An Efficient Evaluation of A Fuzzy Equi–Join Using Fuzzy Equality Indicators

Weining Zhang* Ke Wang†

Abstract

We propose a new measure of fuzzy equality comparison based on the similarity of possibility distributions. we define a type of fuzzy equi–join based on the new fuzzy equality comparison, and allow threshold values to be associated with predicates of the join condition. A sort–merge join algorithm based on a partial order of intervals is used to evaluate the fuzzy equi-join. In order for the evaluation to be efficient, we identify various mappings, called fuzzy equality (FE) indicators, that will determine appropriate intervals for fuzzy data with different characteristics. Experiment results from our preliminary simulation of the algorithm show a significant improvement of efficiency when FE indicators are used with the sort–merge join algorithm.

Keywords:

Fuzzy databases, fuzzy equi–join, fuzzy equality indicator, algorithm, performance.

1 Introduction

The AI and the database communities have long recognized that many applications, such as business decision making, medical diagnosis, and criminal justice, have to deal with information that is uncertain or imprecise, and the knowledge–base and database systems should directly support such applications by providing functionalities to store and to manipulate ill–known data. In recent years, various fuzzy data models and fuzzy database systems have been proposed [1, 2, 3, 6, 7, 9, 12, 17, 14, 19]. These models and systems extend relational and object–oriented data models

*Dept. of Math. & CS, Univ. of Lethbridge, Lethbridge, AB T1K 3M4, CANADA. zhang@cs.uleth.ca. The work of this author is supported in part by an NSERC research grant

†Dept. of ISCS, National Univ. of Singapore. wangk@iscs.nus.sg

using the fuzzy set and the possibility theory [15, 16] to provide the ability of representing ill–known data and issuing queries containing soft restrictions. These models can be classified into two categories: the similarity–based and the possibility–based models. In a similarity–based model, some similarity relationships are specified for some attributes so that values of these attributes may be grouped into similarity classes. Each similarity class contains values that are similar to each other to and above a given degree, thus they are indistinct, and form an uncertain representation of a real–world value. In a possibility–based model, an ill–known data is represented by a possibility distribution which describes the possibility for each crisp attribute value to be the actual value of the data. In both types of models, membership degrees may be associated with tuples of a fuzzy relation. While many of these models provide algebraic operations, such as selection, join, and projection, the efficient implementation of these operations has not been sufficiently studied. Some researchers propose to implement fuzzy database systems as the front–end of existing database management systems (DBMS). For example, the OMRON system [9] is implemented as a front–end of a commercial relational DBMS. However, since fuzzy data are more complex than crisp data, and can not be processed directly by the back–end DBMS, it is not clear if such an implementation will be efficient. We believe that efficient implementation of basic algebraic operations in a fuzzy database is an important research issue.

Among the algebraic operations, fuzzy join is an important and expensive one, and its efficient evaluation is more difficult than that of an ordinary join. There are two reasons for the difficulty. The first is that fuzzy joins may have diverse semantics. In a fuzzy join, two tuples may join even if they do not completely satisfy the join condition. The extend to which they do satisfy the join condition is usually represented by some satisfaction degrees, which are used in turn to derive a degree of the resulting tuple from the join to indicate how it is relevant to the join. However, the satisfaction degrees may be measured in different ways so that to result in different meanings of the fuzzy join. Furthermore, various threshold values may be used to restrict the tuples in the result. Since these meanings of fuzzy joins may not be compatible, different algorithms may have to be used according to different meanings. The second reason is the lack of fast access paths. Since the fast access paths in ordinary databases, such as indexing and hashing, rely heavily on the fact that data are crisp, most efficient join algorithms used in conventional relational databases do not apply directly to fuzzy relational databases.

In this paper, we consider the efficient processing of a fuzzy equi–join in a possibility–based fuzzy relational database similar to [9, 8]. Our contribution is the following.

1. We propose a new measure for a fuzzy equality comparison that is based on the similarity of possibility distributions. This new measure provides a natural semantics.

2. We define a new type of fuzzy equi-join that is based on the new fuzzy equality comparison, and allows threshold values to be associated with predicates of the join condition. This allows more flexibility in specifying joins and allows the threshold values to be used in the join algorithm for efficient evaluation.

3. We identify a number of mappings, called fuzzy equality (FE) indicators, which will associate appropriate intervals to fuzzy data so that the evaluation of the join using a sort–merge algorithm is highly efficient.

4. We present experimental results that show a significant improvement of efficiency when FE indicators are used.

There are several works related to ours. In [11, 17], the satisfaction degree of a predicate is measured by both possibility and necessity. Since the possibility measure gives the highest possibility for a predicate to be true, and the necessity measure gives the lowest possibility for the negation of the predicate to be false, the combination of these measures can explicitly represent the uncertainty of the satisfaction degree. However, as pointed in [10], the resulting algebraic operations, such as selection, projection, and join, can not be composed, implying that queries can not be optimized using well–known algebraic transformation techniques.

In [13], the satisfaction degree of a predicate is represented by a possibility distribution, and so are the membership degrees of tuples. Although the resulting algebraic operations can be composed, this method applies only to discrete possibility distributions, and even in that case, since the possibility distributions may contain many elements, the evaluation of a query may be very inefficient.

In [9, 8] the satisfaction degree is measured solely by the possibility. Since the uncertainty is not explicitly represented, and the possibility is the upper–bound, the satisfaction degree obtained using this method may, at times, be counter–intuitive.

Several interval–based join algorithms were proposed in the context of temporal databases [5] or ordinary relational databases [4]. The temporal join algorithms join tuples that have overlapping time periods. The "band join" algorithm in [4] joins a tuple r with a tuple s if the join attribute value of s is within a pre–specified neighborhood of that of r. These algorithms are not suitable for fuzzy equi–join because the join criterion of these algorithms is the overlap of intervals which merely suggests possible joins in a fuzzy equi–join. Furthermore, these algorithms do not use threshold values.

The rest of this paper is organized as follows. In Section 2, we briefly review concepts of fuzzy relations. In Section 3, we present a new fuzzy equality comparison which is then used to define a fuzzy equi–join. In Section 4, we discuss a sort–merge join algorithm for evaluating fuzzy equi–joins. In Section 5, we define the notion of FE indicator. In Section 6, we identify perfect FE indicators for data sets that contain data with identical shape. In Section 7, we identify FE indicators for data sets containing more general types of data. In Section 8, we presents our experimental results. Section 9 concludes the paper.

2 Fuzzy Relations

In this section, we briefly describe the representation of data in a fuzzy relational database which is similar to that presented in [9, 8].

A data is *crisp* if it is certain and precise, and *fuzzy*, otherwise. A *fuzzy (sub)set* F of an ordinary set U is characterized by a membership function

$$\mu_F : U \rightarrow [0, 1].$$

For every (crisp) value $x \in U$, $\mu_F(x)$ is the membership degree of x with respect to (wrt) F, that is, $\mu_F(x) = 1$ (respectively, $0 < \mu_F(x) < 1$, or $\mu_F(x) = 0$) if x is a full member (respectively, a partial member, or not a member) of F. Without loss of generality, x is in F only if $\mu_F(x) > 0$. A fuzzy data v is represented by a possibility distribution restricted by a fuzzy set F in the sense that v is a member of F, and the possibility for v to be a member x of F is exactly $\mu_F(x)$. Thus, the membership function of F is used to represent v. Since an ordinary set is a special case of a fuzzy set, a crisp data can also be represented by a (degenerated) membership function. So, in this paper, all data will be uniformly represented by membership functions.

A membership function can be defined in a number of ways. Over a numerical universe, a membership functions is typically convex (with a convex curve) and normal (at least one member has degree 1). As in [19, 9, 8]. we shall use the following generic parameterized function to define such membership functions.

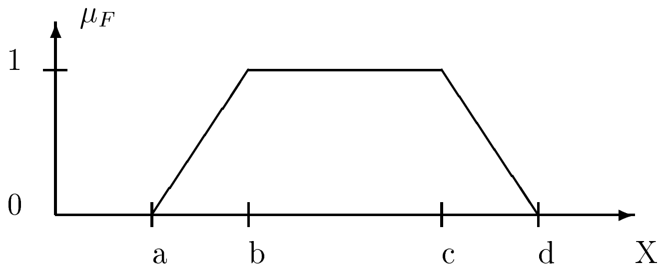

Figure 1: The curve of a generic membership function.

$$MF(a,b,c,d)(x) = \begin{cases} 0, & \text{if } x \le a < b, \text{ or } x < a = b; \\ (x-a)/(b-a), & \text{if } a < x < b; \\ 1, & \text{if } b \le x \le c; \\ (d-x)/(d-c), & \text{if } c < x < d; \\ 0, & \text{if } c < d \le x, \text{ or } c = d < x. \end{cases}$$

where the parameters a, b, c, and d are values in the universe satisfying $a \le b \le c \le d$. In general, the curve of the generic function is a trapezoidal as shown in Figure 1, but can also be some other shapes. For example, $MF(a,b,b,d)$ defines a triangular function since the second and the third parameters are the same. The membership function of a crisp data v is defined by $MF(v,v,v,v)$. If v is defined by $MF(a,b,c,d)$, denoted by $v = MF(a,b,c,d)$, we will use $A(v)$, $B(v)$, $C(v)$ and $D(v)$ to denote the parameters a, b, c, and d, respectively, and define the *support* of v to be the interval $[A(v), D(v)]$, and the *center* of v to be the center of its support. Over a non–numerical universe, a membership function takes the form of $\mu_F = x_1/m_1 + x_2/m_2 + \cdots + x_k/m_k$, where x_i is a value in the universe and m_i is the membership degree of x_i with respect to F. In this case, the degenerated membership function of a crisp value v is $\mu_v = v/1$.

