Efficient Algorithms for Computing a Class of Subsethood and Similarity Measures for Interval Type-2 Fuzzy Sets

Dongrui Wu, *Member, IEEE*, and Jerry M. Mendel, *Life Fellow, IEEE*

***Abstract*— Subsethood and similarity measures are important concepts in fuzzy set (FS) theory. There are many different definitions of them, for both type-1 (T1) FSs and interval type-2 (IT2) FSs. In this paper, Rickard et al.'s definition of IT2 FS subsethood measure, extended from Kosko's T1 FS subsethood measure using the Representation Theorem, and Nguyen and Kreinovich's IT2 FS similarity measure, extended from the Jaccard similarity measure for T1 FSs, are introduced. Efficient algorithms for computing them are also proposed. Simulations demonstrate that our proposed algorithms outperform existing algorithms in the literature.**

***Index Terms*— Interval type-2 fuzzy sets, efficient algorithms, similarity measures, subsethood measures**

I. Introduction

Subsethood [5], [29] and similarity [30], [31] measures are important concepts in fuzzy set (FS) theory. The subsethood of a FS A in another FS B is a quantity in $[0,1]$ indicating the degree of containment of A in B, and the similarity of A to B is a quantity in $[0,1]$ indicating how similar A is to B. Both measures have had wide applications, e.g., subsethood measures of FSs have been used in approximate reasoning [1], classification [10], [14], computing with words [9], [27], control [6], etc, and similarity measures of FSs have been used in approximate reasoning [2], [13], [25], [26], classification [17], clustering [28], computing with words [9], [23], etc.

The most popular subsethood measure for type-1 (T1) FSs was proposed by Kosko [5], which is

$$ss_{_K}(A,B)=\frac{\sum_{i=1}^{N}\min(\mu_A(x_i),\mu_B(x_i))}{\sum_{i=1}^{N}\mu_A(x_i)} \tag{1}$$

and the most popular similarity measure for T1 FSs is the Jaccard similarity measure [3], [23], which is

$$sm_{_J}(A,B)=\frac{\sum_{i=1}^{N}\min(\mu_A(x_i),\mu_B(x_i))}{\sum_{i=1}^{N}\max(\mu_A(x_i),\mu_B(x_i))}. \tag{2}$$

Observe the analogy between $ss_{_K}(A,B)$ and $sm_{_J}(A,B)$.

Rickard et al. [15], [16] and Nguyen and Kreinovich [11] have independently extended Kosko's subsethood measure for T1 FSs to interval type-2 (IT2) FSs [7], and in this paper it is called Rickard et al.'s subsethood measure for IT2 FSs and denoted as $ss_{_R}$ since Rickard et al.'s paper was published a little earlier. Nguyen and Kreinovich [11] also extended the Jaccard similarity measure for T1 FSs to IT2 FSs, and in this paper it is called Nguyen and Kreinovich's similarity measure for IT2 FSs and denoted as $sm_{_{NK}}$. How to compute these measures efficiently is the focus of this paper.

The rest of this paper is organized as follows: Section II introduces the definition of $ss_{_R}$, Rickard et al's exhaustive computation approach for computing it, Nguyen and Kreinovich's fast algorithm, and our efficient algorithm. Their performances are also compared. Section III introduces the definition of $sm_{_{NK}}$, an exhaustive computation approaching for computing it, Nguyen and Kreinovich's fast algorithm, and our efficient algorithm. Their performances are also compared. Finally, Section IV draws conclusions.

II. $ss_{_R}$: Definition and Computation

Rickard et al. [15], [16] extended Kosko's subsethood measure to IT2 FSs based on the Representation Theorem [8] of IT2 FSs. This section presents their results and compares several algorithms for computing $ss_{_R}$.

A. Definition of $ss_{_R}$

Let $\tilde{A}$ and $\tilde{B}$ be two IT2 FSs, and A_e and B_e be embedded T1 FSs [7] of $\tilde{A}$ and $\tilde{B}$, respectively. Then, the subsethood of $\tilde{A}$ in $\tilde{B}$, $ss(\tilde{A},\tilde{B})$, is an interval defined as [15], [16]

$$\begin{aligned} ss_{_R}(\tilde{A},\tilde{B}) &= \bigcup_{\forall A_e,B_e} ss_{_K}(A_e,B_e) \\ &= \bigcup_{\forall A_e,B_e} \frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\mu_{A_e}(x_i)} \\ &\equiv [ss_{_{Rl}}(\tilde{A},\tilde{B}), ss_{_{Rr}}(\tilde{A},\tilde{B})] \end{aligned} \tag{3}$$

where

$$ss_{_{Rl}}(\tilde{A},\tilde{B})=\min_{\forall A_e,B_e}\frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\mu_{A_e}(x_i)} \tag{4}$$

$$ss_{_{Rr}}(\tilde{A},\tilde{B})=\max_{\forall A_e,B_e}\frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\mu_{A_e}(x_i)}. \tag{5}$$

Rickard et al. [15], [16] and Nguyen and Kreinovich [11] independently showed that it is not necessary to enumerate all possible A_e in order to compute $ss_{_{Rl}}(\tilde{A},\tilde{B})$ and $ss_{_{Rr}}(\tilde{A},\tilde{B})$, as suggested by (4) and (5). Their main results are summarized in the following:

Theorem 1: [11], [16] Let

$$I_1\equiv\{x_i|\underline{\mu}_{\tilde{B}}(x_i)\leqslant\underline{\mu}_{\tilde{A}}(x_i)\} \tag{6}$$

$$I_2\equiv\{x_i|\underline{\mu}_{\tilde{B}}(x_i)\geqslant\overline{\mu}_{\tilde{A}}(x_i)\} \tag{7}$$

$$I\equiv\{x_i|\underline{\mu}_{\tilde{A}}(x_i)<\underline{\mu}_{\tilde{B}}(x_i)<\overline{\mu}_{\tilde{A}}(x_i)\} \tag{8}$$

Dongrui Wu is with the Institute for Creative Technologies and the Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, CA 90089 (phone: 213-595-3269; email: dongruiw@usc.edu).

