Missing Values Estimation for Skylines in Incomplete Database

Ali Alwan[1], Hamidah Ibrahim[2], NurIzura Udzir[2], and Fatimah Sidi[2]
[1]Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Malaysia
[2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia

Abstract: *Incompleteness of data is a common problem in many databases including web heterogeneous databases, multi-relational databases, spatial and temporal databases, and data integration. The incompleteness of data introduces challenges in processing queries as providing accurate results that best meet the query conditions over incomplete database is not a trivial task. Several techniques have been proposed to process queries in incomplete database. Some of these techniques retrieve the query results based on the existing values rather than estimating the missing values. Such techniques are undesirable in many cases as the dimensions with missing values might be the important dimensions of the user's query. Besides, the output is incomplete and might not satisfy the user preferences. In this paper we propose an approach that estimates missing values in skylines to guide users in selecting the most appropriate skylines from the several candidate skylines. The approach utilizes the concept of mining attribute correlations to generate an Approximate Functional Dependencies (AFDs) that captured the relationships between the dimensions. Besides, identify the strength of probability correlations to estimate the values. Then, the skylines with estimated values are ranked. By doing so, we ensure that the retrieved skylines are in the order of their estimated precision.*

Keywords: *Skyline Queries, Preference Queries, Incomplete Database, Query Processing, Estimating Missing Values.*

Received August 13, 2015; accepted November 29, 2015

1. Introduction

In many database applications there are many reasons that lead into missing values which make a database incomplete. These reasons include the following [2, 9, 33, 35, 36]:

1. Incomplete data entry: users may be intentionally or accidentally missed some values in one or more attributes (dimensions) when entering data into the database.
2. Inaccurate data from heterogeneous data sources: in most of the online database the data come from different remote resources such as sensor. As a result the received data may miss some values in one or more attributes (dimensions).
3. Integrating heterogeneous schemes: the mediator systems combine the attribute values of local schemas to obtain a universal global scheme. The process of data integration may lead to missing some values in one or more attributes (dimensions) as some attributes may not appear in all local schemas.

For many years there have been various research works focusing on the incomplete database issues including data representation and modeling [15, 17, 30, 32, 36], indexing [26], predicting missing data [4, 18, 19, 20, 21, 23, 35] and query processing [2, 8, 12, 22, 24, 29, 31, 33, 35, 36].

From the previous works it is obvious that more attention has been given to the query processing issues in incomplete database. This is due to the fact that the missing values in the database relation affect negatively on the query output and also lead to high processing cost in obtaining the desired results [12, 33].

In recent years, there has been much interest on a new type of queries named skyline queries. Skyline queries prefer a data item p over the other data item q if and only if p is better than q in all dimensions and not worse than q in at least one dimension. The skyline queries are significant and mostly used in various application domains, like multi-criteria decision making applications [10, 11, 37], decision support system and recommender system, where these systems combine various interests to help users to recommending a strategic decision. Furthermore, e-commerce environment is also a significant area that involves skyline queries. For example, helping customer to make a tradeoff between the price, quality, and efficiency of the products to be purchased.

Several approaches which applied the concept of skyline have been proposed. These include Divide-and-Conquer (D&C), Block Nested Loop (BNL) [7], Bitmap and Index [34], Sort Filter Skyline (SFS) [16], Nearest Neighbor (NN) [25], Linear Elimination Sort Sykline (LESS) [14], Branch and Bound Skyline

(BBS) [28], and Sort and Limit Skyline Algorithm (SaLSA) [3]. These approaches attempt to reduce the searching space and the processing cost by terminating the process of finding skylines as early as possible. Most of the previous techniques assumed that all values of attributes (dimensions) are available and complete for all data items of the database [10, 11, 13, 14, 16, 25, 28, 34, 37]. However, this assumption is not always vaild in the real world database particularly for large databases with high number of dimensions as some values may be missing [5, 6, 24]. The missing values will influence negatively on the process of finding skylines leading to high overhead, due to exhaustive comparison between the data items. Besides, the incompleteness of data leads to lose the transitivity property of skyline technique which is maintained on all existing skyline techniques. This further leads to cyclic dominance between the data items as some data items are incomparable with each other and thus no data item is considered as skyline [24].

As an example, a tourist seeking for a hotel in a specific area that is near to a beach, has high rate, and at the same time cheap in price. Among the set of available hotels, skyline queries would retrieve only those non-dominated hotels that meet the tourist's preferences. Assumed the following three hotels are return as a result of skyline query, h_1(?, 5, 7), h_2 (3, ?, 7), and h_3(6, 9, 3). The symbol (?) is used to represent missing value in the data items. Based on the existing values user is unable to know how far is the hotel h_1 from the beach as the value is unknown and it might not be the best choice. Similarly the user is also unable to know the rate of hotel, h_2, thus h_2 might not be the best choice for the user. Therefore, estimating missing values in the skyline is an issue that needs to be tackled.

In this paper we propose an approach for estimating the missing values of the skylines in incomplete database. The approach utilizes the concept of mining attribute correlations to generate an Approximate Functional Dependencies (AFD) that captures the relationships between the dimensions. Besides, the strength of probably correlations is identified in order to estimate the missing values. Then, the skylines are ranked according to the strength of the generated AFD and the strength of probability correlations. By doing so, we ensure that the retrieved skylines with estimated values are in the order of their estimated precision. The approach consists of four phases, namely: generating AFDs, identifying the strength of probability correlations, imputing the missing values, and ranking the final skylines.

