduce the disk space for spatial uncertain data this paper uses the R*-tree proposed by Guttmann, which adopts the smallest bounding rectangular (MBR) to divide spatial entity by using ―the smallest area criterion, and construct dynamic index tree. The R*-tree can be regarded as an extension of the Btree for multi-dimensional rectangular objects. The R*-tree construction algorithm aims at minimizing the metrics like (i) the area, (ii) the perimeter of each MBR (iii) the overlap between two MBRs in the same node.

3. PROPOSED WORK

a. Uncertain Partitioning based Clustering Approaches

In UK-Medoids, as every object has the same cluster, the expected distance based approach in UK-Means [12], [13] cannot distinguish the two sets of objects having different distributions. In UK-Mediods clustering of uncertain objects is made according to the similarity between their probability distribution. In information theory, the similarity between two distributions can be measured by the Information Entropy deviation (IE). This IE is used to measure the similarity between distributions, and demonstrate the effectiveness of similarity. The PDFs [12], [13] is used over the entire data domain and the difference is captured using the IE deviation.

b. Similarity using IE Deviation

In general, IE between the two probability density functions is defined as follows: In the discrete case, let A and B be two probability distribution functions in a discrete domain D with a finite number of values. The IE deviation between A and B is$IE(A||B) = \sum_{x \in D} F(x) log \frac{A(x)}{B(x)}$. In the continuous case, let A and B be two probability density functions in a continuous domain D with a continuous range of values. The IE deviation between A and B is $IE(A||B) = \sum_{D} F(x) log \frac{A(x)}{B(x)} dx$. In both discrete and continuous cases, IE deviation is defined only in the case where for any x in domain D if A(x) > 0 then B(x) > 0, by convention,$0 \log \frac{0}{p} = 0$, for any p ≠ 0 and the base of log is 2.

c. Partitioning Clustering Methods on Uncertain data

A partitioning clustering method organizes a set of k uncertain objects O into n clusters $C_1, \dots, C_k\ such\ that\ C_i \subseteq O\ (1 \leq i \leq k), C_i \neq \phi, \bigcup_{i=1}^{k} C_i = 0, and\ C_i \cap C_j = \phi\ for\ any\ i \neq j$. Using IE deviation as similarity, a partitioning clustering method tries to partition objects into k clusters and chooses the best n representatives, one for each cluster. To minimize the total IE deviation the computation is $DIE = \sum_{i=1}^{n} \sum_{p \in C_i} IE\ (x||\ c_i)$.

For an object x in cluster$C_i (1 \leq i \leq k)$, the IE deviation $IE(x||c_i)$ between p and the representative $c_i$measures the extra information required to construct p givenc_i. Therefore, $\sum_{p \subset C_i} IE(x||c_i)$ captures the total extra information required to construct the whole cluster C_i using its representativec_i. Summing over all n clusters, the total IE deviation thus measures the quality of the partitioning clustering. The smaller the value of TIE, the better the clustering is performed.

The pseudo code of core uncertain K-Medoids is presented initially with IE deviations as a distance function to calculate the distances between any two uncertain objects. Further refinement is made by incorporate Min-Max bounding box pruning technique into UK-Mediods and final sophistication to the algorithm is made by integrating R*-Tree indexing in to UK-Mediods.

d. Uncertain K-Medoids Method

The Uncertain K-Medoids method consists of two phases, the building phase and the swapping phase. In the building phase, the uncertain K-Medoids method obtains an initial clustering by selecting k representatives one after another. The first representatives C_1 is the one which has the smallest sum of the IE deviation to all other objects in O. That is, $C_1 = argmin_x (\sum_{x' \in O\{x\}} IE(x'||x)$.