Jerry M. Mendel is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 (phone: 213-740-4445; email: mendel@sipi.usc.edu).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

and

$$\mu_{A_l}(x_i) = \begin{cases} \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I_1 \\ \underline{\mu}_{\tilde{A}}(x_i), & x_i \in I_2 \\ \underline{\mu}_{\tilde{A}}(x_i) \text{ or } \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I \end{cases} \tag{9}$$

$$\mu_{A_r}(x_i) = \begin{cases} \underline{\mu}_{\tilde{A}}(x_i), & \overline{\mu}_{\tilde{B}}(x_i) \leqslant \underline{\mu}_{\tilde{A}}(x_i) \\ \overline{\mu}_{\tilde{A}}(x_i), & \overline{\mu}_{\tilde{B}}(x_i) \geqslant \overline{\mu}_{\tilde{A}}(x_i) \\ \overline{\mu}_{\tilde{B}}(x_i), & \underline{\mu}_{\tilde{A}}(x_i) < \overline{\mu}_{\tilde{B}}(x_i) < \overline{\mu}_{\tilde{A}}(x_i) \end{cases} \tag{10}$$

Then,

$$ss_{Rl}(\tilde{A}, \tilde{B}) = \min_{\mu_{A_l}(x_i) \text{ in (9)}} \left[\frac{\sum_{i=1}^{N} \min\left(\mu_{A_l}(x_i), \underline{\mu}_{\tilde{B}}(x_i)\right)}{\sum_{i=1}^{N} \mu_{A_l}(x_i)} \right] \tag{11}$$

$$ss_{Rr}(\tilde{A}, \tilde{B}) = \frac{\sum_{i=1}^{N} \min\left(\mu_{A_r}(x_i), \overline{\mu}_{\tilde{B}}(x_i)\right)}{\sum_{i=1}^{N} \mu_{A_r}(x_i)}. \quad \square \tag{12}$$

B. Rickard et al.'s Exhaustive Computation Approach

Observe from (12) that $ss_{Rr}(\tilde{A}, \tilde{B})$ has a closed-form solution; however, because for each $x_i \in I$, $\mu_{A_l}(x_i)$ in (9) can have two possible values, to compute $ss_{Rl}(\tilde{A}, \tilde{B})$, 2^L evaluations of the bracketed term in (11) have to be performed, where L is the number of elements in I. This approach [16] is referred to in the present paper as the "exhaustive computation approach" for computing $ss_{Rl}(\tilde{A}, \tilde{B})$.

Because 2^L can be a rather large number depending on L, the exhaustive computation approach is not efficient. Two more efficient algorithms are introduced next.

C. Nguyen and Kreinovich's Fast Algorithm

Before introducing Nguyen and Kreinovich's fast algorithm for computing $ss_{Rl}(\tilde{A}, \tilde{B})$, an important result is needed:

Theorem 2: [11] Define

$$r_j = \frac{\underline{\mu}_{\tilde{B}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j)}{\overline{\mu}_{\tilde{A}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j)}, \quad x_j \in I \tag{13}$$

$$I_3 = \{x_j | r_j \leqslant ss_{Rl}(x_j)\} \tag{14}$$

$$I_4 = \{x_j | r_j > ss_{Rl}(x_j)\} \tag{15}$$

Then,

$$ss_{Rl}(\tilde{A}, \tilde{B}) = \frac{\sum_{x_j \in I_1 \cup I_3} \underline{\mu}_{\tilde{B}}(x_j) + \sum_{x_j \in I_2 \cup I_4} \underline{\mu}_{\tilde{A}}(x_j)}{\sum_{x_j \in I_1 \cup I_3} \overline{\mu}_{\tilde{A}}(x_j) + \sum_{x_j \in I_2 \cup I_4} \underline{\mu}_{\tilde{A}}(x_j)}. \quad \square \tag{16}$$

According to Theorem 2, for every $x_j \in I$ we can compute its corresponding r_j. For each x_j whose corresponding r_j is smaller than or equal to $ss_{Rl}(\tilde{A}, \tilde{B})$, $\overline{\mu}_{\tilde{A}}(x_j)$ must be used in computing $ss_{Rl}(\tilde{A}, \tilde{B})$, and for each x_j whose corresponding r_j is larger than $ss_{Rl}(\tilde{A}, \tilde{B})$, $\underline{\mu}_{\tilde{A}}(x_j)$ must be used in computing $ss_{Rl}(\tilde{A}, \tilde{B})$. However, since we do not know $ss_{Rl}(\tilde{A}, \tilde{B})$ *a priori*, we cannot determine when $\overline{\mu}_{\tilde{A}}(x_j)$ or $\underline{\mu}_{\tilde{A}}(x_j)$ should be used.

One approach is to sort these L r_j in ascending order. Then, there must exist a *switch point* k such that

$$\mu_{A_l}(x_j) = \begin{cases} \overline{\mu}_{\tilde{A}}(x_j), & j \leqslant k \\ \underline{\mu}_{\tilde{A}}(x_j), & j > k \end{cases} \tag{17}$$

where[1] $r_k \leqslant ss_{Rl}(\tilde{A}, \tilde{B}) < r_{k+1}$.

The problem then simplifies to determining the switch point k. Since k can only assume $L+1$ values ($k = 0, 1, ..., L$), we can enumerate all these k and substitute (17) into (11) to compute the corresponding $ss^k_{Rl}(\tilde{A}, \tilde{B})$. The final $ss_{Rl}(\tilde{A}, \tilde{B})$ must be the smallest of these $L+1$ resulting $ss^k_{Rl}(\tilde{A}, \tilde{B})$.

Nguyen and Kreinovich's [11] fast algorithm for computing $ss_{Rl}(\tilde{A}, \tilde{B})$ is based on the above idea, and is given in Algorithm 1. It is much faster than the exhaustive computation approach, as demonstrated in Section II-E; however, it first needs to sort r_j, which requires extra memory and is also time-consuming, especially when the set I is large. A more efficient algorithm, which avoids the ranking, is proposed next.

