Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 169 (2005) 47–69
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

A new method to estimate null values in relational database systems based on automatic clustering techniques

Shyi-Ming Chen [a,*], Hsin-Ren Hsiao [b]

[a] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC*
[b] *Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC*

Received 18 October 2003; received in revised form 4 February 2004; accepted 5 February 2004

Abstract

In this paper, we present a new method for estimating null values in relational database systems based on automatic clustering techniques. The proposed method clusters data in advance, such that it only needs to process the most proper clusters instead of all the data in the relational database system for estimating null values. The average estimated accuracy rate of the proposed method is better than the existing methods for estimating null values in relational database systems.

Keywords: Automatic clustering algorithm; Fuzzy relation; Null values; Relational database systems

1. Introduction

It is obvious that data processing is an important activity in business processing. A database system keeps a lot of information for an enterprise when business running. Relational database systems are most widely used in an enterprise. A database system will not operate properly if there exist some null

* Corresponding author.
E-mail address: smchen@et.ntust.edu.tw (S.-M. Chen).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2004.02.012

values of attributes in the system. In [4], Candan et al. provided a unified treatment of null values using constraints. In recent years, many researchers focused on the research topics of estimating null values in relational database systems [5,6,10–12]. In [5], Chen et al. presented a method to estimate null values in the distributed relational database environments. In [6], Chen and Yeh presented a method to generate fuzzy rules from relational database systems for estimating null values. In [10], Hsiao et al. presented a method to estimate null values in relational database systems. In [11], Huang and Chen presented a method to estimate null values in relational database systems using genetic algorithms. In [12], Huang and Chen presented a method to estimate null values in relational database systems with a negative dependency relationship between attributes.

In this paper, we present a new method to estimate null values in relational database systems based on automatic clustering techniques. The proposed method clusters data in advance, such that it only needs to process the most proper clusters instead of the whole data in the relational database system for estimating null values. The average estimated accuracy rate of the proposed method is better than the methods presented in [5,6]. It can estimate null values in relational database systems more accurately.

The rest of this paper is organized as follows. In Section 2, we briefly review an automatic clustering algorithm from [9]. In Section 3, we briefly review Chen-and-Chen's method for estimating null values in relational database systems from [5]. In Section 4, we present a new method for estimating null values in relational database systems. In Section 5, we use an example to illustrate the proposed method. Furthermore, we also compare the average estimated error rate of the proposed method with the existing methods. The conclusions are discussed in Section 6.

2. An automatic clustering algorithm

In recent years, some methods have been proposed for information retrieval [2] and fuzzy query processing [13,16,19] based on clustering techniques. In [18], Selim presented a semi-fuzzy approach for clustering multi-dimensional data. In [3], Can et al. presented an incremental clustering method for document databases. In [7], Hirano et al. presented a comparison of clustering methods for clinical databases. In [9], we have presented an automatic clustering algorithm for fuzzy query processing for relational database systems. The automatic clustering algorithm is reviewed from [9] as follows:

Automatic Clustering Algorithm

/* ***current***: the numerical datum to be clustered;
preceding: the numerical datum preceding ***current*** in the sequence;

pre_preceding: the numerical datum preceding ***preceding*** in the numerical data sequence;

average_dist: the average distance between every pair of neighboring numerical data in the data sequence;

cluster_average_dist: the average distance in a cluster. */

Step 1: Sort the numerical data in an ascending sequence using quick sort [8].

Step 2: Calculate the value of ***average_dist*** of the sorted data sequence;

/* For example, assume that the sorted numerical data sequence is "$x_i = x_j < x_k < x_s = x_m < x_p$". Because there are only four different numerical data (i.e., x_j, x_k, x_s, and x_p) in the sorted data sequence, where the same numerical data (e.g., x_i and x_j are the same numerical data; x_s and x_m are the same numerical data) are only counted once, the value of **average_dist** is calculated as follows:

$$\text{average\_dist} = \frac{(x_k - x_j) + (x_s - x_k) + (x_p - x_m)}{3}. * /$$

put the first numerical datum in the sorted numerical data sequence into the first cluster;

let ***current*** be the second numerical datum in the sorted numerical data sequence;

let ***preceding*** be the first numerical datum in the numerical data sequence;

let ***pre_preceding*** = ***preceding***;

If ***current-preceding*** $\leqslant$ ***average_dist*** then

{

put the numerical datum that ***current*** represents into the cluster;

calculate ***cluster_average_dist***of the cluster, where ***cluster_average_dist*** denotes the average distance of distances between every pair of neighboring elements in a cluster;

let ***preceding*** = ***current***;

let ***current*** be the numerical datum next to ***preceding*** in the sorted numerical data sequence

}

else

{

put the numerical datum that ***current*** represents into a new cluster;

let ***preceding*** = ***current***;

let ***current*** be the numerical datum next to ***preceding*** in the sorted numerical data sequence

}.

Step 3: **If** ***preceding*** is the first element in a cluster **then**

if ***current-preceding*** $\leqslant$ ***average_dist***

and ***current-preceding*** < ***preceding-pre_preceding***

then
{
put the numerical datum that ***current*** represents into the cluster; calculate the value of ***cluster_average_dist*** of the cluster, where ***cluster_average_dist*** denotes the average distance of distances between every pair of neighboring elements in a cluster;
let ***pre_preceding*** = ***preceding***;
let ***preceding*** = ***current***;
let ***current*** be the numerical datum next to ***preceding*** in the sorted numerical data sequence
}
else
{
put the numerical datum that ***current*** represents into a new cluster;
let ***pre_preceding*** = ***preceding***;
let ***preceding*** = ***current***;
let ***current*** be the numerical datum next to ***preceding*** in the sorted numerical data sequence
}
else if ***current-preceding*** $\leqslant$ ***average_dist***
and ***current-preceding*** $\leqslant$ ***cluster_average_dist***
then
{
put the numerical datum that ***current*** represents into the cluster; calculate the value of ***cluster_average_dist*** of the cluster, where ***cluster_average_dist*** denotes the average distance of distances between every pair of neighboring elements in a cluster;
let ***pre_preceding*** = ***preceding***;
let ***preceding*** = ***current***;
let ***current*** be the numerical datum next to ***preceding*** in the sorted numerical data sequence
}
else
{
put the numerical datum that ***current*** represents into a new cluster;
let ***pre_preceding*** = ***preceding***;
let ***preceding*** = ***current***;
let ***current***be the numerical datum next to ***preceding*** in the sorted numerical data sequence
}.

Step 4: **If** no numerical data in the sorted numerical data sequence need to be clustered

then Stop
elsego to **Step3**.

For more details, please refer to [9].

3. A review of Chen-and-Chen's method to estimate null values

In this section, we briefly review the method presented in [5] for estimating null values in relational databases, where a fuzzy similarity matrix [14] is used to represent fuzzy relations and the method is used to deal with one null value in an attribute. The method presented in [5] can be used to estimate one null value for an attribute at one time. Assume that there is a linguistic variable V which contains linguistic terms $v_1, v_2, \ldots$, and v_n. Fig. 1 shows a fuzzy relation, where u_{ij} denotes the closeness degree between v_i and v_j, $u_{ij} \in [0, 1]$, $1 \leqslant i \leqslant n$, and $1 \leqslant j \leqslant n$. The fuzzy relation shown in Fig. 1 is a symmetrical matrix, i.e., $u_{ii} = 1$, $u_{ij} = u_{ji}$, $1 \leqslant i \leqslant n$, and $1 \leqslant j \leqslant n$. For example, we can construct a fuzzy relation for the linguistic variable "Degree" as shown in Fig. 2. From Fig. 2, we can see that the linguistic terms of the linguistic variable "Degree" are Ph.D., Master and Bachelor.