This paper is organized as follows. The previous works related to this work are presented and discussed in section 2. Section 3 gives the basic definitions and notations, which are used throughout the paper. In section 4, we describe our proposed approach for estimating missing values of skylines in incomplete database which has four main phases. Examples are also given to clarify the phases. Section 5 presents a discussion on the strength of the proposed approach. Conclusion is presented in the final section 6.

2. Related Works

Incomplete database or incomplete information becomes a common problem in different real life large database applications. In this context, query processing is challenging as in many cases there is a significant part of the query answer that may be neglected due to the missing values in some dimensions. For that reason several approaches have been proposed tackling the problem of traditional query processing in incomplete database [2, 9, 12, 35, 36]. For many years, there has been much interest in the issues of predicting missing values in databases [2, 26, 27, 30, 33]. The main reason behind this emphasizing is due to missing values influence negatively on the quality of the database as more missing values lead to high rate of bias data. Most importantly, the quality of output of any operation on a database is highly determined by the quality of the given data in the database.

Developing a suitable technique to manipulate an incomplete database is crucial as many real databases have missing attribute values and estimating them requires preprocessing before any operation on the database can be performed. There are three different approaches that can be utilized in processing queries in incomplete database. The first approach retrieves exclusively the most relevant data items with no missing values. This approach is insufficient as many significant data items with missing values are omitted from the final query answer. Ignoring the data items with missing values results into serious reduction in the output size and gives greater possibility of a non significant and inaccurate results when the number of missing data is high [26, 27]. This approach is called retrieve complete data only. The second approach considered the whole data items in the database and retrieved the most relevant complete data items and the whole incomplete data items as well. This approach has significant drawback as some incomplete data items that are retrieved in the query answer might not be relevant to the user. Thus, the size of the query answer is increased significantly without any benefits to the user. This approach is named retrieve all.

The third approach eliminates the problems in the first and second approaches by considering the most relevant complete data items in the database and the most relevant incomplete data items as well. This approach consists of three phases. The first phase retrieves all data items with missing values relevant and irrelevant. Then, the second phase imputes the missing values for every data items. Finally, depending on the estimated values, the third phase determines

which data items are further considered to be in the query answer and which are to be neglected. This approach is called retrieve all relevant and is considered the most appropriate one for the query processing in incomplete database context [1].

In this section we highlight on the prediction methods of handling incomplete data. We classify the prediction methods into two main categories, namely: statistical methods and machine learning methods. Figure 1 illustrates the methods of processing queries in incomplete database.

2.1. Statistical Methods

Many of the prediction approaches that have been proposed are based on statistical methods. The primary aim of this method is preserving the distribution of the whole data by avoiding bias on the data distribution. The predicted values may benefit user at the database level but not at the data item level. In many cases users are more concern on the individual value of the attribute rather than the whole data. Thus, these statistical methods may not be appropriate for preference queries. The statistical methods are as follows [2, 17, 18, 26, 27].

1. Single Imputation (SI) in this method, the missing values for a given attribute are substituted by a single value. The methods that involve single imputation include mean, median, mode, maximum value in the attribute range, minimum value in the attribute range, k-Nearest Neighbor (kNN), and Expectation Maximization (EM).
2. Multiple Imputations (MI) in this method, more than one estimated values are derived in order to fill in the missing values. The computation cost of this method is higher than SI, but is more accurate and productive. This method also needs an appropriate mechanism to reflect the uncertainty of missing data and precisely estimate the missing values.

2. Machine Learning Methods

Machine Learning (ML) methods learn prediction models to handle the database with missing values based on the complete database instances. Some of the ML methods involve probabilistic approach to predict the missing values. ML methods are nonparametric as they do not rely on data distribution during the training or testing process. These methods are also called parameter-free methods or distribution free methods. There are several approaches that have been proposed which belong to the ML methods which include C4.5, auto class, fractioning cases, and CN2 [1, 18, 26, 30, 33]. Figure 1 illustrates the methods of processing queries in incomplete database.

However, preference queries have not received much attention in incomplete database applications in which to evaluate the query, exhaustive comparisons need to be performed in order to determine the relevant data items in the database. Preference queries in incomplete database are fundamentally different from the conventional preference queries in complete database because the transitivity property of preference techniques such as top-k, skyline is no longer hold. Furthermore, the retrieved data items might contain some missing values in one or more dimensions that do not benefit the users. In the following we present and discuss the preference queries proposed by previous researches in incomplete databases. The focus given by the researchers in this environment is highlighted.

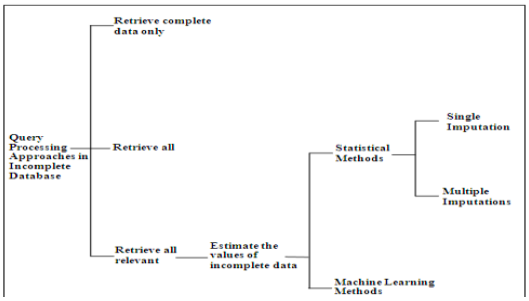


Figure 1. Methods of processing incomplete data.

The first work that tackled the issue of skyline queries in incomplete database is contributed by Khalefa *et al.* [24]. They proposed Iskyline algorithm for processing skyline queries in incomplete data. Iskyline algorithm attempts to divide the initial database into distinct nodes depending on the missing values of dimensions and then applying the conventional skyline technique to retrieve the local skylines of every node.