The rest of the n-1representatives are selected iteratively. In the ith (2 ≤ i ≤ n) iteration, the algorithm selects the representative C_i which decreases the total IE deviation as much as possible. For each object x which has not been selected, a test is made to select the current round. For any other non-selected object x', x' will be assigned to the new representative x if the divergence $IE(x'||x)$ is smaller than the divergence between x' and any previously selected representatives. Therefore, a calculation is made for the contribution of p' to the decrease of the total IE deviation by selecting x as $max\left(0, min_{j=1}^{i-1}\left(IE\left(x'||C_j\right)\right) - IE(x'||x)\right)$. The total decrease of the IE deviation is calculated by selecting x as the sum over the contribution of the non-selected objects, denoted by DIE(x). Then, objects to be selected in the ith iteration is the one that can incur the largest decrease that $C_i = argmax_{x \in O\{c_1, \dots, c_{i-1}\}}(DIE(x))$. In the swapping phase, the uncertain K-Medoids method iteratively improves the clustering by swapping a non-representative object with the representative to which it is assigned. For a non-representative object x, suppose it is assigned to cluster C whose representative is c. The effect of swapping x and c in two cases for all non-selected object x' other than x is considered. If x' currently belongs to c, when c is replaced by x, a reassignment of x' to x or one of the other n − 1 existing representatives, to which x' is the most similar is considered. If x' currently belongs to a representative c' other than c, and $IE(x'||x) < IE(x'||c')$, x' is reassigned to x.

When a reassignment happens, the decrease in the total KL deviation by swapping x and c is recorded. After all non-representative objects are examined; the object is selected by swapping x and c. Then, object is selected by object x_{max} which can make the largest decrease. That is, $x_{max} = argmax_{p \in O\{c_1, \dots, c_k\}}\left(DIE(x)\right)$. A check is made on swapping x_{max} to improve the clusters, i.e.,$DIE(x_{max}) > 0$. If so, the swapping is carried into execution. Otherwise, the method terminates and reports the final clustering. The following algorithm presents the pseudo code of the uncertain K-Medoids method.

Algorithm 1: Uncertain K-Medoids Algorithm

Input: a set $O(o_1, \dots, o_n)$ of uncertain objects, the number of clusters n; Output: n clusters$c_1, \dots, c_n$;

```
1: $C_1 = argmin_x(\sum_{x' \in O\{x\}} IE(x'||x)$
2: i=2;
3: while $i \leq k$ do
4: for $x \in O \backslash \{ c_1, \dots, c_{i-1}\}$ $do$
5:$IE(x) = \sum_{x' \in O\{x, c_1, \dots, c_{i-1}\}}$   $max$
        $\left(0, min_{j=1}^{i-1}\left(IE\left(x'||C_j\right)\right) - IE(x'||x)\right)$
6: $c_i = argmax_{p \in O\{c_1, \dots, c_{i-1}\}}(DIE(x))$
7: $i = i + 1$
8: $c_i = \phi(1 \leq i \leq k)$
9: repeat
10: for $x \in O$ $do$
11: $j = argmin_{i \in [1,k]}\left(IE(x||c_i)\right)$.
12: $x.IE = IE\left(x||c_j\right)$, $c_j = c_j \cup \{x\}$
13: for $x \in O \backslash \{c_1, \dots, c_{nk}\}$ do
14: assume $x \in C$; $DIE(x) = 0$;
$for\ x' \in C\ do$
15: $j = argmin_{i \in [i,k]}\left(IE(x||c_i)\right)$
16:$DIE(x) = DIE(x) + x'.IE - IE(x'||c_j)$
17: for $x' \in O \backslash C \backslash \{c_1, \dots, c_{nk}\}$ do
18: if $IE(x'||x) < x'.IE$ then
19:$DIE(x) = DIE(x) + x'.IE - IE(x'||x)$
20: $x_{max} = argmax_{x \in O\{c_1, \dots, c_n\}}(DIE(x))$
21: if $DIE(x_{max}) > 0$ then
22: replace the center to which $x_{max}$ is assigned in the last iteration by $x_{max}$
23: until $DIE(x_{max}) \leq 0$
24: return $c_1, \dots, c_n$
```

The flaw in Algorithm 1 is that it cannot scale on huge spatial dataset due to its quadratic complexity with respect to the number of objects and it is computationally expensive. To bring down the complexity Min-Max bounding box pruning technique is implanted in to the UK-Medoids. This technique tries to improve the performance of UK-Means [12], [13] efficiently minimize the total dissimilarity with each cluster.