Let $\mathrm{CD}_v(v_i, v_j)$ denote the closeness degree between v_i and v_j of the linguistic variable V. From Fig. 1, we can see that $\mathrm{CD}_v(v_i, v_j) = u_{ij}$, where each value of

	v_1	v_2	$\cdots$	v_n
v_1	1	u_{12}	$\cdots$	u_{1n}
v_2	u_{21}	1	$\cdots$	u_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
v_n	u_{n1}	u_{n2}	$\cdots$	1

Fig. 1. A fuzzy relation for the linguistic variable V.

	Ph.D.	Master	Bachelor
Ph.D.	1	0.6	0.4
Master	0.6	1	0.6
Bachelor	0.4	0.6	1

Fig. 2. A fuzzy relation for the linguistic variable "Degree".

u_{ij} in the fuzzy relation matrix is defined by an experienced database administrator. From Fig. 1, we can also see that $CD_v(v_i, v_i) = 1$, where $1 \leqslant i \leqslant n$, due to the fact that the degree of closeness between each v_i and itself is equal to 1.

In [5], a fuzzy term ranking function can be defined to keep the rank of each fuzzy term in the same linguistic domain. Assume that the linguistic variable V contains fuzzy terms v_i and v_j, and fuzzy term v_i is prior to fuzzy term v_j, then the ranking order between v_i and v_j is defined as follows:

$$\text{Rank}(v_i) > \text{Rank}(v_j).$$

In [5], a rule base is used to indicate the relationships in which some attributes determine other attributes, where all of the rules in a rule base, including the weights of the attributes, are given by domain experts. Table 1 shows a set of fuzzy rules, where the attributes $A_1, A_2, \ldots,$ and A_n determine the attribute B, w_{ij} denotes the weight of attribute A_j appearing in the antecedent portion of rule R_i, $w_{ij} \in [0, 1]$, $\sum_{j=1}^{n} w_{ij} = 1$, $1 \leqslant i \leqslant m$, and $1 \leqslant j \leqslant n$.

The basic idea of estimating null values in relational database systems presented in [5] is that we compare the tuple which has a null value to each rule in the rule base (i.e., Table 1) to see which one is the closest rule. Then, the null value can be estimated by their closeness degrees. Assume that relation R contains attributes $A_1, A_2, \ldots, A_n$ and B, where attributes $A_1, A_2, \ldots,$ and A_n determine attribute B. Let "$r_j.A_k$" denote the value of attribute A_k appearing in the antecedent portion of rule r_j and let "$T_i.A_k$" denote the value of attribute A_k of tuple T_i. If there is a null value in attribute B, then it can be estimated by the following method [5].

Case 1. First, each tuple T_i in relation R is scanned. Assume that attribute B in relation R is in a numerical domain and there exits a rule r_j in the rule base shown as follows:

$$\textbf{IF} A_1 = A_{j1}(W = w_{j1}) \quad \textbf{AND} \quad A_2 = A_{j2}(W = w_{j2}) \quad \textbf{AND} \ \ldots \ \textbf{AND}$$
$$A_n = A_{jn}(W = w_{jn}) \ \textbf{THEN} \ B = t_j,$$

Table 1
A set of fuzzy rules for determining attribute B [5]

Rule 1: IF$A_1 = a_{11}$ $(W = w_{11})$ AND $A_2 = a_{12}$ $(W = w_{12})$ AND $\cdots$ AND $A_n = a_{1n}$ $(W = w_{1n})$ THEN $B_1 = t_1$
Rule 2: IF$A_1 = a_{21}$ $(W = w_{21})$ AND $A_2 = a_{22}$ $(W = w_{22})$ AND $\cdots$ AND $A_n = a_{2n}$ $(W = w_{2n})$ THEN $B_2 = t_2$
$\vdots$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\vdots$
Rule m: IF$A_1 = a_{m1}$ $(W = w_{m1})$ AND $A_2 = a_{m2}$ $(W = w_{m2})$ AND$\cdots$ AND $A_n = a_{mn}$ $(W = w_{mn})$ THEN $B_m = t_m$

where $1 \leqslant j \leqslant m$. Since attribute A_k can be defined either in a numerical or a non-numerical domain, there are two cases to be considered to calculate the closeness degree between tuple T_i and rule r_j:

(1) A_k is in a numerical domain:

The closeness degree between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$\mathrm{CD}v_{ji}(r_j.A_k, T_i.A_k) = \frac{T_i.A_k}{r_j.A_k}.$$

(2) A_k is in a non-numerical domain:

First, check the ranking order of $T_i.A_k$ and $r_j.A_k$, respectively. Then, the closeness degree between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$\mathrm{CD}v_{ji}(r_j.A_k, T_i.A_k) = \begin{cases} \frac{1}{R_{\mathrm{domain}}[r_j.A_k, T_i.A_k]}, & \text{if } \mathrm{Rank}(T_i.A_k) > \mathrm{Rank}(r_j.A_k) \\ R_{\mathrm{domain}}[r_j.A_k, T_i.A_k], & \text{if } \mathrm{Rank}(T_i.A_k) \leqslant \mathrm{Rank}(r_j.A_k), \end{cases}$$

where R is a fuzzy relation, the symbol "domain" denotes an attribute, and $r_j.A_k$ and $T_i.A_k$ are linguistic terms of the attribute "domain". The closeness degree between tuple T_i and rule r_j can be calculated as follows:

$$\mathrm{CD}(r_j, T_i) = \sum_{k=1}^{n} \mathrm{CD}v_{ji}(r_j.A_i, T_k.A_k) * w_{jk},$$

where w_{jk} is the weight of attribute A_k of rule r_j and $1 \leqslant k \leqslant n$. After the closeness degree between tuple T_i and each rule r_j in the rule base has been calculated, where $1 \leqslant j \leqslant m$, the rule whose closeness degree with respect to tuple T_i is closest to 1 is chosen. If rule r_j is the closest one to tuple T_i, then the null value of attribute B can be estimated as follows:

$$T_i.B = \mathrm{CD}(r_j, T_i) * N_j.$$

Case 2. Assume that attribute B is in a non-numerical domain and there exits a rule r_j in the rule base shown as follows:

IF $A_1 = A_{j1}$ $(W = w_{j1})$ **AND** $A_2 = A_{j2}(W = w_{j2})$ **AND** $\cdots$ **AND** $A_n = A_{jn}(W = w_{jn})$**THEN** $B = W_j$.