Algorithm 1: Nguyen and Kreinovich's Fast Algorithm for Computing $ss_{Rl}(\tilde{A}, \tilde{B})$

Find I_1 in (6), I_2 in (7), and I in (8)
$n = \sum_{x_i \in I_1 \cup I} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2} \underline{\mu}_{\tilde{B}}(x_i)$
$d = \sum_{x_i \in I_1 \cup I} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2} \overline{\mu}_{\tilde{A}}(x_i)$
$ss_0 = n/d$
for each $x_j \in I$
  Compute r_j in (13)
end
Sort $\{r_j\}$ in ascending order and call them $r_1, r_2, ..., r_L$
Match $\{x_j\}$ in I with the order of $\{r_j\}$ and call them $x_1, x_2, ..., x_L$
for $j = 1$ to L
  $n = n + \underline{\mu}_{\tilde{B}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j)$
  $d = d + \overline{\mu}_{\tilde{A}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j)$
  $ss_j = n/d$
end
$ss_{Rl}(\tilde{A}, \tilde{B}) = \min_{j=0,1,...L} ss_j$

D. New Efficient Algorithm for Computing $ss_{Rl}(\tilde{A}, \tilde{B})$

Theorem 1 indicates that in computing $ss_{Rl}(\tilde{A}, \tilde{B})$, $\mu_{A_l}(x_i)$ can either be $\underline{\mu}_{\tilde{A}}(x_i)$ or $\overline{\mu}_{\tilde{A}}(x_i)$ for $x_i \in I$. Let I_5 be an arbitrary subset of I, and I_6 be its complementary set in I. Define

$$A_e \equiv \begin{cases} \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I_1 \cup I_6 \\ \underline{\mu}_{\tilde{A}}(x_i), & x_i \in I_2 \cup I_5 \end{cases} \tag{18}$$

Then, it follows from (4), (18) and (11) that

$$ss_{Rl}(\tilde{A}, \tilde{B}) = \min_{I_5, I_6} \left[\frac{\sum_{x_i \in I_1 \cup I_6} \underline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_2 \cup I_5} \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{x_i \in I_1 \cup I_6} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2 \cup I_5} \underline{\mu}_{\tilde{A}}(x_i)} \right] \tag{19}$$

[1]Note that k in (17) is very similar to the switch point used in the Karnik-Mendel (KM) Algorithms [4], [7] or the Enhanced Karnik-Mendel (EKM) Algorithms [21], [24] for computing the centroid of IT2 FSs, type-reduction of IT2 fuzzy logic systems, etc. So, $ss_{Rl}(\tilde{A}, \tilde{B})$ can also be computed using a modified KM Algorithm; however, our experiments showed that this approach is slower than our efficient algorithm. So, its details are not presented here.

Define

$$a \equiv \sum_{x_i \in I_1} \underline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_2} \underline{\mu}_{\tilde{A}}(x_i) \tag{20}$$

$$b \equiv \sum_{x_i \in I_1} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2} \underline{\mu}_{\tilde{A}}(x_i) \tag{21}$$

$$y(I_5, I_6) \equiv \frac{\sum_{x_i \in I_5} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6} \overline{\mu}_{\tilde{A}}(x_i) + b} \tag{22}$$

Then, (19) can be re-expressed as

$$ss_{Rl}(\tilde{A}, \tilde{B}) = \min_{I_5, I_6} y(I_5, I_6). \tag{23}$$

The following theorem specifies the properties of $y(I_5, I_6)$ when $ss_{Rl}(\tilde{A}, \tilde{B})$ is obtained. The proof is given in the Appendix.

Theorem 3: If $y(I_5, I_6)$ cannot be reduced by converting any *single* x_k in I_5 to I_6, i.e., $\forall x_k \in I_5$,

$$\frac{\sum_{x_i \in I_5 \setminus x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \cup x_k} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5 \setminus x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \cup x_k} \overline{\mu}_{\tilde{A}}(x_i) + b} \geqslant y(I_5, I_6) \tag{24}$$

where $I_5 \setminus x_k$ denotes a set obtained by removing x_k from I_5, and $y(I_5, I_6)$ cannot be reduced either by converting any *single* x_k in I_6 to I_5, i.e., $\forall x_k \in I_6$,

$$\frac{\sum_{x_i \in I_5 \cup x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \setminus x_k} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5 \cup x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \setminus x_k} \overline{\mu}_{\tilde{A}}(x_i) + b} \geqslant y(I_5, I_6) \tag{25}$$

Then $ss_{Rl}(\tilde{A}, \tilde{B}) = y(I_5, I_6)$. □

According to Theorem 3, given an arbitrary initialization of $ss_{Rl}(\tilde{A}, \tilde{B})$, one can switch $\mu_{A_l}(x_i)$ from $\underline{\mu}_{\tilde{A}}(x_i)$ to $\overline{\mu}_{\tilde{A}}(x_i)$ or from $\overline{\mu}_{\tilde{A}}(x_i)$ to $\underline{\mu}_{\tilde{A}}(x_i)$ for $\forall x_i \in I$ gradually until any single switch of $\mu_{A_l}(x_i)$ cannot reduce $ss_{Rl}(\tilde{A}, \tilde{B})$ further, as described by Algorithm 2. Observe that this algorithm is very similar to the sequential minimal optimization (SMO) algorithm [12] for training support vector machines (SVMs) [18].

