USING TRIANGULAR NORMS IN A SEGMENT-BASED ASR SYSTEM

Gábor GOSZTOLYA[1], **András KOCSOR**[2]

[1]*Department of Informatics,*
University of Szeged, Szeged, Hungary
E-mail: ggabor@inf.u-szeged.hu

[2]*MTA-SZTE Research Group on Artificial Intelligence*
of the Hungarian Academy of Sciences and University of Szeged,
Szeged, Hungary
E-mail: kocsor@inf.u-szeged.hu

Abstract

In speech recognition there is a constant need to improve the recognition accuracy. There are many ways of doing this, and among them, one might be to increase phoneme recognition rates. Here, however, we decided to investigate the usefulness of triangular norms taken from the field of fuzzy logic. The triangular norms are tools for aggregating one probability factor from multiple probability values, thus they seem to be ideal for constructing hypothesis probabilities. The tests showed that this idea is fruitful: the recognition error rate was reduced by 16% both in isolated word and sentence recognition, without causing any increase in the running times.

Keywords: speech recognition, probability calculation, fuzzy logic, triangular norms

1 Introduction

In the problem of Automatic Speech Recognition (ASR) an important task is to improve the recognition accuracy, but in most cases only tools with small or no further computational needs are permitted. This paper deals with one such idea, that of applying *triangular norms* (t-norms for short) [1, 2] from fuzzy logic to better the probability calculation part.

There are many fields where fuzzy methodology has been applied such as in the field of image processing [3]. We previously used a family of aggregation operators in speech recognition to better the recognition percentage [4]. Now we will test the triangular norms for this task because they provide the largest range of operators that can be used in our case. We experiment with several t-norms and try to determine the best parameters for our needs.

The structure of this paper is as follows. First we define the speech recognition problem in a segment-based environment. Next, we introduce triangular norms and show some of their properties. Then we describe the test environment and the test databases, and analyze the test results. Lastly we draw some conclusions.

2 The Speech Recognition Problem

In speech recognition problems we have a speech signal represented by a series of observations $A = a_1 a_2 \dots a_t$, and a set of possible phoneme sequences (words or word sequences) that will be denoted by W. Our task is to find the word $\hat{w} \in W$ such as

$$\hat{w} = arg \max_{w \in W} P(w|A), \tag{1}$$

which, using Bayes' theorem, is equivalent to

$$\hat{w} = arg \max_{w \in W} \frac{P(A|w) \cdot P(w)}{P(A)}. \tag{2}$$

Further, noting the fact that $P(A)$ is the same for all $w \in W$, we have that

$$\hat{w} = arg \max_{w \in W} P(A|w) P(w). \tag{3}$$

Speech recognition models can be divided into two types – the discriminative and generative ones –, depending on whether they use Eq. (1) or Eq. (3). Throughout this paper we will apply the customary, generative approach [5].

Now let us define w as $o_1 \dots o_n$, where o_j is the jth phoneme of a word (or word sequence) w. Furthermore, let $A_1, \dots, A_n$ be non-overlapping segments of the observation series $A = a_1 \dots a_t$, where $A_j = a_{t_{j-1}} \dots a_{t_j}$, $j \in \{1, \dots, n\}$. An A_j segment is defined by its start and end times and will be denoted by $[t_{j-1}, t_j]$. For a segmentation $A = A_1, \dots, A_n$ we put the values of the time indices belonging to each segment into a vector $S_n = [t_0, \dots, t_n]$ $(1 = t_0 < \dots < t_n = t)$. We make the conventional assumption that the phonemes in a word are independent, thus $P(A|w)$ can be obtained from $P(A_1|o_1), \dots, P(A_n|o_n)$ in some way. Usually we simply multiply these values, but now we will try out some other methods here as well.

The $P(A_j|o_j)$ (or $P([t_{j-1}, t_j], o_j)$) values in effect measure how well the A_j segment represents the o_j phoneme. To calculate the values, many ways can be chosen, but in this paper we opted for the *segment-based approach*. In it this probability is calculated by considering longer, interval-based features which describe the whole A_j segment. In our case it meant that an Artificial Neural Network (or *ANN*) [6] had to be trained on these features, and then its output was normalized to the length of the given segment. We should say here that, of course, any machine-learning algorithm could be used instead. We should also remark that the tests made in this paper can be easily transformed to a frame-based framework.