Because attribute A_k can be either a numerical or a non-numerical domain, where $1 \leqslant k \leqslant n$, there are two cases to be considered to calculate the closeness degree between tuple T_i and rule r_j:

(1) A_k is in a numerical domain:

The closeness degree between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$\mathrm{CD}v_{ji}(r_j.A_k, T_i.A_k) = \frac{T_i.A_k}{r_j.A_k}.$$

(2) A_k is in a non-numerical domain:

First, check the ranking order of $T_i.A_k$ and $r_j.A_k$, respectively. Then, the closeness degree between $r_j.A_k$ and $T_i.A_k$ can be calculated as follows:

$$\mathrm{CD}v_{ji}(r_j.A_k, T_i.A_k) = \begin{cases} \frac{1}{R_{\mathrm{domain}}[r_j.A_k, T_i.A_k]} & \text{if } \mathrm{Rank}(T_i.A_k) > \mathrm{Rank}(r_j.A_k), \\ R_{\mathrm{domain}}[r_j.A_k, T_i.A_k] & \text{if } \mathrm{Rank}(T_i.A_k) \leqslant \mathrm{Rank}(r_j.A_k). \end{cases}$$

The closeness degree between tuple T_i and rule r_j can be calculated as follows:

$$\mathrm{CD}(r_j, T_i) = \sum_{k=1}^{n} \mathrm{CD}v_{ji}(r_j.A_i, T_k.A_k) * w_{jk},$$

where w_{jk} is the weight of attribute A_k of rule r_j and $1 \leqslant k \leqslant n$. After the closeness degree between tuple T_i and each rule r_j in the rule base has been computed, where $1 \leqslant j \leqslant m$, the rule whose closeness degree with respect to tuple T_i closest to 1 is chosen. If rule r_j is the closest one to tuple T_i, then the value of attribute B can be estimated as follows:

$$T_i.B = W_j.$$

For more details, please refer to [5].

4. A new method for estimating null values in relational database systems

In this section, we present a new method to estimate null values in relational database systems based on automatic clustering techniques. Table 2 shows a relation in a relational database system including four attributes: "W", "X", "Y" and "Z". In Table 2, the values of the attribute Z are dependent on the values of the attributes "X" and "Y" (i.e., the attribute "Z" is functional dependent on the attributes "X" and "Y"). That is, the attributes "X" and "Y" are called independent variables and the attribute "Z" is called a dependent variable. Thus, we can apply the concepts of "correlation" and "coefficient of determination" of the regression analysis of statistics [1,15,17] to represent the weights of attributes. The definitions are shown as follows.

Table 2
A relation in a relational database system

W	X	Y	Z
S_1	D_1	E_1	N_1
S_2	D_2	E_2	N_2
⋮	⋮	⋮	⋮
S_m	D_m	E_m	N_m

Definition 1. Assume that X is an independent variable and Y is a dependent variable, then the degree of correlation $r(X, Y)$ between the variables X and Y is denoted by:

$$r(X, Y) = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_i (X_i - \overline{X})^2}\sqrt{\sum_i (Y_i - \overline{Y})^2}}, \tag{1}$$

where X_i and $\overline{X}$ denote a possible value and the mean value of X, respectively, Y_i and $\overline{Y}$ denote a possible value and the mean value of Y, respectively, and $r(X, Y) \in [-1, 1]$.

Definition 2. Assume that X is an independent variable and Y is a dependent variable, then the Coefficient of Determination $\mathrm{COD}(X, Y)$ from X to Y is

$$\mathrm{COD}(X, Y) = (r(X, Y))^2, \tag{2}$$

where $r(X, Y)$ denotes the degree of correlation between the variables X and Y, and $\mathrm{COD}(X, Y) \in [0, 1]$.

If an attribute is in a non-numerical domain, we can assign a real value to each value of the attribute. Therefore, we can represent a tuple in a relation of a relational database system in terms of a vector. For example, the tuple whose EMP-ID is S_2 in Table 2 can be represented by (D_2, E_2, N_2).

According to the values of the attribute "Z", we can partition m tuples in a relation of a relational database system using the automatic clustering algorithm we presented in [9]. Assume that after applying the automatic clustering algorithm described previously, we obtain k clusters (i.e., $\mathrm{cluster}_1, \mathrm{cluster}_2, \ldots,$ and $\mathrm{cluster}_k$), where $1 \leqslant k \leqslant m$, then we can derive some important information for estimating null values of the attribute "Z". This important information includes C_i, $\mathrm{COD}_{i,d}$, $\mathrm{COD}_{i,e}$, $\Delta\mathrm{DS}_i$, and $\Delta\mathrm{ES}_i$, where C_i denotes the cluster center of $\mathrm{cluster}_i$, $\mathrm{COD}_{i,d}$ denotes the "normalized coefficient of determination" from the attribute "X" to the attribute "Z" of $\mathrm{cluster}_i$, $\mathrm{COD}_{i,e}$ denotes the "normalized coefficient of determination" from the attribute "Y" to the attribute "Z" of $\mathrm{cluster}_i$, ΔDS_i denotes the variation of the value of the attribute "Z" for per unit of the value of the attribute "X" of $\mathrm{cluster}_i$, and $\Delta\mathrm{ES}_i$ denotes the variation of the value of the attribute "Z" for per unit of the value of the attribute "Y" of $\mathrm{cluster}_i$, and $1 \leqslant i \leqslant k$.

The algorithm for estimating null values in relational database systems shown in Table 2 based on automatic clustering techniques is now presented as follows:

Step 1: According to the values of the attribute "Z", partition the m tuples of a relation in a relational database system using the automatic clustering algorithm we presented in [9], where the automatic clustering algorithm

builds clusters by using tuples which contain non-null values. Assume that we obtain k clusters (i.e., cluster_1, cluster_2,..., and cluster_k) shown as follows:

$$\begin{aligned}
\text{cluster}_1 &= \{N_{1,1}, N_{1,2}, \ldots, N_{1,p_1}\},\\
\text{cluster}_2 &= \{N_{2,1}, N_{2,2}, \ldots, N_{2,p_2}\},\\
&\vdots\\
\text{cluster}_k &= \{N_{k,1}, N_{k,2}, \ldots, N_{k,p_k}\},
\end{aligned}$$

where $1 \leqslant k \leqslant m$, $N_{i,j}$ denotes the jth element of cluster_i, p_i denotes the number of elements in cluster_i, and $1 \leqslant i \leqslant k$.

Step 2: Calculate the cluster center C_i of cluster_i, where $1 \leqslant i \leqslant k$, shown as follows:

$$C_i = \left(\frac{\sum_{j=1}^{p} D_{i,j}}{p}, \frac{\sum_{j=1}^{p} E_{i,j}}{p}, \frac{\sum_{j=1}^{p} N_{i,j}}{p}\right) = (D_{i\_\text{center}}, E_{i\_\text{center}}, N_{i\_\text{center}}), \quad (3)$$

where p denotes the number of elements in cluster_i, $D_{i,j}$ denotes the attribute value of the attribute "X" of the jth element of cluster_i, $E_{i,j}$ denotes the attribute value of the attribute "Y" of the jth element of cluster_i, $N_{i,j}$ denotes the attribute value of the attribute "Z" of the jth element of cluster_i, D_{i_center} denotes the average attribute value of the attribute "X" of cluster_i, E_{i_center} denotes the average attribute value of the attribute "Y" of cluster_i, N_{i_center} denotes the average attribute value of the attribute "Z" of cluster_i, $1 \leqslant j \leqslant p$, and $1 \leqslant p \leqslant m$.