The work in [22] studied the issue of top-k queries over continuous streaming uncertain and incomplete database systems. In their work two algorithms have been introduced to retrieve the top-*k* query answers, namely: Sorted List Algorithm (SLA) and Early Aggregation Algorithm (EAA). SLA maintains the valid data items to be in FIFO fashion that remove the expired data items from the list before applying top-k process.

Recently, Bharuka and Kumar [5] proposed the Sort based Incomplete Data Skyline (SIDS) algorithm to derive skylines for incomplete data. The input data to the algorithm is assumed to be pre sorted in non increasing order. SIDS employs the round robin fashion in reading the values of the dimensions and determining the next best value in that dimension to be chosen for processing.

To the best of our knowledge the most recent work that tackled the issue of skyline queries in incomplete database is contributed by Bharuka and Kumar [6]. They have designed an approach utilizing top-k frequent skyline technique that tailored to handle skyline queries for complete data. The idea of the proposed approach is identifying the superior skylines based on the relative fractional skyline frequency of

the data item.

However, the previous mentioned works only retrieves the query results according to the current state of the database without estimating the missing values. In other words, the outputs that are derived from the database have some missing values in one or more dimensions. This is undesirable in many cases as the output is incomplete and might not satisfy the user preferences.

The work in [33] addresses the issue of ranking top-k queries in uncertain and incomplete database by introducing a probabilistic model that works under partial orders concept. Under the proposed model several algorithms have been described with the aim of pruning the searching space and filtering the database to efficiently process the query. Furthermore, statistical approach is proposed to handle the incomplete data by imputing the missing values. Lastly, a sampling technique is proposed by employing Markov Chain Monte Carlo (MCMC) method to compute approximate query answers.

3. Preliminaries

In this section, we provide some definitions and notations that are related to skyline queries in incomplete database which are necessary to clarify our proposed approach. Our approach has been developed in the context of incomplete multidimensional relational database, *D*. A relation of the database *D* is denoted by $R(d_1, d_2, \ldots, d_m)$ where *R* is the name of the relation with *m*-arity and $d = \{d_1, d_2, \ldots, d_m\}$ is the set of dimensions.

- Skyline: skyline technique retrieves the skyline, *S*, in a way such that any skyline in *S* is not dominated by any other data items in the dataset.
- Dominate: given two data items p_i and $p_j \in D$ dataset with *d* dimensions, p_i dominates p_j (the greater is better) (denoted by $p_i \succ p_j$) if and only if the following condition holds:
  $p_i.d_k \geq p_j.d_k \wedge \exists d_l \in d, p_i.d_l > p_j.d_l$
- Skyline queries: select a data item p_i from the set of dataset *D* if and only if p_i is as good as p_j (where $i \neq j$) in all dimensions (attributes) and strictly in at least one dimension (attribute). We use $S_{skyline}$ to denote the set of skyline data items, $S_{skyline} = \{p_i \mid \forall p_i, p_j \in D, p_i \succ p_j\}$.
- Incomplete Database: *D* is said to be incomplete if and only if it contains a data item *pi* with missing values in one or more dimensions d_i (attributes); otherwise, it is complete.
- AFD: Given a relation *R*, a subset *X* of its dimensions, and a single dimension d_i of *R*, we say that there is an AFD between *X* and d_i, denoted by $X \; d_i$, if the corresponding functional dependency $X \; d_i$ holds on small fraction of the tuples of *R*. The set of dimensions *X* is called a determining *set* of d_i denoted by DtrSet (d_i).
- θ: is the ratio of the number of data items that needs to be removed from a relation *R* to consider the AFD as a functional dependency. Let *S* be the total number of data items in a relation *R* and let S' be the number of data items that is used to generate the AFD. Let Count= (S-S'), thus, θ=Count / *S*.

4. The Phases of the Proposed Approach

We have proposed an approach for estimating the missing values of skylines in incomplete database. The approach attempts to provide approximate values for the dimension with missing values in the skylines.

4.1. Generating Approximate Functional Dependencies

This section explains the process of analyzing the existing values of skylines in order to identify the relationships between the dimensions of the relation and generate the approximate functional dependency among the dimensions. Since, the size of the skylines is highly affected by the database size and the number of dimensions, therefore, our approach emphasizes on using the derived skylines for estimating the missing values in a database. Analyzing the skylines allows us to capture the relationships between dimensions and determine how the value in one dimension affects the values of the other dimensions. This is conducted by referring to the derived skylines to generate the AFDs that guides us in defining the relationship between dimensions.

Functional dependency is one of the explicit types of attribute correlation, therefore, if there are some known correlations between dimensions it can be added to the deterministic set, which is denoted as DtrSet (*d*). For example, usually there is a correlation between the area and rent dimensions (*area* $\longrightarrow$ *rent*) which holds in the apartments database.

To generate the AFD, we first divide the skylines with missing values into two sets. The first set contains all skylines that have missing values in the identical dimension that needs to be estimated, and the second set contains the remaining skylines. The skyline values of the second set are analyzed to capture the relationships between the dimensions of the relation. Then the AFD is generated that helped us in specifying how the value of one dimension influences the values of the other dimensions. Then to ensure that the derived AFD reflects the relationships between dimensions, we compute the confidence of the generated AFD based on the following formula [1].

$$Conf(\text{AFD}) = 1 - \theta(\text{AFD}) \quad (1)$$

Where θ represents the ratio of the number of data items that needs to be removed from the relation *R* to consider θ as an approximate functional dependency.

