ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

Inclusive Analysis of Incomplete Datasets Using IkNN Search

Baswaraju Swathi[1], Sonia Singh[2], Sujithra K S [3]

Assistant Professor, Dept. of Information Science, Visvesvaraya Technological University, Bangalore, Karnataka, India[1]

Student, Dept. of Information Science, Visvesvaraya Technological University, Bangalore, Karnataka, India [2]

Student, Dept. of Information Science, Visvesvaraya Technological University, Bangalore, Karnataka, India[3]

ABSTRACT: Analyzing and processing any dataset is very important for any organization. It helps the organization in taking important business decisions. These decisions help in increasing the profit of any business organization. However, these data sets also consists of incomplete values or dimensions. These incomplete dimensions are discarded in most of the preprocessing techniques while mining data. The data might be missing due to reasons like failure of data transmission devices, accidental data loss or improper storage. Given a dataset of multi-dimensional objects and a query object, finding k closest objects to the query from the dataset without eliminating the missing value data object is a fundamental problem in data mining. This concept has a significant role in real time applications like image recognition, location based services, etc. In this paper, we study how to retrieve k-closest object to a given query from datasets with incomplete data. The proposed system explains the method for developing an effective indexing structure and also a pruning technique to implement the IkNN retrieval efficiently. For this we develop a LαB index structure using lattice and bucket structures and a lattice partition algorithm to solve IkNN search. Finally the lattice partition algorithm will give the k nearest neighbors to the given query object as the resulting candidate set. The closest data object to the query is placed in the first position of the candidate data set. Therefore, we propose an Analysis technique to process IkNN search on a data set having both complete and incomplete data objects**.**

KEYWORDS: inclusive analysis, incomplete data, indexing, Query processing techniques

I. INTRODUCTION

Research on big data analysis has gained a lot of interest in the past few years. It is evident from a recent IDC forecast [1] that the Big Data technology and its service market will grow at around six times more than the growth rate of the overall information and communication technology (ICT) market, and that its revenue is going to be above 50 billion in the year 2017. Although Bid Data assumes to mean the volume of data, it actually means the analysis on that data.

Analyzing and processing any dataset is very important for any organization as it helps in making key business decisions of an organization and also increases the profit of any business organization. However, these data sets also include incomplete data sets, which are often eliminated in the pre-processing techniques. . Although we can simply perform all the analysis tasks based on complete data sets by removing all the incomplete data, the analysis in incomplete and output is inaccurate. Given a dataset of multi-dimensional objects and a query object, finding k closest objects to the query from the dataset is a fundamental problem in data mining.

IkNN queries refer to finding k closest objects to the query from an incomplete dataset without discarding the incomplete data records. Our objective of this project is to develop and present efficient indexes, pruning techniques and algorithms to support the execution of I*k*NN queries efficiently.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

The proposed system explains the method for developing an effective indexes and pruning technique to implement the IkNN retrieval efficiently. In this system, we build a user interface using java swing. The java client encapsulates the query object given by the data analyst and request the Rserver to perform the search operation. The Rengine executes the search algorithm and provides the resulting candidate set of data objects to the Java client in the form of RList. The java code then converts the Rlist to java list and presents the output to the analyst in the user readable form.

The project mainly concentrates on developing and efficient index and pruning techniques for IkNN search on a dataset for a given query object q. The distance formula used to measure the closeness between query and data object assumes that the distance between objects with missing value dimensions is same as the average distance between objects based on dimensions with observed values. This orders the data objects in the correct order based on its distance from the query object. The results are displayed using Swing user-Interface. Thus, it becomes easier to locate an object in the dataset.

The paper introduces and explains the increasing need for retrieving data from incomplete dataset and the latest techniques that can be used to improve the accuracy of data retrieval. Indexing the dataset and pruning the unmatched data is one of the recently emerging methods to search a data record in a dataset efficiently. This paper explains various indexes to process a search on an incomplete dataset to effectively retrieve the accurate data record for a query.