The *universe* of an attribute A, denoted by $U(A)$, is the set of crisp values that may appear in the attribute. The *domain* of an attribute A, denoted by $\mathcal{D}(A)$, is the set of all (both crisp and fuzzy) values defined over $U(A)$. A fuzzy relation R with a schema $(A_1, \ldots, A_n)$ is a fuzzy set of tuples in $\mathcal{D}(A_1) \times \cdots \times \mathcal{D}(A_n)$.

3 A Fuzzy Equi–join

In this section, we first define a fuzzy equality and then use it to define a fuzzy equi–join. The following example shows the needs for a fuzzy equi–join.

Example 3.1 Consider the following relation R.

NAME	OCCUPATION	AGE
Smith	Engineer	20
Alan	Teacher	About 32
Bill	Lawyer	About 34
Cindy	Teacher	Middle age
Mike	Farmer	58

Let *About* $32 = MF(30, 32, 32, 34)$, *About* $34 = MF(32, 34, 34, 36)$, and $Middle\ age = MF(30, 35, 45, 50)$. The query "Find all pairs of persons from R whose ages are equal to a degree no less than 0.5" requires a join of R with itself on the AGE attribute with a fuzzy equality comparison. Since AGE contains fuzzy values, we must determine the degree for two fuzzy ages, say *About* 32 and $Middle\ age$, to be equal (that is, to satisfy the join condition $AGE = AGE$). □

It is obvious from Example 3.1 that the computation of the satisfaction degree of the fuzzy equality comparison is the key to the meaning of the fuzzy equi–join. As mentioned in the Introduction, the existing methods of measuring the satisfaction degree of the join condition [11, 17, 13, 9, 8] are not satisfactory. In the following, we propose a new measured for the fuzzy equality comparison based on the similarity of fuzzy values.

Definition 3.1 Let $\mathcal{D}$ be a set of values. The fuzzy equality on $\mathcal{D}$ is a mapping

$$\sim=: \mathcal{D} \times \mathcal{D} \rightarrow [0, 1]$$

that for every pair of values $v_1 = MF(a_1, b_1, c_1, d_1)$ and $v_2 = MF(a_2, b_2, c_2, d_2)$ in $\mathcal{D}$, gives

$$(v_1 \sim= v_2) = \frac{\int min(\mu_{v1}(x), \mu_{v2}(x))dx}{\int max(\mu_{v1}(x), \mu_{v2}(x))dx}$$

where $\int$ is over the universe on which the membership functions are defined, and is interpreted as a summation if the universe is discrete. □

Intuitively, $\int min(\mu_{v1}(x), \mu_{v2}(x))dx$ is the accumulated membership degrees of the intersection, and $\int max(\mu_{v1}(x), \mu_{v2}(x))dx$ is that of the union of the two fuzzy sets defining v_1 and v_2. If membership functions are defined using the generic function, the fuzzy equality can be evaluated efficiently by

$$(v_i \sim= v_j) = \begin{cases} 0, & \text{if not } v_i \cap v_j; \\ 1, & \text{if } v_i \equiv v_j; \\ \frac{S_{ij}}{(S_i+S_j-S_{ij})}, & \text{otherwise;} \end{cases} \tag{1}$$

where $\cap$ denotes *overlapping*, and $\equiv$ denotes *equivalence* of two values as defined in Section 4; S_i is the area under μ_{v_i}, S_j is the area under μ_{v_j}, and S_{ij} is the area shared by v_i and v_j. According to this definition, the semantics of predicate $(v_1 \sim= v_2)$ is to determine the degree of similarity of v_1 and v_2. It is easy to see that the fuzzy equality is reflexive (that is $(x \sim= x) = 1$) and symmetric (that is $(x \sim= y) = (y \sim= x)$).

Compared with the existing measures in [11, 17, 13, 9, 8], the new measure seems more natural. On one hand, it results in a crisp degree, therefore, allows the algebraic operations to be composed, and on the other hand, the degree is obtained by considering all possible values in both fuzzy data rather than one best possible value of each fuzzy data. Therefore, it is more intuitive. Furthermore, a fuzzy data can be regarded as the subjective representation of a real–world data viewed by an observer. If the observation is reasonably consistent, one may expect that the similarity of fuzzy data indicates the degree for their corresponding real–world values to be identical. We note that for fuzzy data, the satisfaction degree must always be treated as uncertain.

Notice that, for crisp data, the fuzzy equality is the same as the ordinary equality, that is, it is a "hard" comparison. However, it is possible to make it a soft comparison by fuzzifying the crisp data. For example, for each numeric attribute, a small constant $C > 0$ can be specified, and every crisp value v in the attribute can be converted into a fuzzy value "About v" with a membership function $\mu_v = MF(v-C, v, v, v+C)$. By adjusting C, the degree of fuzziness of the comparison can be adjusted easily. For non–numeric attributes, the fuzzification can be based on a pre-specified proximity relation on the universe of the attribute.

Definition 3.2 A fuzzy equi–join of fuzzy relations R and S on attributes $R.A$ and $S.B$ with a threshold $\gamma \geq 0$, denoted by $R \bowtie_{(R.A \sim= S.B) \geq \gamma} S$, is a fuzzy relation T with the membership function defined by

$$\mu_T(xy) = min(\mu_R(x), \mu_S(y), \mu_q(xy)))$$

Algorithm: Sort–Merge Fuzzy Equi–Join

Sort R on $R.A$ based on $\prec$.
Sort S on $S.B$ based on $\prec$.
For each page P_R of R do
  For each tuple r in P_R do
    For each page P_S in $rng_S(r)$ do
      For each tuple s in P_S do
        If $(r[A] \sim= s[B]) \geq \gamma$ then
          if $d = \min((r[A] \sim= s[B]), \mu_R(r), \mu_S(s)) > 0$
          then output $r \circ s$ with degree d.

Figure 2: Algorithm SMFEJ

where x is a tuple in R, y is a tuple in S, and

$$\mu_q(xy) = \begin{cases} 0, & \text{if } (x[A] \sim= y[B]) < \gamma_i, \\ (x[A] \sim= y[B]), & \text{otherwise.} \end{cases} \qquad \square$$

Since this fuzzy equi–join allows the threshold value to be specified, it is very flexible and can be evaluated more efficiently than existing ones.

4 An Interval–based Fuzzy Join Algorithm

We now present a sort–merge join algorithm for evaluating the fuzzy equi–join. The algorithm, named SMFEJ, is shown in Figure 2, and is essentially the same as the one described in [14]. However, since in [14], the fuzzy join is measured by possibility and does not permit threshold values, it needs to be modified to use the fuzzy equality comparison. We briefly review the algorithm in this section, and consider how to use it to evaluate the fuzzy equi–join efficiently in the next section, Notice that a general sort-merge fuzzy equi–join algorithm that permits arbitrary join attributes is not yet known. The SMFEJ algorithm assumes that fuzzy join attributes have numeric universes and membership functions are defined by the generic parameterized function.

The algorithm has two phases, a sorting phase and a joining phase. In the sorting phase, both relations R and S are sorted on their join attributes according to a partial order defined over the attribute values.

Definition 4.1 Let $I_1 = [l_1, h_1]$ and $I_2 = [l_2, h_2]$ be two intervals on a set with a total order. We say that

1. I_1 *overlaps* I_2, denoted by $I_1 \cap I_2$, if $l_1 < h_2$ and $l_2 < h_1$.

2. I_1 is *equivalent* to I_2, denoted by $I_1 \equiv I_2$, if $l_1 = l_2$ and $h_1 = h_2$.

3. I_1 *precedes* I_2, denoted by $I_1 \prec I_2$, if $l_1 < l_2$, or if $l_1 = l_2$ and $h_1 < h_2$.

4. I_1 *precedes or equals to* I_2, denoted by $I_1 \preceq I_2$, if $I_1 \prec I_2$ or $I_1 \equiv I_2$. □

It is obvious that $\prec$ is a partial order on intervals.

Definition 4.2 Let $I(v)$ denote the interval associated with a value v, and v_1 and v_2 be two values over the same universe. $v_1 \prec v_2$ (respectively, $v_1 \equiv v_2$, $v_1 \preceq v_2$, $v_1 \cap v_2$), wrt to their respective intervals, if $I(v_1) \prec I(v_2)$ (respectively, $I(v_1) \equiv I(v_2)$, $I(v_1) \preceq I(v_2)$, $I(v_1) \cap I(v_2)$). Let t_1 and t_2 be two tuples in the same fuzzy relation. $t_1 \prec t_2$ (respectively, $t_1 \preceq t_2$) wrt attribute A if $t[A] \prec t_2[A]$ (respectively, $t_1[A] \preceq t_2[A]$). □

For example, if $v_1 = MF(5, 6, 8, 9)$, $v_2 = MF(6, 7, 7, 8)$, and $v_3 = MF(9, 10, 11, 12)$, and each value v is associated with the interval $[A(v), D(v)]$, we have $v_1 \preceq v_2 \preceq v_3$ and $v_1 \cap v_2$.

