International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

Survey on Inclusive Analysis of Incomplete Datasets

Mrs. Baswaraju Swathi, Supriya P, Suma M

Information Science, Visvesvaraya Technological University, Bangalore, Karnataka, India

ABSTRACT

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. Incompleteness is the common problem that most of the datasets suffer from. The incompleteness refers to any missing or uncertain data in the datasets. The missing data exists due to failure of data transmission devices, accidental loss of data 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. Further, we explain and discuss the latest techniques used to improve the accuracy of such data retrieval. We then analyze and compare the results obtained, efficiency and performance of all the techniques discussed.

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

I. INTRODUCTION

Processing any dataset and efficiently analysing it is very important as it helps in making key business decisions of a business organisation. There are many data mining techniques to analyse the data sets and provide appropriate decision to be taken to increase the profit of any business organisation. But these data sets also include incomplete data sets which are often eliminated in the pre-processing techniques.

Incompleteness is a common problem that data sets suffer from. Here, incompleteness refers to the case where data has error or certain information is missing, and in this project we focus only on missing data. Missing data is very common in our day-to-day life as well as research. Users tend to skip certain fields when they fill out online forms; participants choose to skip questions on surveys, etc. Although we can simply perform all the analysis tasks based on complete data sets by removing all the incomplete data, the output might be inaccurate. Also a business organisation cannot afford to lose data set whose data is incomplete as all data collected are important to the organisation.

Therefore in this paper, we focus on incomplete data sets and effectively analyse them. Most commonly used algorithm to process a search on incomplete data sets is k-Nearest Neighbor algorithm. The k-Nearest Neighbor search algorithm returns the k objects closest to a query object q from a given object set S. Here, objects in S are multi-dimensional, and they can also be incomplete. Retrieving the data objects without eliminating the missing data is a fundamental problem in data mining and research on information retrieval. The recent efforts made on incomplete data includes the methodologies on indexing incomplete data, incomplete data, querying incomplete data, etc.
The organization of this document is as follows. In Section 2 (**Literature Survey**), various methods used in analysing incomplete data sets for different researches is being discussed. In Section 3 (Conclusion), the conclusion is discussed.

II. LITERATURE SURVEY

A. Nearest Neighbor Queries

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. Nearest Neighbor search done using R-tree is an extension of B-tree for higher dimensions. The decomposition used in R-trees is dynamic, driven by the spatial data objects and with split algorithms, if a region of an n-dimensional space includes dead space, no entry in the R-tree will be introduced. Leaf nodes of the R-tree that consists entries of the form (RECT, oid). These leaf nodes are used as a pointer to a data object. RECT is an n-dimensional Minimal Bounding Rectangle (MBR). It bounds to the corresponding object. A query point P and an object O are enclosed in its MBR. The paper uses two metrics are used for ordering the NN search. The first one is based on the minimum distance of the object O from P. The second metrics is based on the minimum of the maximum possible distance from P to a face of the MBR containing O. These bounds are used by the nearest Neighbor algorithm to order and efficiently prune the paths of the search space in an R-tree. The algorithm finds k Nearest Neighbors to a given query point where k is greater than zero with

- A sorted buffer of at most k current nearest Neighbors is needed
- The MBR [1] pruning is done according to the distance of the furthest nearest Neighbor in this buffer.

The minimum distance of a point P from a spatial object O, denoted by ||(P,O)|| is,

$$||(P, o)|| = min(\sum_{i=1}^{n} |p_i - x_i|^2,$$
$$\forall X = [x_1, \ldots, x_n] \in O).$$

Three strategies used in this paper to prune MBRs during the search is as follows:

- An MBR M with MINDIST (P, M) greater than the MINMAXDIST (P, M') of another MBR M' is discarded because it cannot contain NN. It is used in downward pruning.
- If the distance from P to an Object O that is greater than the MINMAXDIST (P, M) for an MBR, M is discarded as M contains an object O which is closer to P. This is used in Upward Pruning.
- Every MBR M with MINDIST (P, M) greater than the actual distance from P to a given object O is discarded because it cannot enclose an object nearer than O. this is used in Upward Pruning.

