Introducing Relation Compactness for Generating a Flexible Size of Search Results in Fuzzy Queries

Xiaohui Tang Guoqing Chen Qiang Wei

School of Economics and Management, Tsinghua University
Beijing 100084, China
Email: weiq@sem.tsinghua.edu.cn

Abstract-*Generating relevant and manageable search results is of great interest in database queries and information retrieval nowadays. This paper proposes an approach to measuring relation compactness and to generating a compact set of query outcomes under a fuzzy relational data model. The approach has desirable properties that (1) the resultant compact set is unique and contains mutually distinct tuples at the specified degree, and (2) the resultant compact set is information-equivalent to the collection of original query outcomes. Moreover, it is deemed appealing that the approach provides a way to obtain query outcomes of different size in a flexible manner according to the specified degree of compactness or the preferred number of tuples by users.*

Keywords—Tuple closeness, Fuzzy databases and queries, Relation compactness, Information equivalence, Web search

1 Introduction

The commonly used database model nowadays is the relational database (*RDB*) model initiated by Codd [1] which is generally designed under the assumption that the data/information is precise and queries are crisp. However, decision makers often face a growing need for representing and processing imprecise and uncertain information. Particularly in a web environment, approximate match is often preferred in generating search outcomes when exact match is unable to produce satisfactory results from targeted databases.

Since 1980's, a number of fuzzy relational database (*FRDB*) models and fuzzy query techniques [2-6] have been proposed to deal with imprecision and uncertainty of fuzziness type by means of fuzzy logic [7]. For instance, with the data stored in classical *RDB*, Kacprzyk and Zadrozny [8] proposed a system called *FQUERY* that could execute fuzzy queries such as "find (all) records such that most of the (important) clauses are satisfied (to a degree in [0, 1])" for crisp data stored in the Microsoft Access DBMS. In [9], a method to transform query conditions described by fuzzy numbers into the classical *SQL* queries was presented. In [10], the authors extended the *SQL* queries into *SQLf* to allow the expression of imprecise queries. Furthermore, a method was developed to transform the *FRDB* into weighted *RDB* in that a nesting mechanism was introduced to support the expression of some query results such as projection-selection-join operations [11]. In [12-13], we extended basic algebraic operators to deal with fuzzy data modeling and queries in forms of database design as well as equivalent definitions and transformation rules of the algebraic operators.

Notably, when querying databases with data of high volume, the size of the outcomes can easily be very large, even massive in a web database search context. For instance, a fuzzy query such as "*Select* Customer Name *From R Where* Age is about 25 AND Location is near B" requires an approximate match between query conditions and values of corresponding attributes. Another example is a web search, say via Google. Just click and see how many pages of outcomes to appear when keying in *"Population of Beijing"* for an exact match and *Population of Beijing* (notably without quotation marks) for an approximate match. Efforts have been made to deal with queries and web search with imprecise information and approximate match measures, so as to enrich the representation semantics and strengthen the power of search engines [5, 14-16]. Generally speaking, approximate match provides flexible queries, which is considered meaningful and useful in many cases, whereas the size of its outcomes is usually larger than that of the exact match.

Hence, the size of query outcomes becomes an issue of concern. One notable approach was to provide the users with the top-*k* results according to a ranking based on a certain evaluation function *F* between query conditions and all data records [17-18]. While sometimes the top-*k* results are sufficient for fuzzy/soft queries with approximate match, it is desirable and meaningful if a query/search could also guarantee that the selected results have no information loss as far as all the query results are concerned. The focus of this paper is then to provide the users with a compact set of query outcomes, in that the compact set (a) is smaller than and information-equivalent to the set of all query outcomes, and (b) at the same time has its size flexibly specified by users upon their need and preference.

2 Fuzzy Relational Database Model

Fuzzy logic incepted by Zadeh [7] attempts at quantifying and reasoning with imprecision and uncertainty that is common in the real world. Possibility distributions provide a graded semantics to natural language statements such as "the customer is young", which are often used in our daily communications. For example, the linguistic term "Young" can be defined as

$$\pi_{Young}(age) = \begin{cases} 1 & 0 \le age \le 30 \\ (50 - age)/20 & 30 < age \le 50 \\ 0 & age > 50 \end{cases} \quad (1)$$

ISBN: 978-989-95079-6-8

Two specific fuzzy relations are of particular interest, namely, closeness relation and similarity relation. A closeness relation c is a fuzzy relation on $X\times X$ such that $\forall x, y\in X$, $c(x, x) = 1$ (reflexive) and $c(x, y) = c(y, x)$ (symmetric). A similarity relation s is a fuzzy relation on $X\times X$ such that $\forall x, y, z\in X$, $s(x, x) =1$ (reflexive), $s(x, y) = s(y, x)$ (symmetric), and $s(x, z) \geq \sup_{y\in X} \min(s(x, y), s(y, z))$ (sup-min transitive). Apparently, similarity relation is a special case of closeness relation.