Step 3: Based on formula (1), calculate the degree of correlation $r_i(X, Z)$ between the attributes "X" and "Z", and the degree of correlation $r_i(Y, Z)$ between the attributes "Y" and "Z" of cluster_i, respectively, where $1 \leqslant i \leqslant k$. If there is only one element in cluster_i, then let $r_i(X, Z) = r_i(Y, Z) = 0.5000$.

Step 4: Based on formula (2), calculate the coefficient of determination from the attribute "X" to the attribute "Z" and the coefficient of determination from the attribute "Y" to the attribute "Z" of cluster_i, where $1 \leqslant i \leqslant k$, and then normalize them. The normalized coefficient of determination from the attribute "X" to the attribute "Z" of cluster_i is denoted by $\text{COD}_{i,d}$ and the normalized coefficient of determination from the attribute "Y" to the attribute "Z" of cluster_i is denoted by $\text{COD}_{i,e}$.

Step 5: Calculate the variation of the value of the attribute "Z" for per unit of the value of the attribute "X" of cluster_i (i.e., ΔDS_i) and the variation of the value of the attribute "Z" for per unit of the value of the attribute "Y" of cluster_i (i.e., ΔES_i). Assume that there are p elements in cluster_i, $1 \leqslant p \leqslant m$, and assume that the jth element of cluster_i is de-

noted by $(D_{i,j}, E_{i,j}, N_{i,j})$, where $1 \leqslant j \leqslant p$, then ΔDS_i and ΔES_i are calculated as follows:

$$\Delta DS_i = \begin{cases} 0, & \text{if } p = 1 \\ \pm COD_{i,d} \times \dfrac{\sum_{j=1}^{p} \left| N_{i\_center} - N_{i,j} \right|}{\sum_{j=1}^{p} \left| D_{i\_center} - D_{i,j} \right|}, & \text{if } 2 \leqslant p \leqslant m \end{cases} \quad (4)$$

$$\Delta ES_i = \begin{cases} 0, & \text{if } p = 1 \\ \pm COD_{i,e} \times \dfrac{\sum_{j=1}^{p} \left| N_{i\_center} - N_{i,j} \right|}{\sum_{j=1}^{p} \left| E_{i\_center} - E_{i,j} \right|}, & \text{if } 2 \leqslant p \leqslant m \end{cases} \quad (5)$$

The signs of ΔDS_i and ΔES_i are the same as the signs of correlation $r_i(X, Z)$ and $r_i(Y, Z)$, respectively.

Step 6: Let $(d, e, N_{estimated})$ denote the tuple of an employee whose "X" is d, "Y" is e, and "Z" is a null value. Calculate the Euclidean Degree–Experience distance $Dist_i$ between $(d, e, N_{estimated})$ and $(D_{i\_center}, E_{i\_center}, N_{i\_center})$ shown as follows:

$$Dist_i = \sqrt{(D_{i\_center} - d)^2 + (E_{i\_center} - e)^2}, \quad (6)$$

where $1 \leqslant i \leqslant k$. If $Dist_p = Min(Dist_1, Dist_2, \ldots, Dist_k)$, where $1 \leqslant p \leqslant k$, then we let $(d, e, N_{estimated})$ be an element of $cluster_p$.

Step 7: Calculate the estimated value $N_{estimated}$ of the attribute "Salary" as follows:

$$N_{estimated} = N_{p\_center} + \Delta DS_p \times (d - D_{p\_center}) + \Delta ES_p \times (e - E_{p\_center}), \quad (7)$$

where $1 \leqslant p \leqslant k$, D_{p_center} denotes the average value of the attribute "X" of $cluster_p$, E_{p_center} denotes the average value of the attribute "Y" of $cluster_p$, and N_{p_center} denotes the average value of the attribute "Z" of $cluster_p$.

After calculating the value of $N_{estimated}$, we can calculate the estimated error shown as follows:

$$\text{Estimated Error Rate} = \frac{N_{estimated} - \text{Original Value}}{\text{Original Value}} \times 100\%.$$

5. An example to estimate null values in relational database systems

Table 3 [6] shows a relation in a relational database in which the tuple whose "EMP-ID" is S_{23} has a null value in the attribute "Salary", where the attribute

Table 3
A relation contains null values [6]

EMP-ID	Degree	Experience	Salary
S_1	Ph.D.	7.200	63000
S_2	Master	2.000	37000
S_3	Bachelor	7.000	40000
S_4	Ph.D.	1.200	47000
S_5	Master	7.500	53000
S_6	Bachelor	1.500	26000
S_7	Bachelor	2.300	29000
S_8	Ph.D.	2.000	50000
S_9	Ph.D.	3.800	54000
S_{10}	Bachelor	3.500	35000
S_{11}	Master	3.500	40000
S_{12}	Master	3.600	41000
S_{13}	Master	10.000	68000
S_{14}	Ph.D.	5.000	57000
S_{15}	Bachelor	5.000	36000
S_{16}	Master	6.200	50000
S_{17}	Bachelor	0.500	23000
S_{18}	Master	7.200	55000
S_{19}	Master	6.500	51000
S_{20}	Ph.D.	7.800	65000
S_{21}	Master	8.100	64000
S_{22}	Ph.D.	8.500	70000
S_{23}	Master	4.500	**NULL**

"Salary" is functional dependent on the attributes "Degree" and "Experience". We can estimate this null value based on the proposed algorithm. Because the ranking order of the values of the attribute "Degree" is

$$\text{Rank}(\text{Ph.D.}) > \text{Rank}(\text{Master}) > \text{Rank}(\text{Bachelor}),$$

"Bachelor" is assigned to 1.000, "Master" is assigned to 2.000, and "Ph.D." is assigned to 3.000, where these assignments can get the best estimated accuracy rate for estimating null values. Table 4 shows the results after assigning the real values to the attribute values of the attribute "Degree".

The process of estimating null values in the relational database system is illustrated as follows:

[Step 1] Based on the automatic clustering algorithm we presented in [9], we can cluster all of the tuples in Table 4 except the tuple whose EMP-ID is 4_{23} according to the attribute values of "Salary" in Table 4. The cluster results are shown as follows:

Table 4
A relation after assigning real values to the attribute values of the attribute "Degree"

