

# Fuzzy Equi-Join and Its Evaluation

Weining Zhang\*

Ke Wang†

## Abstract

We propose a new measure of fuzzy equality comparison based on the similarity of possibility distributions. we define a 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 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.

## 1 Introduction

The importance of representing and manipulating uncertain or imprecise information in knowledge-base and database systems has long been recognized in the AI community and the database community. Various extensions to the existing database models have been proposed to incorporate ill-known data into the database systems, and to issue and to answer queries with soft restrictions. In recent years, various fuzzy data models and fuzzy database systems have been proposed [1, 5, 2, 3, 6, 10, 8, 11]. These models and systems extend relational and object-oriented data models using the fuzzy set and the possibility theory [9]. 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 [3] 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.

In this paper, we consider the efficient processing of a fuzzy equi-join in a possibility-based fuzzy relational

database similar to [3]. 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 FE indicators, which will associate appropriate intervals to fuzzy data so that the evaluation of the join with a sort-merge algorithm is highly efficient.
4. We present experiment results that show a significant improvement of efficiency when FE indicators are used.

There are several works related to ours. In [5, 10], the satisfaction degree of a predicate is measured by both possibility and necessity. Although the combination of these measures can explicitly represent the uncertainty of the satisfaction degree, as pointed out in [4], the resulting algebraic operations can not be composed, implying that queries can not be optimized using well-known algebraic transformation techniques. In [7], the satisfaction degree and the degree of tuples are themselves possibility distributions which allow algebraic operations to be composed. But the method applies only to discrete possibility distributions, and the evaluation of a query may be very inefficient due to many elements in the possibility distributions. In [3] 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.

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 and identify a number of appropriate FE indicators for fuzzy data with different characteristics. In Section 6, we present our experimental results. Section 7 concludes the paper.

\*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

## 2 Fuzzy Relations

In a fuzzy database, a data is *crisp* if it is certain and precise, and *fuzzy*, otherwise. A fuzzy data is defined by a fuzzy set. 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)  $F$ , where  $\mu_F(x) = 1$  (respectively,  $0 < \mu_F(x) < 1$ , or  $\mu_F(x) = 0$ ) means that  $x$  is a full (respectively, partial, or non) member of  $F$ . If a fuzzy data  $v$  is defined by a fuzzy set  $F$ , then,  $v$  can only have a value in  $F$ , and the possibility for  $v$  to have value  $x$  in  $F$  is  $\mu_F(x)$ . That  $v$  is defined by  $\mu_F$  is denoted by  $v = \mu_F$ . Since a crisp data  $v$  can also be represented by a (degenerated) membership function, all data will be uniformly treated as fuzzy data.

In this paper, we shall use the following generic parameterized function to define membership functions.

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

where the parameters  $a$ ,  $b$ ,  $c$ , and  $d$  are values in the universe satisfying  $a \leq b \leq c \leq d$ . In general, the curve of the generic function is a trapezoidal, 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  $MF(v, v, v, v)$ . Let  $v = MF(a, b, c, d)$ . Then  $A(v)$ ,  $B(v)$ ,  $C(v)$  and  $D(v)$  denote the parameters  $a$ ,  $b$ ,  $c$ , and  $d$ , respectively.

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 values defined over  $U(A)$ . A fuzzy relation  $R$  with a schema  $(A_1, \dots, A_n)$  is a fuzzy set of tuples in  $\mathcal{D}(A_1) \times \dots \times \mathcal{D}(A_n)$  with a membership function

$$\mu_R : \mathcal{D}(A_1) \times \dots \times \mathcal{D}(A_n) \rightarrow [0, 1].$$

A tuple  $t$  is said to be in a fuzzy relation  $R$  if and only if  $\mu_R(t) > 0$ . We shall denote the attribute  $A$  of fuzzy relation  $R$  by  $R.A$ , and the component of tuple  $t$  under attribute  $A$  by  $t[A]$ .

## 3 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 attribute AGE. Since AGE contains fuzzy values, we must determine the degree for two fuzzy ages, say  $About\ 32$  and  $Middle\ age$ , to satisfy the join condition  $AGE = AGE$ .  $\square$

We propose the following new measure for the fuzzy equality comparison.

**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

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

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

If membership functions are defined using the generic function, the fuzzy equality can be evaluated efficiently by

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

where  $S_i$  is the area under  $\mu_{v_i}$ ,  $S_j$  is the area under  $\mu_{v_j}$ , 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  $v_2$ . This new measure allows algebraic operations to be composed, and obtains the degree by considering all rather than just the best possible values in the fuzzy data. Therefore, it is more natural than existing measures [5, 10, 7, 3].

**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

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

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

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

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

A sort-merge fuzzy equi-join algorithm, SMFEJ, is given in Figure 1. The algorithm assumes that fuzzy join attributes

### 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 1: Algorithm SMFEJ

have numeric universes and membership functions are defined by the generic parameterized function. Let  $[l(v), h(v)]$  denote the interval associated with a value  $v$ , the sorting phase is based on the partial order  $\prec$  defined in [8].