In the joining phase, each page of R is read once. For each tuple r in R, the S–tuples that may join with r are in the range of r as defined below.

Definition 4.3 Let S be sorted on $S.B$ according to $\prec$. For any tuple $r \in R$, the *range (of S tuples) of* r, denoted by $rng_S(r)$, is the longest sequence $[s_1, s_2, \ldots, s_k]$ of tuples in S, such that $s_i \preceq s_j$ for all $1 \leq i < j \leq k$, $r[A] \cap s_1[B]$, and $r[A] \cap s_k[B]$. □

Thus, only those pages containing $rng_S(r)$ need to be read into a buffer and those tuples in $rng_S(r)$ need to be scanned to see if they actually join with r.

If the size of the buffer available to the algorithm is big enough to hold the entire $rng_S(r)$ for every $r \in R$, after a tuple of R is processed, either the entire range of the next tuple of R or a large portion of it will already be in the buffer. Thus, the time complexity of the algorithm will be $O(cost(sorting) + n + m)$, where n and m are the sizes of R and S, respectively, in pages, and $cost(sorting)$ is the time spent on sorting R and S including both IO and CPU time. Typically $cost(sorting) = n \log n + m \log m$. In more general cases, let h be the average number of pages of S to be scanned for each page of R. The time complexity of the join will become $O(cost(sorting) + n \times h)$. Clearly, h depends on the threshold of the join, the size of the buffer, and the values in the joining attributes. Although in the worst case, for example, $\gamma = 0$ and only 1 page of buffer is available to S, $h = m$, we expect h to be much smaller than m in practice.

5 Fuzzy Equality Indicators

We now consider how to use SMFEJ algorithm to evaluate fuzzy equi–join efficiently. For practical reasons, we assume a limited buffer space available to the algorithm. Thus, during the joining phase, some pages in $rng_S(r)$ for some tuple r may have to be swapped out of the buffer to make rooms for other pages, and then be swapped back in because they are also in the range of the next R–tuple. In this case, the key to the efficient evaluation of fuzzy equi–join is to determine the appropriate intervals to associate with the fuzzy attribute values, as shown in the following example.

Example 5.1 Assume that R has a tuple r with $r[A] = MF(10, 10, 40, 40)$, and S contains exactly the tuples $s_1, \ldots, s_9$ with $s_1[B] = MF(5, 5, 20, 20)$, $s_2[B] = MF(6, 6, 9, 9)$, $s_3[B] = MF(10, 10, 40, 40)$, $s_4[B] = MF(11, 11, 16, 16)$, $s_5[B] = MF(15, 15, 45, 45)$, $s_6[B] = MF(20, 20, 30, 30)$, $s_7[B] = MF(20, 20, 50, 50)$, $s_8[B] = MF(32, 32, 36, 36)$, and $s_9[B] = MF(35, 35, 60, 60)$. If each join attribute value v is associated with its supports, $[A(v), D(v)]$, we have $s_1 \prec s_2 \prec s_3 \cdots \prec s_9$, $r[A] \cap s_1[B]$, and $r[A] \cap s_9[B]$. Thus, $rng_S(r)$ is $[s_1, \ldots, s_9]$. With a little calculation, we have $(r[A] \sim= s_i[B])$ equal to 0.29, 0, 1, 0.17, 0.71, 0.33, 0.5, 0.13, and 0.1, for $i = 1, 2, \ldots, 9$, respectively. If the join condition is $(R.A \sim= S.B) \geq 0.5$, only s_3, s_5, and s_7 will join with r. If the threshold value is raised from 0.5 to 0.9, only s_3 will join with r. In both cases, however, all S tuples must be scanned.

Now, suppose that $\gamma = 0.5$, and that, based on a method to be discussed later, we assign the interval $[17.5, 32.5]$ to $r[A]$ and the intervals $[8.75, 17.25]$, $[6.75, 8.25]$, $[17.5, 32.5]$, $[12.25, 14.75]$, $[22.5, 37.5]$, $[22.5, 27.5]$, $[27.5, 42.5]$, $[33, 35]$, and $[41.25, 53.75]$ to $s_i[B]$, $i = 1, 2, \ldots, 9$, respectively. Then $s_2 \prec s_1 \prec s_4 \prec s_3 \prec s_6 \prec s_5 \prec s_7 \prec s_8 \prec s_9$. It can be verified that $rng_S(r)$ becomes $[s_3, s_6, s_5, s_7]$, that is, reduced by more than 50%. Thus, the same result can be obtained by scanning a less number of tuples. □

As indicated by Example 5.1, it is possible that not all tuples in $rng_S(r)$ join with r, and the higher the threshold value is, the more such irrelevant tuples are in $rng_S(r)$. Since every tuple in $rng_S(r)$ must be scanned during the join, the efficiency can be improved by moving as many irrelevant tuples out of $rng_S(r)$ as possible. This can be achieved if the assignment of intervals to the attribute values is an appropriate function of the threshold value, so that the sorting will rearrange the tuples appropriately. We now formalize these ideas.

Definition 5.1 Let $\mathcal{D}$ be a set of values, and $\mathcal{I}$ be the set of intervals defined on the set of real numbers $\mathcal{R}$, that is, $\mathcal{I} = \{[x, y] \mid x \leq y \text{ and } x, y \in \mathcal{R}\}$.

1. A mapping $f : \mathcal{D} \times [0, 1] \to \mathcal{I}$ is a *fuzzy equality indicator over* $\mathcal{D}$ (or simply an FE indicator) if for any $v_1, v_2 \in \mathcal{D}$ and $\gamma \in (0, 1]$, $(v_1 \sim= v_2) \geq \gamma$ implies $f(v_1, \gamma) \cap f(v_2, \gamma)$.

2. An FE indicator f is *stronger* than another FE indicator g, if for every $v \in \mathcal{D}$ and every $\gamma \in (0, 1]$, $f(v, \gamma) \subseteq g(v, \gamma)$.

3. An FE indicator f is *perfect* if for every pair of $v_1, v_2 \in \mathcal{D}$ and every $\gamma \in (0, 1]$,

   (a) $f(v_1, \gamma) \cap f(v_2, \gamma)$ if and only if $(v_1 \sim= v_2) \geq \gamma$.

   (b) if $(v_1 \sim= v_2) < \gamma$, then for every $v \in \{x \mid x \in \mathcal{D} \text{ and } (x \sim= v_2) \geq \gamma\}$, either $f(v_1, \gamma) \prec f(v, \gamma)$ or $f(v_1, \gamma) \succ f(v, \gamma)$. □

Intuitively, if f is an FE indicator over the domain of the join attributes of a fuzzy equi–join, by assigning intervals to join attribute values using f, it guarantees that after sorting according to $\prec$, for every tuple r in R, every relevant S–tuple is in $rng_S(r)$. However, it does not guarantee that every tuple in $rng_S(r)$ joins with r unless f is a perfect FE indicator. If both f and g are FE indicators over the domains of the join attributes, and f is stronger than g, f will assign smaller intervals to values than g would, thus may move more irrelevant tuples out of $rng_S(r)$ for every r.

The following example will show the effect of a perfect FE indicator.

Example 5.2 Assume that R has tuples r_1 and r_2 with $r_1[A] = MF(3, 5, 5, 7)$, and $r_2[A] = MF(5, 7, 7, 9)$, and S contains exactly the tuples $s_1, \ldots, s_7$ with $s_1[B] = MF(1, 3, 3, 5)$, $s_2[B] = MF(2, 4, 4, 6)$, $s_3[B] = MF(3, 5, 5, 7)$, $s_4[B] = MF(4, 6, 6, 8)$, $s_5[B] = MF(5, 7, 7, 9)$, $s_6[B] = MF(6, 8, 8, 10)$, and $s_7[B] = MF(7, 9, 9, 11)$,

Suppose the join condition is $(R.A \sim= S.B) \geq \gamma$. If $\gamma = 0.1$, using the perfect FE indicator given in Corollary 6.3, we have $s_1 \prec s_2 \prec s_3 \cdots \prec s_7$, $rng_S(r_1)$ consists of $s_1, \ldots s_5$, and $rng_S(r_2)$ consists of $s_3, \ldots s_7$. If $\gamma = 0.5$, the order of S tuples is the same, but $rng_S(r_1)$ becomes s_2, s_3, s_4, and $rng_S(r_2)$ becomes s_5, s_6, s_7. If $\gamma = 0.7$, $rng_S(r_1)$ becomes s_3, and $rng_S(r_2)$ becomes s_6. In each case, every tuple in $rng_S(r_i)$ actually joins with r_i, for $i = 1, 2$. □

Notice that although Definition 5.1 specifies the necessary properties for a mapping to become an FE indicator, it can not be used directly to identify an appropriate FE indicator, among possibly infinite number of mappings. In the following sections, we shall identify a number of desirable FE indicators and while doing so, we shall keep in mind the following criteria of the desirability.

1. *Low computational complexity*, that is, the computation of determining the interval should be simple, say, in a constant time.

2. *Strong filtering effect*, that is, the FE indicator should be as strong as possible, and preferably be perfect.