Algorithm 2: New Efficient Algorithm for Computing $ss_{Rl}(\tilde{A}, \tilde{B})$

```
Find I_1 in (6), I_2 in (7), and I in (8)
n = Σ_{x_i ∈ I_1} μ_Ã(x_i) + Σ_{x_i ∈ I∪I_2} μ_B̃(x_i)
d = Σ_{x_i ∈ I_1} μ_Ã(x_i) + Σ_{x_i ∈ I∪I_2} μ̄_Ã(x_i)
ss_Rl(Ã, B̃) = n/d
ss_0 = 1
Denote {x_j} in I as x_1, x_2, ..., x_L
sign_j = −1, j = 1, 2, ..., L
while ss_0 > ss_Rl(Ã, B̃)
    ss_0 = ss_Rl(Ã, B̃)
    for j = 1 to L
        n' = n + sign_j · (μ_B̃(x_j) − μ_Ã(x_j))
        d' = d + sign_j · (μ̄_Ã(x_j) − μ_Ã(x_j))
        if n'/d' < ss_Rl(Ã, B̃)
            sign_j = −sign_j
            n = n'
            d = d'
            ss_Rl(Ã, B̃) = n/d
        end
    end
end
```

E. Comparative Studies

Simulations were performed to compare the three algorithms for computing $ss_{Rl}(\tilde{A}, \tilde{B})$. The platform was a Dell PWS690 running Windows XP X64 and Matlab R2007a with Intel Xeon 5150@2.66GHz processor and 2GB RAM.

N was chosen to be $\{10, 20, 50, 100, 200, 500, 1000\}$. For each N, 10,000 Monte Carlo simulations were used to compute $ss_{Rl}(\tilde{A}, \tilde{B})$, i.e., for each N, 10,000 $\underline{\mu}_{\tilde{B}}(x_i)$ were generated from random IT2 FSs with trapezoidal upper and lower membership functions, and 10,000 pairs of $\{\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)\}$ were generated in a similar way.

The total computation time of 10,000 Monte Carlo simulations for the three algorithms are shown in Table I for different N. Observe that both Nguyen and Kreinovich's fast algorithm and our efficient algorithm outperform the exhaustive computation approach significantly, and our efficient algorithm is faster than Nguyen and Kreinovich's fast algorithm, though the performance improvement decreases as N increases[2].

TABLE I

TOTAL COMPUTATION TIME (10,000 MONTE CARLO RUNS) OF THE THREE ALGORITHMS USED TO COMPUTE $ss_{Rl}(\tilde{A}, \tilde{B})$.

N	Exhaustive Computation (s)	Fast Algorithm t_{NK} (s)	Efficient Algorithm t_{WM} (s)	$\frac{t_{NK}-t_{WM}}{t_{NK}}$
10	31.1902	0.5897	0.4221	28.42%
20	—	0.6022	0.4421	26.59%
50	—	0.6944	0.5336	23.17%
100	—	0.8111	0.6350	21.71%
200	—	0.9973	0.8136	18.42%
500	—	1.5823	1.4332	9.42 %
1,000	—	2.5202	2.3276	7.64 %

III. sm_{NK}: DEFINITION AND COMPUTATION

Nguyen and Kreinovich [11] extended the Jaccard similarity measure for T1 FSs to IT2 FSs. This section presents their results and compares several algorithms for computing sm_{NK}.

A. Definition

Let $\tilde{A}$ and $\tilde{B}$ be two IT2 FSs, and A_e and B_e be their embedded T1 FSs. Then, the similarity of $\tilde{A}$ in $\tilde{B}$,

[2]For our efficient algorithm, the number of *while* iterations is $O(N^a)$, where a is a very small positive number, i.e., the number of *while* iterations increases slightly as N increases. Meanwhile, the time to finish each *while* iteration is $O(N)$, so the total computational cost of the *while* loop is $O(N^{1+a})$. For the fast algorithm, the *for* loop has computational cost $O(N)$. So, as N increases, the computational cost saving of the efficient algorithm, gained by not computing and sorting $\{r_j\}$, diminishes.

$sm_{NK}(\tilde{A},\tilde{B})$, is an interval defined as [11]

$$\begin{aligned} sm_{NK}(\tilde{A},\tilde{B}) &= \bigcup_{\forall A_e,B_e} sm_K(A_e,B_e) \\ &= \bigcup_{\forall A_e,B_e} \frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\max(\mu_{A_e}(x_i),\mu_{B_e}(x_i))} \\ &\equiv [sm_{NKl}(\tilde{A},\tilde{B}), sm_{NKr}(\tilde{A},\tilde{B})] \end{aligned} \tag{26}$$

where

$$sm_{NKl}(\tilde{A},\tilde{B}) = \min_{\forall A_e,B_e} \frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\max(\mu_{A_e}(x_i),\mu_{B_e}(x_i))} \tag{27}$$

$$sm_{NKr}(\tilde{A},\tilde{B}) = \max_{\forall A_e,B_e} \frac{\sum_{i=1}^{N}\min(\mu_{A_e}(x_i),\mu_{B_e}(x_i))}{\sum_{i=1}^{N}\max(\mu_{A_e}(x_i),\mu_{B_e}(x_i))} \tag{28}$$

Nguyen and Kreinovich [11] showed that it is not necessary to enumerate all possible A_e and B_e in order to compute $sm_{NKl}(\tilde{A},\tilde{B})$ and $sm_{NKr}(\tilde{A},\tilde{B})$, as suggested by (27) and (28). Their main results are summarized in the following:

Theorem 4: [11] Let

$$I_1 \equiv \{x_i | \underline{\mu}_{\tilde{B}}(x_i) > \overline{\mu}_{\tilde{A}}(x_i)\} \tag{29}$$

$$I_2 \equiv \{x_i | \underline{\mu}_{\tilde{A}}(x_i) > \overline{\mu}_{\tilde{B}}(x_i)\} \tag{30}$$

$$I \equiv \{x_i | \max(\underline{\mu}_{\tilde{A}}(x_i), \underline{\mu}_{\tilde{B}}(x_i)) \leqslant \min(\overline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{B}}(x_i))\} \tag{31}$$

i.e., I is the set of all x_i whose $[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]$ and $[\underline{\mu}_{\tilde{B}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)]$ overlap. Define

$$\mu_{A_l}(x_i) = \begin{cases} \underline{\mu}_{\tilde{A}}(x_i), & x_i \in I_1 \\ \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I_2 \\ \underline{\mu}_{\tilde{A}}(x_i) \text{ or } \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I \end{cases} \tag{32}$$

$$\mu_{B_l}(x_i) = \begin{cases} \underline{\mu}_{\tilde{B}}(x_i), & \mu_{A_l}(x_i) = \overline{\mu}_{\tilde{A}}(x_i) \\ \overline{\mu}_{\tilde{B}}(x_i), & \mu_{A_l}(x_i) = \underline{\mu}_{\tilde{A}}(x_i) \end{cases} \tag{33}$$

$$\mu_{A_r}(x_i) = \begin{cases} \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I_1 \\ \underline{\mu}_{\tilde{A}}(x_i), & x_i \in I_2 \\ \min(\overline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)), & x_i \in I \end{cases} \tag{34}$$

$$\mu_{B_r}(x_i) = \begin{cases} \underline{\mu}_{\tilde{B}}(x_i), & x_i \in I_1 \\ \overline{\mu}_{\tilde{B}}(x_i), & x_i \in I_2 \\ \min(\overline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)), & x_i \in I \end{cases} \tag{35}$$