In this paper, the extended-possibility-based model is considered to be the underlying fuzzy database model since it facilitates handling both imprecision in attribute values (e.g., Age is young) and fuzziness in domain elements (e.g., between classic and gold for the domain of (customer) Class) in terms of possibility distributions (including fuzzy sets and linguistic terms) and closeness relations, respectively [4, 19]. In light of data representation, this model is deemed to be a general setting compared with two known models, namely, the so-called possibility-based model [4] where attribute values can be possibility distributions, and the so-called similarity-based model [3] where domain elements can be associated by similarity relations (reflective, symmetric and sup-min transitive). Specifically, in the extended-possibility-based model, a relation R is a subset of $\Pi(D_1) \times \Pi(D_2) \times \ldots \times \Pi(D_g)$, where $\Pi(D_i) =\{\pi_{Ai} \mid \pi_{Ai}$ is a possibility distribution of attribute A_i on domain $D_i\}$, and a closeness relation c_i is associated with domain D_i ($1 \leq i \leq g$). In addition, an g-tuple t of R is of the form: $t(\pi_{A1}, \pi_{A2}, \ldots, \pi_{Ag})$. An example tuple recording a customer's Name, Age and Class can be (Tony, {0.7/28, 1.0/33, 0.8/36}, diamond). The closeness relation c_{Class} on domain D_{Class}={classic, silver, gold, diamond} can be pre-defined by the managers as shown in Table 1.

Table 1: A closeness relation c_{Class}

c_{Class}	classic	silver	gold	diamond
classic	1.0	0.5	0.0	0.0
silver		1.0	0.75	0.25
gold			1.0	0.75
diamond				1.0

Given two g-tuples $t_p(\pi_{A1}, \pi_{A2}, \ldots, \pi_{Ag})$, $t_q(\pi'_{A1}, \pi'_{A2}, \ldots, \pi'_{Ag})$, where π_{Ai} and π'_{Ai} are normalized ($1 \leq i \leq g$), an example measure for the closeness of two values is [22]:

$$E_c(\pi_{A_i}, \pi'_{A_i}) = \sup_{\substack{x,y\in D_i \\ c_i(x,y)\geq\alpha_i}} \min(\pi_{A_i}(x), \pi'_{A_i}(y)) \qquad (2)$$

Here c_i is a closeness relation on domain D_i, $\alpha_i \in [0, 1]$ is a threshold specified by experts or the database managers. Other measures can be found in [19-21]. Then, the closeness of two tuples t_p and t_q can be considered as:

$$F_c(t_p, t_q) = \min(E_c(\pi_{A_1}, \pi'_{A_1}), E_c(\pi_{A_2}, \pi'_{A_2}), \ldots, E_c(\pi_{A_g}, \pi'_{A_g})) \qquad (3)$$

3 Tuple Extraction

Regarding tuple extraction in fuzzy relational databases, there are two issues to consider. One is to evaluate tuple closeness, the other is to extract representative tuples.

3.1 The evaluation of tuple closeness

For the fuzzy database, given an attribute A with domain D, let $\pi_1, \pi_2, \ldots, \pi_n$ be n possibility distributions as values of A, then a closeness relation on $\Pi(D)\times\Pi(D)$ is a mapping from $\Pi(D)\times\Pi(D)$ to [0,1], which can be represented by an $n\times n$ fuzzy matrix M where $e_{ij} = E_c(\pi_i, \pi_j)$, $1\leq i, j \leq n$, as follows:

$$M = \begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1n} \\ e_{21} & e_{22} & \ldots & e_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ e_{l1} & e_{l2} & \ldots & e_{nn} \end{pmatrix} \qquad (4)$$

Further, for $0 \leq \lambda \leq 1$, where λ is a threshold which is usually determined by users or domain experts, its λ-cut (M_λ) is a classical relation defined by:

$$(e_{ij})_\lambda = \begin{cases} 1 & e_{ij} \geq \lambda \\ 0 & e_{ij} < \lambda \end{cases} \qquad (5)$$

Moreover, it is known that if M is a closeness relation, then its transitive closure M^+ (i.e., M^+ is a series of max-min compositions onto M itself with $M^+ = M^p = M^{p+1}$, $p\geq 1$) is a similarity relation, and M^+ converges within n-1 compositions [23-25]:

$$M^+ = M^p, \ p \leq l - 1 \qquad (6)$$

Furthermore, the equivalence classes of $(M^+)_\lambda$ constitute a partition of the domain concerned [26].

Definition 1: Given a relation $R = \{t_1, t_2, \ldots, t_n\} \subseteq \Pi(D_1) \times \Pi(D_2) \times\ldots\times\Pi(D_g)$, where tuple t_i is $t_i(\pi_{i1}, \pi_{i2}, \ldots, \pi_{ig})$, and threshold $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_g)$ with equivalence classes of $(M^+)_{\lambda i}$, $i = 1, 2, \ldots, n$, then two tuples $t_i(\pi_{i1}, \pi_{i2}, \ldots, \pi_{ig})$ and $t_j(\pi_{j1}, \pi_{j2}, \ldots, \pi_{jg})$ are called $\boldsymbol{\lambda}$-close if for all $k \in \{1, 2, \ldots, g\}$, π_{ik} and π_{jk} are in the same equivalence class. □

Example 1: Given threshold $\boldsymbol{\lambda} = (0.7, 0.8)$ and a relation $S \subseteq \Pi(D_1) \times \Pi(D_2)$ as shown in Table 2. For the sake of simplicity, assume that a_1, a_2, a_3, and a_4 are mutually distinct; so are b_1, b_2, b_3 and b_4. Then the $\boldsymbol{\lambda}$-close tuples can be generated as follows.