3. *Smaller interval for higher threshold*, that is, the size of the interval generated by the FE indicator for any value should decrease as the threshold increases.

Since whether a particular mapping is desirable or not will in general depend on the characteristics of the mapping and the type of values allowed in the join attributes, We shall consider a type of mappings, called the $\mathcal{F}$ mappings, and three types of attribute values distinguished by the shapes of their membership functions. More specifically, an $\mathcal{F}$ mapping is of the form

$$\mathcal{F}(v,\gamma) = [A(v) + \Phi(v,\gamma), D(v) - \Phi(v,\gamma)]$$

where Φ is a monotonically increasing function of γ, and $\Phi(v,0) \geq 0$. The $\mathcal{F}$ mappings have the following properties.

1. *Bounded with a fixed center.* For every value v, $\mathcal{F}(v,\gamma)$ is contained in the support of v with the center at $A(v) + \frac{1}{2}(D(v) - A(v))$.

2. *Monotonically decreasing.* For any value v and $0 < \gamma' \leq \gamma \leq 1$, $\mathcal{F}(v,\gamma) \subseteq \mathcal{F}(v,\gamma')$.

The set of allowed values in the join attributes may be one of the three types, namely, it may have values whose membership functions have the identical shape, similar shapes, or arbitrary shapes. Formally, the identical shape and similar shape are defined as follows.

Definition 5.2 Two values, v_1 and v_2, in the same domain have *the identical shape* if $B(v_1) - A(v_1) = B(v_2) - A(v_2)$, $C(v_1) - B(v_1) = C(v_2) - B(v_2)$, and $D(v_1) - C(v_1) = D(v_2) - C(v_2)$; and *similar shapes* if $B(v_1) - A(v_1) = B(v_2) - A(v_2)$, and $D(v_1) - C(v_1) = D(v_2) - C(v_2)$. □

In the rest of the paper, for any universe U of joining attributes, we use $\mathcal{D}$ to denote the set of all values whose membership functions can be defined over U using the generic function. We use $\mathcal{D}_I$ (respectively, $\mathcal{D}_S$) to denote a set that contains all values in $\mathcal{D}$ whose membership functions have an identical shape (respectively, similar shapes). For convenience, we call $\mathcal{D}_I$ (respectively $\mathcal{D}_S$, and $\mathcal{D}$) the set of identical (respectively similar, and arbitrary) data. For the purpose of subsequent sections, it is not important to know the shapes of the data in $\mathcal{D}_I$ and $\mathcal{D}_S$. However, it should be pointed out that a $\mathcal{D}$ may contain several $\mathcal{D}_S$s and each $\mathcal{D}_S$ may contain several $\mathcal{D}_I$s. In fact, a $\mathcal{D}$ is partitioned by $\mathcal{D}_S$s with different collections of similar shapes and each $\mathcal{D}_S$ is partitioned by $\mathcal{D}_I$s with different shapes. Thus, without loss of generality, we may assume that $\mathcal{D}_I \subset \mathcal{D}_S \subset \mathcal{D}$. As examples, the join attribute values in Example 5.1 form a $\mathcal{D}_S$ and those in Example 5.2 form a $\mathcal{D}_I$.

6 An FE Indicator Over a Set of Identical Data

In this section, we first discuss some properties of the data in $\mathcal{D}_I$ and then use these properties to show that a specific mapping is a perfect FE indicator for $\mathcal{D}_I$.

Consider any two values w_i and w_j in $\mathcal{D}_I$. Since they have the same shape, we have $A(w_i) - A(w_j) = B(w_i) - B(w_j) = C(w_i) - C(w_j) = D(w_i) - D(w_j) = k$, where k is the *distance* between w_i and w_j. This property leads to a simple computation of the satisfaction degree of fuzzy equality. Assume that $w_i = MF(a_i, b_i, c_i, d_i)$, $w_j = MF(a_j, b_j, c_j, d_j)$, $a_i \leq a_j$, and $w_i \cap w_j$. If $b_j \leq c_i$, the area shared by w_i and w_j has a trapezoidal shape, and

$$(w_i \sim= w_j) = \frac{2(d_i - a_j) - [(d_i - a_i) - (c_i - b_i)]}{2[(d_i - a_i) + (c_i - b_i)] - 2(d_i - a_j) + [(d_i - a_i) - (c_i - b_i)]} \tag{2}$$

If $b_j \geq c_i$, the shared area has a triangular shape, thus

$$(w_i \sim= w_j) = \frac{(d_i - a_j)^2}{2[(d_i - a_j)^2 - (c_i - b_i)^2] - (d_i - a_j)^2} \tag{3}$$

Notice that the formulas are expressed so that each part in a pair of parentheses results in a positive value (which represents the length of a line segment on the domain of the membership functions). Since values have the identical shape, the formulas can be expressed mainly by using parameters of one value. In the case of $b_j = c_i$, both formulas give the same result:

$$(w_i \sim= w_j) = \frac{(d_i - a_i) - (c_i - b_i)}{(d_i - a_i) + 3(c_i - b_i)} = \theta.$$

It is straightforward to see that if $b_j \leq c_i$, $(w_i \sim= w_j) \geq \theta$; and if $b_j \geq c_i$, $(w_i \sim= w_j) \leq \theta$. Notice that if w_i and w_j do not overlap with each other, $(w_i \sim= w_j) = 0$.

The following Lemma characterizes the relationship between the positions of values in $\mathcal{D}_I$ and the fuzzy equality among these values.

Lemma 6.1 For any $v_1, v_2, v_3 \in \mathcal{D}_I$ such that $A(v_1) \leq A(v_2) \leq A(v_3)$, $(v_2 \sim= v_3) \geq (v_1 \sim= v_3)$ and $(v_1 \sim= v_2) \geq (v_1 \sim= v_3)$.

Proof. Since the values have identical shape, $A(v_1) \leq A(v_2) \leq A(v_3)$ implies that for each $X \in \{B, C, D\}$, $X(v_1) \leq X(v_2) \leq X(v_3)$. The Lemma is trivially true if v_1 and v_3 do not overlap, since then $(v_1 \sim= v_3) = 0$. So let us assume $v_1 \cap v_3$, which also implies that $v_1 \cap v_2$ and $v_2 \cap v_3$. We need to consider three cases.

Case 1: $C(v_1) \leq B(v_3) \leq C(v_2)$.

In this case, $(v_1 \sim= v_3)$ will be computed using Formula (3), while $(v_2 \sim= v_3)$ will be computed using Formula (2). Since all three values have the same shape, it follows that $(v_1 \sim= v_3) \leq (v_2 \sim= v_3)$. Now consider $(v_1 \sim= v_2)$. Two cases are possible, namely, either $B(v_2) \leq C(v_1)$ or $B(v_2) \geq C(v_1)$. In the former case, $(v_1 \sim= v_2)$ will be computed using Formula (2), thus is at least as large as $(v_1 \sim= v_3)$ since all three values have identical shape. In the latter case, $(v_1 \sim= v_2)$ is computed using the same Formula (3) as $(v_1 \sim= v_3)$ does. However, since $A(v_2) \leq A(v_3)$, $[D(v_1) - A(v_3)] \leq [D(v_1) - A(v_2)]$, thus, $(v_1 \sim= v_3) \leq (v_1 \sim= v_2)$.

Case 2: $B(v_3) \leq C(v_1)$.

In this case, $(v_1 \sim= v_3)$, $(v_2 \sim= v_3)$, and $(v_1 \sim= v_2)$ are computed using Formula (2). Since $[D(v_1) - A(V_3)] \leq [D(v_2) - A(v_3)]$, $(v_1 \sim= v_3) \leq (v_2 \sim= v_3)$. Since $[D(v_1) - A(v_2)] \leq [D(v_1) - A(v_2)]$, $(v_1 \sim= v_3) \leq (v_2 \sim= v_1)$.

Case 3: $B(v_3) \geq C(v_2)$.

This case is similar to Case 2, except that Formula (3) is used instead. $\square$

The following theorem identifies a perfect FE indicator over $\mathcal{D}_I$.

Theorem 6.2 The mapping

$$g(v, \gamma) = \begin{cases} g_1(v, \gamma) & \text{if } 0 < \gamma < \theta, \\ g_2(v, \gamma) & \text{if } \theta \leq \gamma \leq 1. \end{cases}$$

is a perfect FE indicator over $\mathcal{D}_I$, where

$$\begin{aligned} g_1(v, \gamma) &= [A(v) + \frac{1}{2}\sqrt{\frac{2\gamma}{1+\gamma}[(D(v) - A(v))^2 - (C(v) - B(v))^2]}, \\ & \quad D(v) - \frac{1}{2}\sqrt{\frac{2\gamma}{1+\gamma}[(D(v) - A(v))^2 - (C(v) - B(v))^2]}] \end{aligned} \tag{4}$$

$$
\begin{aligned}
g_2(v,\gamma) &= [A(v)+\frac{1+3\gamma}{4+4\gamma}(D(v)-A(v))-\frac{1-\gamma}{4+4\gamma}(C(v)-B(v)), \\
&\quad D(v)-\frac{1+3\gamma}{4+4\gamma}(D(v)-A(v))+\frac{1-\gamma}{4+4\gamma}(C(v)-B(v))] \\
\theta &= \frac{(D(v)-A(v))-(C(v)-B(v))}{(D(v)-A(v))+3(C(v)-B(v))}
\end{aligned}
\tag{5}
$$

Proof. We need to show that for any v_i and v_j in $\mathcal{D}_I$, first, when $0<\gamma\leq\theta$, $g_1(v_i,\gamma)\cap g_1(v_j,\gamma)$ if and only if $(v_i \sim= v_j)\geq\gamma$, and when $\theta\leq\gamma\leq 1$, $g_2(v_i,\gamma)\cap g_2(v_j,\gamma)$ if and only if $(v_i \sim= v_j)\geq\gamma$; second, if $(v_i \sim= v_j)<\gamma$, then either $g_k(v_i,\gamma)\prec g_k(v_m,\gamma)$ or $g_k(v_i,\gamma)\succ g_k(v_m,\gamma)$ for all $v_m\in\{x \mid x\in\mathcal{D}_I \text{ and } (x\sim= v_j)\geq\gamma\}$, where $k=1,2$ depending on whether γ is less than θ.