EMP-ID	Degree	Experience	Salary
S_1	3.000	7.200	63000
S_2	2.000	2.000	37000
S_3	1.000	7.000	40000
S_4	3.000	1.200	47000
S_5	2.000	7.500	53000
S_6	1.000	1.500	26000
S_7	1.000	2.300	29000
S_8	3.000	2.000	50000
S_9	3.000	3.800	54000
S_{10}	1.000	3.500	35000
S_{11}	2.000	3.500	40000
S_{12}	2.000	3.600	41000
S_{13}	2.000	10.000	68000
S_{14}	3.000	5.000	57000
S_{15}	1.000	5.000	36000
S_{16}	2.000	6.200	50000
S_{17}	1.000	0.500	23000
S_{18}	2.000	7.200	55000
S_{19}	2.000	6.500	51000
S_{20}	3.000	7.800	65000
S_{21}	2.000	8.100	64000
S_{22}	3.000	8.500	70000
S_{23}	2.000	4.500	**NULL**

$$
\begin{aligned}
\text{cluster}_1 &= \{(1.000, 0.500, 23000)\},\\
\text{cluster}_2 &= \{(1.000, 1.500, 26000)\},\\
\text{cluster}_3 &= \{(1.000, 2.300, 29000)\},\\
\text{cluster}_4 &= \{(2.000, 2.000, 37000), (1.000, 3.500, 35000),\\
&\qquad (1.000, 5.000, 36000)\},\\
\text{cluster}_5 &= \{(1.000, 7.000, 40000), (2.000, 3.500, 40000),\\
&\qquad (2.000, 3.600, 41000)\},\\
\text{cluster}_6 &= \{(3.000, 1.200, 47000)\},\\
\text{cluster}_7 &= \{(2.000, 7.500, 53000), (3.000, 2.000, 50000), (3.000, 3.800,\\
&\qquad 54000), (3.000, 5.000, 57000), (2.000, 6.200, 50000), (2.000,\\
&\qquad 7.200, 55000), (2.000, 6.500, 51000)\},\\
\text{cluster}_8 &= \{(3.000, 7.200, 63000), (3.000, 7.800, 65000), (2.000, 8.100,\\
&\qquad 64000)\},\\
\text{cluster}_9 &= \{(2.000, 10.000, 68000), (3.000, 8.500, 70000)\}.
\end{aligned}
$$

Then, based on Table 4, we can get the EMP-ID of the elements of each cluster shown as follows:

$$\begin{aligned}
&\text{cluster}_1 = \{S_{17}\},\\
&\text{cluster}_2 = \{S_6\},\\
&\text{cluster}_3 = \{S_7\},\\
&\text{cluster}_4 = \{S_2, S_{10}, S_{15}\},\\
&\text{cluster}_5 = \{S_3, S_{11}, S_{12}\},\\
&\text{cluster}_6 = \{S_4\},\\
&\text{cluster}_7 = \{S_5, S_8, S_9, S_{14}, S_{16}, S_{18}, S_{19}\},\\
&\text{cluster}_8 = \{S_1, S_{20}, S_{21}\},\\
&\text{cluster}_9 = \{S_{13}, S_{22}\}.
\end{aligned}$$

[Step 2] Based on formula (2), we can calculate the cluster center C_i of each cluster_i, where $1 \leqslant i \leqslant 9$, shown as follows:

$$\begin{aligned}
C_1 &= (1.000, 0.500, 23000),\\
C_2 &= (1.000, 1.500, 26000),\\
C_3 &= (1.000, 2.300, 29000),\\
C_4 &= ((2.000 + 1.000 + 1.000)/3, (2.000 + 3.500 + 5.000)/3,\\
&\quad (37000 + 35000 + 36000)/3)\\
&= (1.333, 3.500, 36000),\\
C_5 &= ((1.000 + 2.000 + 2.000)/3, (7.000 + 3.500 + 3.600)/3,\\
&\quad (40000 + 40000 + 41000)/3)\\
&= (1.667, 4.700, 40333.333),\\
C_6 &= (3.000, 1.200, 47000),\\
C_7 &= ((2.000 + 3.000 + 3.000 + 3.000 + 2.000 + 2.000 + 2.000)/7, (7.500\\
&\quad + 2.000 + 3.800 + 5.000 + 6.200 + 7.200 + 6.500)/7, (53000+\\
&\quad 50000 + 54000 + 57000 + 50000 + 55000 + 51000)/7)\\
&= (2.429, 5.457, 52857.143),\\
C_8 &= ((3.000 + 3.000 + 2.000)/3, (7.200 + 7.800 + 8.100)/3,\\
&\quad (63000 + 65000 + 64000)/3)\\
&= (2.667, 7.700, 64000),\\
C_9 &= ((2.000 + 3.000)/2, (10.000 + 8.500)/2, (68000 + 70000)/2)\\
&= (2.500, 9.250, 69000).
\end{aligned}$$

Table 5 summarizes the results of the calculations. Fig. 3 shows each cluster center C_i, where $1 \leqslant i \leqslant 9$, where the symbol • denotes a tuple

Table 5
The clustering results and each cluster center

$cluster_i$	EMP-ID of each element	Cluster center C_i
$cluster_1$	S_{17}	(1.000, 0.500, 23000)
$cluster_2$	S_6	(1.000, 1.500, 26000)
$cluster_3$	S_7	(1.000, 2.300, 29000)
$cluster_4$	S_2, S_{10}, S_{15}	(1.333, 3.500, 36000)
$cluster_5$	S_3, S_{11}, S_{12}	(1.667, 4.700, 40333.333)
$cluster_6$	S_4	(3.000, 1.200, 47000)
$cluster_7$	S_5, S_8, S_9, S_{14}, S_{16}, S_{18}, S_{19}	(2.429, 5.457, 52857.143)
$cluster_8$	S_1, S_{20}, S_{21}	(2.667, 7.700, 64000)
$cluster_9$	S_{13}, S_{22}	(2.500, 9.250, 69000)

of the relation shown in Table 6 and the symbol ☆ denotes a cluster center.

[Step 3] Based on formula (1), we can calculate the degree of correlation r_i (Degree, Salary) between the attributes "Degree" and "Salary" and the degree of correlation r_i (Experience, Salary) between the attributes "Experience" and "Salary" of $cluster_i$, respectively, where $1 \leqslant i \leqslant 9$. For example, the degree of correlation r_4 (Degree, Salary) between the attributes "Degree" and "Salary" and the degree of correlation and r_4 (Experience, Salary) between the attributes "Experience" and "Salary" of $cluster_4$ are calculated as follows:

(a) Calculate the degree of correlation between the attributes "Degree" and "Salary": Because the values of the attribute "EMP-ID" of the tuples in $cluster_4$ are S_2, S_{10} and S_{15}, from Table 4, we can see that

(i) The values of the attributes "Degree" and "Salary" of the tuple whose EMP-ID $= S_2$ are 2.000 and 37000, respectively. Thus, we let $X_1 = 2.000$ and $Y_1 = 37000$.
(ii) The values of the attributes "Degree" and "Salary" of the tuple whose EMP-ID $= S_{10}$ are 1.000 and 35000, respectively. Thus, we let $X_2 = 1.000$ and $Y_2 = 35000$.
(iii) The values of the attributes "Degree" and "Salary" of the tuple whose EMP-ID $= S_{15}$ are 1.000 and 36000, respectively. Thus, we let $X_3 = 1.000$ and $Y_3 = 360004$.