II. RELATED WORK

In [1] this paper aims in finding the k-Nearest Neighbor objects to a given point. It implements the traditional kNN search. The paper presents an efficient nearest Neighbor algorithm. The algorithm finds the nearest Neighbor to the query. It then generalises it to finding the k-nearest Neighbors. The paper uses effective branch and bound search algorithm to process exact k-NN queries for R-trees, introduce several metrics for ordering and pruning the search tree. Future scope for the nearest neighbor queries is aimed to define, then analyse other metrics and characterising the behaviour of the algorithm in dynamic as well as static database. [2]. this paper addresses the problem of continuous monitoring of top-k Queries over multiple non-synchronized streams. This paper proposes an exact algorithm which builds on generating multiple instances of the same object in a way that enables efficient object pruning. This paper does tracking of top-k items over multiple data streams in a sliding window. Each stream represents one particular dimension of interest. This paper explores the design of efficient data structures tailored to maintaining the dominance set of a dynamic database in case of interval dominance check, as part of future work In [3 The author proposes a solution to solve the similarity query which is a fundamental problem in data mining and information retrieval research. The existing work on querying incomplete data addresses the problem where the data values on certain dimensions are unknown. The paper explains a probabilistic framework to model this problem so that the users can find objects in the database that are similar to the query with probability guarantee. In [4] Incomplete datasets, that's is databases that are missing data, are present in many research domains. It is important to derive techniques to access these databases efficiently. This paper utilizes two popularly employed indexing techniques, bitmaps and quantitation to correctly and efficiently answer queries in the presence of missing data. In [5] The author in addresses the problem where data values are uncertain and unknown on dimension incomplete database. The author introduces clustering, indexing, segmentation and searching as a part of the proposed work and finally probabilistic approach clustering forms group of certain attributes using 'CLIHD' algorithm. In [6] With the development of sensor and database technology, people get more focus on the big data issue. But too often the data is difficult to analyse. Affinity Propagation (AP) is a relatively new clustering Algorithm that has been introduced. However many datasets suffer from incompleteness due to various reasons, Therefore some strategies should be employed to make AP applicable to such incomplete datasets. [7] The author proposes and evaluates two indexing schemes for improving the efficiency of data retrieval in high-dimensional databases that are incomplete. These schemes are novel in that the search keys might have missing attributes. Author currently extends the work reported here in several ways. Firstly, he examines the impact of partial queries, the queries that may contain missing information and then he also plans to study a number of other indexing approaches. [8] The author discusses an important query for uncertain datasets in data mining which is called the Probabilistic k-Nearest-Neighbor Query [8]. It computes the probabilities of sets of k objects for being the closest to a given query point. Evaluating such query is expensive, because there is an exponentially more number of k object-sets, for which large number of numerical integration is required. The proposed solution can be applied to uncertain data with arbitrary

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

probability density functions. Different from the exact database, evaluating T-k-PNN requires probability information, and performs expensive numerical integration. Thus, the paper proposed various pruning techniques with consideration of both distance and probability constraints. As shown by the experimental results, with the k-bound filtering technique, a lot of unqualified objects can be pruned.

III. PROPOSED ALGORITHM

A. *Problem Formulation:*

Initial battery energy (IBE) is 50Jules for each node. This section introduces a distance function formula to measure the closeness between objects even with missing data values, which helps us to formalize the I*k*NN retrieval. The paper introduces and explains the increasing need for retrieving data from incomplete dataset and the latest techniques that can be used to improve the accuracy of data retrieval.