The decision on accepting the generated AFD is subject to the value of the Conf (AFD) whether it is greater than or equal to a specified threshold value. The value of the threshold can be controlled and in our experiments we have specified it to be 50%. The lower value of θ reflects that there are relationships between the dimensions and the AFD can be derived. From the value of θ we can determine whether there are inherent relationships between two dimensions and the value in one dimension determines the value of the other dimension or not.

It is not always obvious that there are predefined functional dependencies that reflect the correlations between dimensions particularly for autonomous databases. Thus, generating the AFD helps us in capturing the implicit correlations between dimensions to estimate the values in the dimensions with missing values. Algorithm 1 depicts the details steps for generating the approximate functional dependency in our proposed approach. Here, d_k is the dimension to be estimated while d_l is the dimension used to estimate the value of d_k. We first read a data item from the set of skylines (step 2). If $a_i.d_k$ is a missing value and $a_i.d_l$ is a non-missing value (step 5) then the data item a_i is inserted into set, $S1$ (step 6). Else, if both $a_i.d_k$ and $a_i.d_l$ are non missing values (step 8) then the data item a_i is inserted into set, S2 (step 9). The above steps are repeated until all data items in *Sky* are examined. By doing so, we ensure that the data items with missing value in the dimension d_k are gathered in the same set. While the remaining data items are gathered in another set. The number of data items in set $S2$ is calculated (step 11). Each data item a_i of the set $S1$ is analyzed (step 13). For each data item a_j of the set $S2$ (step 14), if the value of the dimension d_l of a_i and a_j is similar (step 16), then the value of Count is increased by 1 (step 17). The steps (13-17) are repeated until all the data items in $S2$ are scanned. Then, the difference between the total number of data items of set $S2$ and the number of data items that have similar values in the dimension d_l is computed (step 19). The ratio of the data items that have similar value in the dimension d_l to the number of data items in set $S2$ is calculated, and the degree of confidence is computed as well (step 21). If the confidence value is greater than the given threshold value (step 22), then the AFD based on the dimension d_k is generated (step 23).

Algorithm 1: Generating approximate functional dependency.

Input: A set of skylines with missing values, Sky, a threshold value,
T, the dimension to be estimated d_k, the dimension that is used to estimate d_k, say d_l

Output: AFD (If any)
1. *BEGIN*
2. *FOR each a_i in Sky DO*
3. *BEGIN*
4. *IF $a_i.d_k$ is a missing value and $a_i.d_l$ is a non-missing value THEN*
5. *Insert a_i into set, S1*
6. *ELSE*
7. *IF both $a_i.d_k$ and $a_i.d_l$ are non-missing values THEN*
8. *Insert a_i into set, S2*
9. *END*
10. *Total = no. of data items in S2*
11. *Count = 0*
12. *FOR each a_i in S1 DO*
13. *FOR each a_j in S2 DO*
14. *BEGIN*
15. *IF $a_i.d_l == a_j.d_l$ THEN*
16. *Count = Count + 1*
17. *END*
18. *Diff = Total - Count*
19. *Ratio = (Diff / Total)*
20. *Conf = 1 - Ratio*
21. *IF Conf (AFD) >= T THEN*
22. *Generate AFD based on d_k*
23. *END*

We clarify how the AFD is constructed using the following example. Most of the time there is an implicit relationship between the number of rooms in a house and the monthly rental. Thus, an AFD *no.of.rooms* $\rightarrow$ *rent* can be constructed, which indicates that the number of rooms in the house most of the time determines the *rent* rate. The *no.of.rooms* dimension is called deterministic dimension that determines the values of the other dimension *rent*. We can conclude that there is a relationship between the *no.of.rooms* dimension and the *rent* dimension. According to the existing values of the *no.of.rooms* and the *rent*, the confidence, Conf AFD, between these two dimensions is computed and the AFD is generated to estimate the missing values of the *rent* dimension.

4.2. Identifying the Strength of Probability Correlations

The second phase aims at identifying the strength of correlations between two dimensions by computing the strength of probability correlations between the dimensions. That means, after generating the AFD between dimensions we have to determine the strength of correlations between the two dimensions, for instance d_i and d_j, and to which extent the value of d_i influences the value of d_i. The strength of probability correlations between d_i and d_j is denoted as $P(d_i, d_j)$ and is formulated as follows [36]:

$$P(d_i, d_j) = \frac{|d_j|}{|d_j, d_i|} \quad (2)$$

where |,| refers to the number of distinct values. The value of $P(d_i, d_j)$ indicates the strength that every distinct value in d_j is associated with a unique value in d_i.

Let $S(d_i, d_j)$ denotes the correlation between the dimensions d_i and d_j based on the non-missing values in both dimensions and d_i is the dimension of interest which contain missing values. For every data item a_m, where the value of d_i is missing and the value of d_j is

not, the value of $a_m.d_i$ can be estimated using the existing values in the skylines where d_i exists. By doing so, we obtained a number of estimated values of d_i along with their relative frequencies. Then, based on the highest frequency value, the missing value of d_i can be estimated. In case if there are other correlations such as $P(d_i, d_k)$ we repeat the process to obtain the estimated values for $a_m.d_i$ with their relative frequencies.

Algorithm 2 shows the detail steps of identifying the strength of probability correlations. We first analyzed each generated AFD between the dimensions d_i and d_j in the set of AFDs (step 2). Then, the strength of probability correlations between the dimensions d_i and d_j is computed based on the given formula in Equation (2) (step 3). This process is repeated for each generated AFD.

Algorithm 2: Identifying the strength of probability correlations.