e. Min-Max Bounding Box (MM-BB) Pruning Technique

For a given objecto_i, the set Q_x stores the set of candidate cluster representatives that are potentially the closest too_i. In the Min Max Bounding Box technique [12], [13], for an object o_i and a cluster representative c_j, certain points in MBR_i are geometrically determined. The distance from those points to c_j is computed to establish bounds on DIE. The formulation is $MinD\left(o_i, c_j\right) = min_{xMBR_i} d(x, c_j)$,

$MaxD\left(o_i, c_j\right) = max_{xMBR_i} d(x, c_j)$, $MinMaxD(o_i) = min_{c_j \in C}\left\{MaxD\left(o_i, c_j\right)\right\}$. It should be obvious that$MinD\left(o_i, c_j\right) \leq DIE\left(o_i, c_j\right) \leq MaxD\left(o_i, c_j\right)$. Then if $MinD\left(o_i, c_q\right) > MaxD\left(o_i, c_q\right)$ for some cluster representativesc_j and c_q, it is derived as $DIE\left(o_i, c_p\right) > DIE\left(o_i, c_q\right)$ without computing that exact values of the DIEs. So, object o_i will not be assigned to cluster p, thus we can prune away cluster p without bothering to compute$DIE\left(o_i, c_p\right)$.

As an optimization pruning is done on cluster p if $\left(o_i, c_p\right) > MinMaxD(o_i)$, depending on data distribution, the pruning condition $MinD\left(o_i, c_j\right) > MinMaxD(o_i)$ potentially removes many clusters. This avoids many DIE computations instead of computing MinD and MaxD. The pseudo code for MM-BB technique is shown as Algorithm 2. This pseudo code is inserted in algorithm 1 between the lines 4 and 5.

Algorithm 2: Min-Max Bounding Box (Mm-Bb) Pruning Technique

```
1:$Q_x \leftarrow C$ // candidate cluster
2:for all$\in C$ do // for a fixed object $o_j$
3:compute $MinD\left(o_i, c_j\right)$ and $MaxD\left(o_i, c_j\right)$
4:compute $MinMaxD(o_i)$
5:for all $c_j \in C$ do
6:if $MinD\left(o_i, c_j\right) > MinMaxD(o_i)$ then
7:Remove $c_j$ from $q_x$
8:if $|Q_x| = 1$ then // only one candidate remains
9:$DIE(x) \leftarrow j$ where $c_j \in Q_x$
10:else
11:for all $c_j \in Q_x$ do // remaining candidates
```

MM-BB pruning technique improves the performance of UK-Medoids significantly by making use of efficiently evaluable bounds on DIE to avoid many DIE computations. This scenario dominates the execution time of UK-Medoids algorithm. To further reduce the execution time, pruning costs and to take the advantage of spatial distribution, the R*-tree indexing was incorporated into UK-Medoids algorithm.

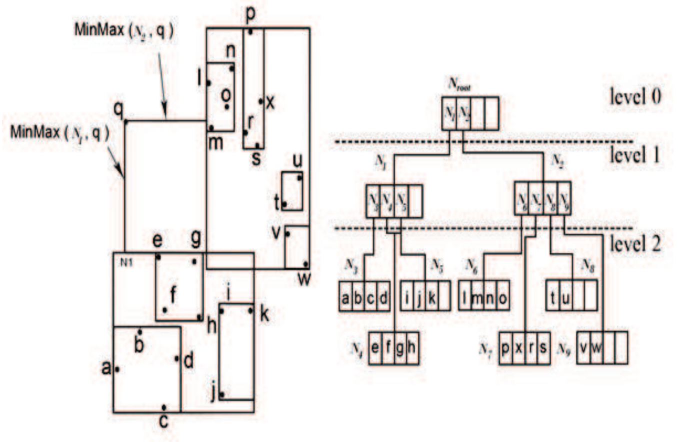

Fig. 1 (a). R*-Tree nodes (b) Structure

f. R*-Tree

The R*-Tree [12], [13] is the most vigorous method which is underlined by the fact that for every query file and every data file less disk accesses are required than by any other variants. The gain in efficiency of the R*-tree for smaller query rectangles is higher than for larger query rectangles, because storage utilization gets more important for larger query rectangles. This emphasizes the goodness of the order preservation of R*-tree. R*-tree has the best storage utilization by using the reinsert the average insertion cost is not increased.