Then,

$$sm_{NK}(\tilde{A},\tilde{B}) \equiv [sm_{NKl}(\tilde{A},\tilde{B}), sm_{NKr}(\tilde{A},\tilde{B})] \tag{36}$$

where

$$\begin{aligned} &sm_{NKl}(\tilde{A},\tilde{B}) \\ &= \min_{\substack{\mu_{A_l}(x_i) \text{ in (32)} \\ \mu_{B_l}(x_i) \text{ in (33)}}} \left[\frac{\sum_{i=1}^{N}\min\left(\mu_{A_l}(x_i),\mu_{B_l}(x_i)\right)}{\sum_{i=1}^{N}\max\left(\mu_{A_l}(x_i),\mu_{B_l}(x_i)\right)} \right] \end{aligned} \tag{37}$$

$$sm_{NKr}(\tilde{A},\tilde{B}) = \frac{\sum_{i=1}^{N}\min\left(\mu_{A_r}(x_i),\mu_{B_r}(x_i)\right)}{\sum_{i=1}^{N}\max\left(\mu_{A_r}(x_i),\mu_{B_r}(x_i)\right)}. \quad \square \tag{38}$$

B. The Exhaustive Computation Approach

Observe from (38) that $sm_{NKr}(\tilde{A},\tilde{B})$ has a closed-form solution; however, because for each $x_i \in I$, $\mu_{A_l}(x_i)$ in (32) can have two possible values, to compute $sm_{NKl}(\tilde{A},\tilde{B})$, 2^L evaluations of the bracketed term in (37) have to be performed, where L is the number of elements in I. This approach is referred to in this paper as the "exhaustive computation approach" for computing $sm_{NKl}(\tilde{A},\tilde{B})$.

Because 2^L can be a very large number depending on L, the exhaustive computation approach is not efficient. Two faster algorithms for computing $sm_{NKl}(\tilde{A},\tilde{B})$ are introduced next.

C. Nguyen and Kreinovich's Fast Algorithm

Without loss of generality, assume $\overline{\mu}_{\tilde{A}}(x_i) \geqslant \overline{\mu}_{\tilde{B}}(x_i)$ for $\forall x_i$. If this is not true, then we can swap $[\underline{\mu}_{\tilde{A}}(x_j), \overline{\mu}_{\tilde{A}}(x_i)]$ and $[\underline{\mu}_{\tilde{B}}(x_j), \overline{\mu}_{\tilde{B}}(x_i)]$, and this does not affect the value of $sm_{NKl}(\tilde{A},\tilde{B})$.

Theorem 5: [11] Define

$$r_j = \frac{\underline{\mu}_{\tilde{B}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j)}{\overline{\mu}_{\tilde{A}}(x_j) - \overline{\mu}_{\tilde{B}}(x_j)}, \quad x_j \in I \tag{39}$$

$$I_3 = \{x_j | r_j \leqslant sm_{Rl}(x_j)\} \tag{40}$$

$$I_4 = \{x_j | r_j > sm_{Rl}(x_j)\} \tag{41}$$

Then,

$$ss_{Rl}(\tilde{A},\tilde{B}) = \frac{\sum_{x_j \in I_1 \cup I_4} \underline{\mu}_{\tilde{A}}(x_j) + \sum_{x_j \in I_2 \cup I_3} \underline{\mu}_{\tilde{B}}(x_j)}{\sum_{x_j \in I_1 \cup I_4} \overline{\mu}_{\tilde{B}}(x_j) + \sum_{x_j \in I_2 \cup I_3} \overline{\mu}_{\tilde{A}}(x_j)}. \quad \square \tag{42}$$

Theorem 5 is the basis of Nguyen and Kreinovich's [11] fast algorithm for computing $sm_{NKl}(\tilde{A},\tilde{B})$, as shown in Algorithm 3. The rationale is similar to that explained in Section II-C. Algorithm 3 is much faster than the exhaustive computation approach, as demonstrated in Section III-E; however, it needs to swap $[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]$ and $[\underline{\mu}_{\tilde{B}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)]$ and to rank r_j, which requires extra memory and is also time-consuming. A more efficient algorithm, which avoids these steps, is proposed next.

D. New Efficient Algorithm for Computing $sm_{NKl}(\tilde{A},\tilde{B})$

Theorem 4 indicates that in computing $sm_{NKl}(\tilde{A},\tilde{B})$, $\mu_{A_l}(x_i)$ can either be $\underline{\mu}_{\tilde{A}}(x_i)$ or $\overline{\mu}_{\tilde{A}}(x_i)$ for $x_i \in I$. Let I_5 be an arbitrary subset of I, and I_6 be its complementary set in I. Define

$$A_e \equiv \begin{cases} \underline{\mu}_{\tilde{A}}(x_i), & x_i \in I_1 \cup I_5 \\ \overline{\mu}_{\tilde{A}}(x_i), & x_i \in I_2 \cup I_6 \end{cases} \tag{43}$$

$$B_e \equiv \begin{cases} \overline{\mu}_{\tilde{B}}(x_i), & \mu_{A_e}(x_i) = \underline{\mu}_{\tilde{A}}(x_i) \\ \underline{\mu}_{\tilde{B}}(x_i), & \mu_{A_e}(x_i) = \overline{\mu}_{\tilde{A}}(x_i) \end{cases} \tag{44}$$