Input: A set of AFDs, AFDs = {S_1, S_2, …, S_n}
Output: The strength of probability correlations, $P(d_i, d_j)$
1. BEGIN
2. FOR each $S_i(d_i, d_j)$ in AFDs DO
3. $P(d_i, d_j) = | d_j|/| d_j, d_i|$
4. END

4.3. Imputing the Missing Values

This phase attempts to impute the missing values of the dimensions in the skylines with the estimated values. This is simply achieved by referring to the dimensions which have missing values and replaced their values with the estimated values. In this process there might be many estimated values for a dimension that need to be considered. Selecting the appropriate estimated value from several alternative values is based on the frequency of the value. The following example explains the process of estimating the missing values where one AFD is generated. Assume a house with the following details a = (*no.of.rooms* = 4, *area* = 700, *rent* = null). Assume that the no.of.rooms dimension has correlation with the *rent* dimension based on the generated AFD and its strength is 0.80. By analyzing the derived skylines with *no.of.rooms* = 4 we obtained the following estimated values for the *rent* with their frequencies (1000, 0.25), (1200, 30), (1300, 0.45). Based on this fact, the value 1300 is selected as the estimated value for the *rent* dimension as it has the highest frequent value when the value of *no.of.rooms* is 4 and the area is 700.

However, in some cases there might be more than one AFD that can be generated between the dimensions. In this case, the results of the AFDs are combined to estimate the missing value. For example, assume a house with the following details a = (*no.of.rooms* = 6, *area* = 700, *rent* = null). Assume the no.of.rooms dimension has correlation with the *rent* dimension based on the generated AFD and its strength is 0.85, while the *area* dimension and the *rent* dimension have correlations based on the generated AFD and its strength is 0.70. By analyzing the derived skylines with *no.of.rooms* = 6 we obtained the following estimated values for the *rent* with their frequencies (1200, 0.45), (1400, 0.55). Similarly, by analyzing the derived skylines for the *area* = 700 the *rent* has the following values with their relative frequencies (1000, 0.25), (1200, 0.25), (1400, 50). Combining the obtained results of both dimensions to introduce an overall a.rent with their strength probability {(1000, 0.175), (1200, 0.5575), (1400, 0.8175)}, where for example the pair (1400, 0.8175) is obtained by adding the results of multiplying the frequency of the pairs (1400, 0.55) and (1400, 0.50), with the probability strength 0.85 and 0.70, respectively. This pair is calculated as follow, (0.55*0.85) + (0.5*0.70) = 0.8175. Then the value 1400 is considered as the estimated value for the rent dimension when the number of rooms in the house is 6 as it has the highest frequency.

4.4. Ranking the Final Skylines

This section presents the last phase of the proposed approach for estimating missing values of skylines in incomplete database. This phase emphasizes at ranking the skylines in a way such that the skylines with the estimated values that have the highest value of strength of probability correlations are placed at the top of the skyline set. This is to ensure that the skylines are retrieved in the order of their precision to help users in selecting the most appropriate skyline. In addition, some skylines might have one dimension with estimated values, while other skylines might have estimated values in more than one dimension. Hence, the skylines which have small number of dimensions with estimated values are listed prior to the skylines which have large number of dimensions with estimated values. Besides, if a skyline has more than one generated AFDs which results into more than one value of the strength of probability correlations, then the strength probability correlations with the highest value is selected and used in the comparison against the other skylines. For example, assuming a skyline, $s1$ has two AFDs, AFD1 between dimensions $d1$ and $d3$ with $P(d1, d3)$ = 0.65 while AFD2 between dimensions $d2$ and $d4$ with $P(d2, d4)$ = 0.80. Assumed another skyline, $s2$ has one AFD only between dimensions $d1$ and $d3$ which results into the following strength of probability correlations, $P(d1, d3)$ = 0.65. Then, $s1$ is compared to $s2$ with respect to the value of the strength of probability correlations $P(d2, d4)$ = 0.80. Therefore, $s1$ is listed as the first choice to the user while $s2$ comes later.

Algorithm 3 illustrates the detail steps of ranking the final skylines. Each data item in the set of skylines, *Sky*, is analyzed (step 2). The highest value of the

strength of probability correlations, p_1 (step 3) and the number of dimensions with estimated values, m_1 (step 4) of a_i are determined. For each data item a_j (step 5), the highest value of the strength of probability correlations, p_2 (step 8), and the number of dimensions with estimated values, m_2 (step 9) of a_j are determined as well. If p_1 is less than p_2 and the number of dimensions with estimated values of a_i is greater than a_j (step 10), then swap a_i with a_j (step 11).

Algorithm 3: Ranking the final skylines algorithm.

Input: A set of skylines, Sky, and the values of the strength of probability correlations, P

Output: A set of ranked skylines
1. BEGIN
2. FOR each a_i in Sky DO
3. Let p_1 = the highest value of P of a_i
4. Let m_1 = the number of dimensions with estimated values of a_i
5. FOR each a_j in Sky DO
6. BEGIN
7. IF i <> j THEN
8. Let p_2 = the highest value of P of a_j
9. Let m_2 = the number of dimensions with estimated values of a_j
10. IF $p_1 < p_2$ AND $m_1 >= m_2$ THEN
11. Swap (a_i, a_j)
12. END
13.END

5. Experiment Setting

To evaluate the accuracy of our proposed approach for estimating the missing values of skylines, we have compared between the real and the estimated values. Let a_i represents the actual value while e_i is the estimated value in the derived skyline. Then, the relative error, re_i, can be calculated as follows [22]:

$$re_i = \frac{1}{a_i} \cdot |a_i - e_i| \quad (3)$$

The above formula measures the relative error between the estimated value and the real value of a skyline. In the experiment, the final relative error is computed as below:

$$\frac{\sum_{j=1}^{m} re_i}{m} \quad (4)$$

Where m is the number of estimated values.