A spatial access method is required to handle both spatial objects and point's objects efficiently. Points can be considered as degenerated rectangles and in this spatial data the rectangles are very small relatively to the data space.

Fig. 1 is illustrated with an example for R*-Tree by assuming a capacity of four entries per node. Points that are nearby in space (a, b, c, d) are inserted into the same leaf node (N3). Leaf nodes are recursively grouped in a bottom-up manner according to their vicinity, up to the top-most level that consists of a single root. Each node is represented as a minimum bounding rectangle (MBR) enclosing all the points in its sub-tree. The nodes of an R*-Tree are meant to be compact, have small margin and achieve minimal. Additionally, in practice, nodes at the same level contain a similar number of data points, due to a minimum utilization constraint. These properties imply that the R*-Tree provides a natural way to partition P according to object proximity and group cardinality criteria.

However R*-Trees have been used exclusively for processing spatial queries such as range search, and spatial joins. The most common such metric is MinMax (N, q), which is defined as the minimum possible distance between q and any point in the sub-tree rooted at node N. The new concepts incorporated in the R*-tree are based on the reduction of the area, margin and overlap of the directory rectangles. Since all the three values are reduced, the R*-Tree is very vigorous against spatial data distribution. Furthermore, due to the fact of the concept of forced reinsert, splits can be prevented, the structure is reorganized dynamically. The pseudo code mentioned in Algorithm 3 is integrated in Algorithm1 after line 24.

Algorithm 3. R*-Tree

Insert_Procedure

IP1:Invoke ChooseSubtree (ST1), with the level as a parameter, to find an appropriate node M, where to place the new entry A.

IP2:If M< Rstar_tree_minimum_bounding _rectangle entries, accommodate A in M.

If M has Rstar_tree_minimum_bounding_ rectangle entries, invoke Overflow _Procedure with the level of N as a parameter (for Reinsertion/Split)

IP3:If OverflowProcedure (OP) was called and Split was performed, proliferate OP upwards if necessary. If it caused a split of the root, create a new root.

IP4: Adjust all covering rectangles in the insertion path.

Choose_Subtree

ST1: Set M to be the root

ST2: If M is in desired tree level, return M. If the child pointers in M, determine minimum overlap cost; choose the entry in M whose rectangle needs the least overlap extension to include the new data rectangle. Resolve association main by choosing the entry whose rectangle needs least area extension, the entry with the rectangle of minimum area.

If the child-points in M do not point to leaves, determine the minimum area cost, choose the entry in M whose rectangle needs least area extension to include the new data rectangle. Resolve association by choosing the entry with the rectangle of smallest area.

ST3: Set M to be the child-node pointer of the chosen entry and repeat from ST2.

Overflow_Procedure

OP: If the level is not the root level and this is the first call of overflow treatment in the given level during the insertion of one data rectangle, then invoke ReinsertProcedure(RP1) else invoke SplitProcedure (SP1).

Reinsert_Procedure

RP1: For all Rstar_tree_minimum_ bounding_rectangle + 1 entries of a Node M, compute the distance between the enters of the rectangles and center of the bounding rectangle of M.

RP2: Sort the entries in decreasing order of their distance computed in RP1

RP3: Remove the first Rstar_tree_ minimum_bounding_rectangle entries from M and adjust the bounding rectangle of M

RP4: In the sort, defined in RP2, starting with the maximum distance, invoke InsertProcedure(IP1) to reinsert the entries

Split_Procedure

S1: Determine the axis, to which the split is performed.