Now we will define the set of possible hypotheses. The hypothesis space is a Cartesian product space where the first dimension is a set of word prefixes, while the second is a set of segmentations. Given a set of words (or word sequences) W, we use $Pref_k(W)$ to denote the k-long prefixes of all the words in W having at least k phonemes. Let $S^k = \{[t_0, t_1, \ldots, t_k] : 1 = t_0 < t_1 < \cdots < t_k \leq t\}$ be the set of sub-segmentations made of k segments over the observation series $a_1 \ldots a_t$. The hypotheses will be the elements of $H = \bigcup_{k=0}^{\infty}(Pref_k(W) \times S^k)$. We will denote the root of the tree – the initial hypothesis – by $h_0 = (\emptyset, [t_0])$, $h_0 \in H$. $Pref_1(W) \times S^1$ will contain the first-level nodes. For a $(o_1 \ldots o_j, [t_0, \ldots, t_j])$ leaf we link all $(o_1 \ldots o_j o_{j+1}, [t_0, \ldots, t_j, t_{j+1}]) \in Pref_{j+1}(W) \times S^{j+1}$ nodes.

Now we need to assign probabilities to the nodes of this search tree. For this we will use a function $T : [0, 1]^2 \to [0, 1]$. Usually T is the simple multiplication operator (i.e. it supplies the product of its two parameters), but here we will also test various other operators. Now let $P(h_0) = 1$. After, let

$$P(o_1 \ldots o_{j+1}, [t_0, \ldots, t_{j+1}]) = T(P(o_1 \ldots o_j, [t_0, \ldots, t_j]), P([t_j, t_{j+1}], o_{j+1})). \quad (4)$$

We look for a leaf with the highest probability. Of course, any search method can be applied, but here we chose the multi-stack decoding algorithm [7].

In the following we will test different families of triangular norms for this T.

3 Triangular norms

The *triangular norms* are standard aggregation operators of fuzzy sets [1, 2]. We would like to apply them to speech recognition in hypothesis probability calculations (i.e. use each in Eq. (4) as T), but first we need to define some basic terms.

Definition 1 *A triangular norm (t-norm) is a binary operation T on the interval $[0,1]$, i.e. a function $T : [0,1]^2 \to [0,1]$, such that for all $x, y, z \in [0,1]$ the following four axioms are satisfied:*

(T1)	$T(x, y) = T(y, x)$.	(commutativity)
(T2)	$T(x, T(y, z)) = T(T(x, y), z)$.	(associativity)
(T3)	$T(x, y) \leq T(x, z)$ whenever $y \leq z$.	(monotonicity)
(T4)	$T(x, 1) = x$	(boundary condition)

Such a t-norm is the product operator (T_P). Now we make some more definitions:

Definition 2 *A t-norm T is continuous if and only if it is continuous in each component, i.e. if for all $x_0, y_0 \in [0,1]$ both the vertical section $T(x_0, \ldots)$ and the horizontal section $T(\ldots, y_0)$ are continuous functions in one variable.*

Definition 3 *A t-norm T is Archimedean if for all $x, y \in]0,1[$ there is an $n \in \mathbb{N}$ such that $T(T(\ldots T(x_1, x_2), \ldots, x_{n-1}), x_n) < y$, where $x_1 = \ldots = x_n = x$.*

These two properties seem to be important for a T function in Eq. (4). We should expect that a slightly different phoneme probability affects the hypothesis probability by only a little bit too. In other words, there are no sudden gaps between these kind of hypotheses. This way, T should be continuous. On the other hand, if T satisfies the Archimedean property, then the longer a word is, the closer the result (and hence the probability of the word pronounced) is to zero, which is also desirable. Another reason for anticipating these properties is that our default operator, T_P is also both continuous and Archimedean. Thus in the following we will use triangular norms which fulfil both these requirements.

3.1 Common triangular norms

Now we introduce the triangular norms that are common in the literature, and which have been tested here. One of the basic t-norms is the Lukasiewicz t-norm, which is

$$T_L(x, y) = max(x + y - 1, 0). \tag{5}$$

However, there exist t-norm families, that is triangular norms with a single parameter. Following Klement, Mesiar and Pap [2], we list the common ones.

