ELSEVIER

Information Sciences 132 (2001) 195–220

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Centroid of a type-2 fuzzy set

Nilesh N. Karnik, Jerry M. Mendel *

Department of Electrical Engineering-Systems, Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564, USA

Received 7 August 1999; received in revised form 8 October 2000; accepted 8 November 2000

Abstract

In this paper, we introduce the centroid and generalized centroid of a type-2 fuzzy set (both of which are essential for implementing a type-2 fuzzy logic system), and explain how to compute them. For practical use, we show how to compute the centroid of interval and Gaussian type-2 fuzzy sets. An exact computation procedure is provided for an interval type-2 set, whereas an approximation is provided for both interval and Gaussian type-2 sets. Examples are given that compare the exact computational results with the approximate results. © 2001 Elsevier Science Inc. All rights reserved.

Keywords: Type-2 fuzzy sets; Centroid; Interval type-2 sets; Gaussian type-2 sets

1. Introduction

We may view the output of a type-1 fuzzy logic system (FLS) – the defuzzified output – as analogous (not equal!) to the mean of a probability density function. Just as variance provides a measure of dispersion about the mean, and is always used to capture more about probabilistic uncertainty in practical statistical-based designs, a FLS also needs some measure of dispersion to capture more about its uncertainties than just a single number. Type-2 fuzzy logic provides this measure of dispersion, and seems to be as fundamental to the design of systems that include linguistic and/or numerical uncertainties, that translate into rule or input uncertainties, as variance is to the mean.

* Corresponding author. Tel.: +1-213-740-4445; fax: +1-213-740-4651.
E-mail address: mendel@sipi.usc.edu (J.M. Mendel).

0020-0255/01/$ - see front matter
PII: S 0 0 2 0 - 0 2 5 5 (0 1) 0 0 0 6 9 - X

The concept of a *type-2 fuzzy set* (or type-2 set, for short) was introduced by Zadeh [24] as an extension of the concept of an ordinary fuzzy set [henceforth called a *type-1 fuzzy set* (or type-1 set, for short)]. A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership value for each element of this set is a fuzzy set in $[0, 1]$, unlike a type-1 set where the membership value is a crisp number in $[0, 1]$.

Type-2 fuzzy sets allow us to handle linguistic uncertainties (as typified by the adage "words can mean different things to different people [18]") as well as numerical uncertainties. A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal [6], "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner". According to John [7], "Type-2 fuzzy sets allow for linguistic grades of membership, thus assisting in knowledge representation, and they also offer improvement on inferencing with type-1 sets".

Mizumoto and Tanaka [19] have studied the set theoretic operations of type-2 sets, properties of membership grades of such sets, and, have examined the operations of algebraic product and algebraic sum for them [20]. Nieminen [21] has provided more details about algebraic structure of type-2 sets. Dubois and Prade [5] and Kaufmann and Gupta [15] have discussed the *join* and *meet* operations between fuzzy numbers under minimum t-norm. Dubois and Prade [3–5] have also discussed fuzzy valued logic and have given a formula for the composition of type-2 relations as an extension of the type-1 sup-star composition, but this formula is only for minimum t-norm. Karnik and Mendel [11,13] have provided a general formula for the extended sup-star composition of type-2 relations. Type-2 fuzzy sets have already been used in a number of applications, including decision making [1,23], solving fuzzy relation equations [22], and pre-processing of data [8].

The output of a type-1 fuzzy logic system is a type-1 fuzzy set. This set is usually defuzzified and, as is well known, many of the most useful defuzzification methods involve a centroid calculation [17,2]. Recently, a type-2 FLS [10,11,14] has been developed, and its output is a type-2 fuzzy set. A major calculation in a type-2 FLS is *type-reduction* [12], which is an extension (using the Extension Principle [24]) of a type-1 defuzzification procedure. Consequently, in order to implement a type-2 FLS, one needs algorithms for computing the centroid of a type-2 fuzzy set.

We believe that the concept of a centroid of a type-2 fuzzy set is new, and present general results for it in Section 2, including a way to approximate its calculation, since its calculation is, in general, quite complex. We focus on the centroid of interval and Gaussian type-2 fuzzy sets in Sections 3 and 4, respectively, because these type-2 sets are quite useful in practical applications. Conclusions are given in Section 5.

In the sequel, we use the following notation and terminology. A is a type-1 fuzzy set, and the membership grade (a synonym for the degree of membership) of $x \in X$ in A is $\mu_A(x)$, which is a crisp number in $[0,1]$. A type-2 fuzzy set, denoted $\tilde{A}$, is characterized by a *type-2 membership function* $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e., $\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \, \forall x \in X, \forall u \in J_x \subseteq J_x[0,1]\}$, in which $0 \leqslant \mu_{\tilde{A}}(x,u) \leqslant 1$. At each value of x, say $x = x'$, the 2D plane whose axes are u and $\mu_{\tilde{A}}(x'u)$ is called a vertical slice of $\mu_{\tilde{A}}(x,u)$. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x,u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x' \in X$ and $\forall u \in J_{x'} \subseteq [0,1]$, i.e., $\mu_{\tilde{A}}(x = x', u) \triangleq \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u$, where $J_{x'} \subseteq [0,1]$. Because $\forall x' \in X$, we drop the prime notation on $\mu_{\tilde{A}}(x')$, and refer to $\mu_{\tilde{A}}(x)$ as a secondary membership function; it is a type-1 fuzzy set, which we also refer to as a *secondary set*. Based on the concept of secondary sets, we can reinterpret a type-2 fuzzy set as the union of all secondary sets, i.e., *we can re-express $\tilde{A}$ in a vertical-slice manner*, as $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | \, \forall x \in X\}$. The domain of a secondary membership function is called the *primary membership* of x; J_x is the primary membership of x, where $J_x \subseteq [0,1]$ for $\forall x \in X$. The *amplitude* of a secondary membership function is called a *secondary grade*; $f_x(u)$ is a secondary grade. Assume that each of the secondary membership functions of a type-2 fuzzy set has only one secondary grade that equals 1. A *principal membership function* is the union of all such points at which this occurs, i.e., $\mu_{\text{principal}}(x) = \int_{x \in X} u/x$, where $f_x(u) = 1$, and is associated with a type-1 fuzzy set.

When the secondary membership functions (MFs) of a type-2 fuzzy set are type-1 Gaussian MFs, we call the type-2 fuzzy set a *Gaussian type-2 set* (regardless of the shape of the principal MF). When the secondary MFs are type-1 interval sets, we call the type-2 set an *interval type-2 set*. Finally, $\sqcap$ denotes *meet* operation, and, $\sqcup$ denotes *join* operation, where meet and join are defined and explained in great detail in [11,13].

Fig. 1 shows an example of a Gaussian type-2 set, where the secondary MF for every point is a Gaussian type-1 set contained in $[0,1]$. Intensity of the shading is approximately proportional to secondary grades. Darker areas indicate larger secondary grades. Fig. 2 shows an interval type-2 set, where the secondary MF for every point is a crisp set, the domain of which is an interval contained in $[0,1]$. Because all the secondary grades are unity, the shading is uniform all over. In the sequel, we represent an interval set just by its domain interval, which can be represented by its left and right end-points as $[l,r]$, or by its center and spread as $[c-s, c+s]$, where $c = (l+r)/2$ and $s = (r-l)/2$.

2. Centroid of a type-2 FS: general results

The centroid of a type-1 set A, whose domain, $x \in X$, is discretized into N points, $x_1, x_2, \ldots, x_N$, is given as

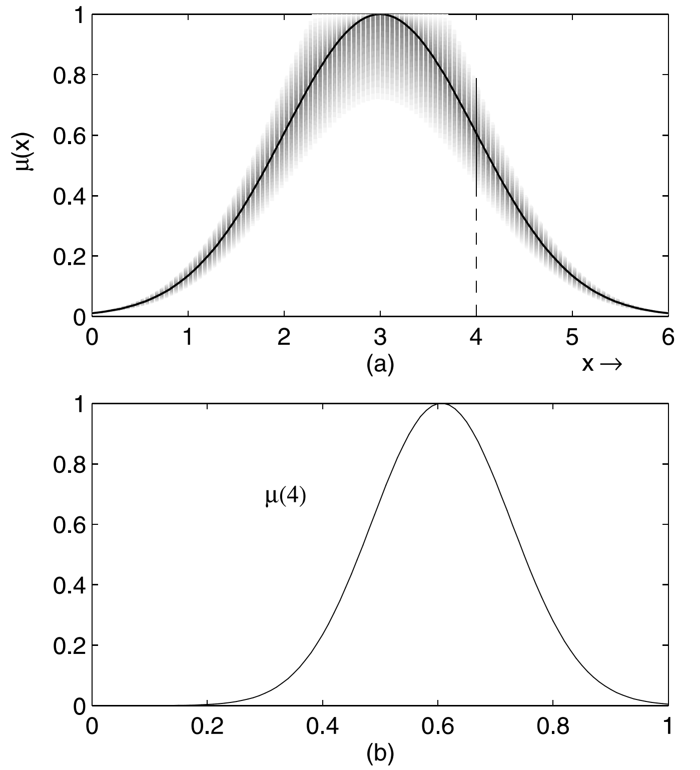


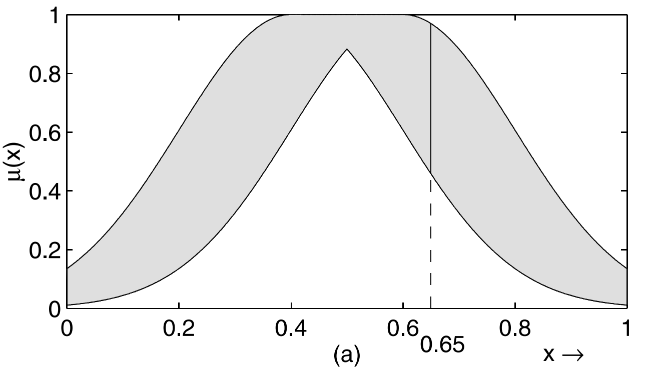
Fig. 1. (a) Pictorial representation of a Gaussian type-2 set. The standard deviations of the secondary Gaussians decrease by design, as x moves away from 3. The *principal* membership function, is indicated with a thick line; it is a Gaussian because of the way the set is constructed. The flat portion from about 2.5 to 3.5 appears because primary memberships cannot be greater than 1 and so the Gaussians have to be "clipped". The primary membership corresponding to $x = 4$ is also shown. The secondary membership function at $x = 4$ is shown in (b), and is also Gaussian.