S1a: For each axis: Sort the entries by the lower then by the upper value of their rectangles and determine all distributions as mentioned above. Compute S, the sum of all margin-values of the different distributions

S1b: Choose the axis with the minimum S as split axis

S2: Besides the chosen split axis, choose the distribution with the minimum overlap-value. Resolve associations by choosing the distribution with minimum area-value

S3: Apportion the entries into two groups.

g. Group-based pruning

Integrating the R*-tree with UK-Medoids and MMBB pruning the multi-level grouping of the uncertain objects had already been processed. The information kept at each node helps to do pruning in batch, thereby further boosting the performance of the pruning algorithms. Instead of repeating the cluster assignment to each uncertain object one after another as in UK-Medoids, the tree is recursively traversed down, starting from the root node. Each entry is examined at the root node. Each entry A represents a group of uncertain objects. The MBR of A is readily available from the R*-tree. So, the cost of the R*-Tree construction is averaged over all iterations which is very negligible.

4. EXPERIMENTAL RESULTS

The algorithms described in the previous section have been implemented on java using JDK 1.6.0 and a series of experiments were performed on a PC with Intel(R) corei3, 2.93 GHz and 4GB of main memory, running on windows 7 operating system. This section focuses on the algorithm runtime performance aforementioned and subsequent integration of pruning and indexing techniques.

The spatial dataset Forest Cover Type is used for all the experiments to evaluate the performance of data uncertainty. Forest Cover Type is a benchmark quandary extracted from http://kdd.ics.uci.edu /databases/covertype/covertype.html UCI KDD Archive. This problem relates to the actual forest cover type for given observation that was determined from US Forest Service (USFS) Region to Resource Information System (RIS). Forest Cover Type includes 581,012 samples represented in point valued data with 7 cover type; each sample has 54 attributes including 10 remotely sensed data and 44 cartographic data. Data preprocessing is applied to this data set and is transformed into many uncertain data sets by replacing each data point with an MBR and also generates the PDF. All the prosposed algorithms in the previous section are experimented with only ten percent of the available data object due to main memory constriction.

For the chosen training set, a set of n MBRs are generated in m-dimensional space $[0,100]^{m}$. Each MBR's side length is generated randomly and is bounded by variable l. The MBR is divided into s grid cells, each corresponding to a PDF sample point. Each sample point is associated with randomly generated probability value, normalized so that the sum of probabilities of the MBR is equal to 1. These probabilities give a discredited representation of the PDF f_i of the corresponding object. For all the algorithms this paper uses the same dataset with variable c as random points to serve as the initial cluster center. The dataset, initial cluster representative and number of clusters to be generated will be fed as inputs to the algorithms. For plainness in scheming the graph the algorithms Uncertain K-Medoids is cited as ukm, Uncertain K-Medoids with Min-Max bounding box pruning is cited as mm-ukm, and Uncertain K-Medoids with pruning and R*-tree indexing is cited as rmm-ukm. The following decisive factors are some of the substantiation to rationalize the competence of the anticipated algorithms.

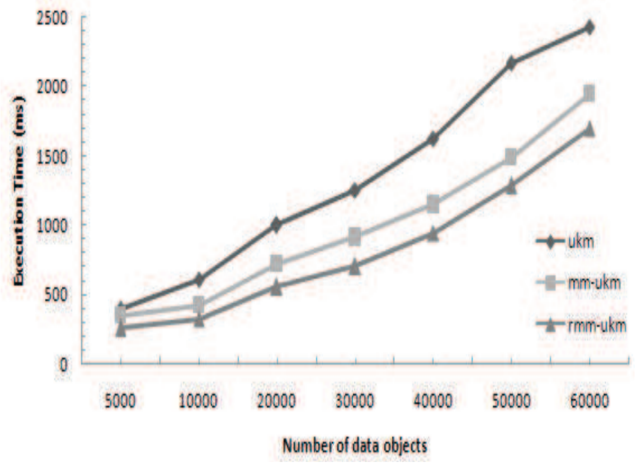

Fig. 2. Execution Time Effectiveness

a. Execution Time

One of an obligatory constraint for any algorithm was to assess its execution time recital when applied over its implanted datasets or databases. To evaluate the algorithms competence proposed in this paper, Fig. 2 is symbolized with the number of uncertain spatial data objects on horizontal axis and execution time measured in milliseconds on vertical axis. The execution time is a combinable constituent with pruning, tree construction and indexing. It is evident from Fig. 2 that as the number of data objects are increased the execution time also increases, but an important observation is that mm-ukm torrent about 50% of the time and rmm-ukm torrents about 60% when compared with the ukm at any surveillance of uncertain objects. Therefore it is one of substantiation that rmm-ukm is praiseworthy to model the spatial data objects.