Note that most of these t-norm families can have other λ values correspoding to some basic t-norms, but here we omit the full listing due to lack of space. Schweizer-Sklar t-norms ($\lambda \in \mathbb{R}, \lambda \neq 0$):

$$T_\lambda^{SS}(x,y) = (\max((x^\lambda + y^\lambda - 1), 0))^{1/\lambda} \tag{6}$$

Hamacher t-norms ($\lambda > 0$):

$$T_\lambda^{H}(x,y) = \frac{x \cdot y}{\lambda + (1-\lambda)(x + y - x \cdot y)} \tag{7}$$

Yager t-norms ($\lambda > 0$):

$$T_\lambda^{Y}(x,y) = \max(1 - ((1-x)^\lambda + (1-y)^\lambda)^{1/\lambda}, 0) \tag{8}$$

Dombi t-norms ($\lambda > 0$):

$$T_\lambda^{D}(x,y) = \frac{1}{1 + ((\frac{1-x}{x})^\lambda + (\frac{1-y}{y})^\lambda)^{1/\lambda}} \tag{9}$$

Sugeno-Weber t-norms ($\lambda > -1$):

$$T_\lambda^{SW}(x,y) = \max(\frac{x + y - 1 + \lambda xy}{1 + \lambda}, 0) \tag{10}$$

Aczél-Alsina t-norms ($\lambda > 0$):

$$T_\lambda^{AA}(x,y) = e^{-((-\log x)^\lambda + (-\log y)^\lambda)^{1/\lambda}} \tag{11}$$

Mayor-Torrens t-norms ($\lambda > 0$):

$$T_\lambda^{MT}(x,y) = \begin{cases} \max(x + y - \lambda, 0) & \text{if } \lambda \in]0,1] \text{ and } (x,y) \in [0,\lambda]^2, \\ \min(x,y) & \text{otherwise.} \end{cases} \tag{12}$$

Table 1 describes the intervals in which these t-norm families satisfy the continuity or Archimedean property, or both. The reader should note here that T^{MT} with $\lambda = 1$ is the same as T_L.

4 Experimental results

We performed two experiments in order to test the above mentioned triangular norms. In the first one we sought to check their capabilities during the recognition of isolated words. We used a corpus of 500 children uttering 60 words each, making a total of 30,000 utterances of 2,000 different Hungarian

	T_λ^{SS}	T_λ^{H}	T_λ^{Y}	T_λ^{D}	T_λ^{SW}	T_λ^{AA}	T_λ^{MT}
cont.	$\lambda \in \mathbb{R}\backslash 0$	$\lambda > 0$	$\lambda > 0$	$\lambda > 0$	$\lambda > -1$	$\lambda > 0$	$0 \leq \lambda \leq 1$
Arch.	$\lambda \in \mathbb{R}\backslash 0$	$\lambda > 0$	$\lambda > 0$	$\lambda \geq 0$	$\lambda \geq -1$	$\lambda \geq 0$	$\lambda = 1$
both	$\lambda \in \mathbb{R}\backslash 0$	$\lambda > 0$	$\lambda > 0$	$\lambda > 0$	$\lambda > -1$	$\lambda > 0$	$\lambda = 1$

Table 1. The intervals where the triangular norm families we deal with are continuous, Archimedean, and both continuous and Archimedean.

words. 24,000 utterances were used for training, while the remaining 6,000 words were used for testing purposes. Many of the young speakers had just learned to read and some of them had difficulties with pronunciation, which led to a diverse database. Moreover, many used words were very similar to each other, which made the recognition task difficult. Testing was done in the framework of the OASIS Speech Laboratory [8]. The diversity of the database led to a basic word recognition rate of 92.17% in the OASIS system, while the *HTK* system [10] we used as a reference achieved a score of 92.60%.

In the other test we sought to test the performance of these triangular norms in sentence recognition. For it we trained the phoneme recognition ANN on a large, general database. 332 people of various ages spoke 12 sentences and 12 words each, which were recorded on different computers and sound cards via different microphones. [9] This way we fulfilled our goal of training a speaker-independent phoneme classifier. We should also mention here that the resulting neural networks can be used in any context.