Table 2: A relation S with fuzziness

	A_1	A_2
t_1	$0.3/a_1, 0.7/a_2, 0.7/a_3, 0.7/a_4$	$0.53/b_1, 0.59/b_2, 0.62/b_3, 0.85/b_4$
t_2	$0.6/a_1, 0.4/a_2, 0.1/a_3, 0.8/a_4$	$0.92/b_1, 0.67/b_2, 0.09/b_3, 0.06/b_4$
t_3	$0.2/a_1, 0.6/a_2, 0.1/a_3, 1.0/a_4$	$0.45/b_1, 0.75/b_2, 0.06/b_3, 0.75/b_4$
t_4	$0.5/a_1, 0.1/a_2, 1.0/a_3, 0.9/a_4$	$0.61/b_1, 0.66/b_2, 0.35/b_3, 0.69/b_4$
t_5	$0.9/a_1, 0.6/a_2, 0.2/a_3, 0.2/a_4$	$0.56/b_1, 0.32/b_2, 0.98/b_3, 0.30/b_4$
t_6	$0.4/a_1, 0.2/a_2, 0.2/a_3, 0.4/a_4$	$0.81/b_1, 0.30/b_2, 0.93/b_3, 0.99/b_4$

First, for A_1 we can have:

$$M = \begin{pmatrix} 1 & 0.7 & 0.7 & 0.7 & 0.6 & 0.4 \\ 0.7 & 1 & 0.8 & 0.8 & 0.6 & 0.4 \\ 0.7 & 0.8 & 1 & 0.9 & 0.6 & 0.4 \\ 0.7 & 0.8 & 0.9 & 1 & 0.5 & 0.4 \\ 0.6 & 0.6 & 0.6 & 0.5 & 1 & 0.4 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 1 \end{pmatrix},$$

where each element on (i, j) in the matrix represents the closeness of t_i and t_j on A_1. Then, with $(M^+=M^3=M^2)_{0.7}$, three equivalence classes are obtained: $\{\pi_{11}, \pi_{21}, \pi_{31}, \pi_{41}\}$, $\{\pi_{51}\}$ and $\{\pi_{61}\}$. Consider attribute A_2, in the similar manner, there are three equivalence classes: $\{\pi_{12}, \pi_{22}, \pi_{52}, \pi_{62}\}$, $\{\pi_{32}\}$ and $\{\pi_{42}\}$.

ISBN: 978-989-95079-6-8

That means that values of t_1 and t_2 are in the same class for A_1 (i.e., π_{11}, π_{21}) as well as for A_2 (i.e., π_{12}, π_{22}), indicating that t_1 and t_2 are λ-close.

Finally, it is worthwhile to mention that, in case there are closeness relations on domains, the calculation of M can be readily made in consideration of c_i and α_i in E_c. For instance, suppose we have $c_1 = c_{\text{Class}}$ for A_1 with a_1 = classic, a_2 = silver, a_3 = gold and a_4 = diamond as shown in Table 1. Given α_1 = 0.7, since $c_1(a_2, a_3) = c_1(a_3, a_4) = 0.75$, we have $e'_{45} = 0.6 \neq e_{45} = 0.5$. In general, $M \subseteq M'$. □

3.2 Relation compactness

In addition to closeness of any two tuples, it is of interest to investigate the compactness of a given relation R composed of n tuples in the fuzzy relational database. Here the compactness of a relation refers to the degree of non-redundancy. For a given relation, the more redundant tuples it contains, the lower the degree of compactness. Straightforward ways of evaluating the compactness are to use the minimum closeness of any two tuples of R (*e.g.,* $1 - \min_{t,t' \in R} F_c(t, t')$) or the average closeness of any two tuples of R(*e.g.,* $1 - \sum_{t,t' \in R} F_c(t, t')/n(n-1)$) to reflect the compactness of R in whole. But it shall be noted that neither $1 - \min_{t,t' \in R} F_c(t, t')$ nor $1 - \sum_{t,t' \in R} Fc(t, t')/n(n-1)$ could describe the compactness of R in a satisfactory fashion. For example, suppose that all the n tuples in R are classical tuples for the sake of simplicity, i.e., for any $t, t' \in R$, $F_c(t, t') = 1$ or $F_c(t, t') = 0$. If there are $(n-1)$ tuples which are identical, only one tuple is not identical to them, then $1 - \min_{t,t' \in R} F_c(t, t') = 1$. But we know that there are a lot of tuples whose closeness degrees to each other are 1 in such a situation. On the other hand, if half of the tuples are identical to each other, and the other half of the tuples are also identical to each other (but different from the first half), then, $1 - \sum_{t, t' \in R} Fc(t, t')/n(n-1) = 2 \times (n/2) \times (n/2 - 1)/n(n-1) = 0.5 + 1/(2n - 2) > 0.5$. But in this situation, if n is large enough, e.g., $n = 100$, we will surely think that the compactness degree of R will be very close to 0, as 98% of the tuples can be deleted. Thus, a new measure, relation compactness, is proposed to evaluate a given relation R being compact. This measure not only works well in both of the above situations, but also possesses some good properties. Let us first consider it in classical relations, and then extend it into the relations in fuzzy databases.

In information theory [27], consider a single discrete information source, it may produce different kinds of symbol sets $A = \{a_1, a_2, \ldots, a_n\}$. For each possible symbol set A there will be a set of probabilities *pi* of producing the various possible symbols a_i ($\sum_{i=1}^{n} p_i = 1$), where these symbols are assumed successive and independent. Thus there is an entropy *Hi* for each a_i. The entropy of this given piece of information will be defined as the average of these H_i weighted in accordance with the probability of occurrence of the symbols in question,

$$H = H(p_1, p_2, \ldots, p_n) = -\sum_{i}^{n} p_i H_i = -\sum_{i}^{n} p_i \log p_i \qquad (7)$$

where the default log base is 2.