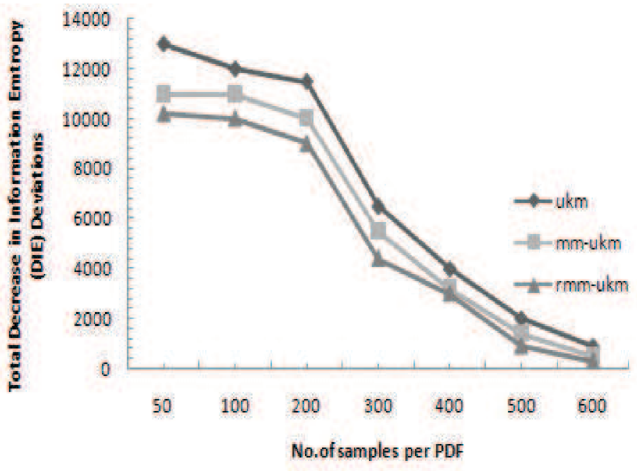

Fig. 3. Pruning Efficiency

b. Pruning Efficiency

Fig. 3 endeavors to study the pruning effectiveness of the anticipated algorithms, the total Decrease in Information Entropy (DIE) deviations is symbolized on vertical axis and number of samples per PDF is symbolized on horizontal axis. An important harmony amongst all the anticipated algorithms was as the number of samples per pdf increase the DIE deviations are abridged. It is an experimental fact that ukm doesn't employ any pruning technique and the other two algorithms exploit with the pruning technique so, the DIE deviations are more massive in ukm, and about 20% torrent is observed in mm-ukm and about 30% torrent is observed in rmm-ukm. It is another substantiation that entropy calculations dominate ukm and effective pruning technique considerably reduces the tree construction time and indexing.

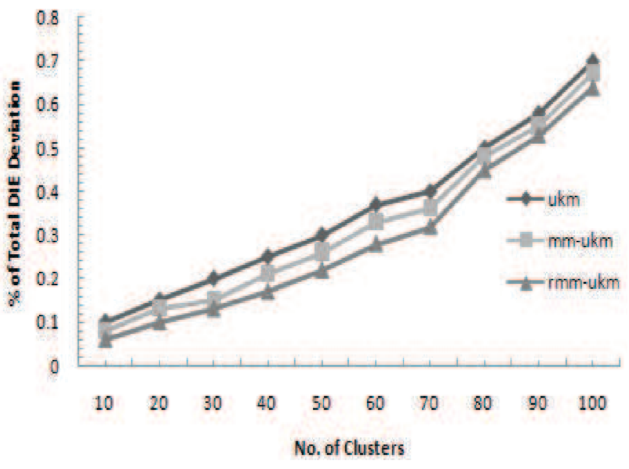

Fig. 4. Cluster Generation on Uncertain Spatial Data Objects

c. Cluster Generation

It is observed from Fig. 4 that the cluster arrangement in rmm-ukm is seen to be consistently better than mm-ukm and ukm over the whole range of total percentage of DIE deviations. This is because with a large number of clusters, cluster representatives are generally less spread out. It is therefore less likely that the integration of pruning and indexing will be able to prune all but one cluster for a given object. Hence total DIE deviations will have to be computed to determine the cluster assignment. The total percentage of DIE deviations in the algorithms increases when there is more number of clusters.

This substantiation is evidently revealed in Fig. 4 with number of clusters on horizontal axis and percentage of total DIE deviations in vertical axis. It is also verified in the anticipated algorithms as the number of cluster patterns increases the total percentage of DIE deviations, but when indexing and pruning are integrated the overall DIE are significantly abridged.

d. Block Size of R*-Tree

The R*-Tree is constructed with density of objects, granularity of the groups, size of MBR of each group that affect the block size in each level of R*-tree. It is to inspect from Fig. 5 with execution time on vertical axis and block size of R*-Tree on horizontal axis that the execution time increases slightly with block size.