Symbol Notation	Description
O	a *d*-dimensional data object with missing values on some dimensions
B	a bitmap corresponding to an object *o* with *d* bits
o[i]	the *i*-th dimensional value of an object *o*
iset(o)	the set of the dimensions *i* where *o* is observed
	the distance between two objects *o* and *p*
Sk	the global/final result of an I*k*NN query
α(o)	the α value of an object o

Table 1: summarizes the symbols used frequently

We use a dash "-" to represent a missing value in the data object *o*. In addition, we use a bit string with *d* bits, to denote whether the dimensional values of the object *o* are missing. The *i*-th bit of β is on (= 1) if its *i*-th dimensional value is observed; or else its *i*-th bit is off (= 0). Further, *iset*(*o*) is the set of the dimensions *i* on which *o* is observed. Consider a data object A (-, 8, -, 15), its bit string value β = (0101) and iset (B1) = {d2, d4}.

In our project, we use and apply the distance function defined by Dixon, that is stated below for I*k*NN search.

$$\mathfrak{D}(o,p) = \frac{d}{|Iset(o) \cap Iset(p)|} \sum_{i=1}^{d} \delta_i(o,p)^2$$

where δ is defined as

$$\delta_i(o,p) = \begin{cases} o.[i] - p.[i], & \text{if } i \in Iset(o) \cap Iset(p) \\ 0, & \text{otherwise} \end{cases}$$

The distance function defined computes the distance between two incomplete objects, normalizes it to compensate for the missing values i.e. according to the distance definition, the distance between objects with missing value dimensions is equivalent to the average distance between objects based on dimensions with observed values. Further, if there are no missing values, Ð (*o*, *p*) is the square of Euclidean distance between objects, and aslo ranks the objects in the same order as the Euclidean distance does.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

Data Object	Dim-1	Dim-2	Dim-3	Dim-4
A1	3	4	10	2
B1	-	8	9	15
C1	-	9	-	27
D1	-	-	-	23
E1	96	24	30	-
A2	60	27	34	46
B2	-	26	53	17
C2	-	24	-	81
D2	-	-	-	11
E2	88	55	26	-
A3	56	13	21	7
B3	-	89	46	24
C3	-	17	-	35
D3	-	-	-	9
E3	66	43	22	-
A4	96	45	29	33
B4	-	2	91	54
C4	-	82	-	54
D4	-	-	-	85
E4	26	58	69	-
A5	6	87	37	29
B5	-	82	43	38
C5	-	2	-	39
D5	-	-	-	46
E5	16	36	48	-

L-1		L-2		L-3				L-4	
B1		B1		B1		B2		B1	
α	OBJECT	α	OBJECT	α	OBJECT	α	OBJECT	α	OBJECT
9	D3(-,-,-,09)	18	C1(-,09,-,27)	10.7	B1(-,08,09,15)	33.3	E5(16,36,48,-)	4.8	A1(03,04,10,02)
11	D2(-,-,-,11)	20.5	C5(-,02,-,39)	32	B2(-,26,53,17)	43.7	E3(66,43,22,-)	24.3	A3(56,13,21,07)
23	D1(-,-,-,23)	26	C3(-,17,-,35)	49	B4(-,02,91,54)	50	E1(96,24,30,-)	39.8	A5(06,87,37,29)
46	D5(-,-,-,46)	52.5	C2(-,24,-,81)	50.7	B3(-,82,46,24)	51	E4(26,58,69,-)	41.8	A2(60,27,34,46)
85	D4(-,-,-,85)	68	C4(-,82,-,54)	54.3	B5(-,82,43,38)	56.3	E2(88,55,26,-)	50.8	A4(96,45,29,33)

Fig 1.2 Sample data and derived lattices and buckets and alpha value

The LαB index is an efficient structure to support incomplete data IkNN retrieval. The index organizes and clusters incomplete data objects in a two-layer structure, a lattice layer and a bucket layer. For the coarse (lattice) layer, LαB employs the lattice structure to cluster the buckets based on the total number of observed dimensions. In the fine (bucket) layer, for every lattice, LαB partitions objects *o* in the lattice into buckets based on *Iset* (*o*) sets, and the objects in the same buckets share the same *Iset* (*o*) set.