Similarly to [28], if there is a relation S with a set of classical tuples, which can be divided into m classes $C_1, C_2, \ldots, C_m$, the probability of a random tuple belonging to class C_i is n_i/n, where n_i is the number of tuples in C_i, and n is the number of tuples in S. Here, n_i/n is also called the probability of class C_i in S. The expected information for classifying this given relation S is:

$$H(n_1, n_2, \ldots, n_m) = -\sum_{i=1}^{m} p_i H_i = -\sum_{i=`}^{m} \frac{n_i}{n} \log \frac{n_i}{n} \qquad (8)$$

Definition 2: Let $R = \{t_1, t_2, \ldots, t_n\}$ be a classical relation with n tuples, R be divided into m classes $C_1, C_2, \ldots, C_m$ according to tuple identity (i.e., every tuple in C_i is identical to each other), and n_i be the number of tuples in C_i. Then the relation compactness of R is defined as

$$RC'(n_1, n_2, \ldots, n_m) = H(n_1, n_2, \ldots, n_m)/\log n$$
$$= -(\sum_{i=1}^{m} p_i H_i)/\log n = -(\sum_{i=1}^{m} \frac{n_i}{n} \log \frac{n_i}{n})/\log n \qquad (9)$$

For a relation R, its degree of compactness (RC') reflects the extent to which the tuples of R are not redundant, measured by the "amount of information" in R. In other words, the higher RC' is, the less redundant the tuples in R, meaning that the more information R contains.

Next, let us consider the fuzzy extension. Suppose there are n tuples with fuzziness involved in $R = \{t_1, t_2, \ldots, t_n\}$, where these tuples can be close to each other. To describe the degree at which a given relation with fuzziness is compact, the concept of relation compactness in the classical context could be extended to cope with the closeness of these tuples. A tuple t_k may not totally belong to class C_i, but belong to class C_i at a certain degree, e.g., $O_k^i \in [0, 1]$. Thus, we will use $\sum_{k=1}^{n_i} O_k^i$ instead of n_i in the extended relation compactness. Concretely, suppose we have n_i λ-close tuples in class $C_i = \{t_1, t_2, \ldots, t_{ni}\}$ of relation R. For each tuple t_k, $1 \le k \le n_i$, the degree of t_k belonging to C_i (or the closeness of t_k to C_i) can be defined as:

$$O_k^i = \sum_{j=1}^{n_i} F_c(t_k, t_j)/n_i \qquad (10)$$

Definition 3: Given a relation with fuzziness $R = \{t_1, t_2, \ldots, t_n\}$, where R can be divided into m classes $C_1, C_2, \ldots, C_m$ according to λ-close (i.e., every tuple in C_i is λ-close to each other), and n_i is the number of tuples in C_i. The relation compactness in R is,

$$RC(n'_1, n'_2, \ldots, n'_m) = -(\sum_{i=1}^{m} \frac{n'_i}{n} \log \frac{n'_i}{n})/\log n \qquad (11)$$

where $C_i = \{t_1, t_2, \ldots, t_{ni}\}$, $1 \le i \le m$, $1 < n_i \le n$, O_k^i is the degree of t_k belonging to C_i, and $n'_i = \sum_{k=1}^{n_i} O_k^i$ is the Σcount operation for the "effective number" of tuples in class C_i. □

As the classical relation is a special case of a relation with fuzziness, if all the tuples in $C_i = \{t_1, t_2, \ldots, t_{ni}\}$ are classical tuples, then they are equal to each other, i.e., $F_c(t_k, t_j) = 1$, $O_k^i = 1$, and $n'_i = n_i$ ($1 \le j, k \le n_i$). So $RC'(n_1, n_2, \ldots, n_m)$ is a special case of $RC(n'_1, n'_2, \ldots, n'_m)$. Note that one may also use some

ISBN: 978-989-95079-6-8

other measures (e.g., [19]) instead of O_k^i for the degree at which t_k belongs to C_i in (10). For *RC* in the fuzzy database context, the following proposition holds.

Proposition 1: Give a relation $R = \{t_1, t_2, ..., t_n\} \subseteq \Pi(D_1) \times \Pi(D_2) \times \ldots \times \Pi(D_g)$, where it is divided into m classes $C_1, C_2, ..., C_m$ according to $\boldsymbol{\lambda}$-close, $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, ..., \lambda_g\}$, and n_i is the number of tuples in C_i $(1 \le i \le m)$, then

(1) R is not $\boldsymbol{\lambda}$-close, i.e., $m = n$, $n'_1 = n'_2 = ... = n'_m = 1$, if and only if $RC(n'_1, n'_2, ..., n'_m) = 1$.

(2) Suppose $\min_{i=1}^{g} \lambda_i \ge e^{-1} \approx 0.37$, if R is totally $\boldsymbol{\lambda}$-close, i.e., $m = 1$, $n'_1 = n$, then $RC(n'_1)$ decreases when any two of the tuples's closeness $F_c(t_k, t_j)$ increases $(1 \le k, j \le n)$. Especially, if $\boldsymbol{\lambda} = \mathbf{1}$, then $RC(n'_1) = RC'(n_1) = 0$.