In the next step we combined this phoneme classification method with a simple language model. The sentences spoken were restricted to those of medical reports. 150 randomly selected sentences were recorded, which were then used as the test database. The language model was a simple word 2-gram; i.e. the probability of the next word depends only of the last word spoken, and it is calculated by a statistical investigation of texts in a similar field. Thus we carried out this investigation on all the available almost 9,000 reports, which contained 2,500 different words in 95,000 sentences.

The performance of a speech recognition system can be easily measured on word recognition tasks: we only have to compute the ratio of the correctly recognized and the tested words. However, we cannot use this method on sentence recognition as we do now because only one badly identified word would ruin the whole sentence. We cannot compare the two sentences word by word either, because one incorrectly inserted or skipped word would also corrupt the calculated performance ratio. For this reason, usually the edit distance of the two sentences (the original and the resulted) is calculated; that is, we construct

		T_λ^{SS}	T_λ^{H}	T_λ^{Y}	T_λ^{D}	T_λ^{AA}
Word	interval	$[-3, 1]$	$]0, 15]$	$[180, 320]$	$]0, 5]$	$]0, 20]$
	step	0.01	0.02	0.5	0.01	0.05
Sentence	interval	$[-2, 0.5]$	$]0, 2]$	—	$[0.1, 0.5]$	$[0.9, 1.4]$
	step	0.01	0.01	—	0.02	0.02

Table 2. The tested interval and the step sizes of the triangular norm families.

the resulting sentence from the original by using the following operations: inserting and deleting words, and replacing one word with another one. These operations have some cost (in our case 3, 3 and 4, respectively), and then we choose an operation set with the lowest cost. Then we can calculate the following measures:

$$Correctness = \frac{N - S - D}{N} \tag{13}$$

and

$$Accuracy = \frac{N - S - D - I}{N}, \tag{14}$$

where N is the total number of words in all the original sentences, S is the number of substitutions, D is the number of deletions and I is the number of insertions. Under these circumstances, the baseline was correctness = 92.03% and accuracy = 91.69%, which is probably due to the large number of words and the simple nature of the language model.

The testing was carried out in a similar way for both test environments. First we tested the Lukasiewicz t-norm (T_L). The next step was to test the t-norm families that had a λ parameter. For each t-norm family the tested interval was first determined by some rough tests in the region where it is both continuous and Archimedean, then a suitable step size was assigned to it for which we increment the λ value inside this interval. Unfortunately the Sugeno-Weber family did not produce any acceptable results at this stage, so we excluded it from further tests. Neither did the Yager family on the sentence recognition task, so we omitted these as well. Table 2 shows the tested intervals and the step sizes for both test series.

Next the testing was done on these the intervals and with these step sizes. The results can be seen on Figure 1 and Figure 2, while the best performances of the t-norms tested are listed in Table 3 and Table 4. Note that $T_L = T_1^{MT}$, so the best performance of T^{MT} is also below 1% by all measures. The reason for this is probably that the Lukasiewicz t-norm is a rather drastic one for probabilities appearing in a speech recognition environment: to get a result