The Fig 1.2 shows the dataset of 25 records with 4 dimensions and missing values. Based on the problem formulation structure we have defined, the figure shows the derived lattices, buckets. The alpha value of an object α (o) is the average of all the non-missing dimensional values of the data object o. Consider an object A (2, 3, 4,-), then the value α (A)=(2+3+4)/3=3.

B. *Proposed Solution:*

The project mainly concentrates on developing and efficient index and pruning techniques for IkNN search on a dataset for a given query object q. Unlike the traditional approaches, the proposed system does an inclusive analysis of the missing value data objects in the dataset.

The distance formula used to measure the closeness between query and data object assumes that the distance between objects with missing value dimensions is same as the average distance between objects based on dimensions with observed values. This orders the data objects in the correct order based on its distance from the query object.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

The LαB index arranges the records in the dataset in the order that is easier to prune the records based on alpha and distance pruning. Thus, it becomes easier to locate an object in the dataset.

IV. PSEUDO CODE

The proposed system explains the method for developing an effective indexes and pruning technique to implement the IkNN retrieval efficiently. In this system, we build a user interface using java swing. The java client encapsulates the query object given by the data analyst and request the Rserver to perform the search operation.

Fig 1.3 User-Interface for Query Input

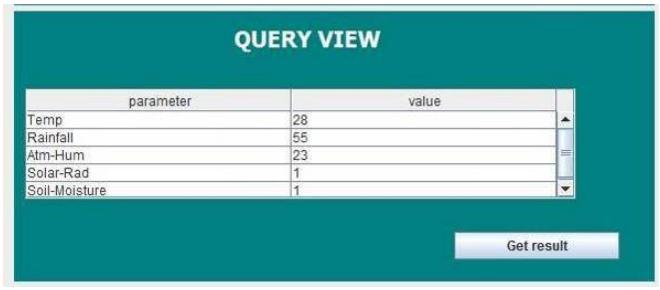


Fig 1.3 User-Interface for Query View

The user is allowed to select the crop as input. Then the parameter values associated with the crop is taken as Query Input.

On clicking the Submit button User-Interface submits the user input and executes the IKNN algorithm.

Initially the datasets are organized based on their Alpha values. These Alpha values are the Average of the observed dimensions in their respected lattices and buckets.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

```
for(i in 1:nrow(example2))
{
   row<-example2[i,]
   rowna<-row[!is.na(row)]
   x=length(rowna)
   if(x==1)
      L1<-c(L1,i)
   else if(x==2)
      L2<-c(L2,i)
   else if(x==3)
      L3<-c(L3,i)
   else if(x==4)
      L4<-c(L4,i)
   else if(x==5)
      L5<-c(L5,i)
}
```

Fig 1.4 The above figure shows the Lattice code Snippet.

```
bucket<-function(en,latt)
{
   for(j in 1:ncol(en$extemp))
   {
      vec1<-en$extemp[,j]
      ind1<-which(!is.na(vec1))
      count=0
      for(l in 1:length(latt))
      {
         vec=example2[latt[l],]
         indr<-which(!is.na(vec))
         if(isTRUE(identical(indr,ind1)))
         {
            count=count+1
            en$exbuck[count,j]<-latt[l]
         }
      }
   }
}
```

Fig 1.4 the above figure shows the Bucket code Snippet.