$$c_A = \frac{\sum_{i=1}^{N} x_i \mu_A(x_i)}{\sum_{i=1}^{N} \mu_A(x_i)}. \tag{1}$$

Similarly, the centroid of a type-2 set $\tilde{A}$, $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | \, x \in X\}$, whose x-domain is discretized into N points, so that $\tilde{A} = \sum_{i=1}^{N} [\int_{u \in J_{x_i}} f_{x_i}(u)/u]/x_i$, can be defined using the Extension Principle as follows:

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} [f_{x_1}(\theta_1) \bigstar \cdots \bigstar f_{x_N}(\theta_N)] \Bigg/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i}. \tag{2}$$

$C_{\tilde{A}}$ is a type-1 fuzzy set. Let $\theta = [\theta_1, \ldots, \theta_N]^{\mathrm{T}}$,



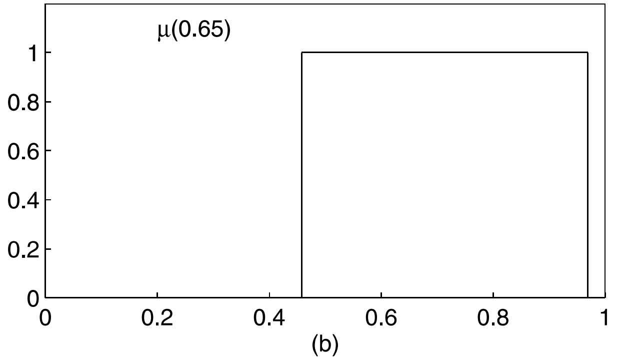


Fig. 2. (a) An interval type-2 set. The primary membership at $x = 0.65$ is also shown. The secondary membership function at $x = 0.65$ is shown in (b), and equals 1, i.e., the secondary MF is an interval type-1 set.

$$a(\theta) \triangleq \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i}, \tag{3}$$

$$b(\theta) \triangleq f_{x_1}(\theta_1) \bigstar \cdots \bigstar f_{x_N}(\theta_N). \tag{4}$$

The computation of $C_{\tilde{A}}$ involves computing the tuple (a, b) many times. Suppose, for example, (a, b) is computed α times; then, we can view the computation of $C_{\tilde{A}}$ as the computation of the α tuples $(a_1, b_1), (a_2, b_2), \ldots, (a_\alpha, b_\alpha)$.

Eq. (2) can also be described in words as follows. To find the centroid, we consider every possible combination $\{\theta_1, \ldots, \theta_N\}$ such that $\theta_i \in J_{x_i}$. For every such combination, we perform the type-1 centroid calculation in (1) by using θ_i in place of $\mu_{\tilde{A}}(x_i)$; and, to each point in the centroid, we assign a membership grade equal to the t-norm of the membership grades of the θ_i in the J_{x_i}. If more than one combination of θ_i gives us the same point in the centroid, we keep the one with the largest membership grade.

$C_{\tilde{A}}$ can also be expressed as

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} b(\theta)/a(\theta). \tag{5}$$

If two or more combinations of vector θ give the same point in the centroid set, $a(\theta)$, we keep the one with the largest value of $b(\theta)$.

Every combination $\{\theta_1, \ldots, \theta_N\}$ can be thought to form the membership function of some type-1 set A' which has the same domain as $\tilde{A}$. We call A' an *embedded type-1 set* in $\tilde{A}$. The centroid $C_{\tilde{A}}$ is a type-1 set whose elements are the centroids of all the embedded type-1 sets in $\tilde{A}$. The membership of an embedded set centroid in $C_{\tilde{A}}$ is calculated as the t-norm of the secondary grades corresponding to $\{\theta_1, \ldots, \theta_N\}$ that make up that embedded set.

The sequence of computations needed to obtain $C_{\tilde{A}}$ is as follows:

1. Discretize the x-domain into N points, $x_1, \ldots, x_N$.
2. Discretize each J_{x_j} (the primary memberships of x_j) into a suitable number of points, say M_j $(j = 1, \ldots, N)$.
3. Enumerate all the embedded sets; there will be $\prod_{j=1}^{N} M_j$ of them.
4. Compute the centroid using (2), i.e., compute the α tuples (a_i, b_i), $i = 1, 2, \ldots, \prod_{j=1}^{N} M_j$, where a_i and b_i are given in (3) and (4), respectively. In this case $\alpha = \prod_{j=1}^{N} M_j$.

Observe that if the domain of $\tilde{A}$ and/or $\mu_{\tilde{A}}(x)$ is continuous, the domain of $C_{\tilde{A}}$ is also continuous. The number of all the embedded type-1 sets in $\tilde{A}$, in this case, is uncountable; therefore, the domains of $\tilde{A}$ and each $\mu_{\tilde{A}}(x)$ have to be discretized for the calculation of $C_{\tilde{A}}$ (as explained in Section 2.1, we always use minimum t-norm for the centroid calculation of a type-2 set having a continuous domain, even though we use product t-norm everywhere else). In step 3 above, $\prod_{j=1}^{N} M_j$ can be very large even for small M_j and N. If, however, the secondary membership functions have a regular structure (e.g., interval or Gaussian sets), we can obtain the exact or approximate centroid without having to do all the calculations. For interval secondary MFs, the exact centroid can be obtained relatively easily, without performing computations for all the combinations, by using the computational procedure described in Section 3.1. For Gaussian and interval secondary MFs, under certain conditions, described in Section 4, centroids can be computed approximately.

2.1. Centroid computation using product t-norm

Calculation of the centroid of a type-2 set which has a continuous domain and not all of whose secondary grades are unity, using product t-norm, gives us an unexpected result. In the following, we discuss the problem and suggest a remedy.

We concentrate on type-2 sets having a continuous domain whose secondary membership functions are such that, for any domain point, only one primary membership has a secondary grade equal to one. Let $\tilde{A}$ be such a set. In the discussion associated with (2), we assumed that the domain of $\tilde{A}$ is discretized into N points. The true centroid of $\tilde{A}$ (assuming $\tilde{A}$ has a continuous domain) is the limit of $C_{\tilde{A}}$ in (2) as $N \to \infty$. When we use the product t-norm $\lim_{N\to\infty} \mathscr{T}_{i=1}^{N} f_{x_i}(\theta_i) = \lim_{N\to\infty} \prod_{i=1}^{N} f_{x_i}(\theta_i)$.

Let B be an embedded type-1 set in $\tilde{A}$. The centroid of B is computed as

$$c_B = \frac{\sum_{i=1}^{N} x_i \mu_B(x_i)}{\sum_{i=1}^{N} \mu_B(x_i)} \tag{6}$$

and the membership of c_B in $C_{\tilde{A}}$ (denoted as $\mu_C(c_B)$] is

$$\mu_C(c_B) = \prod_{i=1}^{N} f_{x_i}(\theta_i), \tag{7}$$

where $\{\theta_1, \ldots, \theta_N\}$ are the primary memberships that make up the type-1 set B. Also, let A denote the principal membership function of $\tilde{A}$. Obviously, $\mu_C(c_A) = 1$. Observe that:

1. $\lim_{N\to\infty} \mu_C(c_B)$ is non-zero only if B differs from A in *at most a finite number of points*. For all other embedded sets B, the product of an infinite number of quantities less than one will cause $\mu_C(c_B)$ to go to zero as $N \to \infty$.

2. For any embedded set B, whose membership function differs from that of A in only a finite number of points (i.e., when $\mu_B(x) \neq \mu_A(x)$, for only a finite number of points x), $c_B = c_A$. This can be explained as follows. The (true) centroid of B is the limit of (6) as $N \to \infty$, i.e., $c_B = \int_x x\mu_B(x)\,dx / \int_x \mu_B(x)\,dx$, where $x \in B$. Since A and B share the same domain (both are embedded sets in $\tilde{A}$), $x \in A \iff x \in B$; and since $\mu_A(x)$ and $\mu_B(x)$ differ only in a finite number of points, $\int_x x\mu_B(x)\,dx = \int_x x\mu_A(x)\,dx$ and $\int_x \mu_B(x)\,dx = \int_x \mu_A(x)\,dx$; therefore, $c_B = c_A$.

From these two observations, we can see that the only point having non-zero membership in $C_{\tilde{A}}$ is equal to c_A; and its membership grade is equal to the supremum of the membership grades of all the embedded type-1 sets which have the same centroid, which is equal to 1 (since $\mu_C(c_A) = 1$). In other words, $C_{\tilde{A}} = 1/c_A = c_A$, i.e., the centroid of $\tilde{A}$ will be equal to a crisp number … the centroid of its principal membership function!