(3) $0 \le RC(n'_1, n'_2, ..., n'_m) \le 1$.

(4) Suppose $(n'_p + n'_q)/n \le e^{-1} \approx 0.37$, if n is fixed, $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, ..., \lambda_g\}$, $\lambda_i = \lambda$, $i = 1, 2, ..., g$, then *RC* decreases when any two of the original classes (e.g., C_p, C_q) are merged into one class C'_p according to $\boldsymbol{\lambda}$-close.

For different n and m, the value of *RC* is different. If $m = 2$, then $RC = -((n'_1/n)\log(n'_1/n) + ((n-n'_1)/n)\log((n-n'_1)/n))/\log n$, with different n, *RC* is shown in Figure 1. If $m = 3$, then $RC = -((n'_1/n)\log(n'_1/n) + (n'_2/n)\log(n'_2/n) + ((n-n'_1-n'_2)/n)\log((n-n'_1-n'_2)/n))/\log n$, with $n = 100$, *RC* is shown in Figure 2. Figures 1 and 2 reflect that RC has a characteristic of convexity.

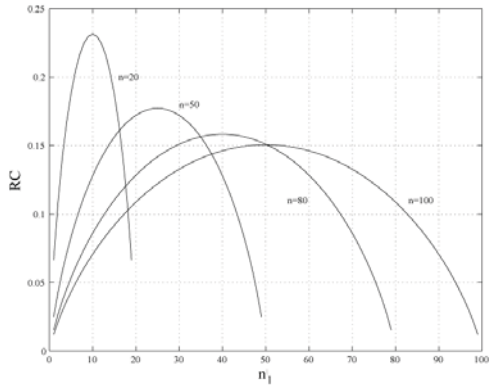

Figure 1: The value of *RC* ($m = 2$)

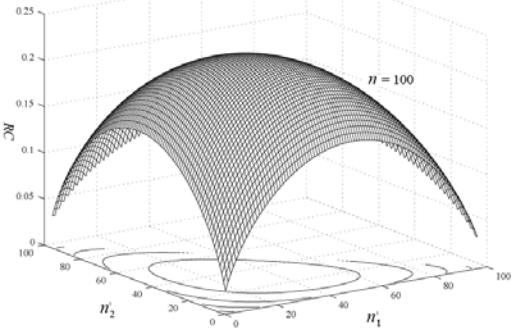

Figure 2: The value of *RC* ($m = 3$; $n = 100$)

3.3 Extraction of representative tuples

When it is known which tuples in a relation with fuzziness are $\boldsymbol{\lambda}$-close, then how to extract the representative tuples is the next problem of concern. It is considered ideal to have fewer tuples if they carry the same amount of original information. Usually, due to existence of tuple closeness, obtaining a smaller set of query outcomes becomes an effort of extracting representative tuples in light of "information equivalence".

Here we consider a center-based method for extracting the representative tuples. Conceptually, all the tuples in a same equivalence class are regarded to express approximately the same information and therefore one of them could be extracted to represent the class. Suppose we have n_i $\boldsymbol{\lambda}$-close tuples in class $C_i = \{t_1, t_2, ..., t_{ni}\}$ of a relation R. Then, we will keep the tuple whose closeness to C_i is the highest. This means, tuple t_p will be kept if $O_p^i = \max_{k=1}^{n_i} O_k^i$, $1 \le p \le n_i$.

Theorem 1: Suppose there are h $\boldsymbol{\lambda}$-close tuples in class $C = \{t_1, t_2, ..., t_h\}$ of a relation R, $C_i = C - \{t_i\}$, $1 \le i \le h$, and the relation compactness of C and C_i are *RC* and RC_i respectively. Then, tuple t_p $(1 \le p \le h)$ will be retained if and only if $RC_p = \max_{i=1}^{h} RC_i$.

Theorem 1 indicates that we can extract the representative tuples in light of relation compactness. It will keep the tuple that, without this tuple, the relation compactness of the other tuples will be the highest.

Example 2: Suppose we have had the $\boldsymbol{\lambda}$-close tuples of a fuzzy query about customers' age and salary as shown in Table 3.

Table 3: Customer's age and salary (in part)

	Age	Salary($)
t_1	0.6/25, 0.9/35, 0.6/40	0.8/4000, 0.9/5000, 0.8/6000
t_2	0.7/28, 0.8/35, 0.7/42	0.7/4500, 0.8/5000, 0.7/6000
t_3	0.6/25, 0.9/35, 0.8/42	0.9/4000, 0.7/5000, 0.6/6000

According to (2): $e_{12}(\pi_{11}, \pi_{21}) = 0.8$, $e_{13}(\pi_{11}, \pi_{31}) = 0.9$, $e_{23}(\pi_{21}, \pi_{31}) = 0.8$; $e_{12}(\pi_{12}, \pi_{22}) = 0.8$, $e_{13}(\pi_{12}, \pi_{32}) = 0.8$, $e_{23}(\pi_{22}, \pi_{32}) = 0.7$. Then, according to (3) and (10): $F_c(t_1, t_2) = 0.8$, $F_c(t_1, t_3) = 0.8$, $F_c(t_2, t_3) = 0.7$, $F_c(t_1, t_1) = F_c(t_2, t_2) = F_c(t_3, t_3) = 1$. $O_1 = 0.87$, $O_2 = 0.83$, $O_3 = 0.83$. Thus, t_2 and t_3 will be eliminated, t_1 will be kept.