The k-NN algorithm is implemented and thoroughly tested and algorithm scales up with both the number of NN requested and with size of the data sets while experimented with real and synthetic data sets. 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.

B. Evaluating Top-k Queries over Incomplete Data Streams

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. Consider several scenarios where attributes of objects are incomplete like Internet service providers, sensors etc. The attributes of objects arrives in different streams at different time instants. Using different algorithms best (largest) score for each attribute is calculated and updated in top k list of objects.

Initially the proposed system continuously monitors aggregation queries over incomplete data streams in a sliding window model. Then perform dominance check to retain only those objects which are necessary for providing exact results. Later an algorithm is proposed that focus on the accuracy in the top k results and minimization of storage minimisation of storage consumptions.

Algorithms under Comparison

- SORTED LIST ALGOROTHM [2]: This is the algorithm based on multiple sorted lists, each list for each attribute of object.

Once the **<p:id; p:value(i); p:ti>** value is received from the i^{th} stream, **<p:id; p:value(i))** is inserted in the ith list which is sorted based on the value field.

When a new tuple**<p:id; p:value(i); p:ti>** arrives, p's new score is computed by random access to all other attribute lists, result list is then updated accordingly. If p's score is higher than the least score in this list, then p is inserted to the result view. Similarly whenever a tuple expires, the score of its corresponding object decreases. The result view is updated if this object was part of the result view.

- EARLY AGGREGATION ALGORITHM (EAA) [2]: It uses the interval dominance check to discard instances which are not part of the k dominance set. EAA provides exact results and retains the least number of objects

Interval Dominance: Given two object instances p_i and q_j we say p_i dominates q_j

$p^i \geq q^j$ iff $p^i.t > q^j.t$ and p^i:currentscore > q^j:bestscore.

The correlation appearance between different streams, which is usual in real world scenarios, to estimate best score in less optimistic way than considering the best. This method provides highly accurate results while reducing the number of retained objects dramatically. As a part of future work, exploring the design of efficient data structures tailored to maintaining the dominance set of dynamic data sets. Measure of interest in the experiment results:

- Memory Consumption: This paper report on the number of items retained for each of the algorithms under comparison as a measure of storage. Items are tuples in case of SLA, as we keep tuples separately without aggregating them. We ignore the kmax materialized aggregated results, as kmax << W. For EAA, an item is an aggregated object instance.
- We report on the precision, i.e., the number of relevant data points among returned top-k results as the effectiveness metric. The relevance is defined by the SLA method which keeps all valid tuples. Consider that SLA reports A as the set of top-k results, then this set is the ground truth. So if set B is returned as the top-k in another algorithm, this algorithm's precision is calculated as: precision =|A∩B|

The experiments in this paper provides highly accurate results while reducing the number of retained objects dramatically. 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

C. Searching Dimension Incomplete Databases

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. Missing dimension information poses great computational challenge, since all possible combinations of missing dimensions need to be examined when evaluating the similarity between the query and the data objects. This paper specifies the development of the lower and upper bounds of the probability that a data object is similar to the query. These bounds enable efficient filtering of irrelevant data objects without explicitly examining all missing dimension combinations. The author introduces a probability triangle inequality. It is employed to further prune the search space and speed up the query process. The proposed probabilistic framework and techniques can be applied to both whole and subsequence queries. Extensive experimental results on real-life data sets demonstrate the effectiveness and efficiency of this approach. There are three major steps in the approach for the whole sequence matching: 1) pruning with probability triangle inequality [3]; 2) pruning with probability lower and upper bounds [3]; and 3) naive probability verification [3].