This problem occurs because, under the product t-norm, $\lim_{N\to\infty} \mathscr{T}_{i=1}^{N} f_{x_i}(\theta_i) = \lim_{N\to\infty} \prod_{i=1}^{N} f_{x_i}(\theta_i) = 0$, unless only a finite number of $f_{x_i}(\theta_i)$ are less than 1. The minimum t-norm does not cause such a problem. To avoid this problem, *we will always use the minimum t-norm to calculate the centroid of a type-2 set having a continuous domain.*

2.2. Generalized centroid

The centroid of a type-1 fuzzy set, in (1), is a weighted average, of the general form

$$y(z_1,\dots,z_N,w_1,\dots,w_N)=\frac{\sum_{l=1}^{N} w_l z_l}{\sum_{l=1}^{N} w_l}, \tag{8}$$

where $z_l \in \Re$ and $w_l \in [0,1]$ for $l=1,\dots,N$. So far, we have only considered the extension of (8) when w_l becomes a type-1 set, because most type-1 defuzzification methods fit this case. However, a recently defined center-of-sets defuzzifier [11,14] requires both w_l and z_l to become type-1 sets; hence, we are motivated to generalize the extension of (1) to (2) to the extension of (8). The result of doing this is called a *generalized centroid*, which we denote as GC.

If each z_l is replaced by a type-1 fuzzy set $Z_l \subset \Re$ with associated membership function $\mu_{Z_l}(z_l)$, and each w_l is replaced by a type-1 fuzzy set $W_l \subset [0,1]$ with associated membership function $\mu_{W_l}(w_l)$, then the extension of (8) – the generalized centroid – is

$$GC=\int_{z_1\in Z_1}\cdots\int_{z_N\in Z_N}\int_{w_1\in W_1}\cdots\int_{w_N\in W_N} \mathscr{T}_{l=1}^{N}\mu_{Z_l}(z_l)\bigstar\mathscr{T}_{l=1}^{N}\mu_{W_l}(w_l)\Big/\frac{\sum_{l=1}^{N} w_l z_l}{\sum_{l=1}^{N} w_l}, \tag{9}$$

where $\mathscr{T}$ and $\bigstar$ both indicate the t-norm used ... product or minimum. Observe that GC is a type-1 fuzzy set. In this case, we let $\theta=[z_1,\dots,z_N,w_1,\dots,w_N]^{\mathrm{T}}$ and re-express $a(\theta)$ and $b(\theta)$ in (3) and (4), as

$$a(\theta)=\sum_{l=1}^{N} w_l z_l \Big/ \sum_{l=1}^{N} w_l \tag{10}$$

and

$$b(\theta)=\mathscr{T}_{l=1}^{N}\mu_{Z_l}(z_l)\bigstar\mathscr{T}_{l=1}^{N}\mu_{W_l}(w_l). \tag{11}$$

The sequence of computations needed to obtain GC is as follows:
1. Discretize the domain of each type-1 fuzzy set Z_l into a suitable number of points, say N_l $(l=1,2,\dots,N)$.
2. Discretize the domain of each type-1 fuzzy set W_l into a suitable number of points, say M_l $(l=1,2,\dots,N)$.
3. Enumerate all the possible combinations $\theta=[z_1,\dots,z_N,w_1,\dots,w_N]^{\mathrm{T}}$ such that $z_l\in Z_l$ and $w_l \in W_l$. The total number of combinations will be $\prod_{j=1}^{N} M_j N_j$.
4. Compute the generalized centroid using (9), i.e., compute the α tuples (a_i,b_i), $i=1,2,\dots,\prod_{j=1}^{N} M_j N_j$, where a_i and b_i are given in (10) and (11), respectively. In this case $\alpha=\prod_{j=1}^{N} M_j N_j$.

2.3. Computational complexity

The centroid and generalized centroid have high computation complexity. As described at the beginning of this section, the centroid $C_{\tilde{A}}$, is a collection of the centroids of all embedded type-1 fuzzy sets. The centroid or generalized centroid operations for each type-1 set can be processed in parallel. The number of parallel processors depends on the sampling rates in the x-domain and primary MF domain. The computations for a in (3) show the computational complexity in terms of multiplications, additions, and divisions; and, the computations for b in (4) give the number of operations for t-norm. The number of parallel processors equals the number of (a_i, b_i) tuples.

For the type-2 centroid, if the x-domain is sampled to N points, and each primary MF is sampled to M_i points, then there are $N-1$ t-norm (minimum) operations, N multiplications, $2(N-1)$ additions and one division, and, $\prod_{i=1}^{N} M_i$ parallel processors are required. A similar analysis applies to the generalized centroid. Presently, parallel processing is not available for most researchers, so the computational complexity of centroid and generalized centroid computation is high. This motivates us to develop some approximate results.

2.4. Approximate result

In this section, we present a result that lets us approximate the generalized centroid of a type-2 fuzzy set as an affine combination of type-1 sets, provided that the amount of type-2 uncertainty is small.

Theorem 2.1. *Consider the generalized centroid in* (9). *If each Z_l is a type-1 set with support $[c_l - s_l, c_l + s_l]$, and if each W_l is also a type-1 set with support $[h_l - \Delta_l, h_l + \Delta_l]$, then*

$$GC \approx \sum_{l=1}^{N} \left[\underline{Z}_l \left(\frac{h_l}{\sum_l h_l} \right) + \underline{W}_l \left(\frac{c_l - \mathscr{C}}{\sum_l h_l} \right) \right] + \mathscr{C}, \tag{12}$$

where

$$\underline{Z}_l = Z_l - c_l, \tag{13}$$

$$\underline{W}_l = W_l - h_l, \tag{14}$$

and

$$\mathscr{C} = \frac{\sum_{l=1}^{N} h_l c_l}{\sum_{l=1}^{N} h_l}, \tag{15}$$

provided that

$$\frac{\sum_{l=1}^{N} \Delta_l}{\sum_{l=1}^{N} h_l} \ll 1. \tag{16}$$

The approximation improves as $\sum_{l=1}^{N} \Delta_l / \sum_{l=1}^{N} h_l$ *grows smaller, and the result is exact when* $\sum_{l=1}^{N} \Delta_l = 0$, *i.e., when* $\Delta_l = 0$ *for* $l = 1, \ldots, N$.

See Appendix B for the proof. Note that Theorem 2.1 *is true regardless of the specific nature of the* Z_l's or W_l's. This theorem is important, because, in general, it is much easier to compute the RHS of (12) than the RHS of (9). To compute (12), we need to use the algebraic sum operation given in Theorem A.1 or A.2 in Appendix A.

For the centroid of a type-2 fuzzy set, $Z_l = c_l$, so $\underline{Z}_l = 0$, and we obtain the following.

Corollary 2.1. *Consider the centroid in* (2), *when each* J_{x_i} *is a type*-1 *set with support* $[h_i - \Delta_i, h_i + \Delta_i]$; *then,*

$$C_{\tilde{A}} \approx \sum_{i=1}^{N} \underline{J}_{x_i} \left(\frac{x_i - \mathscr{C}}{\sum_i h_i} \right) + \mathscr{C}, \tag{17}$$

where

$$\underline{J}_{x_i} = J_{x_i} - h_i, \tag{18}$$

and

$$\mathscr{C} = \frac{\sum_{i=1}^{N} h_i x_i}{\sum_{i=1}^{N} h_i}, \tag{19}$$

provided that

$$\frac{\sum_{i=1}^{N} \Delta_i}{\sum_{i=1}^{N} h_i} \ll 1. \tag{20}$$

We illustrate the application of Theorem 2.1 and Corollary 2.1 in Sections 3 and 4 below. In these sections we comment on the constraint in (16) (or (20)).

3. Centroid of an interval type-2 fuzzy set

In this section, we focus on an interval type-2 fuzzy set. We first present a computational procedure that lets us compute the generalized centroid (which is given by (9)) exactly, without actually having to consider the centroids of all the embedded type-1 sets. Then, we present an approximate result that gives an expression for the generalized centroid if the amount of type-2 uncertainty is small in a sense to be defined.

Consider the generalized centroid in (9). If each Z_l and W_l $(l = 1, \ldots, N)$ is an interval type-1 set, then, using the fact that $\mu_{Z_l}(z_l) = \mu_{W_l}(w_l) = 1$, (9) can be rewritten as

$$GC = \int_{z_1 \in Z_1} \cdots \int_{z_N \in Z_N} \int_{w_1 \in W_1} \cdots \int_{w_N \in W_N} 1 \Big/ \frac{\sum_{l=1}^{N} w_l z_l}{\sum_{l=1}^{N} w_l} = [y_l, y_r]. \tag{21}$$

3.1. Exact result: computational procedure for generalized centroid

We present an iterative procedure to compute the generalized centroid GC, when each Z_l in (21) is an interval type-1 set, having center c_l and spread s_l $(s_l \geqslant 0)$, and when each W_l is also an interval type-1 set with center h_l and spread Δ_l $(\Delta_l \geqslant 0)$ (we assume that $h_l \geqslant \Delta_l$, so that $w_l \geqslant 0$ for $l = 1, \ldots, N$).

We make the following observations:

1. Since $Z_1, \ldots, Z_N, W_1, \ldots, W_N$ are interval type-1 sets, GC will also be an interval type-1 set, i.e., it will be a crisp set, $[y_l, y_r]$. So, to find GC we need to compute just the two end-points of the interval, y_l and y_r.