The Rengine executes the search algorithm and provides the resulting candidate set of data objects to the Java client in the form of RList. The java code then converts the Rlist to java list and presents the output to the analyst in the. user readable form

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

Algorithm 1 Lattice Partition Algorithm (LP)

Input: an incomplete dataset S, a query object q, a parameter k, a $L\alpha B$ index L
Output: the result set S_k of an IkNN query on S

```
1: initialize a max-heap S_k ← ∅
2: for each bucket B ∈ L do
3:     q_α ← (Σ_{i∈Iset(B)} q.[i]) / |Iset(B)|
4:     if | S_k | = k then
5:         α_U ← q_α + D(S_k.top,q)/d , α_L ← q_α − D(S_k.top,q)/d   // update α_U, α_L by Eq. 4
6:     pos = BinarySearch(B, q_α)                                    // find the position of q in bucket B
7:     if pos ≥ 0 then
8:         Search(S_k, k, B, pos, q_α, 0)                            // visit objects from B[pos] to B[0]
9:     if pos+1< B.size then
10:        Search(S_k, k, B, pos+1, q_α, 1)                          // visit objects from B[pos+1] to
           B[B.size−1]
11: return S_k

Search(S_k, k, B, pos, q_α, tag)
12: i ← pos, flag ← true
13: while (i ≥ 0 ∧ tag = 0) or (i < B.size ∧ tag = 1) do
14:    get the object o in B[i]
15:    if | S_k | < k then
16:        D(o, q) ← Get-Dist(o, q)
17:        S_k ← S_k + {o}
18:        if | S_k | = k then
19:            α_U ← q_α + D(S_k.top,q)/d , α_L ← q_α − D(S_k.top,q)/d   // update α_U, α_L by Eq. 4
20:    else if (α(o) > α_L ∧ tag = 0) or (α(o) < α_U ∧ tag = 1) then
21:        D(o, q) ← Get-Dist(o, q) by partial distance pruning          // Heuristic 2
22:        if D(o, q) ≠ -1 then
23:            S_k ← S_k − { S_k.top} + {o}
24:            α_U ← q_α + D(S_k.top,q)/d , α_L ← q_α − D(S_k.top,q)/d   // update α_U, α_L by Eq. 4
25:    else
26:        flag ← false, break                                           // Heuristic 1
27:    if tag = 0 then i ← i − 1
28:    if tag = 1 then i ← i + 1
```

Fig 1.5 Proposed system for searching an incomplete datasets

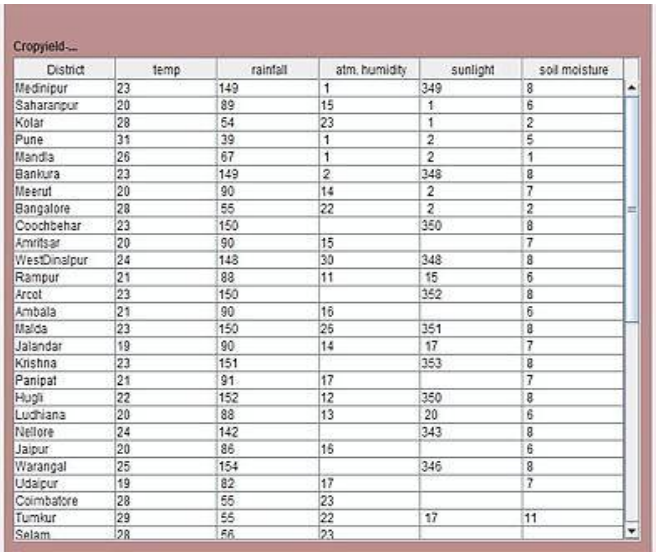
Unlike the existing approach, this proposed system does not eliminate missing data records from data analysis to search for the k data objects closest to the query object.

V. RESULTS

The backend Rserve does the inclusive analyses of missing value data objects for the IkNN retrieval as follows. Identify the query object and introduce a method to measure the closeness between the query object and data objects even with incomplete data values. The closeness is measured using the distance formula given by Dixon.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India



Cropyield-...