Therefore, we can get

$$\overline{X} = (2.000 + 1.000 + 1.000)/ = 1.333,$$

$$\overline{Y} = (37000 + 35000 + 36000)/3 = 36000.$$

Based on formula (1), we can get the degree of correlation r_4 (Degree, Salary) between the attributes "Degree" and "Salary" of $cluster_4$ shown as follows:

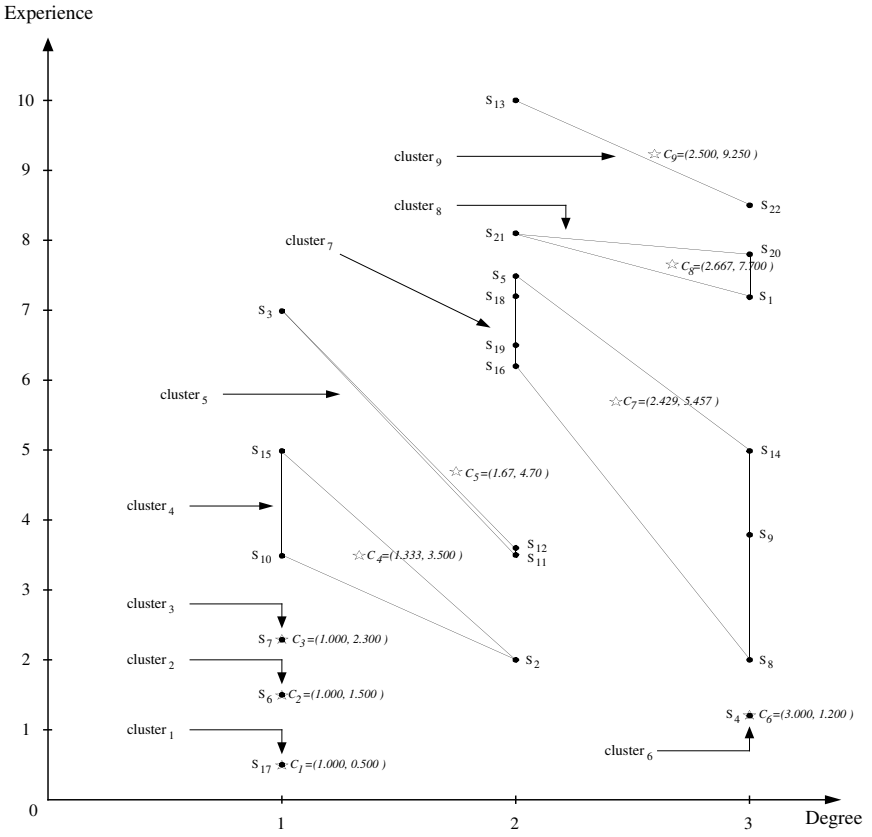

Fig. 3. Diagram representation of each cluster center.

Table 6
The degree of correlation r_i (Degree, Salary) and r_i (Experience, Salary)

Cluster_i	r_i (Degree, Salary)	r_i (Experience, Salary)
cluster_1	0.500	0.500
cluster_2	0.500	0.500
cluster_3	0.500	0.500
cluster_4	0.866	−0.500
cluster_5	0.500	−0.478
cluster_6	0.500	0.500
cluster_7	0.283	0.189
cluster_8	0.000	0.655
cluster_9	1.000	−1.000

$$r_4(\text{Degree}, \text{Salary}) = \frac{\sum_{i=1}^{3}(X_i - 1.333)(Y_i - 36000)}{\sqrt{\sum_{i=1}^{3}(X_i - 1.333)^2}\sqrt{\sum_{i=1}^{3}(Y_i - 36000)^2}} = 0.866.$$

(b) Calculate the degree of correlation between the attributes "Experience" and "Salary": Because the values of the attribute "EMP-ID" of the tuples in cluster$_4$ are S_2, S_{10} and S_{15}, from Table 4, we can see that

(i) The values of the attributes "Experience" and "Salary" of the tuple whose EMP-ID $= S_2$ are 2.000 and 37000, respectively. Thus, we let $X_1 = 2.000$ and $Y_1 = 37000$.
(ii) The values of the attributes "Experience" and "Salary" of the tuple whose EMP-ID $= S_{10}$ are 3.500 and 35000, respectively. Thus, we let $X_2 = 3.500$ and $Y_2 = 35000$.
(iii) The values of the attributes "Experience" and "Salary" of the tuple whose EMP-ID $= S_{15}$ are 5.000 and 36000, respectively. Thus, we let $X_3 = 5.000$ and $Y_3 = 36000$.

Therefore, we can get:

$$\overline{X} = (2.000 + 3.500 + 5.000)/3 = 3.500,$$
$$\overline{Y} = (37000 + 35000 + 36000)/3 = 36000.$$

Based on formula (1), we can get the degree of correlation r_4(Experience, Salary) between the attributes " Degree" and "Salary" of cluster$_4$ shown as follows:

$$r_4(\text{Experience}, \text{Salary}) = \frac{\sum_{i=1}^{3}(X_i - 3.500)(Y_i - 36000)}{\sqrt{\sum_{i=1}^{3}(X_i - 3.500)^2}\sqrt{\sum_{i=1}^{3}(Y_i - 36000)^2}} = -0.500.$$

In the same way, we can obtain the degree of correlation r_i (Degree, Salary) between the attributes "Degree" and "Salary" and the degree of correlation r_i (Experience, Salary) between the attributes "Experience" and "Salary" of each cluster$_i$, where $1 \leqslant i \leqslant 9$, as shown in Table 6.

[Step 4] Based on formula (2), we can calculate the values of $\text{COD}_{i,d}$ and $\text{COD}_{i,e}$ of cluster$_i$, respectively, where $1 \leqslant i \leqslant 9$. For example, $\text{COD}_{4,d}$ and $\text{COD}_{4,e}$ are calculated as follows:

(a) Calculate the coefficient of determination from the attribute "Degree" to the attribute "Salary":

$$\text{COD}_4(\text{Degree}, \text{Salary}) = (r_4(\text{Degree}, \text{Salary}))^2 = (0.866)^2 = 0.750.$$

(b) Calculate the coefficient of determination from the attribute "Experience" to the attribute "Salary":

$$\begin{aligned}\mathrm{COD}_4(\text{Experience}, \text{Salary}) &= (r_4(\text{Experience}, \text{Salary}))^2 = (-0.500)^2 \\ &= 0.250.\end{aligned}$$

After normalizing the value of COD_4 (Degree, Salary) and COD_4 (Experience, Salary), we can get the normalized value of $\mathrm{COD}_{4,d}$ and $\mathrm{COD}_{4,e}$, respectively, shown as follows:

$$\begin{aligned}\mathrm{COD}_{4,d} &= \frac{\mathrm{COD}_4(\text{Degree}, \text{Salary})}{\mathrm{COD}_4(\text{Degree}, \text{Salary}) + \mathrm{COD}_4(\text{Experience}, \text{Salary})} \\ &= \frac{0.750}{0.750 + 0.250} = 0.750, \\ \mathrm{COD}_{4,e} &= \frac{\mathrm{COD}_4(\text{Experience}, \text{Salary})}{\mathrm{COD}_4(\text{Degree}, \text{Salary}) + \mathrm{COD}_4(\text{Experience}, \text{Salary})} \\ &= \frac{0.250}{0.750 + 0.250} = 0.250.\end{aligned}$$

In the same way, we can obtain the value of $\mathrm{COD}_{i,d}$ and $\mathrm{COD}_{i,e}$ of each $\mathrm{cluster}_i$, where $1 \leqslant i \leqslant 9$, as shown in Table 7.