2. Let $y \triangleq \sum_{l=1}^{N} z_l w_l / \sum_{l=1}^{N} w_l$. Since $w_l \geqslant 0$ for all l, the partial derivative $\partial y / \partial z_k = w_k / \sum_l w_l \geqslant 0$; therefore, y always increases with increasing z_k, and, for any combination of $\{w_1, \ldots, w_N\}$ chosen so that $w_l \in W_l$, y is maximized when $z_l = c_l + s_l$ for $l = 1, \ldots, N$, and y is minimized when $z_l = c_l - s_l$ for $l = 1, \ldots, N$. y_r is, therefore, obtained by maximizing

$$\left[\sum_l w_l (c_l + s_l)\right] \Big/ \left[\sum_l w_l\right] \tag{22}$$

subject to the constraints $w_l \in W_l$ for $l = 1, \ldots, N$; and, y_l is obtained by minimizing

$$\left[\sum_l w_l (c_l - s_l)\right] \Big/ \left[\sum_l w_l\right] \tag{23}$$

again subject to the constraints $w_l \in W_l$ for $l = 1, \ldots, N$.

From these two observations, it is clear that in order to compute GC, we only need to consider the problems of maximizing and minimizing the generalized centroid

$$y(w_1, \ldots, w_N) = \frac{\sum_{l=1}^{N} z_l w_l}{\sum_{l=1}^{N} w_l} \tag{24}$$

subject to the constraints $w_l \in [h_l - \Delta_l, h_l + \Delta_l]$ for $l = 1, \ldots, N$, where, $h_l \geqslant \Delta_l$ for $l = 1, \ldots, N$. As explained in observation 2 above, we set $z_l = c_l + s_l$

$(l = 1, \ldots, N)$, when maximizing $y(w_1, \ldots, w_N)$, and $z_l = c_l - s_l$ $(l = 1, \ldots, N)$, when minimizing $y(w_1, \ldots, w_N)$.

Differentiating $y(w_1, \ldots, w_N)$ w.r.t. w_k gives us

$$\frac{\partial}{\partial w_k} y(w_1, \ldots, w_N) = \frac{\partial}{\partial w_k}\left[\frac{\sum_{l=1}^{N} z_l w_l}{\sum_{l=1}^{N} w_l}\right] = \frac{z_k - y(w_1, \ldots, w_N)}{\sum_{l=1}^{N} w_l}. \tag{25}$$

Since $\sum_{l=1}^{N} w_l > 0$, it is easy to see from (25) that

$$\frac{\partial}{\partial w_k} y(w_1, \ldots, w_N) \gtreqless 0 \quad \text{if } z_k \gtreqless y(w_1, \ldots, w_N). \tag{26}$$

Equating $\partial y / \partial w_k$ to zero does not give us any information about the value of w_k when $y(w_1, \ldots, w_N)$ is maximized or minimized, because

$$y(w_1, \ldots, w_N) = z_k \quad \Rightarrow \quad \frac{\sum_l z_l w_l}{\sum_l w_l} = z_k \quad \Rightarrow \quad \frac{\sum_{l \neq k} z_l w_l}{\sum_{l \neq k} w_l} = z_k, \tag{27}$$

where the summations are from 1 to N. Observe that w_k no longer appears on the RHS of (27). Eq. (26), however, gives us the direction in which w_k should be changed to increase or decrease $y(w_1, \ldots, w_N)$. Observe, from (26), that if $z_k > y(w_1, \ldots, w_N)$, $y(w_1, \ldots, w_N)$ increases as w_k increases; and, if $z_k < y(w_1, \ldots, w_N)$, $y(w_1, \ldots, w_N)$ increases as w_k decreases.

Recall that the maximum value that w_k can attain is $h_k + \Delta_k$ and the minimum value that it can attain is $h_k - \Delta_k$. The discussion in the previous paragraph, therefore, implies that $y(w_1, \ldots, w_N)$ attains its maximum value if: (1) $w_k = h_k + \Delta_k$ for those values of k for which $z_k > y(w_1, \ldots, w_N)$, and, (2) $w_k = h_k - \Delta_k$ for those values of k for which $z_k < y(w_1, \ldots, w_N)$. Similarly, $y(w_1, \ldots, w_N)$ attains its minimum value, if: (1) $w_k = h_k - \Delta_k$ for those values of k for which $z_k > y(w_1, \ldots, w_N)$, and, (2) $w_k = h_k + \Delta_k$ for those values of k for which $z_k < y(w_1, \ldots, w_N)$.

The maximum of $y(w_1, \ldots, w_N)$ can be obtained by the iterative procedure given next. We set $z_l = c_l + s_l$ $(l = 1, \ldots, N)$; and, without loss of generality, assume that the z_l's are arranged in ascending order, i.e., $z_1 \leqslant z_2 \leqslant \cdots \leqslant z_N$.

1. Set $w_l = h_l$ for $l = 1, \ldots, N$, and compute $y' = y(h_1, \ldots, h_N)$ using (24).
2. Find k $(1 \leqslant k \leqslant N - 1)$ such that $z_k \leqslant y' \leqslant z_{k+1}$.
3. Set $w_l = h_l - \Delta_l$ for $l \leqslant k$ and $w_l = h_l + \Delta_l$ for $l \geqslant k + 1$, and compute $y'' = y(h_1 - \Delta_1, \ldots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \ldots, h_N + \Delta_N)$ using (24). (Since the z_l's are arranged in ascending order, observe, from (26), the discussion after (27), and the fact $z_k \leqslant y' \leqslant z_{k+1}$, that $y'' \geqslant y'$, because we are decreasing w_l for $l \leqslant k$ and increasing w_l for $l \geqslant k + 1$.)
4. Check if $y'' = y'$. If yes, stop. y'' is the maximum value of $y(w_1, \ldots, w_N)$. If no, go to step 5.
5. Set y' equal to y''. Go to step 2.

It can easily be shown that *this iterative procedure converges in at most N iterations*, where one iteration consists of one pass through steps 2–5 (step 1 is an initialization). At any iteration, let k' be such that $z_{k'} \leqslant y'' \leqslant z_{k'+1}$. Since $y'' \geqslant y'$, $k' \geqslant k$. If k' is the same as k, the algorithm converges at the end of the next iteration. This can be explained as follows: $k' = k$ implies that both y' and y'' are in $[z_k, z_{k+1}]$. Note that, it is still possible that $y'' \neq y'$. If this happens, however, observe from step 3 that $y'' = y(h_1 - \Delta_1, \ldots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \ldots, h_N + \Delta_N)$; and, because of step 5, for the next iteration $y' = y(h_1 - \Delta_1, \ldots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \ldots, h_N + \Delta_N)$. The index k chosen for the next iteration will, therefore, be the same as the index k chosen for the current iteration ($k' = k$); consequently, at the end of the next iteration, we will have $y'' = y(h_1 - \Delta_1, \ldots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \ldots, h_N + \Delta_N) = y'$, and the algorithm will converge. Since k can take at most $N - 1$ values, the algorithm converges in at most N iterations.

The minimum of $y(w_1, \ldots, w_N)$ can be obtained by using a procedure similar to the one described above. Only two changes need to be made: (1) we must set $z_l = c_l - s_l$ for $l = 1, \ldots, N$; and, (2) in Step 3, we must set $w_l = h_l + \Delta_l$ for $l \leqslant k$ and $w_l = h_l - \Delta_l$ for $l \geqslant k + 1$, to compute the generalized centroid $y'' = y(h_1 + \Delta_1, \ldots, h_k + \Delta_k, h_{k+1} - \Delta_{k+1}, \ldots, h_N - \Delta_N)$.

Although we have developed an exact computation procedure for the generalized centroid, some iterations are still required, and no closed-form formula is available for computing it. To simplify the computation further, we develop an approximate result for the generalized centroid.

3.2. Approximate result for generalized centroid

When each Z_l and W_l is an interval type-1 set, we obtain the following corollary to Theorem 2.1.

Corollary 3.1. *When each Z_l and W_l in Theorem* 2.1 *is an interval type*-1 *set*, *GC in* (12) *is approximately an interval type*-1 *set with center $\mathscr{C}$ and spread $\mathscr{S}$*, *where $\mathscr{C}$ is given by* (15), *and*

$$\mathscr{S} = \frac{\sum_{l=1}^{N} [(h_l s_l) + |c_l - \mathscr{C}|\Delta_l]}{\sum_{l=1}^{N} h_l} \tag{28}$$

provided that condition (16) *is satisfied.*

Proof. Observe, from Theorem A.1 that when each Z_l and W_l in Theorem 2.1 is an interval type-1 set, $\underline{Z}_l$ is a zero mean interval type-1 set with domain $[-s_l, s_l]$ and $\underline{W}_l$ is a zero mean interval type-1 set with domain $[-\Delta_l, \Delta_l]$. So, applying Theorem A.1 to (12), the result in Corollary 3.1 follows. □

3.3. Centroid for an interval type-2 set

For an interval type-2 fuzzy set, (2) reduces to

$$C_{\tilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} 1 \bigg/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} = [c_l, c_r]. \tag{29}$$

Let $J_{x_i} \triangleq [L_i, R_i]$. To use the computational procedure described in Section 3.1, note that x_i plays the role of c_i; $s_i = 0$ for all i, since the x_i's are crisp; $(L_i + R_i)/2 = h_i$; and, $(R_i - L_i)/2 = \Delta_i$.

For the centroid of an interval type-2 set, $s_l = 0$ in (28), so Corollary 3.1 simplifies as follows.