Before we continue, the following properties of mappings g_k, where $k=1,2$, are worthwhile to mention.

Prop 1: $g_k(v,0)=[A(v),D(v)]$, $g_2[v,1]=[A(v)+(D(v)-A(v))/2, D(v)-(D(v)-A(v))/2]$, and $g_1(v,1)=g_2(v,1)$ if v has a triangular shape.

Prop 2: For any $0\leq\gamma\leq\gamma'\leq 1$, $g_k(v,\gamma')\subseteq g_k(v,\gamma)$, and they have the same center $A(v)+(D(v)-A(v))/2$.

Prop 3: For any two values $v_i,v_j\in\mathcal{D}_I$, and $0\leq\gamma\leq 1$, $g_k(v_i,\gamma)$ and $g_k(v_j,\gamma)$ have the same length, and if $A(v_i)\leq A(v_j)$, $g_k(v_i,\gamma)\preceq g_k(v_j,\gamma)$.

Prop 4: $g_1(v,\theta)=g_2(v,\theta)$.

We now proceed with the proof. Without loss of generality, let $v_i=MF(a_i,b_i,c_i,d_i)$, $v_j=MF(a_j,b_j,c_j,d_j)$, and $a_i\leq a_j$.

Let $0<\gamma<\theta$.

(**If**) Assume that $(v_i\sim= v_j)\geq\gamma$. We show that $g_1(v_i,\gamma)\cap g_1(v_j,\gamma)$, that is,

$$a_j+\frac{1}{2}\sqrt{\frac{2\gamma}{1+\gamma}[(d_j-a_j)^2-(c_j-b_j)^2]}\leq d_i-\frac{1}{2}\sqrt{\frac{2\gamma}{1+\gamma}[(d_i-a_i)^2-(c_i-b_i)^2]}$$

or equivalently

$$(d_i-a_j)\geq\frac{1}{2}\sqrt{\frac{2\gamma}{1+\gamma}[(d_i-a_i)^2-(c_i-b_i)^2]} \tag{6}$$

As mentioned before, $(v_i\sim= v_j)$ must be computed using one of the Formulas (3) and (2). If Formula (3) must be used, $(v_i\sim= v_j)\geq\gamma$ implies that

$$\frac{(d_i-a_j)^2}{2[(d_i-a_i)^2-(c_i-b_i)^2]-(d_i-a_j)^2}\geq\gamma$$

which, by solving for $(d_i - a_j)$, gives the inequality (6). If Formula (2) must be used, we can find a value $v_k \in \mathcal{D}_I$, such that $A(v_k) \geq A(v_j)$, $(v_i \sim= v_k) \geq \gamma$, and the fuzzy equality must be computed using Formula (3). According to what we just proved, $(v_i \sim= v_k) \geq \gamma$ implies $g_1(v_i, \gamma) \cap g_1(v_k, \gamma)$. Since $A(v_i) \leq A(v_j) \leq A(v_k)$, by Lemma 6.1, $g_1(v_i, \gamma) \preceq g_1(v_j, \gamma) \preceq g_1(v_k, \gamma)$. Thus $g_1(v_i, \gamma) \cap g_1(v_k, \gamma)$ implies, $g_1(v_i, \gamma) \cap g_1(v_j, \gamma)$.

(**Only if**) Assume that $g_1(v_i, \gamma) \cap g_1(v_j, \gamma)$, we prove that $(v_i \sim= v_j) \geq \gamma$.

Again, $(v_i \sim= v_j)$ may be computed using either Formula (3) or Formula (2). If the former is used, since by assumption, inequality (6) must hold, we directly have

$$(v_i \sim= v_j) \geq \frac{\frac{\gamma}{1+\gamma}(d_i - a_i + c_i - b_i)}{(d_i - a_i + c_i - b_i) - \frac{\gamma}{1+\gamma}(d_i - a_i + c_i - b_i)} = \gamma.$$

If Formula (2) must be used, we already have $(v_i \sim= v_j) \geq \theta \geq \gamma$.

Let $\theta \leq \gamma \leq 1$.

(**If**) Assume that $(v_i \sim= v_j) \geq \gamma$. In this case, Formula (2) must be used to compute $(v_i \sim= v_j)$, and we have

$$(v_i \sim= v_j) = \frac{2(d_i - a_j) - [(d_i - a_i) - (c_i - b_i)]}{2(d_i - a_i + c_i - b_i) - 2(d_i - a_j) + [(d_i - a_i) - (c_i - b_i)]} \geq \gamma$$

By solving for $(d_i - a_j)$, we get

$$(d_i - a_j) \geq \frac{1+3\gamma}{2+2\gamma}(d_i - a_i) - \frac{1-\gamma}{2+2\gamma}(c_i - b_i)$$

which is equivalent to

$$a_j + \frac{1+3\gamma}{4+4\gamma}(d_j - a_j) - \frac{1-\gamma}{4+4\gamma}(c_j - b_j) \leq d_i - \frac{1+3\gamma}{2+2\gamma}(d_i - a_i) - \frac{1-\gamma}{2+2\gamma}(c_i - b_i)$$

and thus $g_2(v_i, \gamma) \cap g_2(v_j, \gamma)$.

(**Only if**) Now assume that $g_2(v_i, \gamma) \cap g_2(v_j, \gamma)$, that is,

$$(d_i - a_j) \geq \frac{1+3\gamma}{2+2\gamma}(d_i - a_i) - \frac{1-\gamma}{2+2\gamma}(c_i - b_i)$$

We claim that in this case, $(v_i \sim= v_j)$ must be computed using Formula (2). To see this, notice that $(d_i - a_j)$ is monotonically increasing with respect to γ. Thus for $\theta \leq \gamma \leq 1$, $g_2(v_i, \gamma) \cap g_2(v_j, \gamma)$

implies that

$$(d_i - a_j) \geq \frac{1+3\theta}{2+2\theta}(d_i - a_i) - \frac{1-\theta}{2+2\theta}(c_i - b_i) = (d_i - a_i) - (c_i - b_i)$$

If $(v_i \sim= v_j)$ must computed using Formula (3), we must have $(d_i - a_j) < (d_i - a_i) - (c_i - b_i)$, which implies that $g(v_i, \gamma)$ does not overlap with $g_2(v_j, \gamma)$, a contradiction to previous assumption. Now, since $(v_i \sim= v_j)$ must be computed using Formula (2), we have

$$(v_i \sim= v_j) \geq \frac{\frac{2\gamma}{1+\gamma}(d_i - a_i + c_i - b_i)}{2(d_i - a_i + c_i - b_i) - \frac{2\gamma}{1+\gamma}(d_i - a_i + c_i - b_i)} = \gamma$$

Finally, consider any $u, v, w \in \mathcal{D}_I$. Let $(u \sim= w) < \gamma$ and $(v \sim= w) \geq \gamma$. If $A(u) \leq A(w)$ and $A(v) \leq A(w)$, by Lemma 6.1, we must have $A(u) \leq A(v) \leq A(w)$. By Prop 3, $g_k(u, \gamma) \preceq g_k(v, \gamma) \preceq g_k(w, \gamma)$, where $k = 1$ if $0 < \gamma < \theta$, and $k = 2$ if $\theta \leq \gamma \leq 1$. Similarly, if $A(u) \geq A(w)$ and $A(v) \geq A(w)$, then $g_k(w, \gamma) \preceq g_k(v, \gamma) \preceq g_k(u, \gamma)$. □

Notice that g has to be defined in terms of two mappings g_1 and g_2, since, as shown by the following example, neither g_1 nor g_2 is a perfect FE indicator over $\mathcal{D}_I$.