Then, $sm_{NKl}(\tilde{A},\tilde{B})$ can be re-written as

$$\begin{aligned} &sm_{NKl}(\tilde{A},\tilde{B}) \\ &= \min_{I_5,I_6} \left[\frac{\sum_{x_i \in I_1 \cup I_5} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2 \cup I_6} \underline{\mu}_{\tilde{B}}(x_i)}{\sum_{x_i \in I_1 \cup I_5} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_2 \cup I_6} \overline{\mu}_{\tilde{A}}(x_i)} \right] \end{aligned} \tag{45}$$

Algorithm 3:* Nguyen and Kreinovich's Fast Algorithm for Computing $sm_{NKl}(\tilde{A},\tilde{B})$

for $i = 1$ to N
  if $\overline{\mu}_{\tilde{A}}(x_i) < \overline{\mu}_{\tilde{B}}(x_i)$
    Swap $[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]$ and $[\underline{\mu}_{\tilde{B}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)]$
  end
end
$n = \sum_{i=1}^{N} \underline{\mu}_{\tilde{B}}(x_i)$
$d = \sum_{i=1}^{N} \overline{\mu}_{\tilde{A}}(x_i)$
$sm_0 = n/d$
Find I_1 in (29)
Construct I' as the complement of I_1 in $\{x_i\}_{i=1}^{N}$
for each $x_j \in I'$
  Compute r_j in (39)
end
Sort $\{r_j\}$ in *descending* order and call them $r_1, r_2, ..., r_L$
Match $\{x_j\}$ in I with the order of $\{r_j\}$ and call them $x_1, x_2, ..., x_L$
for $j = 1$ to L
  $n = n - \underline{\mu}_{\tilde{B}}(x_j) + \underline{\mu}_{\tilde{A}}(x_j)$
  $d = d - \overline{\mu}_{\tilde{A}}(x_j) + \overline{\mu}_{\tilde{B}}(x_j)$
  $sm_j = n/d$
end
$sm_{NKl}(\tilde{A},\tilde{B}) = \min_{j=0,1,...L} sm_j$

* This algorithm is not exactly the same as that in [11] because the latter has some typographical errors.

Define

$$a \equiv \sum_{x_i \in I_1} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_2} \underline{\mu}_{\tilde{B}}(x_i) \tag{46}$$

$$b \equiv \sum_{x_i \in I_1} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_2} \overline{\mu}_{\tilde{A}}(x_i) \tag{47}$$

$$y(I_5, I_6) \equiv \frac{\sum_{x_i \in I_5} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_6} \overline{\mu}_{\tilde{A}}(x_i) + b} \tag{48}$$

Then, (45) can be re-expressed as

$$sm_{NKl}(\tilde{A},\tilde{B}) = \min_{I_5, I_6} y(I_5, I_6) \tag{49}$$

The following theorem specifies the properties of $y(I_5, I_6)$ when $sm_{NKl}(\tilde{A},\tilde{B})$ is obtained. Its proof is very similar to that in the Appendix, and is left to the reader as an exercise.

Theorem 6: If $y(I_5, I_6)$ cannot be reduced by converting any *single* x_k in I_5 to I_6, i.e., $\forall x_k \in I_5$,

$$\frac{\sum_{x_i \in I_5 \setminus x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \cup x_k} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5 \setminus x_k} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_6 \cup x_k} \overline{\mu}_{\tilde{A}}(x_i) + b} \geqslant y(I_5, I_6) \tag{50}$$

and $y(I_5, I_6)$ cannot be reduced either by converting any *single* x_k in I_6 to I_5, i.e., $\forall x_k \in I_6$,

$$\frac{\sum_{x_i \in I_5 \cup x_k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I_6 \setminus x_k} \underline{\mu}_{\tilde{B}}(x_i) + a}{\sum_{x_i \in I_5 \cup x_k} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I_6 \setminus x_k} \overline{\mu}_{\tilde{A}}(x_i) + b} \geqslant y(I_5, I_6) \tag{51}$$

Then $sm_{NKl}(\tilde{A},\tilde{B}) = y(I_5, I_6)$. □

According to Theorem 6, given an arbitrary initialization of $sm_{NKl}(\tilde{A},\tilde{B})$, one can switch $\mu_{A_l}(x_i)$ from $\underline{\mu}_{\tilde{A}}(x_i)$ to $\overline{\mu}_{\tilde{A}}(x_i)$ or from $\overline{\mu}_{\tilde{A}}(x_i)$ to $\underline{\mu}_{\tilde{A}}(x_i)$ [$\mu_{B_l}(x_i)$ is updated accordingly] for $\forall x_i \in I$ gradually until any single switch of $\mu_{A_l}(x_i)$ cannot reduce $sm_{NKl}(\tilde{A},\tilde{B})$ further, as described by Algorithm 4.

Algorithm 4: New Efficient Algorithm for Computing $sm_{NKl}(\tilde{A},\tilde{B})$

Find I_1 in (29), I_2 in (30), and I in (31)
$n = \sum_{x_i \in I_1} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{x_i \in I \cup I_2} \underline{\mu}_{\tilde{B}}(x_i)$
$d = \sum_{x_i \in I_1} \overline{\mu}_{\tilde{B}}(x_i) + \sum_{x_i \in I \cup I_2} \overline{\mu}_{\tilde{A}}(x_i)$
$sm_{NKl}(\tilde{A},\tilde{B}) = n/d$
$sm_0 = 1$
Denote $\{x_j\}$ in I as $x_1, x_2, ..., x_L$
$\text{sign}_j = -1$, $j = 1, 2, ..., L$
while $sm_0 > sm_{NKl}(\tilde{A},\tilde{B})$
  $sm_0 = ss_{NKl}(\tilde{A},\tilde{B})$
  for $j = 1$ to L {
    $n' = n + \text{sign}_j \cdot (\underline{\mu}_{\tilde{B}}(x_j) - \underline{\mu}_{\tilde{A}}(x_j))$
    $d' = d + \text{sign}_j \cdot (\overline{\mu}_{\tilde{A}}(x_j) - \overline{\mu}_{\tilde{B}}(x_j))$
    if $n'/d' < sm_{NKl}(\tilde{A},\tilde{B})$
      $\text{sign}_j = -\text{sign}_j$
      $n = n'$
      $d = d'$
      $sm_{NKl}(\tilde{A},\tilde{B}) = n/d$
    end
  end
end