Corollary 3.2. *Consider the centroid in* (29), *where each* J_{x_i} *is an interval type*-1 *set with support* $[h_i - \Delta_i, h_i + \Delta_i]$. *Then* $C_{\tilde{A}}$ *in* (29) *is approximately an interval type*-1 *set with center* $\mathscr{C}$ *and spread* $\mathscr{S}$, *where* $\mathscr{C}$ *is given by* (19), *and*

$$\mathscr{S} = \frac{\sum_{i=1}^{N} |c_i - \mathscr{C}| \Delta_l}{\sum_{i=1}^{N} h_i} \tag{30}$$

provided that condition (20) *is satisfied.*

The proof of this corollary is very similar to the proof of Corollary 3.1, and uses Corollary 2.1; hence, we leave it to the reader. Note that, even though we have an exact computational procedure for computing the centroid of an interval type-2 set, it requires up to M iterations of the 4-step iterative procedure, whereas Corollary 3.2 only requires two computations, $\mathscr{C}$ and $\mathscr{S}$.

For interval sets, condition (20) means the average spread of N interval sets is much less than the average of the centers of these N sets.

Example 3.1. In Fig. 3, we show an interval type-2 fuzzy set $\tilde{A}$. The x-domain $[0, 10]$ is uniformly sampled for $N = 101$ points (0:0.1:10), i.e., $x_i = 0.1(i - 1)$. The centroid of this type-2 fuzzy set using the exact computation procedure is $[4.0388 - 0.3765, 4.0388 + 0.3765] = [3.6623, 4.4153]$, and that using the approximate result in Corollary 3.2 is $[4.0125 - 0.3710, 4.0125 + 0.3710] = [3.6415, 4.3835]$. From (20), $(\sum_{i=1}^{N} \Delta_i / \sum_{i=1}^{N} h_i) = 0.1747 \ll 1$ where h_i and Δ_i are the mean and standard of each $\mu_{\tilde{A}}(x_i)$ $(i = 1, 2, \ldots, 101)$. Observe that the approximate results approximate the exact results very well when condition (20) is satisfied.

4. Centroid of a Gaussian type-2 fuzzy set

Unfortunately, except for the interval type-2 set, we do not have an exact procedure to compute the centroid of any other type-2 fuzzy set; hence, for all

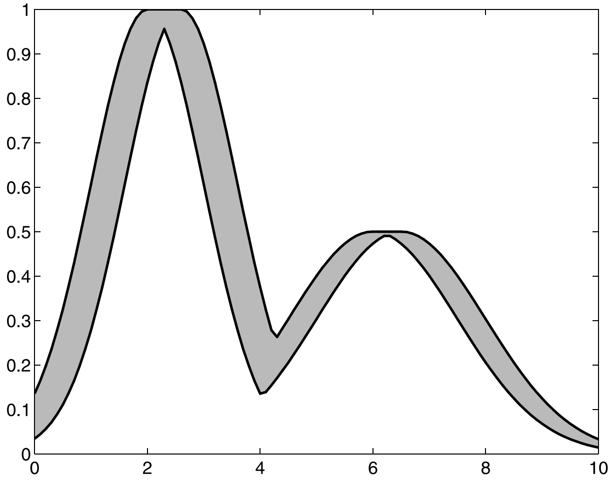


Fig. 3. The interval type-2 fuzzy set used in Example 3.1.

other type-2 sets we focus on an approximate result. An approximate result for the generalized centroid of a Gaussian type-2 fuzzy set is given in the following corollary to Theorem 2.1.

Corollary 4.1. *If each Z_l is a Gaussian type*-1 *set*, *with mean m_l and standard deviation σ_l, and if each W_l is also a Gaussian type*-1 *set with mean h_l and standard deviation Δ_l, then GC is approximately a Gaussian type*-1 *set*, *with mean $\mathscr{M}$ and standard deviation Σ, where*

$$\mathscr{M} = \frac{\sum_{l=1}^{N} h_l m_l}{\sum_{l=1}^{N} h_l} \tag{31}$$

and

$$\Sigma = \begin{cases} \dfrac{\sqrt{\sum_{l=1}^{N}\left[(h_l\sigma_l)^2 + [m_l - \mathscr{M}]^2 \Delta_l^2\right]}}{\sum_{l=1}^{N} h_l} & \text{if product t-norm is used}, \\ \dfrac{\sum_{l=1}^{N}[(h_l\sigma_l) + |m_l - \mathscr{M}|\Delta_l]}{\sum_{l=1}^{N} h_l} & \text{if minimum t-norm is used} \end{cases} \tag{32}$$

provided that

$$\frac{k\sum_{l=1}^{N}\Delta_l}{\sum_{l=1}^{N} h_l} \ll 1, \tag{33}$$

where k is the number of standard deviations of a Gaussian considered significant (*generally*, $k = 2$ *or* 3). *The Gaussian approximation improves as* $k(\sum_{l=1}^{N}\Delta_l/$

$\sum_{l=1}^{N} h_l$) *grows smaller, and the result is exact when* $\sum_{l=1}^{N} \Delta_l = 0$, *i.e., when* $\Delta_l = 0$ *for* $l = 1, \ldots, N$.

Proof. Observe, from Theorem A.2, (13) and (14), that $\underline{Z}_l$ is a Gaussian type-1 set with mean 0 and standard deviation σ_l and $\underline{W}_l$ is a Gaussian type-1 set with mean 0 and standard deviation ζ_l; therefore, applying Theorem A.2 now to (12), the result in Corollary 3.1 follows. □

A sufficient condition for satisfying (33) is that the Gaussian W_l are narrow, i.e., $k\Delta_l/h_l \ll 1$ for $l = 1, \ldots, N$. Observe, however, that there is no condition on the standard deviation of the Z_l; consequently, when all the W_l are crisp numbers, the corollary gives an exact result.

Recall that we will use only minimum *t*-norm for the centroid calculation of a type-2 set with a continuous domain (if the domain is discrete, however, product *t*-norm may be used). From Corollary 4.1, we get the following result for the centroid of a Gaussian type-2 set.

Corollary 4.2. *The centroid of a Gaussian type*-2 *set* $\tilde{A}$ *is approximately a Gaussian type*-1 *set with mean* $\mathscr{M}(C_{\tilde{A}})$, *where*

$$\mathscr{M}(C_{\tilde{A}}) = \frac{\sum_{i=1}^{N} x_i m(x_i)}{\sum_{i=1}^{N} m(x_i)}, \tag{34}$$

and standard deviation $\Sigma(C_{\tilde{A}})$ (*under minimum t-norm*), *where*

$$\Sigma(C_{\tilde{A}}) = \frac{\sum_{i=1}^{N} \left|x_i - \mathscr{M}(C_{\tilde{A}})\right| \sigma(x_i)}{\sum_{i=1}^{N} m(x_i)}, \tag{35}$$

as long as the standard deviations of the secondary memberships are small compared to their means, *i.e.*, *if*

$$k\frac{\sum_{i=1}^{N} \sigma(x_i)}{\sum_{i=1}^{N} m(x_i)} \ll 1 \tag{36}$$

is satisfied, *where k has the same meaning as in Corollary* 4.1.

Proof. Observe that: (1) the x_i's in (2), which are crisp numbers, correspond to the z_l's in (9); (2) the J_{x_i}'s in (2) correspond to the W_l's in (9); and, (3) the sum in (2) goes from 1 to N instead of from 1 to M. If we denote the mean and the standard deviation of $\mu_{\tilde{A}}(x_i)$ as $m(x_i)$ and $\sigma(x_i)$, respectively, then using Corollary 4.1, $C_{\tilde{A}}$ is approximately a Gaussian type-1 set with mean $\mathscr{M}(C_{\tilde{A}})$ (given in (34)) and standard deviation $\Sigma(C_{\tilde{A}})$ (given in (35)) provided (36) is satisfied, i.e., if standard deviations of the secondary membership functions are small compared to their means. □

Comment 1. See Fig. 1 for an example of a type-2 set, which can be made to satisfy condition (36) easily. In this set the standard deviation of every secondary MF is proportional to its mean. If we set the constant of proportionality to a small value, (36) can be satisfied.

Comment 2. Because the secondary MF at each x is a Gaussian type-1 set, the primary membership which has a secondary grade equal to unity is $m(x)$; and, since the principal membership function is the set of those primary memberships for which the secondary grades are equal to 1, *$m(x)$ for $x \in \tilde{A}$ is the same as the principal membership function of $\tilde{A}$*. Observe, therefore, from