The approaches were implemented using VB.NET 2010 programming language. Comprehensive experiments have been carried out on Pentium V 2Duo 2.4GHz PC with 4GB memory and Windows XP service pack 3 platform. The experiment aims at investigating to what extent the estimated value is near to the real value. For our experiments we have decided to use two different types of dataset, namely: synthetic and real datasets. For the synthetic dataset we have built a generator that randomly generates a set of uniformly and independent data. In addition, three multidimensional real datasets have been used in the experiments to illustrate the benefits of our proposed approach. The first real dataset is the NBA statistics, the second real data set is the CoIL 2000 Insurance Company and the third real dataset is Movie Lens dataset. These real datasets have been used in most previous works related to the area of preference queries and particularly in skyline queries [3, 10, 11, 24, 28, 37]. Table 1 summarized the range of parameter values for the datasets.

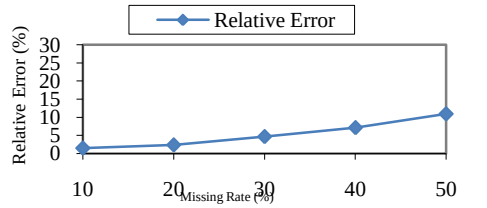
Table 1. The parameters setting of the real and synthetic datasets for incomplete database.

Parameter Settings	Dataset Name			
	Synthetic	**NBA**	**CoIL 2000**	**MovieLens**
No. of dimensions (*d*)	9	9	21	3
Missing Rate (%)	10%, 20%, 30%, 40%, 50%	10%, 20%, 30%, 40%, 50%	10%, 20%, 30%, 40%, 50%	10%, 20%, 30%, 40%, 50%
Dataset size (K)	300K	100K	250K	1200K

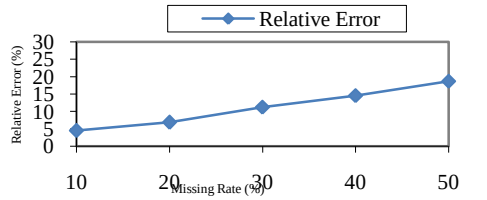
5.1. Experiment Results

In this section we highlight the experiment results on the synthetic and real datasets. The experiments emphasize on measuring the relative error between the real and the estimated values of the skylines. The relative error is computed as follow, we first find the relative error for every dimension with missing value in the skyline. Then, the average of the relative errors of the whole skylines is computed. In the experiments we have investigated the effect of the missing rate on the accuracy of the estimated values of the skylines, thus, the missing rate is varied in the range of 10%-50% for every dataset considered in this research. Figure 2-a demonstrates the results for the synthetic dataset by varying the missing rate and fixing the number of dimensions and the dataset size. From the results it can be seen that when the missing rate is 10% the relative error is almost 2%, while when the missing rate is 20% the relative error between the real and the estimated values is almost 3%. The value of relative error is continuously increased with respect to the missing rate in the dataset and it reached up to 10% when the missing rate is 50%. From the results it can be concluded that the relative error is quite low for the synthetic dataset. This is because the ranges of the values for the synthetic dataset fall in the range of 0-9, and the number of dimensions is large which results into identifying large number of skylines. Hence, the process of estimating the missing values achieved low relative error rate. Figure 2-b illustrates the experiment results on the NBA dataset that shows the relative error between the real and the estimated values. Also, in this experiment the dataset size is fixed to 100K and the number of dimensions is fixed to 9, and the missing rate is varied from 10%-50%. From the figure it is clear that the relative error reached up to 5% when the missing rate is 10%, and the relative error raised up to

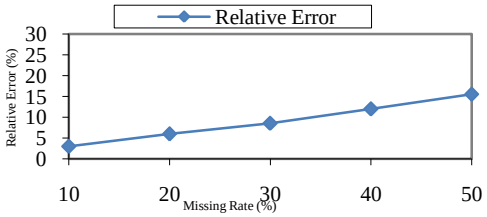
18% when the missing rate is 50%. From this experiment it can be concluded that our approach achieved reasonable rate in the relative error. This is because the NBA dataset has wide range of values for most of the dimensions and the number of dimensions is large.



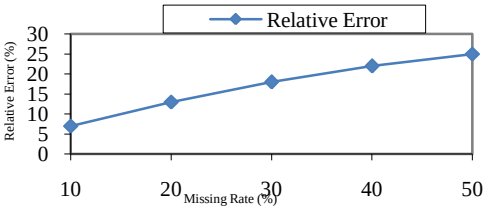
a) Synthetic



b) NBA.



c) CoIL Insurance Company.



d) MovieLens.

Figure 2. The effect of the missing rate on the relative error.