E. Comparative Studies

Simulations were performed to compare the three algorithms for computing $sm_{NKl}(\tilde{A},\tilde{B})$. The platform was the same as that described in Section II-E. This simulation setup was also similar, except that here 10,000 pairs of $\{\underline{\mu}_{\tilde{B}}(x_i), \overline{\mu}_{\tilde{B}}(x_i)\}$ instead of $\underline{\mu}_{\tilde{B}}(x_i)$ were also generated from random IT2 FSs.

The total computation time of 10,000 Monte Carlo simulations for the three algorithms are shown in Table II for different N. Observe that both our efficient algorithm and Nguyen and Kreinovich's fast algorithm outperform the exhaustive computation approach significantly. Our efficient algorithm is also faster than Nguyen and Kreinovich's fast algorithm.

TABLE II
TOTAL COMPUTATION TIME (10,000 MONTE CARLO RUNS) OF THE THREE ALGORITHMS FOR $sm_{NKl}(\tilde{A},\tilde{B})$.

N	Exhaustive Computation (s)	Fast Algorithm t_{NK} (s)	Efficient Algorithm t_{WM} (s)	$\frac{t_{NK}-t_{WM}}{t_{NK}}$
10	131.7550	0.5042	0.4214	16.43%
20	—	0.5228	0.4279	18.16%
50	—	0.6347	0.5317	16.23%
100	—	0.7515	0.6421	14.56%
200	—	0.9798	0.8592	12.30%
500	—	1.6897	1.5251	9.74%
1,000	—	2.9063	2.6480	8.89%

IV. CONCLUSIONS

Subsethood and similarity measures are important concepts in FS theory. In this paper, Rickard et al.'s definition of IT2 FS subsethood measure and Nguyen and Kreinovich's IT2 FS similarity measure have been introduced. Efficient algorithms for computing them were also proposed. Simulations demonstrated that our proposed algorithms outperform existing algorithms in the literature.

$$\frac{\sum_{x_i\in I_5\setminus x_k}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{x_i\in I_6\cup x_k}\underline{\mu}_{\tilde{B}}(x_i)+a}{\sum_{x_i\in I_5\setminus x_k}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{x_i\in I_6\cup x_k}\overline{\mu}_{\tilde{A}}(x_i)+b}=\frac{\sum_{x_i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{x_i\in I_6}\underline{\mu}_{\tilde{B}}(x_i)+a+[\underline{\mu}_{\tilde{B}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)]}{\sum_{x_i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{x_i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b+[\overline{\mu}_{\tilde{A}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)]}$$
$$=\frac{y(I_5,I_6)[\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b]+[\underline{\mu}_{\tilde{B}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)]}{\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b+[\overline{\mu}_{\tilde{A}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)]} \quad (52)$$

Finally, note that although we have studied Rickard et al.'s IT2 FS subsethood measure and Nguyen and Kreinovich's IT2 FS similarity measure in this paper, this does not mean that they are always the best subsethood and similarity measure for IT2 FSs, e.g., in [9], [27] we have shown that Vlachos and Sergiadis's IT2 FS subsethood measure [19] outperforms $ss_{_R}$ in computing with words, and in [9], [23] we have shown that a similarity measure constructed based on the average cardinality [20], [22] may be the best in computing with words; however, $ss_{_R}$ and $sm_{_{NK}}$ may be more suitable for applications other than computing with words, and hence efficient algorithms for computing them are useful.

APPENDIX

A. Proof of Theorem 3

The left hand side of (24) can be rewritten as (52) on top of this page,

where the last line follows from (22). Substituting (52) into (24) and rearranging, it follows that

$$\underline{\mu}_{\tilde{B}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)\geqslant y(I_5,I_6)[\overline{\mu}_{\tilde{A}}(x_k)-\underline{\mu}_{\tilde{A}}(x_k)] \quad (53)$$

Similarly, from (25) it is obtained that

$$\underline{\mu}_{\tilde{A}}(x_k)-\underline{\mu}_{\tilde{B}}(x_k)\geqslant y(I_5,I_6)[\underline{\mu}_{\tilde{A}}(x_k)-\overline{\mu}_{\tilde{A}}(x_k)] \quad (54)$$

To show that $ss_{_{Rl}}(\tilde{A},\tilde{B})=y(I_5,I_6)$, i.e., $y(I_5,I_6)$ is the global minimum, it only needs to show that for any $I_5'\subseteq I_5$ and any $I_6'\subseteq I_6$,

$$\frac{\sum_{i\in I_5\cup I_6'\setminus I_5'}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6\cup I_5'\setminus I_6'}\underline{\mu}_{\tilde{B}}(x_i)+a}{\sum_{i\in I_5\cup I_6'\setminus I_5'}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6\cup I_5'\setminus I_6'}\overline{\mu}_{\tilde{A}}(x_i)+b}\geqslant y(I_5,I_6) \quad (55)$$

i.e., the switch of *any arbitrary number* of x_i from I_5 to I_6 and *any arbitrary number* of x_i from I_6 to I_5 will increase $y(I_5,I_6)$. The correctness of (55) is shown in (56) on the next page.

The next-to-the-last line of (56) has made use of (53) and (54).

REFERENCES

[1] H. Bustince, "Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets," *International Journal of Approximate Reasoning*, vol. 23, no. 3, pp. 137–209, 2000.

[2] V. V. Cross and T. A. Sudkamp, *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications*. Heidelberg, NY: Physica-Verlag, 2002.

[3] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bulletin de la Societe de Vaud des Sciences Naturelles*, vol. 44, p. 223, 1908.