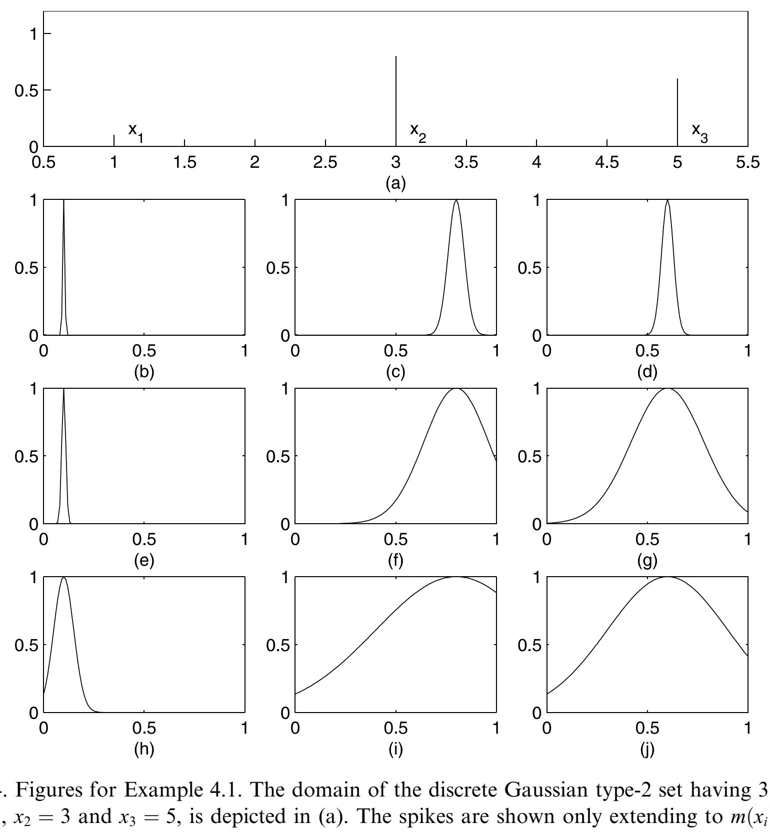


Fig. 4. Figures for Example 4.1. The domain of the discrete Gaussian type-2 set having 3 points, $x_1 = 1$, $x_2 = 3$ and $x_3 = 5$, is depicted in (a). The spikes are shown only extending to $m(x_i)$ for illustrative purposes.The secondary MFs of x_1, x_2 and x_3 for case 1 are depicted in (b), (c) and (d), respectively; the secondary MFs for case 2 are depicted in (e), (f) and (g); and, those for case 3 are depicted in (h), (i) and (j).

(34), that the mean of the approximate centroid, $\mathscr{M}(C_{\tilde{A}})$, corresponds to the centroid of the principal membership function of $\tilde{A}$.

Example 4.1. Now, we demonstrate the use of Corollary 4.2 with an example. Consider a Gaussian type-2 set with a discrete x-domain consisting of only 3 points, $x_1 = 1$, $x_2 = 3$ and $x_3 = 5$ (see Fig. 4(a)). Suppose that $m(x_1) = 0.1$, $m(x_2) = 0.8$ and $m(x_3) = 0.6$. We consider three cases:

1. If $\sigma(x_i) = 0.05m(x_i)$ for $i = 1, 2, 3$, the secondary MFs of x_1, x_2 and x_3 are shown in Figs. 4(b), (c) and (d), respectively; and, the true centroid and the approximation in Corollary 4.2 are as shown in Fig. 5(a). In this case, when $k = 2$, $k[\sum_i \sigma(x_i)]/[\sum_i m(x_i)] = 0.1$.
2. If $\sigma(x_1) = 0.3m(x_1)$, $\sigma(x_2) = 0.1m(x_2)$ and $\sigma(x_3) = 0.2m(x_3)$, the secondary MFs of x_1, x_2 and x_3 are shown in Figs. 4(e), (f) and (g), respectively; and, the true centroid and the approximation in Corollary 4.2 are as shown in Fig. 5(b). In this case, when $k = 2$, $k[\sum_i \sigma(x_i)]/[\sum_i m(x_i)] = 0.3066$.

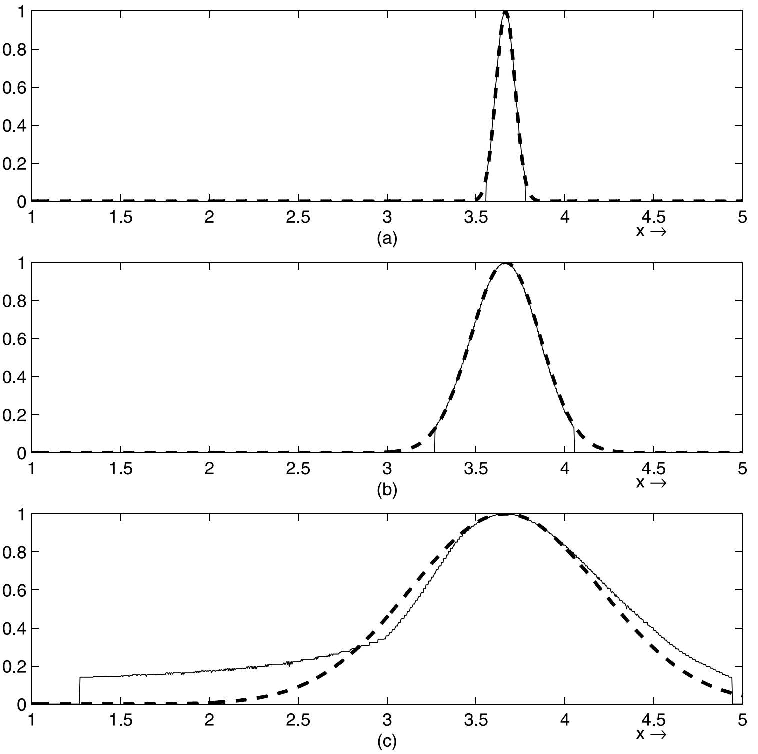


Fig. 5. True and approximate centroids of the Gaussian type-2 sets depicted in Fig. 4 for three choices of $\sigma(x_i)$ $(i = 1, 2, 3)$: (a) $\sigma(x_i) = 0.05m(x_i)$ for $i = 1, 2, 3$; (b) $\sigma(x_1) = 0.3m(x_1)$, $\sigma(x_2) = 0.1m(x_2)$ and $\sigma(x_3) = 0.2m(x_3)$; (c) $\sigma(x_i) = 0.5m(x_i)$ for $i = 1, 2, 3$.

3. If $\sigma(x_i) = 0.5m(x_i)$ for $i = 1, 2, 3$, the secondary MFs of x_1, x_2 and x_3 are shown in Figs. 4(h), (i) and (j), respectively; and, the true centroid and the approximation in Corollary 4.2 are as shown in Fig. 5(c). In this case, when $k = 2$, $k[\sum_i \sigma(x_i)]/[\sum_i m(x_i)] = 1$.

When computing the true centroids, only primary membership values between $m(x_i) \pm 2\sigma(x_i)$ were considered. Observe that, although the domain of the type-2 set is discrete, that of its centroid is continuous, because the secondary memberships of x_1, x_2 and x_3 have continuous domains.

Observe, also, that the approximation in the first two cases is much closer to the true centroid than that in the third case; however, although a smaller value for $[\sum_i \sigma(x_i)]/[\sum_i m(x_i)]$ will generally give a better approximation, it is not at all easy to predict how close the actual centroid of a given Gaussian type-2 set will be to its approximation. The same can be said about the approximation in Corollary 4.1.

5. Conclusions

In this paper, we introduced the concepts of a centroid and a generalized centroid of a type-2 fuzzy set and explained how to compute them. We have shown that the centroid operation for a general type-2 fuzzy set is computationally intensive, but that it can be computed efficiently for interval type-2 fuzzy sets.

Using interval type-2 sets implies that we are associating equal uncertainty with all the primary memberships (i.e., all the secondary grades are unity). This is a very reasonable starting point, since, in general, we do not have any additional information about the levels of uncertainty associated with different primary memberships, something that is required to use other kinds of type-2 sets. The use of Gaussian type-2 sets, for example, implies that the most certain primary memberships lie on the principal membership function, and the uncertainty decreases away from the principal membership function.

We have presented a very efficient computational procedure for the centroid of interval type-2 fuzzy sets. This procedure makes interval type-2 sets an attractive choice when using type-2 FLSs. We also presented an approximate result that can be used with type-2 fuzzy sets for which the type-2 uncertainty is quantifiably "small"; it leads to tremendous savings in computation. For an approximate result for the centroid of a triangle type-2 fuzzy set, see [9].

The centroid and generalized centroid operations are very important in the implementation of type-2 FLSs [11,14,16], because they are needed to implement type-reduction methods [12].

An open research issue is how to reduce the computational complexity of the centroid of arbitrary type-2 fuzzy sets.

Acknowledgements

This work was supported by the National Science Foundation under Grant MIP 9419386. The authors wish to thank Qilian Liang who assisted in the final preparation of this paper.

Appendix A. Operations on type-2 sets

Here, we present some background materials about type-2 fuzzy sets that are needed in this paper. Recall that the secondary MFs of type-2 sets are type-1 sets; therefore, in order to perform operations like union and intersection on type-2 sets, we need to be able to perform *t*-conorm and *t*-norm operations between type-1 sets. This is done using Zadeh's Extension Principle [5,11,24].

A binary operation $*$ between two crisp numbers can be extended to two type-1 sets $F = \int_v f(v)/v$ and $G = \int_w g(w)/w$ as

$$F * G = \int_v \int_w [f(v) \bigstar g(w)]/(v * w), \tag{A.1}$$

where $\bigstar$ denotes a chosen *t*-norm. We will generally use product or minimum *t*-norm. For example, the extension of the *t*-conorm (we generally use the maximum *t*-conorm) operation to type-1 sets is

$$F \sqcup G = \int_v \int_w [f(v) \bigstar g(w)]/(v \vee w). \tag{A.2}$$

This is called the *join* operation [19]. Similarly, the extension of the *t*-norm operation to type-1 sets, which is also known as the *meet* operation [19], is

$$F \sqcap G = \int_v \int_w [f(v) \bigstar g(w)]/(v \bigstar w). \tag{A.3}$$

We next show an example of the *meet* operation under product *t*-norm, when the sets involved are interval type-1 sets.