Given X_{rv}, $\delta^2(Q_o, X_o)$ is a real value and $\delta^2(Q_l, X_l)$ is a random variable. We can determine the lower and upper bounds on the distance between Xo and Q as follows:

$$\delta_{LB_o}(Q, X_o) = \min_{|Q_o|=|X_o|} \delta(Q_o, X_o),$$

$$\delta_{UB_o}(Q, X_o) = \max_{|Q_o|=|X_o|} \delta(Q_o, X_o),$$

The process of the subsequence query problem can be divided as follows:

- Step 1. Firstly, tackle all the case, which matches the subsequence from the first element of the target dimension incomplete sequence.
- Step 2. Remove the first element of target dimension incomplete sequence and repeat Step 1; terminate until there is no element left in target sequence.

This approach achieves satisfactory performance in querying dimension incomplete data for both whole sequence matching and subsequence matching; both the probability triangle inequality and the probability bounds have a good pruning power and improve query efficiency significantly.

This paper provides future scope of developing index structures to utilize probability triangle inequality to further improve the efficiency of the query process. Also wide range of distance functions can be incorporated to extend query strategy.

D. Indexing Incomplete Databases

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 [4] and quantitation [4] to correctly and efficiently answer queries in the presence of missing data. The performance of Bitmap indexes and quantization based indexes is evaluated and compared over variety of analysis parameters for real and synthetic data sets. The goal of this paper is to provide techniques that access databases efficiently in the presence of missing data and we make use of indexing techniques that exhibit better performance than existing techniques and sequential scan when the database attributes that are specified in a search key have missing data.

Using equality encoded bitmaps, bit Bi,j [x]is 1 if recorded x has value j for attribute Ai and 0 otherwise, Using this encoding, if if Bi,j [x] = 1 then Bi,k[x] = 0 for all k _= j.

$$\begin{cases} (\bigcup_{j=v_1}^{v_2} B_{i,j}) \vee B_{i,0} & \text{if} \quad \mathrm{v}_2 - \mathrm{v}_1 \le \lfloor \mathrm{C_i}/2 \rfloor \\ \bigcup_{j=1}^{v_1-1} B_{i,j} \vee \bigcup_{j=v_2+1}^{C_i} B_{i,j} & otherwise \end{cases}$$

$$\begin{cases} (\bigcup_{j=v_1}^{v_2} B_{i,j}) & \text{if} \quad \mathrm{v}_2 - \mathrm{v}_1 \le \lfloor \mathrm{C_i}/2 \rfloor \\ \bigcup_{j=1}^{v_1-1} B_{i,j} \vee \bigcup_{j=v_2+1}^{C_i} B_{i,j} \oplus B_{i,0} & otherwise \end{cases}$$

Interval Evaluation for Bitmap Equality Encoding [4]

With equality encoded bitmaps a point query is executed by ANDing together the bit vectors corresponding to the values specified in the search key Bitmap Equality are optimal for point queries. However with missing data when missing data means a query match we need to use two bitmaps instead of one to answer the query. Range queries are executed by first ORing together all bit vectors specified by each range in the search key and then ANDing the answer together. If the query range for an attribute queried includes more than half the cardinality then we execute the query by taking the compliment of the ORed bitmaps.

The techniques presented in this paper are easy to apply the effective indexing of missing data. This techniques exhibit linear performance for query execution time with respect to database and query dimensionality. This is done by essentially indexing attributes independently. There are several areas in which the techniques proposed here could be improved .The biggest weakness of the range encoded bitmaps is the inability to compress them.

E. Searching Dimension in Incomplete Database by Using Hybrid Indexing Method

Incompleteness of dimension information is a common problem these days in most of the databases. Such databases includes web heterogonous databases, multi-relational databases, spatial and temporal databases and data integration. 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' [5] algorithm. Simultaneously missing ratio will be evaluated on the basis of standard measure such as, precision and recall. **Clustering:** Clustering is nothing but common technique in data mining to discover hidden patterns from massive datasets. With the development of privacy-maintaining data mining application. CLIHD can be expanded as "clustering incomplete high dimensional database". **Indexing:** Indexing is the process which representation of data in sequence format. Due to indexing result will show in minimum span of time. **Hybrid Indexing [5]:** Hybrid Index scheme is used to apply for indexing that provides identical identity to each of the entry in high-dimensional database. When simultaneously enter data in database, there may be possibility to lose data and their dimensions. But due to BR-tree, ids are provided to each tuple entry. The BR-tree is shown to be more efficient in supporting range queries and have lower insertion and storage costs.