[4] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, pp. 195–220, 2001.

[5] B. Kosko, "Fuzziness vs. probability," *International Journal of General Systems*, vol. 17, pp. 211–240, 1990.

[6] C.-T. Lin, "Adaptive subsethood for neural fuzzy control," *International Journal of Systems Science*, vol. 27, no. 10, pp. 937–955, 1996.

[7] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[8] J. M. Mendel, R. I. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Trans. on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[9] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, NJ: Wiley-IEEE Press, 2010.

[10] F. Morabito and M. Versaci, "On the use of fuzzy subsethood and supersethood for the plasma classification problem in tokamak reactors," in *Proc. 2nd Int'l Conf. on Neural Networks and Artificial Intelligence*, Minsk, Belarus, October 2001.

[11] H. T. Nguyen and V. Kreinovich, "Computing degrees of subsethood and similarity for interval-valued fuzzy sets: Fast algorithms," in *Proc. 9th Int'l Conf. on Intelligent Technologies*, Samui, Thailand, October 2008, pp. 47–55.

[12] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods: Support vector machines*, B. Scholkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[13] S. Raha, N. Pal, and K. Ray, "Similarity-based approximate reasoning: Methodology and application," *IEEE Trans. on Systems, Man, and Cybernetics–A*, vol. 32, no. 4, pp. 541– 547, 2002.

[14] K. Rasmani and Q. Shen, "Subsethood-based fuzzy rule models and their application to student performance classification," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Reno, NV, May 2005, pp. 755–760.

[15] J. T. Rickard, J. Aisbett, G. Gibbon, and D. Morgenthaler, "Fuzzy subsethood for type-n fuzzy sets," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, New York, May 2008.

[16] J. T. Rickard, J. Aisbett, and G. Gibbon, "Fuzzy subsethood for fuzzy sets of type-2 and generalized type-n," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 1, pp. 50–60, 2009.

[17] K. Saastamoinen and J. Ketola, "Medical data classification using logical similarity based measures," in *Proc. IEEE Int'l Conf. on Cybernetics and Intelligent Systems*, Bangkok, Thailand, June 2006, pp. 1–5.

[18] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag, 1995.

[19] I. Vlachos and G. Sergiadis, "Subsethood, entropy, and cardinality for interval-valued fuzzy sets – an algebraic derivation," *Fuzzy Sets and Systems*, vol. 158, pp. 1384–1396, 2007.

[20] D. Wu and J. M. Mendel, "Cardinality, fuzziness, variance and skewness of interval type-2 fuzzy sets," in *Proc. 1st IEEE Symposium on Foundations of Computational Intelligence*, Honolulu, HI, April 2007, pp. 375–382.

[21] ——, "Enhanced Karnik-Mendel Algorithms for interval type-2 fuzzy sets and systems," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, San Diego, CA, June 2007, pp. 184–189.

[22] ——, "Uncertainty measures for interval type-2 fuzzy sets," *Information Sciences*, vol. 177, no. 23, pp. 5378–5393, 2007.

[23] ——, "A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 8, pp. 1169–1192, 2009.

[24] ——, "Enhanced Karnik-Mendel Algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.

[25] ——, "Perceptual reasoning for perceptual computing: A similarity-based approach," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 6, pp. 1397–1411, 2009.

$$
\begin{aligned}
&\frac{\sum_{i\in I_5\cup I_6'\setminus I_5'}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6\cup I_5'\setminus I_6'}\underline{\mu}_{\tilde{B}}(x_i)+a}{\sum_{i\in I_5\cup I_6'\setminus I_5'}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6\cup I_5'\setminus I_6'}\overline{\mu}_{\tilde{A}}(x_i)+b}\\
=&\frac{\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\underline{\mu}_{\tilde{B}}(x_i)+a+\sum_{i\in I_5'}\left[\underline{\mu}_{\tilde{B}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{B}}(x_i)\right]}{\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b+\sum_{i\in I_5'}\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\overline{\mu}_{\tilde{A}}(x_i)\right]}\\
=&\frac{y(I_5,I_6)\left[\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b\right]+\sum_{i\in I_5'}\left[\underline{\mu}_{\tilde{B}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{B}}(x_i)\right]}{\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b+\sum_{i\in I_5'}\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\overline{\mu}_{\tilde{A}}(x_i)\right]}\\
\geqslant&\frac{y(I_5,I_6)\left[\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b\right]+y(I_5,I_6)\sum_{i\in I_5'}\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+y(I_5,I_6)\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\overline{\mu}_{\tilde{A}}(x_i)\right]}{\sum_{i\in I_5}\underline{\mu}_{\tilde{A}}(x_i)+\sum_{i\in I_6}\overline{\mu}_{\tilde{A}}(x_i)+b+\sum_{i\in I_5'}\left[\overline{\mu}_{\tilde{A}}(x_i)-\underline{\mu}_{\tilde{A}}(x_i)\right]+\sum_{i\in I_6'}\left[\underline{\mu}_{\tilde{A}}(x_i)-\overline{\mu}_{\tilde{A}}(x_i)\right]}\\
=&y(I_5,I_6). \qquad (56)
\end{aligned}
$$

[26] ——, "Similarity-based perceptual reasoning for perceptual computing," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Jeju Island, South Korea, August 2009, pp. 700–705.

[27] ——, "Interval type-2 fuzzy set subsethood measures as a decoder for perceptual reasoning," Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, Tech. Rep. USC-SIPI Report 398, 2010.

[28] M.-S. Yang and D.-C. Lin, "On similarity and inclusion measures between type-2 fuzzy sets with an application to clustering," *Computers and Mathematics with Applications*, vol. 57, pp. 896–907, 2009.

[29] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[30] ——, "Similarity relations and fuzzy orderings," *Information Sciences*, vol. 3, pp. 177–200, 1971.

[31] R. Zwick, E. Carlstein, and D. Budescu, "Measures of similarity among fuzzy concepts: A comparative analysis," *International Journal of Approximate Reasoning*, vol. 1, pp. 221–242, 1987.