[Step 5] Based on formula (4) and formula (5), we can calculate the value of $\Delta\mathrm{DS}_i$ and $\Delta\mathrm{ES}_i$ of each $\mathrm{cluster}_i$, where $1 \leqslant i \leqslant 9$. For example, the value of $\Delta\mathrm{DS}_4$ and $\Delta\mathrm{ES}_4$ are calculated as follows:

(i) From Table 5, we can see that the EMP-ID of the tuples in $\mathrm{cluster}_4$ are S_2, S_{10} and S_{15}, and the cluster center C_4 of $\mathrm{cluster}_4$ is (1.333, 3.500, 36000). Thus, $D_{4\_\mathrm{center}} = 1.333$, $E_{4\_\mathrm{center}} = 3.500$ and $N_{4\_\mathrm{center}} = 36000$.
(ii) From Table 4, we can see that $D_{4,1} = 2.000$, $D_{4,2} = 1.000$, $D_{4,3} = 1.000$, $E_{4,1} = 2.000$, $E_{4,2} = 3.500$, $E_{4,3} = 5.000$, $N_{4,1} = 37000$, $N_{4,2} = 35000$ and $N_{4,3} = 36000$.
(iii) From Table 7, we can see that $\mathrm{COD}_{4,d} = 0.750$ and $\mathrm{COD}_{4,e} = 0.250$.

Table 7
The value of $\mathrm{COD}_{i,d}$ and $\mathrm{COD}_{i,e}$ of each $\mathrm{cluster}_i$

$\mathrm{Cluster}_i$	$\mathrm{COD}_{i,d}$	$\mathrm{COD}_{i,e}$
$\mathrm{cluster}_1$	0.500	0.500
$\mathrm{cluster}_2$	0.500	0.500
$\mathrm{cluster}_3$	0.500	0.500
$\mathrm{cluster}_4$	0.750	0.250
$\mathrm{cluster}_5$	0.522	0.478
$\mathrm{cluster}_6$	0.500	0.500
$\mathrm{cluster}_7$	0.692	0.308
$\mathrm{cluster}_8$	0.000	1.000
$\mathrm{cluster}_9$	0.500	0.500

Based on formula (4), we can get the value of ΔDS_4 shown as follows:

$$\Delta DS_4 = COD_{4,d} \times \frac{\sum_{j=1}^{3} |N_{4\_center} - N_{4,j}|}{\sum_{j=1}^{3} |D_{4\_center} - D_{4,j}|} = 0.750 \times \frac{1000 + 1000 + 0}{0.667 + 0.333 + 0.333} = 1127.820.$$

Based on formula (5), we can get the value of ΔES_4 shown as follows:

$$\Delta ES_4 = -COD_{4,e} \times \frac{\sum_{j=1}^{3} |N_{4\_center} - N_{4,j}|}{\sum_{j=1}^{3} |E_{4\_center} - E_{4,j}|} = -0.250 \times \frac{1000 + 1000 + 0}{1.500 + 0.00 + 1.500} = -166.667.$$

In the same way, we can get the values of ΔDS_i and ΔES_i of each $cluster_i$, where $1 \leqslant i \leqslant 9$, as shown in Table 8.

[Step 6] Based on formula (6), we can calculate the Euclidean Degree-Experience distance $Dist_i$ between $S_{23}(2.000, 4.500, N_{estimated})$ and the cluster center C_i of $cluster_i$, where $1 \leqslant i \leqslant 9$. The results are shown as follows:

$$Dist_1 = \sqrt{(1.000 - 2.00)^2 + (0.500 - 4.500)^2} = 4.123,$$

$$Dist_2 = \sqrt{(1.000 - 2.000)^2 + (1.500 - 4.500)^2} = 3.162,$$

$$Dist_3 = \sqrt{(1.000 - 2.000)^2 + (2.300 - 4.500)^2} = 2.417,$$

Table 8
ΔDS_i and ΔES_i of each $cluster_i$

$Cluster_i$	ΔDS_i	ΔES_i
$cluster_1$	0.000	0.000
$cluster_2$	0.000	0.000
$cluster_3$	0.000	0.000
$cluster_4$	1127.820	−166.667
$cluster_5$	523.678	−138.443
$cluster_6$	0.000	0.000
$cluster_7$	3047.114	431.578
$cluster_8$	0.000	2000.000
$cluster_9$	1000.000	−666.667

$$\mathrm{Dist}_4 = \sqrt{(1.333 - 2.000)^2 + (3.500 - 4.500)^2} = 1.202,$$

$$\mathrm{Dist}_5 = \sqrt{(1.667 - 2.000)^2 + (4.700 - 4.500)^2} = 0.388,$$

$$\mathrm{Dist}_6 = \sqrt{(3.000 - 2.000)^2 + (1.200 - 4.500)^2} = 3.448,$$

$$\mathrm{Dist}_7 = \sqrt{(2.429 - 2.000)^2 + (5.457 - 4.500)^2} = 1.049,$$

$$\mathrm{Dist}_8 = \sqrt{(2.667 - 2.000)^2 + (7.700 - 4.500)^2} = 3.269,$$

$$\mathrm{Dist}_9 = \sqrt{(2.500 - 2.000)^2 + (9.250 - 4.500)^2} = 4.776.$$

Because Dist_5 has the smallest value among the values of Dist_1, Dist_2,..., and Dist_9, we let S_{23} (2.000, 4.500, $N_{\mathrm{estimated}}$) be an element of $\mathrm{cluster}_5$.

[Step 7] Based on formula (7), the estimated value $N_{\mathrm{estimated}}$ of the attribute "Salary" of the tuple whose EMP-ID = S_{23} can be calculated as follows:

Table 9
A relation in a relational database system [5,6]

EMP-ID	Degree	Experience	Salary
S_1	Ph.D.	7.200	63000
S_2	Master	2.000	37000
S_3	Bachelor	7.000	40000
S_4	Ph.D.	1.200	47000
S_5	Master	7.500	53000
S_6	Bachelor	1.500	26000
S_7	Bachelor	2.300	29000
S_8	Ph.D.	2.000	50000
S_9	Ph.D.	3.800	54000
S_{10}	Bachelor	3.500	35000
S_{11}	Master	3.500	40000
S_{12}	Master	3.600	41000
S_{13}	Master	10.000	68000
S_{14}	Ph.D.	5.000	57000
S_{15}	Bachelor	5.000	36000
S_{16}	Master	6.200	50000
S_{17}	Bachelor	0.500	23000
S_{18}	Master	7.200	55000
S_{19}	Master	6.500	51000
S_{20}	Ph.D.	7.800	65000
S_{21}	Master	8.100	64000
S_{22}	Ph.D.	8.500	70000

Table 10
A comparison of the estimated results of the proposed method with the existing methods