Example 6.1 Consider three values $v_1 = MF(10, 20, 40, 50)$, $v_2 = MF(35, 45, 65, 75)$, and $v_3 = MF(20, 30, 50, 60)$. Obviously, v_1, v_2, and v_3 have the identical shape, and $\theta = 0.2$. Notice that $(v_1 \sim= v_2) = 0.1034$, and $(v_1 \sim= v_3) = 0.5$. If $\gamma = 0.1034$, we have $\gamma < \theta$ and $(v_1 \sim= v_2) \geq \gamma$, but $g_2(v_1, 0.1034) = [17.8113, 42.1887]$ does not overlap with $g_2(v_2, 0.1034) = [42.8113, 67.1887]$. Thus, when $\gamma < \theta$, g_2 is not even an FE indicator, let alone a perfect one. If $\gamma = 0.52$, we have $\gamma > \theta$, and $(v_1 \sim= v_3) < \gamma$, but $g_1(v_1, 0.52) = [24.327, 35.673]$ will still overlap with $g_1(v_3, 0.52) = [34.327, 45.673]$. Thus, when $\gamma \geq \theta$, g_1 is not a perfect FE indicator. In addition, for any value $v = MF(a, a, d, d)$, $g_1(v, \gamma) = [a, d]$ disregarding what value γ is, thus is not very useful at all. □

Depending on the shape of the values in $\mathcal{D}_I$, variants of the perfect FE indicator may also exist, as indicated by the following Corollaries.

Corollary 6.3 If the shape of values in $\mathcal{D}_I$ is triangular, the mapping $g_3(v, \gamma) = [A(v) + \sqrt{\frac{\gamma}{2+2\gamma}}(D(v) - A(v)), D(v) - \sqrt{\frac{\gamma}{2+2\gamma}}(D(v) - A(v))]$ is a perfect FE indicator over $\mathcal{D}_I$. □

An example of $\mathcal{D}_I$ that has triangular shaped values is a set of fuzzy numbers.

Corollary 6.4 If the shape of the values in $\mathcal{D}_I$ is rectangular, the mapping $g_4(v, \gamma) = [A(v) + \frac{\gamma}{1+\gamma}(D(v) - A(v)), D(v) - \frac{\gamma}{1+\gamma}(D(v) - A(v))]$ is a perfect FE indicator over $\mathcal{D}_I$. □

7 FE Indicators Over Sets of Similar or Arbitrary Data

We now consider data sets $\mathcal{D}_S$ and $\mathcal{D}$. Unfortunately, none of the $\mathcal{F}$ mappings is a perfect FE indicator over these types of data sets, as indicated by the following proposition.

Proposition 7.1 No $\mathcal{F}$ mapping is a perfect FE indicator over $\mathcal{D}_S$ (respectively, $\mathcal{D}$).

Proof. We prove that for any $\mathcal{F}$ mapping f, there are two values v_1 and v_2 in $\mathcal{D}_S$, such that for some $0 < \gamma \leq 1$, $f(v_1, \gamma) \cap f(v_2, \gamma)$, but $(v_1 \sim= v_2) < \gamma$.

Notice that for any value v and $0 < \gamma \leq \gamma' \leq 1$, $f(v, \gamma) \subseteq f(v, \gamma')$ and they have the same center. Since the center of $f(v, \gamma)$ is $A(v) + \frac{1}{2}(D(v) - A(v))$, if two values v_1 and v_2 in $\mathcal{D}_S$ have the same center, $f(v_1, \gamma) \cap f(v_2, \gamma)$ for every $0 < \gamma \leq 1$. Notice that in this case, since v_1 and v_2 have the similar shapes, either the graphs of v_1 and v_2 are identical to each other or one of them contains the other completely.

Now, since $\mathcal{D}_S$ contains all values that have a shape similar to a given shape, there must be two values v_1 and v_2, such that they have the same center, and $S_1/S_2 < m$ for some $0 < m \leq 1$, where S_1 and S_2 are the areas of the graphs of v_1 and v_2, respectively. Then, for $\gamma = m$, we have $f(v_1, \gamma) \cap f(v_2, \gamma)$ but $(v_1 \sim= v_2) < \gamma$.

Since $\mathcal{D}$ has partitions that are $\mathcal{D}_S$, the same holds for $\mathcal{D}$ as well. □

Thus, at the best, we can only identify the strongest FE indicators for $\mathcal{D}_S$ and $\mathcal{D}$ among $\mathcal{F}$ mappings. Although one may consider mappings other than $\mathcal{F}$ mappings for identifying perfect FE indicators for $\mathcal{D}$ or $\mathcal{D}_S$, the task will be very difficult due to too many mappings to consider. Besides, such perfect FE indicators may not exist because whether or not two arbitrary values v_1 and v_2 in such a set will satisfy $(v_1 \sim= v_2) \geq \gamma$ will depend, in general, on the parameters of both values, rather than any one value alone. Notice that as indicated by Example 6.1, none of g, g_1 and g_2 mappings is a strongest FE indicator for $\mathcal{D}_S$ or $\mathcal{D}$. In the following, we shall restrict our attention to the following subset of $\mathcal{F}$ mappings, the f_k mappings, of the form

$$f_k(v, \gamma) = [A(v) + \frac{\gamma}{k}(D(v) - A(v)), D(v) - \frac{\gamma}{k}(D(v) - A(v))],$$

where $k \geq 2$ is an integer. As a special case, let $f_\infty(v, \gamma) = [A(v), D(v)]$. The following lemma describes some useful properties of f_k mappings over $\mathcal{D}$.

Lemma 7.2 Consider the data set $\mathcal{D}$ and the f_k mappings.

1. The mapping f_∞ is an FE indicator over $\mathcal{D}$.

2. If both f_i and f_j are FE indicators over $\mathcal{D}$ and $i < j$, f_i is stronger than f_j.

3. There exists a unique strongest FE indicator over $\mathcal{D}$ among f_k mappings.

Proof. We prove the statements in the given order.

1. For any two values v_1 and v_2 in $\mathcal{D}$, assume that $(v_1 \sim= v_2) \geq \gamma$, for some $\gamma > 0$. By Definition 3.1, $\int Min(\mu_{v_1}(x), \mu_{v_2}(x))dx > 0$. This implies that for at least one value of x, say $x = i$, both $\mu_{v_1}(i) > 0$ and $\mu_{v_2}(i) > 0$. Thus $i \in [A(v_1), D(v_1)]$ and $i \in [A(v_2), D(v_2)]$. That is, $f_\infty(v_1, \gamma)$ and $f_\infty(v_2, \gamma)$ overlap with each other. Therefore, f_∞ is an FE indicator.

2. Let f_i and f_j be FE indicators over $\mathcal{D}$, where $i < j$. For any $v \in \mathcal{D}$ and any $1 \geq \gamma \geq 0$, we have $\gamma \cdot (D(v) - A(v))/i \geq \gamma \cdot (D(v) - A(v))/j$. Thus, $A(v) + \gamma \cdot (D(v) - A(v))/i \geq A(v) + \gamma \cdot (D(v) - A(v))/j$, and $D(v) - \gamma \cdot (D(v) - A(v))/i \leq D(v) - \gamma \cdot (D(v) - A(v))/j$. That is, $f_i(v, \gamma) \subseteq f_j(v, \gamma)$. By Definition 5.1, f_i is stronger than f_j.

3. Statement 1 indicates the existence of FE indicators in f_k family, and Statement 2 implies that no two FE indicators in the family will be equally strong when the threshold value is greater than 0. It follows that there is a unique strongest FE indicator in the family of mappings. □

Since $\mathcal{D}_S$ is a subset of a $\mathcal{D}$, these properties hold for $\mathcal{D}_S$ as well.

The following theorems identify strong FE indicators among f_k mappings for data sets $\mathcal{D}_S$ and $\mathcal{D}$, respectively.

Theorem 7.3 Among f_k mappings, f_2 is the strongest FE indicator over $\mathcal{D}_S$.

Proof. We only need to show that f_2 is an FE indicator over $\mathcal{D}_S$. Then it is the strongest by Lemma 7.2. By Definition 5.1, we need to show that, for any two values $v_1, v_2 \in \mathcal{D}_S$, and any threshold value $0 < \gamma \leq 1$, if $(v_1 \sim= v_2) \geq \gamma$, $f_2(v_1, \gamma) \cap f_2(v_2, \gamma)$.

Let $v_1 = MF(a_1, b_1, c_1, d_1)$, $v_2 = MF(a_2, b_2, c_2, d_2)$ and $a_1 \leq a_2$. Assume $(v_1 \sim= v_2) \geq \gamma$ which, by Equation (1), can be written as $S_{12}/(S_1 + S_2 - S_{12}) \geq \gamma$. We shall prove that $f_2(v_1, \gamma) \cap f_2(v_2, \gamma)$, or equivalently $\gamma[(d_2 - a_2) + (d_1 - a_1)] \leq 2(d_1 - a_2)$. Since the inequality must hold for all γ less than or equal to $(v_1 \sim= v_2)$, it is sufficient to show that it holds for $\gamma = S_{12}/(S_1 + S_2 - S_{12})$. Thus we need to show that

$$S_{12}[d_2 - a_1 + 3(d_1 - a_2)] \leq 2(S_1 + S_2)(d_1 - a_2) \tag{7}$$

Notice that $S_i = (d_i - a_i + c_i - b_i)/2$ for $i = 1, 2$, and S_{12} is calculated based on the shape of the area shared by both values.

Since $(v_1 \sim= v_2) \geq 0$, v_1 and v_2 has a non–empty overlap. We need to consider two cases according to how the two values overlap.

Case 1: $a_2 \leq d_1$ and $c_1 \leq b_2$.