Precision: Precision represent that at what percent of relevant data collect from retrieved data.

Precision (%) = (existing data/total data)*100
Recall: Recall represent that at what percent of retrieved data collect from relevant field.
Recall (%)=(missing data/total data)*10

Mosaic: Mosaic scheme introduced the process of forming single data set for single attribute. Due to this process data integrity will maintained. Storage cost will increase. So that user can search data easily according to query. Therefore, there are as many inverted indexes as that number of dimensions in the search key. **R+ Tree** [5]: In this method incomplete database value may replace by 0 or 1. So that it is easy to verify the value of is available or not.

The author, in this paper explains an approach for mitigating the problem of missing dimension information in the datasets. The performance of clustering of partial or random data is improved with the help of CLIHD (clustering incomplete high dimensional database) algorithm. The hybrid index is the combination of three method i.e. BR tree, MOSAIC and R+ tree method. This approach achieves satisfactory performance in querying incomplete data.

F. *K*-Nearest Neighbor Intervals Based AP Clustering Algorithm for Large Incomplete Data

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) [6] 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.

AP clustering for inter-valued data: Let the complete data set X = $\{x_{1,}x_{2,\ldots\ldots,}x_N\}$, where X_i€R, The goal of AP is to find optimal exemplar set X_N = $\{ x_{c1,}x_{c2,\ldots\ldots\ldots,}x_{cK} \}$ 1<K<N, by minimizing the clustering error function. AP algorithm takes each data point as the candidate exemplar and calculates the attractiveness information between sample points that is the similarity between any two sample points. Similarity is usually set as

$$\|\overline{x_\iota}, \overline{x_\jmath}\|_2^2 = \sum_{l=1}^{M} |x_{il}^{+} - x_{il}^{+}|^2 + \sum_{l=1}^{M} |x_{il}^{-}|^2 + \sum_{l=1}^{M} [\frac{|x_{il}^{+} - x_{il}^{+}| + |x_{il}^{-} - x_{il}^{-}|}{2}]^2 \qquad (1)$$

AP Algorithm for Incomplete Data Based on K-Nearest Neighbor Intervals (kNNI-AP) k-nearest Neighbor intervals (kNNI) of missing attributes are proposed.

Let X = $\{ x_1, x_2, \ldots\ldots\ldots, x_n\}$ be a multidimensional incomplete sample with some missing attribute values. For an incomplete $X_i = \{x_{i1}, x_{i2}, \ldots\ldots\ldots, x_{iM}\}$,the K-nearest neighbors should be found first. The distance between sample A and B can be obtained as follows.

$$\|x_a - x_b\|_2 = \sqrt{\sum_{j=1}^{M} d_j(x_{aj}, x_{bj})^2} \ *\frac{M}{\omega} \qquad (2)$$

Where,

$$d_j(x_{aj,}x_{bj}) = \begin{cases} 0, & (1 - m_{aj})(1 - m_{bj}) = 0 \\ d_n(x_{aj,}x_{bj}), & others \end{cases} \qquad (3)$$

$$d_N(x_{aj,}x_{bj}) = \frac{|x_{aj} - x_{bj}|}{\max(x_j) - \min(x_j)} \qquad (4)$$

In terms of misclassification ratio, KNNI-AP is always the best performer except for incomplete Wine dataset with 25% missing attributes. KNNI-AP [6] is better than the results of IPDS-AP and KNNI-FCM

The result of KNNI-AP algorithm is general, simple and appropriate for the AP clustering with incomplete data, the final clustering results depend on the choice of K and P for KNNI-AP.