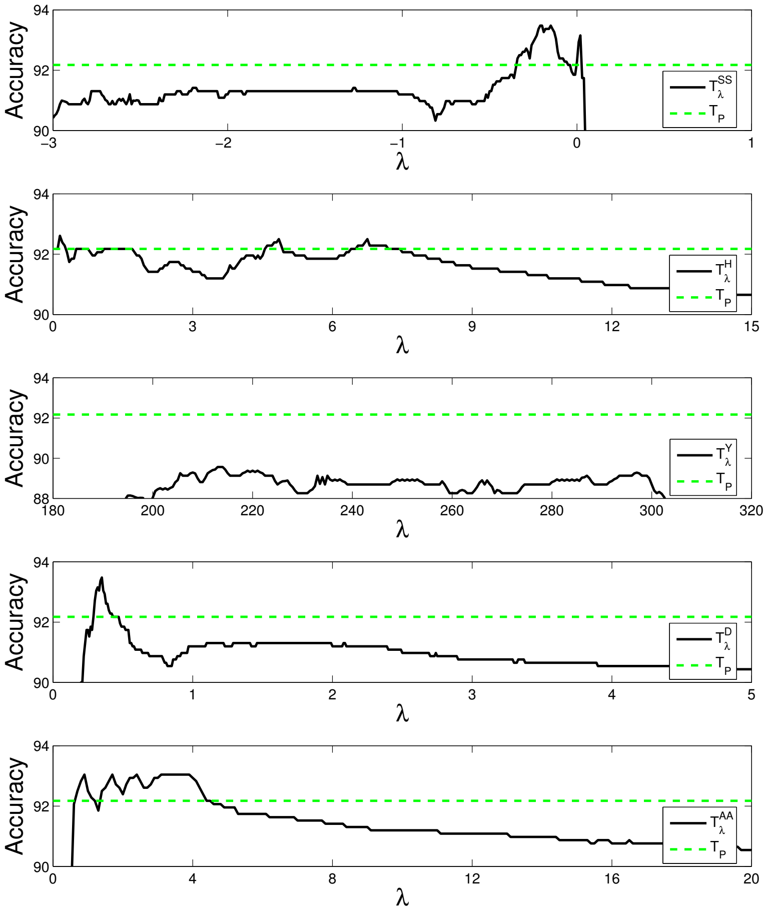

Figure 1. Recognition accuracy using the Schweizer-Sklar, Hamacher, Yager, Dombi and Aczél-Alsina t-norm families, relative to the product operator.

t-norm family	T_P	T_L	T_λ^{SS}	T_λ^{H}
Best performance	92.17%	0.26%	93.47%	92.60%
Rel. error reduction	—	—	16.60%	5.49%
t-norm family	T_P	T_λ^{Y}	T_λ^{D}	T_λ^{AA}
Best performance	92.17%	89.45%	93.47%	93.04%
Rel. error reduction	—	—	16.60%	11.11%

Table 3. Recognition percentages and relative error reduction rates when the triangular norms were applied.

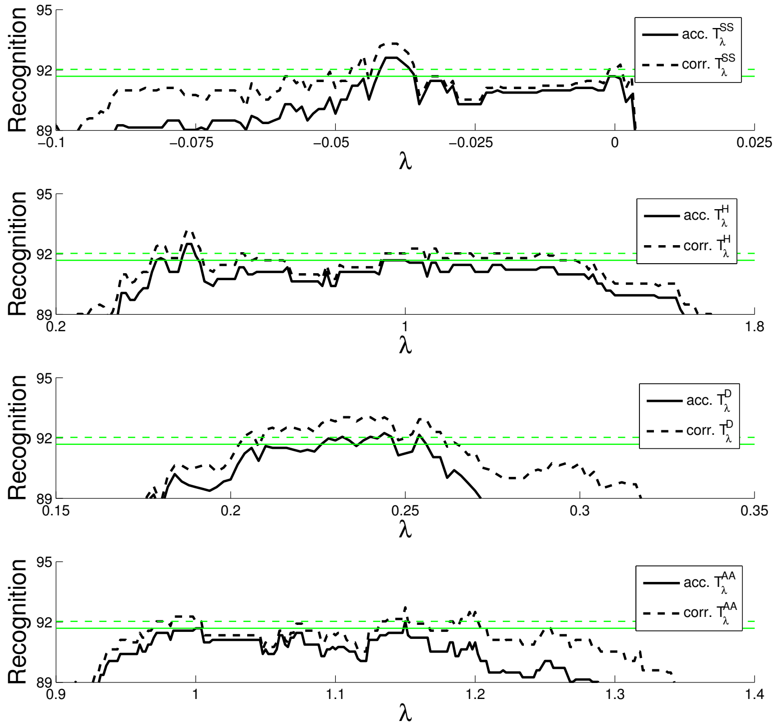

Figure 2. Recognition accuracy using the Schweizer-Sklar, Hamacher, Yager, Dombi and Aczél-Alsina t-norm families, relative to the product operator. The sides of the tested regions where there was no information shown, was omitted.

t-norm family	T_P	T_L	T_λ^{SS}	T_λ^{H}
Best accuracy	91.69%	0.69%	92.61%	92.50%
Relative error reduction	—	—	11.07%	9.74%
Best correctness	92.03%	0.81%	93.31%	93.19%
Relative error reduction	—	—	16.06%	14.55%
t-norm family	T_P	T_λ^{Y}	T_λ^{D}	T_λ^{AA}
Best accuracy	91.69%	—	92.25%	92.03%
Relative error reduction	—	—	6.73%	9.74%
Best correctness	92.03%	—	93.03%	92.73%
Relative error reduction	—	—	12.54%	8.78%

Table 4. Recognition percentages and relative error reduction rates when the triangular norms were applied.

greater than 0, the sum of the two parameters must be over 1. But the probability value of a hypothesis (one of these parameters) is usually very close to 0, while the probability of the next phoneme (the other parameter) is rarely greater than 0.5. Thus practically all hypotheses had a probability of 0, resulting in just a few hits which were just lucky guesses. The bad performance of the Sugeno-Weber t-norm family can be similarly explained: the probability values generated were 0 too often to get an acceptable result.