In this case, the area shared by both values has a triangular shape. With a little calculation, we have $S_{12} = (d_1 - a_2)/2(d_1 - a_2 + b_2 - c_1)$. By substituting the formulae of S_1, S_2, and S_{12} into the inequality (7), and multiplying both sides by $2(d_1 - a_2 + b_2 - c_1)/(d_1 - a_2)$, we have

$$\begin{aligned}
LHS &= (d_1 - a_2)[(d_2 - a_1) + 3(d_1 - a_2)] \\
&\leq (d_1 - a_2)[(d_2 - a_1) + (d_1 - a_2) + (d_1 - a_1) + (d_2 - a_2)] \\
&\leq 2[(d_1 - a_2) + (b_2 - c_1)][(d_1 - a_1) + (d_2 - a_2) + (c_1 - b_1) + (c_2 - b_2)] \\
&\leq RHS
\end{aligned}$$

Thus, the inequality (7) holds.

Case 2: $a_1 \leq a_2$ and $c_1 \geq b_2$.

In this case, the shared area has a general trapezoidal shape. Depending on whether v_2 is completely contained in v_1, we have two subcases.

Subcase 2.1: $d_1 \leq d_2$, that is, v_2 is not completely contained in v_1.

Substituting S_1, S_2 and $S_{12} = (d_1 - a_2 + c_1 - b_2)/2$ into the inequality (7), and multiply both sides by 2, we have

$$\begin{aligned}
LHS &= [(d_1 - a_2) + (c_1 - b_2)][3(d_1 - a_2) + (d_2 - a_1)] \\
&\leq (d_1 - a_2)[3(d_1 - a_2) + (d_2 - a_1)] + 3(c_1 - b_2)(d_1 - a_2) \\
&\quad +(d_1 - a_2)[(d_2 - d_1) + (c_2 - c_1) + (c_1 - b_2) + (b_2 - b_1) + (a_2 - a_1)] \\
&\leq 2(d_1 - a_2)[(d_1 - a_1 + c_1 - b_1) + (d_2 - a_2 + c_2 - b_2)] \\
&= RHS
\end{aligned}$$

That is, the inequality holds.

Subcase 2.2: $d_1 \geq d_2$, that is, v_2 is completely contained in v_1.

Substituting S_1, S_2 and $S_{12} = (d_1 - a_2 + c_1 - b_2)/2$ into (7), and multiply both sides by 2, we have

$$\begin{aligned}
LHS &= [(d_2 - a_2) + (c_2 - b_2)][3(d_1 - a_2) + (d_2 - a_1)] \\
&\leq (d_1 - a_2)[3(d_2 - a_2) + (d_1 - a_1)] + 3(d_1 - a_2)(c_2 - b_2) \\
&\quad +(d_1 - a_2)[(d_1 - d_2) + (c_1 - c_2) + (c_1 - b_1) + (b_2 - b_1) + (a_2 - a_1)] \\
&\leq 2(d_1 - a_2)[(d_1 - a_1 + c_1 - b_1) + (d_2 - a_2 + c_2 - b_2)] \\
&= RHS
\end{aligned}$$

Thus we have shown that $(v_1 \sim= v_2) \geq \gamma$ implies $f_2(v_1, \gamma) \cap f_2(v_2, \gamma)$, that is, f_2 is an FE indicator over $\mathcal{D}_S$. □

Theorem 7.4 Among f_k mappings, f_3 is the strongest FE indicator over $\mathcal{D}$.

Proof. (Sketch) We shall show that f_3 is an FE indicator but f_2 is not. Thus, by Lemma 7.2, f_3 is the strongest FE indicator in f_k mappings.

First of all, we show that f_2 is not an FE indicator. Recall that $\mathcal{D}$ contains all values that can be defined over a universe. Thus there must be two values $v_1 = MF(a_1, b_1, c_1, d_1)$ and $v_2 = MF(a_2, b_2, c_2, d_2)$ in $\mathcal{D}$ such that $a_1 < a_2$, $b_2 < b_1$, $c_1 = c_2$, $d_1 = d_2$, and the following inequality is satisfied.

$$ABD + A^2D + ABC + A^2C + A^2B > 2(A^2 + B^2)D$$

where $A = a_2 - a_1$, $B = b_1 - b_2$, $C = c_1 - b_1$, and $D = d_1 - a_2$. Notice that given v_1 and v_2, it is necessary that A, B, and C are all positive, and $D > C$. It is straightforward to verify that among other possible choices of values, if $A > 2$, $B = 1$, $D = C + 1$, and $C = 0$, the inequality will hold. It can be shown that if $(v_1 \sim= v_2) = \gamma$, then $f_2(v_1, \gamma)$ and $f_2(v_2, \gamma)$ do not overlap, and therefore, f_2 is not an FE indicator. For example, for the values chosen above, we may have $v_1 = MF(1, 5, 5, 5)$ and $v_2 = MF(4, 4, 5, 5)$. Then, $(v_1 \sim= v_2) = 7/17$. The left end–point of $f_2(v_2, 7/17)$ is $4 + (7/34)$, and the right end–point of $f_2(v_1, 7/17)$ is $4 + (6/34)$. Thus, the two intervals do not overlap.

To show that f_3 is an FE indicator, we show that for any two values $v_1, v_2 \in \mathcal{D}$, and any threshold γ, whenever $(v_1 \sim= v_2) \geq \gamma$, $f_3(v_1, \gamma) \cap f_3(v_2, \gamma)$. The proof follows the general approach used to prove Theorem 7.3. Let $v_1 = MF(a_1, b_1, c_1, d_1)$, $v_2 = MF(a_2, b_2, c_2, d_2)$, and $a_1 \leq a_2$. We need to show that if $(v_1 \sim= v_2) \geq \gamma$, $a_2 + \gamma \cdot (d_2 - a_2)/3 < d_1 - \gamma \cdot (d_1 - a_1)/3$, or equivalently, $\gamma \cdot (d_2 - a_2 + d_1 - a_1) < 3(d_1 - a_2)$. Again, we replace γ by $S_{12}/(S_1 + S_2 - S_{12})$. Depending on the

Experiment 1									
		$f_\infty(v,\gamma)$		$f_3(v,\gamma)$		$f_2(v,\gamma)$		$g(v,\gamma)$	
Threshold	# of Join	CPU	I/O	CPU	I/O	CPU	I/O	CPU	I/O
0.1	379573588	1.000	1.000	0.968	0.942	0.952	0.915	0.881	0.788
0.2	318669158	1.000	1.000	0.932	0.886	0.897	0.829	0.796	0.662
0.3	262396178	1.000	1.000	0.889	0.829	0.831	0.739	0.705	0.545
0.4	211771896	1.000	1.000	0.840	0.769	0.754	0.646	0.611	0.440
0.5	167719006	1.000	1.000	0.784	0.708	0.665	0.549	0.517	0.349
0.6	126680774	1.000	1.000	0.720	0.646	0.560	0.444	0.417	0.263
0.7	91196660	1.000	1.000	0.648	0.582	0.445	0.340	0.318	0.190
0.8	57147246	1.000	1.000	0.563	0.512	0.313	0.231	0.212	0.119
0.9	27095108	1.000	1.000	0.474	0.444	0.166	0.119	0.107	0.057
1.0	1214944	1.000	1.000	0.376	0.375	0.005	0.003	0.005	0.003

Table 1: Experiment 1 results

shapes of v_1 and v_2, there are 8 cases to consider. For each case, the formulae that calculate S_{12}, S_1 and S_2 are substituted into the inequality and the inequality is then shown to be satisfied. Due to the limited space, the proof is omitted here. Interested readers may refer to [18]. □

Notice that among the three FE indicators identified, namely, g for $\mathcal{D}_I$, f_2 for $\mathcal{D}_S$, and f_3 for $\mathcal{D}$, g is the strongest, then f_2, and f_3 is the weakest. Also g and f_2 are no longer FE indicators for data sets more general than $\mathcal{D}_I$ and $\mathcal{D}_S$, respectively.

8 Experiment Results

We have conducted preliminary experiments to study the performance of Algorithm SMFEJ using various types of data and the FE indicators identified in the previous section.

The performance study is based on a simulation of Algorithm SMFEJ on synthetic data. The experiments are performed using a SUN SPARCStation 5, and the performance of the algorithm is measured by the number of I/O pages read from the inner relation, as the I/O cost, and the number of comparisons made, as the CPU cost[1]. Only the costs during the join phase of the algorithm is measured. For simplicity, the I/O costs of reading the outer relation and of writing results are

[1]A more accurate measurement of CPU time will take into consideration the time needed to apply FE indicators on joining attribute values. However, one may decide to invoke the FE indicator once for each tuple in the process of sorting, and store the interval together with the tuple when the sorted relation is written back to the disk. Since in general FE indicators are constant time functions, the additional CPU cost is insignificant. The storage implication is a few bytes per tuple, and is justified for large relations.