District	temp	rainfall	atm. humidity	sunlight	soil moisture
Medinipur	23	149	1	349	8
Saharanpur	20	89	15	1	6
Kolar	28	54	23	1	2
Pune	31	39	1	2	5
Mandla	26	67	1	2	1
Bankura	23	149	2	348	8
Meerut	20	90	14	2	7
Bangalore	28	55	22	2	2
Coochbehar	23	150		350	8
Amritsar	20	90	15		7
WestDinajpur	24	148	30	348	8
Rampur	21	88	11	15	6
Arcot	23	150		352	8
Ambala	21	90	16		6
Malda	23	150	25	351	8
Jalandar	19	90	14	17	7
Krishna	23	151		353	8
Panipat	21	91	17		7
Hugli	22	152	12	350	8
Ludhiana	20	88	13	20	6
Nellore	24	142		343	8
Jaipur	20	86	16		6
Warangal	25	154		346	8
Udaipur	19	82	17		7
Coimbatore	28	55	23		
Tumkur	29	55	22	17	11
Selam	28	56	23		

Fig 1.6 output of IKNN search for crop yield analysis

Assume a use case scenario of Crop yield Analysis, With the sample dataset as given below.

We then develop a LαB index structure-using lattice and bucket structures and a lattice partition algorithm to solve IkNN search. Finally the lattice partition algorithm will give the k nearest neighbors to the given query object as the resulting candidate set.

The closest data object to the query is placed in the first position of the candidate data set. Therefore, we propose an Analysis technique to process IkNN search on a data set having both complete and incomplete data objects

VI. Conclusion and Future Work

The The *k* nearest neighbor (*k*NN) query plays an important role in many real-life applications including image recognition and location-based services. However, it is common that in most of the practical applications, *k*NN retrieval struggles with incomplete data. We study the problem of *incomplete k nearest neighbor* (I*k*NN) *search*, which focuses on the *k*NN query for incomplete data. In this paper we initially develop an efficient index, i.e., LαB index, using lattice/bucket structures and values of incomplete objects, for indexing incomplete data. Then, based on L B index, we propose LP algorithm to tackle I*k*NN retrieval efficiently, which utilizes two pruning heuristics, i.e., value pruning and partial distance pruning. For the usecase scenario discussed above in result section the results obtained are satisfactory.

Consequently, the exact algorithms for kNN queries on this database may not be able to return the answers immediately, since the query is costly on the high dimensional database. In view of this, we propose an approximate algorithm. Also, we aim to solve the IkNN search algorithm given that the query is incomplete i.e. few object values in the query is missing

As the future work, we intend to explore other interesting queries (e.g., reverse k nearest neighbor search) on incomplete data.

References

[1]. Nick Roussopoulos Stephen Kelly Fredrick Vincent: “Nearest Neighbor Queries”.
[2].Parisa Haghani, Sebastian Michel and Karl Aberer: “Evaluating Top-k Queries over Incomplete Data Streams”.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017
8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)
Organized by
Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

[3].Wei Cheng, Xiaoming Jin, Jian-Tao Sun, Xuemin Lin, Xiang Zhang, and Wei Wang: "Searching Dimension Incomplete Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No.3, March 2014.
[4].Guadalupe Canahuate, Michael Gibas, and Hakan Ferhatosmanoglu: "Indexing Incomplete Databases".
[5]. Yogitha. M kapse and Anthara Bhattacharya: "Searching dimension in incomplete database by using hybrid Indexing Method", International Journal of advanced research in Computer Science and Management Studies, Volume 3,Issue 6, June 2015.
[6] Cheng Lu, Shiji Song and Cheng Wu: " -Nearest Neighbor Intervals Based AP Clustering Algorithm for Large Incomplete Data", Hindawi Publishing Corporation, Mathematical Problems in Engineering, Volume 2015, Article ID 535932,
[7]. Beng Chin Ooi, Cheng Hian Goh and Kian-Lee Tan: "Fast High-Dimensional Data Search in Incomplete Databases".
[8]. Reynold Cheng, Lei Chen, Jinchuan Chen and Xike Xie: "Evaluating Probability Threshold k-Nearest-Neighbor Queries over Uncertain Data".