Figure 2-c depicts the results of the relative error on CoIL 2000 for an insurance company. In this experiment we have tested our proposed approach for estimating missing values of the skylines by varying the missing rate and fixing both the dataset size and the number of dimensions. From the experiment results it can be concluded that the relative error rate is 4% when the missing rate is 10% and the relative error continuously raised and reached up to 15% when the missing rate is 50%. Our proposed approach achieved good results in estimating the missing values on CoIL 2000 Insurance dataset. This is because CoIL 2000 Insurance dataset has high number of dimensions that results into large number of identified skylines. Hence, the values of the identified skylines can be employed to estimate the missing values. Finally, Figure 2-d demonstrates the experiment results for estimating the missing values of skylines on the Movie Lens dataset. In this experiment both the dataset size and the number of dimensions are fixed, while the missing rate is varied in the range of 10%-50%. The relative error is almost 7% when the missing rate is 10%, and the ratio of the relative error constantly increased and reached up to 23% when the missing rate reached up to 50%.

From the experiment results it can be concluded that our approach is scalable and achieved less than 25% relative error. Besides, increasing the missing rate has influence on the accuracy of the proposed approach. We also observed that the number of identified skylines has high impact on the relative error. That means, when the number of identified skylines is huge, this leads to generate more AFDs that captured the relationships between the dimensions. Furthermore, the large number of skylines also leads to high value of the strength of probability correlations between the dimensions. The AFDs and the value of strength of probability correlations highly influence on the estimated values and impact on the relative error rate.

6. Conclusions

In this paper, we have presented and explained the process of estimating missing values of skylines in incomplete database. The approach of estimating missing values of skylines presented in this paper consists of four phases, namely: generating AFDs, identifying the strength of probability correlations, imputing the missing values, and ranking the final skylines. The approach attempts to estimate values for the dimensions with missing values in the skylines. To the best of our knowledge, our approach is the first attempts in estimating missing values in the skylines. Thus, the approach is capable of deriving skylines with no missing values in incomplete database systems providing users with complete query answers.

References

[1] Antova L., Koch C., and Olteanu D., "From Complete to Incomplete Information and Back," *in Proceedings of the ACM SIGMOD International Conference on Management of Data*, Beijing, pp. 713-724, 2007.

[2] Antova L., Koch C., and Olteanu D., "$10^{(10)6}$ Worlds and Beyond: Efficient Representation and Processing of Incomplete Information," *The International Journal on Very Large Data Bases*, vol. 18, no. 5, pp. 1021-1040, 2009.

[3] Bartolini I., Ciaccia P., and Patella M., "SaLSa: Computing the Skyline Without Scanning the Whole Sky," *in Proceedings of the 15th International Conference on Information and Knowledge Managemen*, Arlington, pp. 405-414, 2006.

[4] Batista G. and Monard M., "An Analysis of Four Missing Data Treatment Methods for Supervised

Learning," *Applied Artificial Intelligence Journal*, vol. 17, no. 5-6, pp. 519-533, 2003.

[5] Bharuka R. and Kumar P., "Finding Skylines for Incomplete Data," *in Proceedings of the Twenty-Fourth Australian Database Conference*, Adelaide, pp.109- 117, 2013.

[6] Bharuka R. and Kumar P., "Finding Superior Skyline Points from Incomplete Data," *Proceedings of the 19th International Conference on Management of Data*, Ahmadabad, pp. 35-44, 2013.

[7] Börzsönyi S., Kossmann D., and Stocker K., "The Skyline Operator," *in Proceedings of the 17th International Conference on Data Engineering*, Cancun, pp. 421-430, 2001.

[8] Bruyère V., Decan A., and Wijsen J., "On First-Order Query Rewriting for Incomplete Database Histories," *in Proceedings of the 16th International Symposium on Temporal Representation and Reasoning*, Bressanone-Brixen, pp. 54-61, 2009.

[9] Canahuate G., Gibas M., and Ferhatosmanoglu H., "Indexing Incomplete Databases," *in Proceedings of the 10th International Conference on Advances in Database Technology*, Munich, pp. 884-901, 2006.

[10] Chan C., Jagadish H., Tan K., Anthony K., and, Zhenjie Z., "On High Dimensional Skylines," *in Proceedings of the 10th International Conference on Extending Database Technology*, Munich, pp. 478-495, 2006.

[11] Chan C., Jagadish H., Tan K., Tung A., and Zhenjie Z., "Finding K-dominant Skylines in High Dimensional Space," *in Proceedings of the International Conference on Management of Data*, Chicago, pp. 503-514, 2006.

[12] Cheng W., Jin X., Sun J., Lin X., Zhang X., and Wang W., "Searching Dimension Incomplete Databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 725-738, 2014.

[13] Chomicki J., Godfrey P., Gryz J., and Liang D., "Skyline with Presorting," *in Proceedings of the 19th International Conference on Data Engineering*, Bangalore, pp. 717-719, 2003.

[14] Chomicki J., Godfrey P., Gryz J., and Liang D., "Skyline with Presorting: Theory and Optimizations," *in Proceedings of the International IIS: IIPWM' 05*, Gdansk, pp. 595-604, 2005.

[15] George A., "Efficient High Dimension Data Clustering Using Constraint-Partitioning K-Means Algorithm," *The International Arab Journal of Information Technology*, vol. 10, no. 5, pp. 467- 476, 2013.

[16] Godfrey P., Shipley R., and Gryz J., "Maximal Vector Computation in Large Data Sets," *in Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, pp. 229-240, 2005.

[17] Green T. and Tannen V., "Models for Incomplete and Probabilistic Information," *IEEE Data Engineering Bulletin*, vol. 29, no. 1, pp. 17-24, 2006.

[18] Grzymala-Busse J. and Hu M., "A Comparison of Several Approaches to Missing Attribute Values in Data Mining," *in Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing*, Banff, pp. 378- 385, 2000.

[19] Grzymala-Busse J., "Rough Set Approach to Incomplete Data," *in Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing*, Zakopane, pp. 50-55, 2004.