Example A.1. Let F and G be two interval type-1 sets with domains $[l_f, r_f]$ and $[l_g, r_g]$, respectively. Using (A.3), the *meet* between F and G, under product *t*-norm, can be obtained as

$$F \sqcap G = \int_{v \in F} \int_{w \in G} (1 \times 1)/(vw). \tag{A.4}$$

Observe, from (A.4), that: (a) each term in $F \sqcap G$ is equal to the product vw for some $v \in F$ and $w \in G$, the smallest term being $l_f l_g$ and the largest $r_f r_g$; and, (b) since both F and G have continuous domains, $F \sqcap G$ also has a continuous domain; consequently, $F \sqcap G$ is an interval type-1 set with domain $[l_f l_g, r_f r_g]$, i.e.,

$$F \sqcap G = \int_{u \in [l_f l_g, r_f r_g]} 1/u. \tag{A.5}$$

In a similar manner, the *meet*, $\sqcap_{i=1}^n F_i$, under product *t*-norm of *n* interval type-1 sets $F_1, \ldots, F_n$, having domains $[l_1, r_1], \ldots, [l_n, r_n]$, respectively, is an interval set with domain $[\prod_{i=1}^n l_i, \prod_{i=1}^n r_i]$. It is also easy to see that the *meet* of $F_1, \ldots, F_n$ under minimum *t*-norm is an interval set with domain $[\wedge_{i=1}^n l_i, \wedge_{i=1}^n r_i]$, where $\wedge$ denotes the minimum.

Kaufmann and Gupta [15] give a similar result while discussing properties of fuzzy numbers.

Algebraic operations between type-1 sets can also be defined using (A.1), e.g., the algebraic sum of two type-1 sets $F = \int_v f(v)/v$ and $G = \int_w g(w)/w$ is

$$F + G = \int_v \int_w [f(v) \bigstar g(w)]/(v + w). \tag{A.6}$$

We have the following results about algebraic sums of interval and Gaussian type-1 sets.

Theorem A.1 [11,13]. *Given n interval type-1 numbers $F_1, \ldots, F_n$, with means $m_1, m_2, \ldots, m_n$ and spreads $s_1, s_2, \ldots, s_n$, their affine combination $\sum_{i=1}^n \alpha_i F_i + \beta$, where α_i $(i = 1, \ldots, n)$ and β are crisp constants, is also an interval type-1 number with mean $\sum_{i=1}^n \alpha_i m_i + \beta$ and spread $\sum_{i=1}^n |\alpha_i| s_i$. This is true for any t-norm.*

Theorem A.2 [11,13]. *Given n type-1 Gaussian fuzzy numbers $F_1, \ldots, F_n$, with means $m_1, m_2, \ldots, m_n$ and standard deviations $\sigma_1, \sigma_2, \ldots, \sigma_n$, their affine combination $\sum_{i=1}^n \alpha_i F_i + \beta$, where α_i $(i = 1, \ldots, n)$ and β are crisp constants, is also a Gaussian fuzzy number with mean $\sum_{i=1}^n \alpha_i m_i + \beta$, and standard deviation Σ', where*

$$\Sigma' = \begin{cases} \sqrt{\sum_{i=1}^n \alpha_i^2 \sigma_i^2} & \text{if product } t\text{-norm is used}, \\ \sum_{i=1}^n |\alpha_i| \sigma_i & \text{if minimum } t\text{-norm is used}. \end{cases} \tag{A.7}$$

Using the Extension Principle, an *n*-ary operation $f(\theta_1, \ldots, \theta_n)$ on crisp numbers can be extended to *n* type-1 fuzzy sets $F_1, \ldots, F_n$ as [11]

$$f(F_1, \ldots, F_n) = \int_{\theta_1} \cdots \int_{\theta_n} \mu_{F_1}(\theta_1) \bigstar \cdots \bigstar \mu_{F_n}(\theta_n) / f(\theta_1, \ldots, \theta_n), \tag{A.8}$$

where all the integrals denote logical union, and $\theta_i \in F_i$ for $i = 1, \ldots, n$.

Appendix B. Proof of theorem 2.1

Here, we linearize (9), and express it as an affine combination of type-1 sets. If we let $\gamma_l = z_l - c_l$ and $\Delta_l = w_l - h_l$ for $l = 1, \ldots, N$, (9) becomes

$$Y = \int_{\gamma_1} \cdots \int_{\gamma_N} \int_{\delta_1} \cdots \int_{\delta_N} \mathscr{T}_{l=1}^{N} \mu_{Z_l}(c_l + \gamma_l) \bigstar \mathscr{T}_{l=1}^{N} \mu_{W_l}(h_l + \delta_l) \Bigg/ \frac{\sum_{l=1}^{N}(h_l + \delta_l)(c_l + \gamma_l)}{\sum_{l=1}^{N}(h_l + \delta_l)}, \tag{A.9}$$

where each δ_l takes values in $[-\Delta_l, \Delta_l]$ and each γ_l takes values in $[-s_l, s_l]$.

The term to the right of the slash in (A.9) can be rewritten as

$$\frac{\sum_l w_l z_l}{\sum_l w_l} = \frac{\sum_l h_l c_l + \sum_l h_l \gamma_l + \sum_l \delta_l c_l + \sum_l \delta_l \gamma_l}{\sum_l h_l + \sum_l \delta_l}, \tag{A.10}$$

where the limits on each sum are from 1 to N.

In what follows, we express the term on the RHS of (A.10) as an affine combination of γ_l and δ_l to find an (approximate) expression for Y in (A.9). We expand the denominator of (A.10) by first rewriting it as

$$\frac{1}{\sum_l h_l + \sum_l \delta_l} = \frac{1}{\sum_l h_l}\left(\frac{1}{1 + (\sum_l \delta_l / \sum_l h_l)}\right) \approx \frac{1}{\sum_l h_l}\left(1 - \frac{\sum_l \delta_l}{\sum_l h_l}\right) \tag{A.11}$$

as long as

$$\frac{|\sum_l \delta_l|}{\sum_l h_l} \ll 1, \tag{A.12}$$

which is equivalent to assuming that (since δ_l varies between $-\Delta_l$ and Δ_l)

$$\frac{\sum_l \Delta_l}{\sum_l h_l} \ll 1. \tag{A.13}$$

Using (A.11) in (A.10), we get

$$\frac{\sum_l w_l z_l}{\sum_l w_l} \approx \frac{\sum_l h_l c_l + \sum_l h_l \gamma_l + \sum_l \delta_l c_l + \sum_l \delta_l \gamma_l}{\sum_l h_l}\left(1 - \frac{\sum_l \delta_l}{\sum_l h_l}\right). \tag{A.14}$$

Ignoring all the terms containing powers of $\sum_l \delta_l / \sum_l h_l$ higher than 1, we get

$$\frac{\sum_l w_l z_l}{\sum_l w_l} \approx \frac{\sum_l h_l c_l}{\sum_l h_l} - \frac{\sum_l \delta_l}{\sum_l h_l}\left(\frac{\sum_l h_l c_l}{\sum_l h_l}\right) + \frac{\sum_l h_l \gamma_l}{\sum_l h_l} - \frac{\sum_l \delta_l}{\sum_l h_l}\left(\frac{\sum_l h_l \gamma_l}{\sum_l h_l}\right) + \frac{\sum_l \delta_l c_l}{\sum_l h_l} + \frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}. \tag{A.15}$$

Let

$$\mathscr{C} = \frac{\sum_l h_l c_l}{\sum_l h_l}; \tag{A.16}$$

then (A.15) can be rewritten as

$$\frac{\sum_l w_l z_l}{\sum_l w_l} \approx \mathscr{C} - \mathscr{C}\frac{\sum_l \delta_l}{\sum_l h_l} + \frac{\sum_l \delta_l c_l}{\sum_l h_l} + \frac{\sum_l h_l \gamma_l}{\sum_l h_l} - \frac{\sum_l \delta_l}{\sum_l h_l}\left(\frac{\sum_l h_l \gamma_l}{\sum_l h_l}\right) + \frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}. \tag{A.17}$$

Next we focus on the last two terms in (A.17). Clearly,

$$\left|\frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}\right| \leqslant \left|\max_l \gamma_l \left(\frac{\sum_l \delta_l}{\sum_l h_l}\right)\right|. \tag{A.18}$$

Since γ_l takes values between $\pm s_l$, (A.18) is equivalent to

$$\left|\frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}\right| \leqslant \max_l s_l \left|\frac{\sum_l \delta_l}{\sum_l h_l}\right|. \tag{A.19}$$

Similarly,

$$\left|\frac{\sum_l h_l \gamma_l}{\sum_l h_l}\right| \leqslant \max_l s_l. \tag{A.20}$$

Consequently,

$$\left| - \frac{\sum_l \delta_l}{\sum_l h_l}\left(\frac{\sum_l h_l \gamma_l}{\sum_l h_l}\right) + \frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}\right| \leqslant \left|\frac{\sum_l \delta_l}{\sum_l h_l}\left(\frac{\sum_l h_l \gamma_l}{\sum_l h_l}\right)\right| + \left|\frac{\sum_l \delta_l \gamma_l}{\sum_l h_l}\right| \leqslant 2\max_l s_l \left|\frac{\sum_l \delta_l}{\sum_l h_l}\right|. \tag{A.21}$$

Observe, from (A.17) and (A.21), that if condition (A.12) is satisfied, we can ignore the last two terms on the RHS of (A.17) in comparison with $\sum_l h_l \gamma_l / \sum_l h_l$, which, according to (A.20), takes values in $[-\max_l s_l, \max_l s_l]$. Doing this gives us

$$\frac{\sum_l w_l z_l}{\sum_l w_l} \approx \sum_{l=1}^{N}\left[\gamma_l\left(\frac{h_l}{\sum_l h_l}\right) + \delta_l\left(\frac{c_l - \mathscr{C}}{\sum_l h_l}\right)\right] + \mathscr{C}. \tag{A.22}$$