With smaller blocks, the number of nodes in the R*-Tree increases, and so does the levels of the R*-Tree. This substantiation has a positive effect on pruning cost reduction because a deeper R*-Tree allows more chance for batching the pruning computation, which can be applied to a larger number of nodes at more dissimilar granularities.

e. Levels of R*-Tree

Fig. 6 is symbolized with percentage of decrease in Information Entropy on vertical axis and Levels of R*-Tree on horizontal axis. On the other hand, smaller blocks imply a larger number of R*-Tree nodes, this has the adverse effect of more processing and thus a slower execution time. So, when the block size is extremely small, such cost would increase up to a level that the time saving brought about by R*-tree advancement cannot compensate.

The substantiation from all the anticipated algorithms is that as the levels in R*-Tree increases the percentage of DIE deviations is dropped consistently. When pruning is integrated in ukm the percentage of DIE deviations is dropped by 20% in each level and when pruning and indexing are integrated in ukm the percentage of DIE deviations are dropped by 40% in each level.

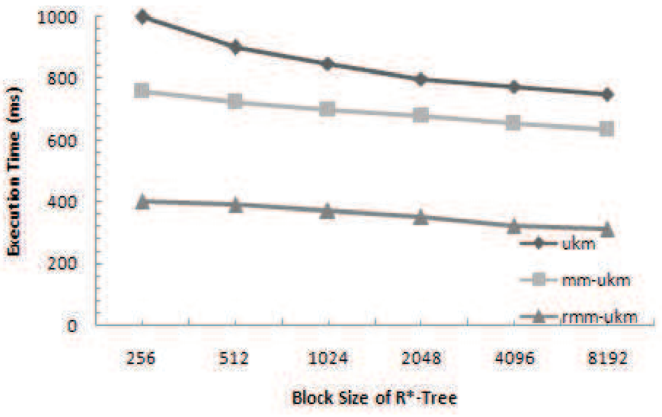

Fig. 5. Block Size of R*-Tree

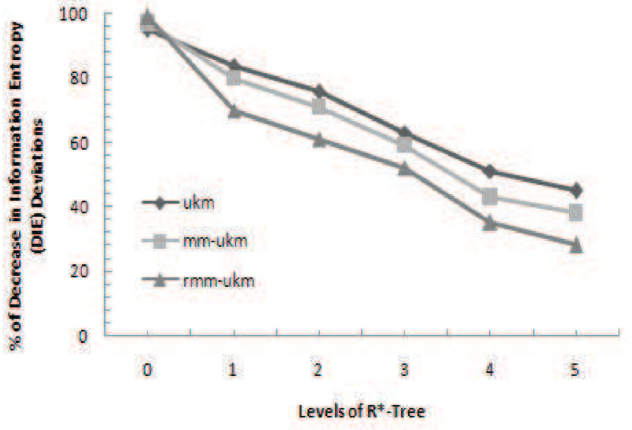

Fig. 6. Levels of R*-Tree

5. JUSTIFICATION FOR RESULTS

All the selected datasets originally contain deterministic values; hence uncertainty was synthetically generated for each object in the dataset. It is prominent that anticipated algorithms performed similarly for all the PDFs computations, since they employ similar clustering scheme; the only divergence is due to integration of pruning and indexing techniques in ukm. The new techniques bring down the execution of ukm and also reduce the computation time of PDF. A consistent demise pattern is observed in DIE deviations which substantiate that the integration being practical on uncertain spatial data and affirms the completeness in assortment of the anticipated algorithms.

6. CONCLUSION

This paper proposes a realistic way to model uncertainties in spatial data under the umbrella of uncertain spatial data mining with given perspective and parameter. The novel approach proposed in this paper accommodates tuples having numerical attributes with uncertainty described by arbitrary pdf. Performance is an issue to provoke privileged number of cluster progression even if more complicated computations of Information entropy deviations are involved.