EMP-ID	Degree	Experience	Salary	Chen-and-Chen's method [5]		Chen-and-Yeh's method [6]		The proposed method	
				Salary (estimated)	Estimated error	Salary (estimated)	Estimated error	Salary (estimated)	Estimated error
S_1	Ph.D.	7.200	63000	63000	+0.000	65000	+3.175	63000.000	+0.000
S_2	Master	2.000	37000	33711	−8.889	30704	−17.016	37002.256	+0.006
S_3	Bachelor	7.000	40000	46648	+16.620	35000	−12.500	39665.621	−0.836
S_4	Ph.D.	1.200	47000	36216	−22.945	46000	−2.128	47000.000	+0.000
S_5	Master	7.500	53000	56200	+6.038	54500	+2.830	52431.645	−1.072
S_6	Bachelor	1.500	26000	27179	+4.535	26346	+1.331	26000.000	+0.000
S_7	Bachelor	2.300	29000	29195	+ 0.672	28500	−1.724	29000.000	+0.000
S_8	Ph.D.	2.000	50000	39861	−20.278	50000	+0.000	53105.080	+6.210
S_9	Ph.D.	3.800	54000	48061	−10.998	55000	+1.852	53881.920	−0.219
S_{10}	Bachelor	3.500	35000	32219	−7.946	31538	−9.891	35624.436	+1.784
S_{11}	Master	3.500	40000	40544	+1.360	41590	+3.975	40673.849	+1.685
S_{12}	Master	3.600	41000	41000	0.000	45159	+10.144	40660.005	−0.829
S_{13}	Master	10.000	68000	64533	−5.099	65000	−4.412	68000.000	+0.000
S_{14}	Ph.D.	5.000	57000	55666	−2.34	55000	−3.509	54399.814	−4.562
S_{15}	Bachelor	5.000	36000	35999	−0.003	35000	−2.778	35374.435	−1.738
S_{16}	Master	6.200	50000	51866	+3.732	48600	−2.800	51870.594	+3.741
S_{17}	Bachelor	0.500	23000	24659	+7.213	25000	+8.696	23000.000	+0.000
S_{18}	Master	7.200	55000	55200	+0.364	52400	−4.727	52302.172	−4.905
S_{19}	Master	6.500	51000	52866	+3.659	49500	−2.941	52000.067	+1.961
S_{20}	Ph.D.	7.800	65000	65000	+0.000	65000	+0.000	64200.000	−1.231
S_{21}	Master	8.100	64000	58200	−9.063	58700	−8.281	64800.000	+1.250
S_{22}	Ph.D.	8.500	70000	67333	−3.810	65000	−7.143	70000.000	+0.000
Average estimated error rate (%)				6.162		5.084		1.456	

$$\begin{aligned} N_{\text{estimated}} &= N_{5_{\text{center}}} + \Delta DS_5 \times (2.000 - D_{5_{\text{center}}}) + \Delta \text{ES}_5 \times (4.500 - E_{5_{\text{center}}}) \\ &= 40333.333 + 523.678 \times (2.000 - 1.667) \\ &\quad + (-138.443) \times (4.500 - 4.700) \\ &= 40186.637. \end{aligned}$$

Therefore, the estimated value of the attribute "Salary" of the tuple whose EMP-ID $= S_{23}$ is 40186.637.

Table 9 [5,6] shows a relation of a relational database system. We apply the proposed method to estimate the attribute value of the attribute "Salary" of each tuple shown in Table 9 and compare the estimated error of each tuple with the ones in [5,6], as shown in Table 10, where we made 22 experiments and each experiment consider 1 null value and 21 non-null values of the attribute "Salary" of Table 9. From Table 10, we can see that the average estimated error rate of the proposed method is smaller than the ones presented in [5,6]. That is, the average estimated accuracy rate of the proposed method is better than the ones presented in [5,6].

6. Conclusions

In this paper, we have presented a new method to estimate null values in relational database systems based on automatic clustering techniques. The proposed method clusters data in advance, such that it only needs to process the most proper clusters instead of all the data in the relational database systems. From Table 10, we can see that the average estimated error rate of the proposed method is smaller than the ones presented in [5,6]. That is, the average estimated accuracy rate of the proposed method is better than the ones presented in [5,6]. The proposed method provides a useful way to estimate null values in relational database systems.

Acknowledgements

This work was supported in part by the National Science Council, Republic of China, under Grant NSC 91-2213-E-011-052.

References

[1] M.L. Bernson, D.M. Levine, M. Goldstein, Intermediate Statistical Methods and Applications, Prentice-Hall, New Jersey, 1983.
[2] S.K. Bhatia, J.S. Deogun, Conceptual clustering in information retrieval, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 28 (3) (1998) 427–436.

[3] F. Can, E.A. Fox, C.D. Snavely, R.K. France, Incremental clustering for very large databases: initial MARIAN experience, Information Sciences 84 (1-2) (1995) 101–114.
[4] K.S. Candan, J. Grant, V.S. Subrahmanian, A unified treatment of null values using constraints, Information Sciences 98 (1-4) (1997) 99–156.
[5] S.M. Chen, H.H. Chen, Estimating null values in the distributed relational databases environments, Cybernetics and Systems: An International Journal 31 (8) (2000) 851–871.
[6] S.M. Chen, M.S. Yeh, Generating fuzzy rules from relational database systems for estimating null values, Cybernetics and Systems: An International Journal 29 (6) (1998) 363–376.
[7] S. Hirano, X. Sun, S. Tsumoto, Comparison of clustering methods for clinical databases, Information Sciences 159 (3-4) (2004) 155–165.
[8] E. Horowitz, S. Sahni, Fundamentals of Data Structures, Computer Science Press, New York, 1982.
[9] H.R. Hsiao, S.M. Chen, A new automatic clustering algorithm for fuzzy query processing, in: Proceedings of the 6th Conference on Artificial Intelligence and Applications, Kaohsiung, Taiwan, Republic of China, 2001, pp. 550–555.
[10] H.R. Hsiao, S.M. Chen, A new method to estimate null values in relational database systems, in: Proceedings of the 2002 International Conference on Information Management, Taipei, Taiwan, Republic of China, 2002.
[11] C.M. Huang, S.M. Chen, A new method to estimate null values in relational database systems using genetic algorithms, in: Proceedings of the 6th Conference on Artificial Intelligence and Applications, Kaohsiung, Taiwan, Republic of China, 2001, pp. 599–604.
[12] C.M. Huang, S.M. Chen, Estimating null values in relational database systems with a negative dependency relationship between attributes, in: Proceedings of the 2002 International Conference on Information Management, Taipei, Taiwan, Republic of China, 2002.
[13] M. Kamel, B. Hadfield, M. Ismail, Fuzzy query processing using clustering techniques, Information Processing and Management 26 (2) (1990) 279–293.
[14] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, New Jersey, 1995.
[15] S.W. Lee, S.M. Chen, A new method to generate fuzzy rules from relational database systems, in: Proceedings of the 9th National Conference on Fuzzy Theory and Its Applications, Tao-Yuan, Taiwan, Republic of China, 2001, pp. 702–707.
[16] Y.S. Lin, S.M. Chen, Using automatic clustering techniques for fuzzy query processing in relational database systems, in: Proceedings of the 11th National Conference on Information Management, Kaohsiung, Taiwan, Republic of China, 2000.
[17] W. Mendenhall, R.J. Beaver, Introduction to probability and statistics, Wadsworth, Belmont, CA, 1994.
[18] S.Z. Selim, M.A. Ismail, Soft clustering of multidimensional data: a semi-fuzzy approach, Pattern Recognition 17 (5) (1984) 559–568.
[19] M.S. Yeh, S.M. Chen, A new method for fuzzy query processing using automatic clustering techniques, Journal of Computers 6 (1) (1994) 1–10.