In the future, the author focuses on the work that selects K and P with theoretical basis and the improvements on the similarity measurements of AP when the missing percentage is large, which will be helpful to extend KNNI-AP to solve clustering incomplete data with various missing percentage.

G. Fast High-Dimensional Data Search in Incomplete Databases

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.

Firstly, the paper defines a multi-dimensional index structure, called the Bit string-augmented R-tree (BR-tree) [7]. Let (x1,x2,….xk) be the search key corresponding to a tuple t. The paper introduces a bit string y1….yk as follows:

Yi=1 if xi is known
Yi = 0 otherwise.

Author define a mapping function f [7] on the search

In addition, a one-dimensional index is built on each dimension of the search key. It comprises a family of multiple one-dimensional one-attribute (MOSAIC) indexes [7]. The MOSAIC structure outperforms the BR-tree in retrieval time for point queries, as well as in range queries over incomplete databases for dimension-unrestricted data distributions. This index allows for rapid identification of the set of tuples having a given value in the dimension being indexed. The processing point and range queries are very straightforward in this method.

Consider a range query, say ([Xi, Yi], [X2, Y2], . . . , [Xk, Yk]). Then for any arbitrary dimension, say i, the set of tuples that can contribute to the final result are those indexed by values in the range [Xi, Yi] and those indexed by -1. This summarizes that only 2k sub queries need to be developed and constructed which is significantly smaller than the 2k sub queries that are needed if the bit string augmented multi-dimensional index had been used. Consider if the results obtained from a sub query i is given by Resi. The final result is given by the intersection of all tuples in all dimensions (Res1, Res2…Resk)

In comparison with a single k-dimensional index, this approach is less space efficient. In fact, it has been recognized that the approach may not be efficient for high-dimensional data search as it is to a single high-dimensional index. But, this may no longer be true for incomplete datasets, where the choice of a multi-dimensional index requires that a query is to be rewritten into a number of sub queries. Thus, it can be concluded that having a family of multiple one-dimension single-attribute indexes may is not a bad approach to solve the problem

The experiment results in this paper shows that the proposed schemes are effective in reducing the search time for a wide range of queries as compared to exhaustive search. The BR-tree index that is used in the paper is more efficient in supporting range queries and has lesser insertion and storage costs as compared to the MOSAIC structure explained. 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.

H. Evaluating Probability Threshold k-Nearest-Neighbor Queries over Uncertain Data

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. A user may not be concerned about the exact probability values. The author handles this query efficiently in three steps discussed later.

For S to be a query answer, the distance of any object oh (where oh /∈ S) from q must be greater than that of oi (where oi ∈ S). Now, at distance r, the pdf that object oi ∈ S has the k-th shortest distance from q is the product of the following factors:

- the pdf that oi has a distance of r from q, i.e., di(r);
- • the probability that all objects in S other than oi have shorter distances than r, i.e., Qoj∈S∧oj6=oi Dj (r); and
- the probability that objects in D − S have longer distances than r, i.e.,Qoh∈D−S (1 − Dh(r)).

The author then presents three methods to efficiently process a T-k-PNN query.

The first method, called k-*bound filtering*, effectively removes all objects that have no chance to be a query answer. The author calls the objects that arenot pruned by the k-bound filtering as the *candidate objects*.

After k-bound filtering, one has to still need consider the k-subsets of the candidate objects while k-bound filtering utilizes distance information or pruning, the PCS makes use of the probability information of uncertain data to remove unqualified k-subsets.

The third method, called *verification*, is useful for handling k-subsets that are not filtered by the previous two methods. Verification process determines whether a k-subset is a query answer, by making use of the uncertainty pdf of objects returned by k-bound filtering. The author uses two kinds of verification *lower-bound verification* and *upper-bound* verification. It quickly computes the lower and upper bounds of qualification probabilities of k-subsets. These bounds are then used to determine how the k-subset should be handled.

The proposed solution can be applied to uncertain data with arbitrary 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. CONCLUSION

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.

IV. 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”.

[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: “*K*-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”.