Uncertain K-Medoids algorithm is primarily designed to handle tremendous uncertain spatial data. This algorithm is experientially verified to be highly effective and empirically good in lessening the execution time, but hoist quadratic intricacy. The Min Max bounding box pruning is devised into the UK-Medoids to shrink the convolution and improve efficiency, but hoist pruning outlay. To purge pruning overheads, accomplish improvement in execution time and save substantial amount of entropy computations R*-Tree was implanted in UK-Medoids algorithm. This anticipated algorithm is pragmatic in exploiting spatial data uncertainties with remarkable higher accuracies. The execution times are of an order of magnitude comparable to classical algorithms.

Future work may address the issues involved in modeling uncertain data with soft computing based partitioned clustering. To support proximity based queries and building indexes for multi dimensional spatial data Hilbert R-tree, priority R-tree are the substitutes.

7. REFERENCES

[1] B. Kao, S.D. Lee, D.W. Cheung, W.-S. Ho, and K.F. Chan, "Clustering Uncertain Data Using Voronoi Diagrams," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 333-342, Dec. 2008.

[2] Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: SIGMOD. (1990) 322–331.

[3] C. Bohm, P. Kunath, A. Pryakhin, and M. Schubert, "Querying Objects Modeled by Arbitrary Probability Distributions," Proc. 10th Int'l Symp. Spatial and Temporal Databases (SSTD), 2007.

[4] C.C. Aggarwal and P.S. Yu, "A Framework for Clustering Uncertain Data Streams," Proc. 24th IEEE Int'l Conf. Data Eng. (ICDE), 2008.

[5] Ester, M., Kriegel, H.P., Xu, X.: Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. In: SSD. (1995) 67–82.

[6] Francesco Gullo, Giovanni Ponti, Andrea Tagarelli: Clustering Uncertain Data Via K-Medoids. SUM 2008: 229-242.

[7] G. Cormode and A. McGregor, "Approximation algorithms for clustering uncertain data," in *PODS*, M. Lenzerini and D. Lembo, Eds. Vancouver, BC, Canada: ACM, 9th–11th Jun. 2008, pp. 191–200.

[8] Jaiwei Han, Micheline Kamber, Book Title —Data Mining Concept and Techniques, Morgan Kaufmann (An Imprint of ELSEVIER) Publication, ISBN: 1-55860-489-8, 2003.

[9] M. Chau, Reynold Cheng, B. Kao and J. Ng. Data with uncertainty Mining: An Example in Clustering Location Data. In the Methodologies for Knowledge Discovery and Data Mining, Pacific-Asia Conference (PAKDD 2006), Singapore, 2006.

[10] Ng R.T. and Han J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining, Proc. 20th Int. Conf. on Very Large Data Bases, 144-155. Santiago, Chile.

[11] Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB. (1994) 144–155.

[12] Ramachandra Rao Kurada, Durga Sri B, "A Novel Approach by Applying Partitioning Clustering over Uncertain Spatial Data Objects using Pruning Techniques and R*-tree Indexing" in ICCCE 2012, 12 & 13 April 2012, Dr. MGR University, Chennai, Published by Coimbatore Institute of Information Technology, ISBN No: 978-1-4675-2248-9.

[13] Ramachandra Rao Kurada, Durga Sri B, "Unsupervised Classification of Uncertain Data Objects in Spatial Databases Using Computational Geometry and Indexing Techniques", IJERA, Vol.2 Issue 2, March-April 2012, pp. 806-814.ISSN No.: 2248-9622.

* * *

[1]Ramachandra Rao Kurada, Assistant Professor, Department of Computer Applications, Shri Vishnu Engineering College for Women, Vishnupur, Bhimavaram, W.G. Dt., India, ramachandrarao.kurada@gmail.com.
[2]Aruna Kumar M, Department of Computer Applications, Shri Vishnu Engineering College for Women, Vishnupur, Bhimavaram, W.G. Dt., India, aruna.marii@gmail.com.
[3]Durga Sri B, Assistant Professor, Dept. of Computer Science & Engineering, Sri Sai Aditya Institute of Science & Technology, Surampalem, E.G.Dt. India, durga.sree.14@gmail.com