Also, from the viewpoint of relation compactness, Table 3 can be considered as a class C, so, $C_1 = \{t_2, t_3\}$, $C_2 = \{t_1, t_3\}$, $C_3 = \{t_1, t_2\}$. For RC_1, $n'_1 = (F_c(t_2, t_2) + F_c(t_2, t_3))=2 + (F_c(t_3, t_3) + F_c(t_2, t_3))/2 = 1.7$. $RC_1 = -(1.7/2)\log(1.7/2)/\log 2 = 0.20$. In the same way, $RC_2 = RC_3 = 0.14$, thus, t_1 will be kept.

As far as the complexity of the tuple extraction is concerned, while introducing possibility distributions enriches the semantics that the data model represents, approximate match may lead to an increase in computational complexity. Further analysis shows that the complexity is polynomial and generally manageable. Suppose there are n tuples in relation R, the matrix M has $n \times n$ entries. For every entry, suppose there are, on average, p values in the two imprecise attribute values to compare, the cost is $O(p^2)$. The total cost to get M is then $O(p^2n^2)$. A reasonable estimate of p could be within 10. Therefore, the total cost to get M could be simplified to be $O(ln^2)$ where $l < 100$. For the transitive closure of M, the total cost is at $O(n^2)$ level for large n [24-25].

Furthermore, in consideration of generating matrix M, though detailed treatments go beyond the scope of this paper, there

ISBN: 978-989-95079-6-8

have been effective approaches in database queries, informational retrieval and web search, using a variety of measures, in respective fields, for keyword similarity, text match, semantic proximity, etc., in order to compare pair-wise closeness of data/information of interest (e.g., [30-34]).

4 Further Discussions

First, whether the extraction result is unique and representative tuples are not redundant is an interesting and important issue. Importantly, it can be proven that the result of the tuple extraction treatment in Section 3 is unique, and the resulting new relation is not $\boldsymbol{\lambda}$-close.

Second, we can also have the number of tuples in the outcome to be controlled by the setting of $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_g)$. When λ_i increases, there will be more classes, thus, after the extraction, more tuples will be kept. Especially, with $\lambda_i = 1$, the treatment turns into the classical situation, in that no tuples will be eliminated.

Third, two representations are information equivalent if the transformation from one to the other entails no loss of information, i.e., if each can be constructed from the other [35]. In other words, the information in one is also inferable from the other, and vice versa [36-37]. Here, "information equivalence" in our discussions could be described as follows. Suppose there is an original query result R_1. The classified relation is R_2 (i.e., the $\boldsymbol{\lambda}$-close tuples are in the same classes), and the resultant relation (compact set) containing representative tuples is R_3, which is not $\boldsymbol{\lambda}$-close.

Compared with original relation R_1 of the query result, R_2 and R_3 contain the classification information. We can say that, in terms of $\boldsymbol{\lambda}$-close, R_2 and R_3 are information equivalent. If we have had R_2, then R_3 could be derived (i.e., via extracting) with respect to tuple closeness/relation compactness. If we have had R_3, suppose $R_3 = \{t_u, t_w, t_x, ...\}$, then every tuple of R_3 is not $\boldsymbol{\lambda}$-close, and can be considered as a class labeled as $C_u, C_w, C_x, ...$. For every tuple t_i of R, if t_i satisfies the fuzzy query conditions, then it could be compared with the tuples in the labeled classes $C_u, C_w, C_x, ...$, if there is a tuple t in one labeled class, e.g., C_u such that t_i and t are $\boldsymbol{\lambda}$-close, then add t_i to C_u. Consequently, we will get relation R_2, as the tuples in R_3 and the equivalence classes in R_2 are bijective.

Finally, let us discuss the approach from a more applied perspective. While searching via a search engine (e.g., Google or the like) or querying available databases for some information, it would be desirable if the database/web search management systems are capable of generating outcomes that are representative and of a manageable size. Even better is to control the query/search outcomes with different compactness degrees or with different sizes. A way to view different results is to specify the degree of relation compactness. If R_C is higher, there will be fewer outcomes, otherwise, more outcomes.

Example 3: Someone uses a search engine to browse for the literature about "fuzzy queries". To help discuss the procedure and treatment details of how the search results can be shortened or extended, we hereby only chose 10 resultant items as R_1, merely for the sake of illustrative simplicity:

a_1: A Fuzzy Database Model for Supporting a Concept-Based Query ...
a_2: Fuzzy Database Query Languages and Their Relational Completeness ...
a_3: FSQL (Fuzzy SQL), a Fuzzy Query Language
a_4: Fuzzy Database Modeling with XML
a_5: Fuzzy database systems - Fuzzy Systems, 1995. International Joint ...
a_6: Amazon.com: Fuzzy Database Modeling of Imprecise and Uncertain ...
a_7: Amazon.com: Fuzzy Database Modeling with XML
a_8: Database Schema with Fuzzy Classification and Classification Query ...
a_9: Type-2 Fuzzy Logic - Publications Database
a_{10}: Fuzzy Clustering for Content-based Indexing in Multimedia Database

Suppose the threshold set by the user is $\lambda = 0.7$ and the closeness of these articles generated with a keywords match technique (e.g., [30-34]) is shown in Table 4.