**Definition 4.1** Let  $v_1$  and  $v_2$  be two values over the same universe.  $v_1$  *precedes*  $v_2$ , denoted by  $v_1 \prec v_2$ , if  $l(v_1) < l(v_2)$ , or if  $l(v_1) = l(v_2)$  and  $h(v_1) < h(v_2)$ .  $v_1$  *precedes or equals*  $v_2$ , denoted by  $v_1 \preceq v_2$ , if  $v_1 \prec v_2$  or  $v_1 \equiv v_2$ .  $v_1$  *overlaps*  $v_2$ , denoted by  $v_1 \cap v_2$ , if  $l(v_1) < h(v_2)$  and  $l(v_2) < h(v_1)$ . 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,  $t_1[A] \preceq t_2[A]$ ).  $\square$

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 4.2** Let  $S$  be sorted on  $S.B$  according to  $\prec$ . The *range (of  $S$  tuples) of a tuple  $r \in R$* , denoted by  $rng_S(r)$ , is the longest sequence  $[s_1, s_2, \dots, s_k]$  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  $r[A] \cap s_k[B]$ .  $\square$

Thus, only those pages containing tuples in  $rng_S(r)$  need to be scanned when  $r$  is processed in the joining phase.

## 5 Fuzzy Equality Indicators and Filters

In practice, a limited buffer space is available, therefore, during joining phase, a page of  $S$  may be swapped in and out of the buffer if it is in ranges of multiple  $R$ -tuples. The key to the efficiency of SMFEJ is to associate appropriate intervals with fuzzy attribute values.

**Example 5.1** Assume that  $R$  has a tuple  $r$  with  $r[A] = MF(10, 10, 40, 40)$ , and  $S$  contains exactly the tuples  $s_1, \dots, s_9$  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

join attribute value  $v$  is associated with interval  $[A(v), D(v)]$ , then  $s_1 \prec s_2 \prec s_3 \dots \prec s_9$ ,  $r[A] \cap s_1[B]$ , and  $r[A] \cap s_9[B]$ . So, all  $S$ -tuples are in  $rng_S(r)$ . For  $i = 1, 2, \dots, 9$ ,  $(r[A] \sim s_i[B])$  equals to 0.29, 0, 1, 0.17, 0.71, 0.33, 0.5, 0.13, and 0.1, respectively. If the join condition is  $(R.A \sim S.B) \geq 0.5$ , only  $s_3, s_5$ , and  $s_7$  join with  $r$ . If the threshold is raised to 0.9, only  $s_3$  joins with  $r$ . But in both cases, all  $S$ -tuples must be scanned. However, if the intervals are assigned according to Theorem 5.5 for  $\gamma = 0.5$ , we have  $s_2 \prec s_1 \prec s_4 \prec s_3 \prec s_6 \prec s_5 \prec s_7 \prec s_8 \prec s_9$  and  $rng_S(r)$  being  $[s_3, s_6, s_5, s_7]$ . Thus, the join only needs to scan about 50% less number of tuples.  $\square$

In general,  $rng_S(r)$  may contain tuples irrelevant to  $r$ , and the number of such tuples may increase with the threshold value. These irrelevant tuples may be removed from  $rng_S(r)$  if the assignment of intervals is by an appropriate function of the threshold value.

**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  $f : \mathcal{D} \times [0, 1] \rightarrow \mathcal{I}$  is an *FE (fuzzy equality) indicator over  $\mathcal{D}$*  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  $f(v_1, \gamma) \succ f(v, \gamma)$ .  $\square$

Intuitively, using an FE indicator to assign intervals to join attribute values will guarantee that after sorting, all  $S$ -tuples relevant to a tuple  $r$  are in  $rng_S(r)$ , using a stronger (or perfect) FE indicator will guarantee that  $rng_S(r)$  contains less (or no) irrelevant tuple. In the following, we shall identify a number of desirable mappings which are efficient to compute, are as strong as possible, and assign smaller intervals for higher threshold. Whether a particular mapping is desirable depends in general on the characteristics of both the mapping and the type of attribute values. We shall consider the  $\mathcal{F}$  mappings of the following form.

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

where  $\Phi$  is a monotonically increasing function of  $\gamma$ , and  $\Phi(v, 0) \geq 0$ . The join attribute values come in three types, namely, their membership functions may have the identical, similar, or arbitrary shapes. These types of values are denoted by  $\mathcal{D}_I$ ,  $\mathcal{D}_S$ , and  $\mathcal{D}$ , respectively.

For  $\mathcal{D}_I$ , we have the following theorem and corollaries (the proofs are omitted due to limited space).