Using (A.22), (A.9) can be rewritten as

$$Y \approx \int_{\gamma_1} \cdots \int_{\gamma_N} \int_{\delta_1} \cdots \int_{\delta_N} \mathscr{T}_{l=1}^{N} \mu_{Z_l}(c_l + \gamma_l) \bigstar \mathscr{T}_{l=1}^{N} \mu_{W_l}(h_l + \delta_l) \Bigg/ \sum_{l=1}^{N}\left[\gamma_l\left(\frac{h_l}{\sum_l h_l}\right) + \delta_l\left(\frac{c_l - \mathscr{C}}{\sum_l h_l}\right)\right] + \mathscr{C}. \tag{A.23}$$

Recall that $\gamma_l = z_l - c_l$ and $\delta_l = w_l - h_l$, and let

$$\underline{Z}_l = Z_l - c_l \quad \text{for } l = 1, \ldots, N \tag{A.24}$$

and

$$\underline{W}_l = W_l - h_l \quad \text{for } l = 1, \ldots, N; \tag{A.25}$$

so that each $\underline{Z}_l$ is a type-1 fuzzy number with support $[-s_l, s_l]$ and each $\underline{W}_l$ is a type-1 fuzzy number with support $[-\Delta_l, \Delta_l]$. Observe, from (A.24) that

$$\begin{aligned} \underline{Z}_l &= \int_{z_l} \mu_{Z_l}(z_l)/(z_l - c_l) \\ &= \int_{z_l} \mu_{Z_l}(\gamma_l + c_l)/\gamma_l \\ &\Rightarrow \mu_{Z_l}(c_l + \gamma_l) = \mu_{\underline{Z}_1}(\gamma_l). \end{aligned} \tag{A.26}$$

Similarly, $\mu_{W_l}(h_l + \delta_l) = \mu_{\underline{W}_1}(\delta_l)$. This means that (A.23) can be rewritten as

$$\begin{aligned} Y \approx & \int_{\gamma_1} \cdots \int_{\gamma_N} \int_{\delta_1} \cdots \int_{\delta_N} \mathcal{T}_{l=1}^{N} \mu_{\underline{Z}_1}(\gamma_l) \bigstar \mathcal{T}_{l=1}^{N} \mu_{\underline{W}_1}(\delta_l) \\ & \bigg/ \sum_{l=1}^{N} \left[\gamma_l \left(\frac{h_l}{\sum_l h_l} \right) + \delta_l \left(\frac{c_l - \mathscr{C}}{\sum_l h_l} \right) \right] + \mathscr{C}. \end{aligned} \tag{A.27}$$

The result in Theorem 2.1 follows by observing that the RHS of (A.27) is equal to

$$\sum_{l=1}^{N} \left[\underline{Z}_l \left(\frac{h_l}{\sum_l h_l} \right) + \underline{W}_l \left(\frac{c_l - \mathscr{C}}{\sum_l h_l} \right) \right] + \mathscr{C}$$

(see (A.8)).

Comment. When all $\Delta_l = 0$, there is only one source of fuzziness in $\sum_l w_l z_l / \sum_l w_l$, namely, the Z_l's. In this case, (9) reduces to

$$Y(Z_1, \ldots, Z_N, h_1, \ldots, h_N) = \int_{z_1} \cdots \int_{z_N} \mathcal{T}_{l=1}^{N} \mu_{Z_l}(z_l) \bigg/ \frac{\sum_{l=1}^{N} h_l z_l}{\sum_{l=1}^{N} h_l}. \tag{A.28}$$

Again, letting $\gamma_l = z_l - c_l$ for $l = 1, \ldots, N$, we have

$$\begin{aligned} & Y(Z_1, \ldots, Z_N, h_1, \ldots, h_N) \\ &\quad = \int_{\gamma_1} \cdots \int_{\gamma_N} \mathcal{T}_{l=1}^{N} \mu_{Z_l}(c_l + \gamma_l) \bigg/ \frac{\sum_{l=1}^{N} h_l (c_l + \gamma_l)}{\sum_{l=1}^{N} h_l} \\ &\quad = \int_{\gamma_1} \cdots \int_{\gamma_N} \mathcal{T}_{l=1}^{N} \mu_{Z_l}(c_l + \gamma_l) \bigg/ \left[\frac{\sum_{l=1}^{N} h_l c_l}{\sum_{l=1}^{N} h_l} + \sum_{l=1}^{N} \gamma_l \left(\frac{h_l}{\sum_{l=1}^{N} h_l} \right) \right] \end{aligned}$$

$$= \int_{\gamma_1} \cdots \int_{\gamma_N} \mathscr{T}_{l=1}^{N} \mu_{Z_l}(c_l + \gamma_l) \Bigg/ \left[\mathscr{C} + \sum_{l=1}^{N} \gamma_l \left(\frac{h_l}{\sum_{l=1}^{N} h_l} \right) \right]$$

$$= \int_{\gamma_1} \cdots \int_{\gamma_N} \mathscr{T}_{l=1}^{N} \mu_{\underline{Z}_l}(\gamma_l) \Bigg/ \left[\mathscr{C} + \sum_{l=1}^{N} \gamma_l \left(\frac{h_l}{\sum_{l=1}^{N} h_l} \right) \right]$$

$$= \sum_{l=1}^{N} \underline{Z}_l \left(\frac{h_l}{\sum_{l=1}^{N} h_l} \right) + \mathscr{C}, \tag{A.29}$$

where $\underline{Z}_l$ and $\mathscr{C}$ are as defined in (A.24) and (A.16), respectively. Observe that, in this case, the result in Theorem 2.1 is exact.

References

[1] J.L. Chaneau, M. Gunaratne, A.G. Altschaeffl, An application of type-2 sets to decision making in engineering, in: J.C. Bezdek (Ed.), Analysis of Fuzzy Information – vol. II: Artificial Intelligence and Decision Systems, CRC Press, Boca Raton, FL, 1987.

[2] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy Control, second ed., Springer, Berlin, 1996.

[3] D. Dubois, H. Prade, Operations on fuzzy numbers, Int. J. Systems Sci. 9 (6) (1978) 613–626.

[4] D. Dubois, H. Prade, Operations in a fuzzy-valued logic, Inform. Control 43 (1979) 224–240.

[5] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, NY, 1980.

[6] E. Hisdal, The IF THEN ELSE statement and interval-valued fuzzy sets of higher type, Int. J. Man–Machine Studies 15 (1981) 385–455.

[7] R.I. John, Type 2 fuzzy sets: an appraisal of theory and applications, Int. J. Uncertainty, Fuzziness Knowledge-Based Systems 6 (6) (1998) 563–576.

[8] R.I. John, P.R. Innocent, M.R. Barnes, Type 2 fuzzy sets and neuro-fuzzy clustering of radiographic tibia images, in: 1998 IEEE International Conference on Fuzzy Systems, Anchorage, AK, USA, May 1998, pp. 1373–1376.

[9] N.N. Karnik, Type-2 fuzzy logic systems, Ph.D. Dissertation, University of Southern California, Los Angeles, CA, 1998.

[10] N.N. Karnik, J.M. Mendel, Introduction to Type-2 Fuzzy Logic Systems, presented at the 1998 IEEE FUZZ Conference, Anchorage, AK, May.

[11] N.N. Karnik, J.M. Mendel, An introduction to type-2 fuzzy logic systems, October 1998, USC Report, http://sipi.usc.edu/~mendel/report.

[12] N.N. Karnik, J.M. Mendel, Type-2 Fuzzy Logic Systems: Type-Reduction, presented at the 1998 IEEE SMC Conference, San Diego, CA, October.

[13] N.N. Karnik, J.M. Mendel, Operations on type-2 fuzzy set, Fuzzy Sets Systems (2000).

[14] N.N. Karnik, J.M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Trans. Fuzzy Systems 7 (6) (1999) 643–658.

[15] A. Kaufmann, M.M. Gupta, Introduction to Fuzzy Arithmetic: Theory and Applications, Van Nostrand Reinhold, NY, 1991.

[16] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Systems 8 (5) (2000).

[17] J.M. Mendel, Fuzzy Logic Systems for Engineering: A Tutorial, Proc. IEEE 83 (3) (1995) 345–377.

[18] J.M. Mendel, Computing with words when words can mean different things to different people, presented at Int'l. ICSC Congress on Computational Intelligence: Methods & Applications, Third Annual Symposium on Fuzzy Logic and Applications, Rochester, NY, June 22–25, 1999.
[19] M. Mizumoto, K. Tanaka, Some properties of fuzzy sets of type-2, Inform. Control 31 (1976) 312–340.
[20] M. Mizumoto, K. Tanaka, Fuzzy sets of type 2 under algebraic product and algebraic sum, Fuzzy Sets Systems 5 (1981) 277–290.
[21] J. Nieminen, On the algebraic structure of fuzzy sets of type-2, Kybernetica 13 (4) (1977).
[22] M. Wagenknecht, K. Hartmann, Application of fuzzy sets of type 2 to the solution of fuzzy equation systems, Fuzzy Sets Systems 25 (1988) 183–190.
[23] R.R. Yager, Fuzzy subsets of type II in decisions, J. Cybernet. 10 (1980) 137–159.
[24] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – 1, Inform. Sci. 8 (1975) 199–249.