Table 4: The closeness degrees among $a_1, a_2, \ldots, a_{10}$

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	1	0.75	0.9	0.6	0.5	0.5	0.5	0.6	0.6	0.6
a_2		1	0.82	0.6	0.6	0.6	0.6	0.5	0.6	0.6
a_3			1	0.6	0.5	0.5	0.6	0.5	0.6	0.6
a_4				1	0.85	0.9	0.9	0.6	0.6	0.6
a_5					1	0.7	0.75	0.6	0.5	0.5
a_6						1	0.78	0.6	0.6	0.6
a_7							1	0.6	0.5	0.6
a_8								1	0.9	0.95
a_9									1	0.7
a_{10}										1

Now, let us consider how to obtain the outcomes with $RC|_{0.7}$ being 1 and 0.6 respectively. In doing so, the database management systems or search processing functions could execute the extracting process with $\lambda = 0.7$, $\lambda_1 = 0.8$, $\lambda_2 = 0.85$, $\lambda 3 = 0.9$ and $\lambda 4 = 1$ at the back-end as follows.

For $\lambda = 0.7$, R_1 can be classified as $\{a_1, a_2, a_3\}$, $\{a_4, a_5, a_6, a_7\}$ and $\{a_8, a_9, a_{10}\}$, extracting the representative tuples as described in Section 3.3, the final outcome is $S = \{a_3, a_4, a_8\}$, $RC(S)|_{0.7} = 1$.

For $\lambda_1 = 0.8$, R_1 can be classified as $\{a_1, a_2, a_3\}$, $\{a_4, a_5, a_6, a_7\}$ and $\{a_8, a_9, a_{10}\}$, $S_1 = \{a_3, a_4, a_8\}$, $RC(S_1)|_{0.8} = RC(S_1)|_{0.7} = 1$.

For $\lambda_2 = 0.85$, R_1 can be classified as $\{a_1\}$, $\{a_2\}$, $\{a_3\}$, $\{a_4, a_5, a_6, a_7\}$ and $\{a_8, a_9, a_{10}\}$, $S_2 = \{a_1, a_2, a_3, a_4, a_8\}$, $RC(S_2)|_{0.85} = 1$, $RC(S_2)|_{0.7} = 0.6$.

In the same manner, for $\lambda_3 = 0.9$, $S_3 = \{a_1, a_2, a_3, a_4, a_5, a_8\}$, $RC(S_3)|_{0.9} = 1$, $RC(S_3)|_{0.7} = 0.57$.

For $\lambda_4 = 1$, $S_4 = R_1 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}\}$, $RC(S_4)|_1 = 1$, $RC(R_1)|_{0.7} = RC(S_4)|_{0.7} = 0.47$.

Thus, if the threshold $\lambda = 0.7$, the original result of the query is compact at a degree of $RC(R_1)|_{0.7} = 0.47$. The back-end can give the user a totally compact outcome $S = \{a_3, a_4, a_8\}$ (with $RC(S)|_{0.7} = 1$) by default, which is composed of the three items:

"FSQL (Fuzzy SQL), a Fuzzy Query Language"
"Fuzzy Database Modeling with XML"

ISBN: 978-989-95079-6-8

"Database Schema with Fuzzy Classification and Classification Query ..."

If user wants to review more outcomes, e.g., $RC|_{0.7} = 0.6$, then the DBMS can generate $S_2 = \{a_1, a_2, a_3, a_4, a_8\}$, with $RC(S_2)|_{0.7} = 0.6$. Two more specific "fuzzy query" related items are listed: "A Fuzzy Database Model for Supporting a Concept-Based Query ..." and "Fuzzy Database Query Languages and Their Relational Completeness ...". In this way, the list of the literature could be shortened or extended depending upon the need and preference of the user. □

5 Conclusion

This paper has introduced a measure, namely relation compactness, to express the degree of a given relation being compact. Then an approach to evaluating and extracting representative tuples has been proposed. It has been proven that the resultant set of query outcomes is unique, smaller in size, and information equivalent to the original result. Moreover, the approach enables the query/search users to obtain various sizes of results upon their need and preference in light of compactness.

Ongoing research is centering on an application with real data on tuple extraction in both database query and web search contexts.

Acknowledgements

The work was partly supported by the National Natural Science Foundation of China (70890083/70621061) and Tsinghua University's Research Center for Contemporary Management.

References

[1] E.F. Codd, "A relational model for large shared data banks", *Communications of the ACM*, vol. 13, no. 6, pp. 377-387, 1970.

[2] J.F. Baldwin, S.Q. Zhou, "A fuzzy relational inference language", *Fuzzy Sets and Systems*, vol. 14, pp. 155-174, 1984.

[3] B.P. Buckles, F.E. Petry, "A fuzzy representation of data for relational databases", *Fuzzy Sets and Systems*, vol. 7, pp. 213-226, 1982.

[4] H. Prade, C. Testemale, "Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries", *Information Sciences*, vol. 34, pp. 115-143, 1984.

[5] V. Owei, "An Intelligent Approach To Handling Imperfect Information In Concept-Based Natural Language Queries", *ACM Transactions on Information Systems*, vol. 20, no. 3, pp. 291-328, 2002.

[6] Z.M. Ma, W.J. Zhang, W.Y. Ma, "Extending object-oriented databases for fuzzy information modeling", *Information Systems*, vol. 29, no. 5, pp. 421-435, 2004.

[7] L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3-28, 1978.