**Theorem 5.1** 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

$$\begin{aligned} g_1(v, \gamma) &= [A(v) + \frac{1}{2} \left\{ \frac{2\gamma}{1+\gamma} [(D(v) - A(v))^2 \right. \\ &\quad \left. - (C(v) - B(v))^2 \right\}^{1/2}, \\ &\quad D(v) - \frac{1}{2} \left\{ \frac{2\gamma}{1+\gamma} [(D(v) - A(v))^2 \right. \\ &\quad \left. - (C(v) - B(v))^2 \right\}^{1/2}] \\ g_2(v, \gamma) &= [A(v) + \frac{1+3\gamma}{4+4\gamma} (D(v) - A(v)) \\ &\quad - \frac{1-\gamma}{4+4\gamma} (C(v) - B(v)), \\ &\quad D(v) - \frac{1+3\gamma}{4+4\gamma} (D(v) - A(v)) \\ &\quad + \frac{1-\gamma}{4+4\gamma} (C(v) - B(v))] \\ \theta &= \frac{(D(v) - A(v)) - (C(v) - B(v))}{(D(v) - A(v)) + 3(C(v) - B(v))} \quad \square \end{aligned}$$

**Corollary 5.2** 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  $\mathcal{D}_I$ .  $\square$

**Corollary 5.3** 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  $\mathcal{D}_I$ .  $\square$

For  $\mathcal{D}_S$  and  $\mathcal{D}$ , we have the following proposition.

**Proposition 5.4** No  $\mathcal{F}$  mapping is a perfect FE indicator over  $\mathcal{D}_S$  (respectively,  $\mathcal{D}$ ).  $\square$

Thus, we can at the best find the strongest FE indicators for  $\mathcal{D}_S$  and  $\mathcal{D}$  among  $\mathcal{F}$  mappings<sup>1</sup>. Consider a subset of  $\mathcal{F}$  mappings, the  $f_k$  mappings, of the following 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, and  $f_\infty(v, \gamma) = [A(v), D(v)]$ . We have the following theorems.

**Theorem 5.5** Among  $f_k$  mappings,  $f_2$  is the strongest FE indicator over  $\mathcal{D}_S$ .  $\square$

**Theorem 5.6** Among  $f_k$  mappings,  $f_3$  is the strongest FE indicator over  $\mathcal{D}$ .  $\square$

Notice that among the three FE indicators identified,  $g$  is stronger than  $f_2$  which is in turn stronger than  $f_3$ . However,  $g$  and  $f_2$  are no longer FE indicators for data sets more general than  $\mathcal{D}_I$  and  $\mathcal{D}_S$ , respectively.

<sup>1</sup>It is an open problem to find perfect FE indicators that are not  $\mathcal{F}$  mappings.

## 6 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.

Three experiments are performed using a SUN SPARC-Station 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. For simplicity, we measure only the costs incurred during the joining phase, with the I/O costs of reading the outer relation and writing results omitted. For each pair of  $R$  and  $S$  tuples, a comparison is made on the join attribute values to determine if they overlap, and if they do, another comparison is made to determine if they actually join. 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. 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 are obtained before the next page of relation  $S$  is read. 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.

In the experiments, the execution of Algorithm SMFEJ is simulated using different types of synthetic data. For each experiment, both relations have 30,000 randomly generated tuples with the identical 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 easily for various applications. The data in the join attributes are randomly generated with guaranteed characteristics, such as similar or identical shape. For all experiments, the performance of Algorithm SMFEJ with FE indicator  $f_\infty$  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 show the magnitude of the computational cost, the number of pairs of tuples that actually join is also given. Due to limited space, only the result of one experiment is reported here. In this experiment, the data set allowed in the join attributes is  $\mathcal{D}_I$  with a randomly determined shape. We compare 4 FE indicators:  $f_\infty$ ,  $f_3$ ,  $f_2$ , and  $g$ . As shown in Table 1,  $g$  outperforms  $f_2$  which outperforms  $f_3$  which outperforms  $f_\infty$ . 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  $f_\infty$ , 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%.

Experiment 1									
Threshold	# of Join	$f_{\infty}(v, \gamma)$		$f_3(v, \gamma)$		$f_2(v, \gamma)$		$g(v, \gamma)$	
		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

## 7 Conclusion

In this paper, we propose to measure the fuzzy equality by the similarity of possibility distributions, and define a type of fuzzy equi-join based on this new measure of fuzzy equality. Threshold values can be specified with the fuzzy equi-join. A sort-merge join algorithm using a partial order over intervals to sort data is used to evaluate the join. We identify a number of mappings, called FE indicator, that determine appropriate intervals for fuzzy data, so that the fuzzy equi-join can be efficiently evaluated using the sort-merge algorithm. Our results are verified with experiments.

## 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] 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.
- [3] 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.
- [4] F. Petry. *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publishers, 1996.
- [5] 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.
- [6] S. Shenoi and A. Melton. An extended version of the fuzzy relational database model. *Information Sciences*, 51:35–52, 1990.
- [7] M. Umamo and S. Fukami. Fuzzy relational algebra for possibility-distribution-fuzzy-relational model of fuzzy data. *Journal of Intelligent Information Systems*, 3:7–27, 1994.
- [8] 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.
- [9] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Set and Systems*, 1:3–28, 1978.
- [10] M. Zemankova and A. Kandel. Implementing imprecision in information systems. *Information Sciences*, 37, 1985.
- [11] 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*, pp. 139–146, 1995.