The remaining triangular norms produced an acceptable recognition performance, although the Yager t-norm family could not achieve the initial percentage value in the word recognition task. Also, it performed poorly on sentence recognition, which was clear even after the preliminary tests. The others, however, matched or even outperformed the basic product operator. In the first test environment the Hamacher and the Aczél-Alsina family gave a slight improvement of 5.49% and 11.11% (in terms of the relative error reduction). The Schweizer-Sklar [11] and the Dombi [12] t-norm families were the most effective for this task: we were able to achieve a 16.60% relative error reduction using these two t-norms, causing the recognition percentage to increase from 92.17% to 93.47%.

In the second test environment the results were similar. Although the relative error reduction rates were different for the accuracy and correctness scores, there was a definite improvement in the values. Here also the Schweizer-Sklar [11] proved to be the best-working triangular norm family with relative improvements of 11.07% and 16.06% (accuracy and correctness, respectively). The Hamacher family worked almost as well with rates of 9.74% and 14.55%; moreover, the Dombi and Aczél-Alsina families produced good results. Lastly, we should stress here that it is not the best parameter value that is the key issue, because it may vary in different settings (feature set, frame or segment-based model etc.). The actual value of λ can be tuned to suit the particular problem. What is important is the ability of a triangular norm to actually better the recognition scores.

5 Conclusions and Future Work

In this paper we investigated the usefulness of triangular norms for tasks in speech recognition. Several t-norms were tested as hypothesis probability approximators. The results confirm that this approach works, because we were able to reduce the recognition error by 16% without incurring any increase in the running times. The next logical step might be to experiment with full reinforcement aggregation operators (or uninorms) because their full reinforcement property makes them closer to human decision making. This is what we intend to do in the future.

References

[1] Dubois D., Prade H (editors), 2000, *Fundamentals of Fuzzy Sets*, Kluwer Academic.

[2] Klement E.P., Mesiar R., Pap E., 2000, *Triangular Norms*, Kluwer Academic.

[3] Franke K., Koppen M., Nickolay B., 2000, *Fuzzy Image Processing by Using Dubois and Prade Fuzzy Norms*, Proceedings of ICPR, Barcelona, Spain, pp. 3518-3521.

[4] Gosztolya G., Kocsor A., 2004, *Aggregation Operators and Hypothesis Space Reductions in Speech Recognition*, Proceedings of TSD, Brno, Czech Republic, pp. 315-322.

[5] Jelinek F., 1997, *Statistical Methods for Speech Recognition*, The MIT Press, Boston.

[6] Bishop C.M., 1995, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

[7] Bahl L.R., Gopalakrishnan P.S., Mercer R.L., 1993, *Search Issues in Large Vocabulary Speech Recognition*, Proceedings of the 1993 IEEE Workshop on Automatic Speech Recognition, Snowbird, UT.

[8] Kocsor A., Tóth L., Kuba A. Jr., 1999, *An Overview of the Oasis Speech Recognition Project*, Proceedings of ICAI '99, Eger-Noszvaj, Hungary, pp. 95-102.

[9] Vicsi K., Kocsor A., Teleki Cs., Tóth L., 2004, *Beszédadatbázis irodai számítógép-felhasználói környezetben*, Proceedings of MSZNY 2004, Szeged, Hungary, pp. 315-318. (In Hungarian)

[10] Young S. et al., *The HMM Toolkit (HTK) (software and manual)*, http://htk.eng.cam.ac.uk/

[11] Schweizer B., Sklar A., 1961, *Associative functions and statistical triangle inequalities*, Publ. Math. Debrecen 8, pp. 169-186.

[12] Dombi J., 1982, *A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators*, Fuzzy Sets and Systems 8, pp. 149-163.