[8] J. Kacprzyk, S. Zadrozny, "Computing with words in intelligent database querying: standalone and internet-based applications", *Information Sciences*, vol. 134, pp. 71-109, 2001.

[9] S.M. Chen, W.T. Jong, "Fuzzy Query Translation For Relational Database Systems", *IEEE Transactions on System, Man, and Cybernetics-Part B*, vol. 27, no. 4, pp. 714-721, 1997.

[10] P. Bosc, O. Pivert, "SQLf: a relational database language for fuzzy querying", *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 1, pp.1-17, 1995.

[11] P. Bosc, O. Pivert, "About project-selection-join queries addressed to possibilistic relational databases", *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 124-139, 2005.

[12] G.Q. Chen, E.E. Kerre, J. Vandenbulcke, "Normalization based on fuzzy functional dependency in a fuzzy relational data model", *Information Systems*, vol. 21, no. 3, pp. 299-310, 1996.

[13] X.H. Tang, G.Q. Chen, "Equivalence and Transformation of Extended Algebraic Operators in Fuzzy Relational Databases", *Fuzzy Sets and Systems*, vol. 157, no. 12, pp. 1581-1596, 2006.

[14] P. Buche, C. Dervin, O. Haemmerle, R. Thomopoulos, "Fuzzy querying of incomplete, imprecise, and heterogeneously structured data in the relational model using ontologies and rules", *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 3, pp. 373-383, 2005.

[15] D.Y. Choi, "Enhancing the power of Web search engines by means of fuzzy query", *Decision Support Systems*, vol. 35, pp. 31-44, 2003.

[16] S. Fox, K. Karnawat, M. Mydland, S. Dumais, T. White, "Evaluating Implicit Measures to Improve Web Search", *ACM Transactions on Information Systems*, vol. 23, no. 2, pp. 147-168, 2005.

[17] C. Li, K. C. Chang, I. F. Ilyas, S. Song, "RankSQL: Query Algebra and Optimization for Relational Top-k Queries", in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005, pp. 131-142.

[18] H. Yu, S. Hwang, K. C. Chang, "Enabling soft queries for data retrieval", *Information Systems*, vol. 32, pp. 560-574, 2007.

[19] G.Q. Chen, J. Vandenbulcke, E.E. Kerre, "A general treatment of data redundancy in a fuzzy relational data model", *Journal of The American Society for Information Science*, vol. 43, no. 4, pp. 304-311, 1992.

[20] J.C. Cubero, M.A. Vila, "A new definition of fuzzy functional dependency in fuzzy relational databases", *Intelligent Systems*, vol. 9, no. 5, pp. 441-448, 1994.

[21] K.V.S.V.N. Raju, A.K. Majumdar, "Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems", *ACM Transactions on Database Systems*, vol. 13, no. 2, pp. 129-166, 1988.

[22] G.Q. Chen, *Fuzzy logic in data modeling: semantics, constraints, and database design*, Kluwer Academic Publishers, Boston, 1998.

[23] S. Tamura, S. Higuchi, K. Tanaka, "Pattern classification based on fuzzy relations", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 1, no. 1, pp. 61-66, 1971.

[24] H.S. Lee, "An optimal algorithm for computing the max-min transitive closure of a fuzzy similarity matrix", *Fuzzy Sets and Systems*, vol. 123, pp. 129-136, 2001.

[25] H.L. Larsen, R.R. Yager, "Efficient computation of transitive closures", *Fuzzy Sets and Systems*, vol. 38, no. 1, pp. 81-90, 1990.

[26] S. Shenoi, A. Melton, Fan, L.T. "An equivalence classes model of fuzzy relational databases", *Fuzzy Sets and Systems*, vol. 38, pp. 153-170, 1990.

[27] C.E. Shannon, "A Mathematical Theory of Communication", *The Bell System Technical Journal*, vol. 27, pp. 379-423, pp. 623-656, 1948.

[28] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann publishers, 2001.

[29] T.M. Cover, J.A. Thomas, *Elements of Information Theory*, John Wiley, 1991.

[30] Y.Z. Cao, M.S. Ying, G.Q. Chen, "Retraction and Generalized Extension of Computing With Words", *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1238-1250, 2007.

[31] S. Medasani, R. Krishnapuram, and Y. Choi, "Graph matching by relaxation of fuzzy assignments", *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp. 173-182, 2001.

[32] A. Koeller, V. Keelara, "Approximate matching of textual domain attributes for information source integration", in *Proc. of the 2nd international workshop on Information quality in information systems*, 2005, pp. 77-86.

[33] M. Mitra, B.B. Chaudhuri, "Information Retrieval from Documents: A Survey", *Information Retrieval*, vol. 2, pp. 141-163, 2000.

[34] E. Rahm, P.A. Bernstein, "A survey of approaches to automatic schema matching", *The International Journal on Very Large Data Bases*, vol. 10, no. 4, pp. 334-350, 2001.

[35] H.A. Simon, "On the forms of mental representation" in *Minnesota Studies in the Philosophy of Science: Perception and cognition: issues in the foundations of psychology*, C. W. Savage Eds. University of Minnesota Press, 1978.

[36] J.H. Larkin, H.A. Simon, "Why a diagram is (sometimes) worth ten thousand words", *Cognitive Science*, vol. 11, pp. 65-99, 1987.

[37] K. Siau, "Informational and Computational Equivalence in Comparing Information Modeling Methods", *Journal of Database Management*, vol. 15, no. 1, pp. 73-86, 2004.