Experiment 2							
		$f_\infty(v,\gamma)$		$f_3(v,\gamma)$		$f_2(v,\gamma)$	
Threshold	# of Join	CPU	I/O	CPU	I/O	CPU	I/O
0.1	547023456	1.000	1.000	0.994	0.991	0.980	0.978
0.2	465995332	1.000	1.000	0.965	0.958	0.933	0.905
0.3	389639504	1.000	1.000	0.928	0.912	0.877	0.834
0.4	318878030	1.000	1.000	0.887	0.861	0.799	0.730
0.5	245738336	1.000	1.000	0.829	0.793	0.708	0.626
0.6	166331884	1.000	1.000	0.762	0.729	0.601	0.529
0.7	96649582	1.000	1.000	0.693	0.676	0.466	0.410
0.8	44126168	1.000	1.000	0.604	0.605	0.306	0.275
0.9	11282446	1.000	1.000	0.511	0.529	0.149	0.143
1.0	35354	1.000	1.000	0.418	0.436	0.001	0.001

Table 2: Experiment 2 results

omitted. For each pair of R and S tuple, if the values in the join attributes overlap with each other, two comparisons are recorded, one to determine that they are overlap, and the other to determine whether they really join. If the two values do not overlap, one comparison is recorded. The Algorithm SMFEJ is implemented to take advantage of page buffers. For each page of relation R, one page of relation S is read at a time, and all join results that can be obtained from the two pages will be obtained before the next page of relation S is read. In order to illustrate the effect of FE indicators, only a minimum amount of buffer space is assumed, that is, one page for each input relation. It is straightforward to see that a larger buffer space will reduce the I/O cost. However, with more buffer space available to the algorithm, using FE indicators will save more CPU cost than I/O cost.

Each of the three experiments uses a different type of data. For each experiment, both relations have 30,000 randomly generated tuples and same type of data in the join attributes. The universe of the fuzzy join attributes contains 1,000 units, which can be thought of as the interval [1, 1000] of real number. The use of the unit allows the data to be interpreted for various applications easily. For example, if the universe of attribute Age is from 10 years old to 90 years old, with the unit being a year, the universe can be thought of as containing 80 units. However, if the unit is a week, the universe will contain 4,160 units. The data in the join attributes are randomly generated with guaranteed characteristics of the type of data sets chosen for the attributes. The support of the membership functions of the data comes in two types. The size of a small support is between 1/5 to 1/4 of the size of the universe, and that of a large support is between 1/2 to 3/5 of the size of the universe. Each I/O page contains 5 tuples. For all experiments, the performance of Algorithm SMFEJ with FE indicator f_∞ is used as the reference, and the performances of the algorithm with other types of FE indicators are express as a percentage of the reference. To give a feel of the magnitude of the computation cost, the number of pairs of tuples that actually join is also given. The results of the experiments are given in Tables 1–3. In experiment 1, the data set allowed in the join attributes is $\mathcal{D}_I$ with a randomly determined shape. We compare 4 FE indicators: f_∞, f_3, f_2, and g. As expected, g outperforms f_2 which outperforms f_3 which outperforms f_∞. This is because that g is the strongest FE indicator. The effectiveness of using appropriate FE indicator is shown clearly by the increasingly larger percentage of saving obtained for increasingly higher threshold. Compare g with f_∞, the percentage of saving on I/O cost ranges from 21.2% to 99.7% with an average of 57%, and that on CPU cost ranges from 11.9% to 99.5% with an average of 54.3%. In experiment 2, the data set allowed in the join attributes is $\mathcal{D}_S$ with characteristics randomly determined. Since g is no longer an FE indicator in this case, only the rest 3 mappings are compared. Again, f_2 outperforms f_3 which outperforms f_∞. Compare f_2 with f_∞, the percentage of savings on CPU cost ranges from 2% to 99.9% with an average of 41.8% and that on I/O cost ranges from 2.2% to 99.9% with an average of 45.7%. In experiment 3, the data set in the join attribute is $\mathcal{D}$. Only f_2 and f_∞ are compared. The percentage of saving on I/O cost ranges from 2% to 61% with an average 30.2%, and that on CPU cost is similar. All three experiments show that using FE indicators is more efficient, and stronger FE indicators perform better than weaker ones. Although when the threshold is low, the percentage of saving is not significant, if the relations are large, even a small percent of saving, say 10%, will make a noticeable difference.

9 Conclusion

In this paper, we propose a new fuzzy equality comparison operator with a measure that combines the possibility measure with the similarity measure. we define a type of fuzzy equi–join based on the new fuzzy equality comparison operator which allows threshold values to be associated with individual predicates of the join condition. A sort–merge join algorithm based on a partial order of intervals is used to evaluate the fuzzy equi–join. In order to achieve high level of efficiency, various mappings, called FE indicators, that determine appropriate intervals for fuzzy data, are identified for data sets with different characteristics. Experiment results from our preliminary simulation of the algorithm show a significant improvement of efficiency when FE indicators are

Experiment 3					
		$f_\infty(v,\gamma)$		$f_3(v,\gamma)$	
Threshold	# of Join	CPU	I/O	CPU	I/O
0.1	319089228	1.00	1.00	0.98	0.98
0.2	248675130	1.00	1.00	0.94	0.92
0.3	196333906	1.00	1.00	0.89	0.86
0.4	152369916	1.00	1.00	0.84	0.80
0.5	109961740	1.00	1.00	0.77	0.74
0.6	65371268	1.00	1.00	0.70	0.67
0.7	29545404	1.00	1.00	0.62	0.61
0.8	8340254	1.00	1.00	0.54	0.54
0.9	766350	1.00	1.00	0.46	0.47
1.0	30082	1.00	1.00	0.38	0.39

Table 3: Experiment 3 results

used in conjunction with the sort–merge join algorithm.

The efficient evaluation of joins in fuzzy relational databases is still an open area of research. In this paper, we proposed FE indicators of certain characteristics. However, there may be other types of FE indicators that may be better than f_2 and f_3 for $\mathcal{D}_S$ and $\mathcal{D}$ data sets. Our results indicate that the more strongly the data are correlated, say from arbitrary shape to similar shape to identical shape, the more beneficial to use FE indicators. It may be interesting to study other types of data correlations, and the effect that they have on join evaluation. The join algorithm used in this paper is limited to join attributes that have numeric universes, or universes that can be mapped into a numeric one. Finding efficient join algorithms that can be applied to both numeric and discrete attributes is an important issue requiring further research. Due to the nature of uncertainty and imprecision of the data, conventional fast access paths do not handle fuzzy data well. Finding new types of fast access paths that handle both crisp and fuzzy data efficiently is a challenging task.

Acknowledgment

The authors want to thank the anonymous referees for their comments and suggestions that led to improvements of this paper.

References

[1] P. Bosc, M. Galibourg, and G. Hamon. Fuzzy querying with SQL: Extensions and implementation aspects. *Fuzzy Set and Systems*, 28:333–349, 1988.

[2] B. P. Buckles and F. E. Petry. A fuzzy model for relational databases. *Fuzzy Set and Systems*, 7(3):213–226, 1982.

[3] S. K. Chang and J. S. Ke. Translation of fuzzy queries for relational database systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:281–294, 1979.

[4] D. J. DeWitt, J. F. Naughton, and D. A. Schneider. An evaluation of non–equijoin algorithms. In *Int'l Conf. on Very Large Data Base*, pages 443–452, 1991.

[5] H.Gunadhi and A. Segev. Query processing algorithms for temporal intersection joins. In *IEEE Int'l Conf. on Data Engineering*, Kobe, Japan, 1991.

[6] J. Kacprzyk, S. Zadrozny, and A. Ziolkowski. FQUERY III+: A "human–consistent" database querying system based on fuzzy logic with linguistic quantifiers. *Information Systems*, 14(6):443–453, 1989.

[7] D. Li and D. Liu. *A Fuzzy Prolog Database System.* Research Studies Press, Taunton, England, 1990.

[8] H. Nakajima, T. Sogoh, and M. Arao. Fuzzy database language and library — fuzzy extension to SQL. In *Second IEEE International Conference on Fuzzy Systems*, pages 35–52, 1992.

[9] H. Nakajima, T. Sogoh, and M. Arao. Development of an efficient fuzzy SQL for large scale fuzzy relational databases. In *The Fifth IFSA World Congress*, 1993.

[10] F. Petry. *Fuzzy Databases: Principles and Applications.* Kluwer Academic Publishers, 1996.

[11] H. Prade and C. Testemale. Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. *Information Sciences*, 34:115–143, 1984.

[12] S. Shenoi and A. Melton. An extended version of the fuzzy relational database model. *Information Sciences*, 51:35–52, 1990.

[13] M. Umano and S. Fukami. Fuzzy relational algebra for possibility–distribution–fuzzy–relational model of fuzzy data. *Journal of Intelligent Information Systems*, 3:7–27, 1994.

[14] Q. Yang, C. Liu, J. Wu, C. Yu, S. Dao, H. Nakajima, and N. Rishe. Efficient processing of nested fuzzy SQL queries in fuzzy databases. In *IEEE Int'l Conf. on Data Engineering*, pages 131–138, 1995.

[15] L. Zadeh. Fuzzy set. *Information and Control*, 8:338–353, 1965.

[16] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Set and Systems*, 1:3–28, 1978.

[17] M. Zemankova and A. Kandel. Implementing imprecision in information systems. *Information Sciences*, 37, 1985.

[18] W. Zhang. Evaluating a fuzzy equi–join using FE indicators. Technical report, Dept. of Math. & CS, University of Lethbridge, 1997.

[19] W. Zhang, C. Yu, B. Reagan, and H. Nakajima. Context dependent interpretations for linguistic terms in fuzzy relational databases. In *IEEE Int'l Conf. on Data Engineering*, pages 139–146, 1995.