[20] Grzymala-Busse J., "Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction," *Transactions on Rough Sets I*, vol. 3100, pp. 78-95, 2004.

[21] Grzymala-Busse J. and Rzasa W., "Local and Global Approximations for Incomplete Data," *Rough Sets and Current Trends in Computing*, vol. 4259, pp. 21-34, 2008.

[22] Haghani P., Michel S., and Aberer K., "Evaluating Top-k Queries over Incomplete Data Streams, *in Proceedings of the 18th ACM Conference on Information and Knowledge Management*, Hong Kong, pp. 877-886, 2009.

[23] Jonsson P. and Wohlin C., "An Evaluation of K-nearest Neighbour Imputation Using Likert Data," *in Proceedings of the 10th International Symposium on Software Metrics*, Chicago, pp. 108-118, 2004.

[24] Khalefa M., Mokbel M., and Levandoski J., "Skyline Query Processing for Incomplete Data," *in Proceedings of the 24th International Conference on Data Engineering*, Cancun, pp. 556-565, 2008.

[25] Kossmann D., Ramsak F., and Rost S., "Shooting Stars in the Sky: An Online Algorithm for Skyline Queries," *in Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, pp. 275-286, 2002.

[26] Ooi B., Goh C., and Tan K., "Fast High-Dimensional Data Search in Incomplete Databases," *in Proceedings of the 24th International Conference on Very Large Data Base*, San Francisco, pp. 357-367, 1998.

[27] Otsuka S. and Miyazaki N., "An Incomplete Database Approach to Global Query Processing," *in Proceedings of the 13th International Conference on Information Networking*, Tokyo, pp. 337-342, 1998.

[28] Papadias D., Tao Y., Fu G., and Seeger B., "An Optimal and Progressive Algorithm for Skyline Queries," *in Proceedings of the International*

Conference on Management of Data, San Diego, pp. 467-478, 2003.
[29] Razniewski S. and Nutt W., "Completeness of Queries Over Incomplete Databases," *in Proceedings of the 37th International Conference on Very Large Data Base*, Seattle, pp. 749-760, 2011.
[30] Sarma A., Benjelloun O., Halevy A., and Widom J., "Working Models for Uncertain Data," *in Proceedings of the 22rd International Conference on Data Engineering*, Atlanta, pp. 7- 27, 2006.
[31] Shen S., "Database Relaxation: An Approach to Query Processing in Incomplete Databases," *Information Processing and Management Journal*, vol. 24, no. 2, pp. 151-159, 1988.
[32] Soliman M., Ilyas I., and Chang K., "Top-k Query Processing in Uncertain Databases," *in Proceedings of the 23rd International Conference on Data Engineering*, Istanbul, pp. 896-905, 2007.
[33] Soliman M., Ilyas I., and Ben-David S., "Supporting Ranking Queries on Uncertain and Incomplete Data," *Very Large Database Journal*, vol. 19, no. 4, pp. 477-501, 2010.
[34] Tan K., Eng P., and Ooi B., "Efficient Progressive Skyline Computation," *in Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, pp. 301-310, 2001.
[35] Twala B., Cartwright M., and Shepperd M., "Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases," *in Proceedings of the International Symposium on Empirical Software Engineering*, Noosa Heads, pp. 105-114, 2005.
[36] Wolf G., Kalavagattu A., Khatri H., Balakrishnan R., Chokshi B., Fan J., Chen Y., and Kambhampati S., "Query Processing Over Incomplete Autonomous Databases: Query Rewriting Using Learned Data Dependencies," *The International Journal on Very Large Data Bases*, vol. 18, no. 5, pp. 1167- 1190, 2009.
[37] Yiu M. and Mamoulis N., "Efficient Processing of Top-k Dominating Queries on Multi-dimensional Data," *in Proceedings of the 33rd International Conference on Very Large Data Bases*, Trondheim, pp. 483-494, 2007.

Ali Alwan: is currently an assistant professor at Kulliyyah (Faculty) of Information and Communication Technology, International Islamic University Malaysia (IIUM), Malaysia. He received his Master of Computer Science (2009) and Ph.D in Computer Science (2013) from Universiti Putra Malaysia (UPM), Malaysia. His research interests include preference queries, skyline queries, probabilistic and uncertain databases, query processing and optimization and management of incomplete data, data integration, location-based social networks (LBSN), recommendation systems, and data management in cloud computing.

Hamidah Ibrahim: is currently a full professor at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). She obtained her PhD in computer science from the University of Wales Cardiff, UK in 1998. Her current research interests include databases (distributed, parallel, mobile, biomedical, XML) focusing on issues related to integrity constraints checking, cache strategies, integration, access control, transaction processing, and query processing and optimization; data management in grid and knowledge-based systems.

NurIzura Udzir: is an associate professor at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM) since 1998. She received her Bachelor of Computer Science (1996) and Master of Science (1998) from UPM, and her PhD in Computer Science from the University of York, UK (2006). She is a member of IEEE Computer Society. Her areas of specialization are access control, secure operating systems, intrusion detection systems, coordination models and languages, and distributed systems. She is currently the Leader of the Information Security Group at the faculty.

Fatimah Sidi: is currently working as an associate professor in the discipline of Computer Science, at Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). She obtained her PhD in management information system from Universiti Putra Malaysia, Malaysia (UPM) (2008). Her current research interests are Knowledge and Information Management Systems, Data and Knowledge Engineering, Database and Data Warehouse.