Application of Full Reinforcement Aggregation Operators in Speech Recognition

András Kocsor and Gábor Gosztolya
Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences and University of Szeged, Hungary
e-mail: {kocsor, ggabor}@inf.u-szeged.hu

Abstracts: *In speech recognition probably the most important factor is the recognition accuracy. This is why many attempts have been made to improve it. One such idea might be to use some kind of aggregation method for hypothesis probability calculations. The triangular norms are tools for aggregating one probability value from multiple probability values, thus they seem to be good for this task. In this paper, however, we go even further: we apply full reinforcement aggregation operators because they work in a similar way to human reasoning due to their full reinforcement property. The tests also revealed that this idea is fruitful: we were able to reduce the relative error rates by* 16%.

1 Introduction

Fuzzy logic has a wide range of applications. One such area could be the hypothesis probability calculation in the field of automatic speech recognition. This paper deals with applying *full reinforcement aggregation operators* [6] for this task, because this idea could better the recognition rates with small or no further computational needs.

The aggregation task in speech recognition means that we have the probability of each phoneme and want to aggregate them in some way to get a probability value for the word they construct. Fuzzy logic offers a wide range of operators for this purpose. Perhaps the most well-known of them is the group of triangular norms (*t-norms*) which we used for probability calculation earlier [4]. But the full reinforcement aggregation operators (or *uninorms*) are tools more suitable for artificial intelligence purposes: if the values considered are all – or mostly – high ones, the result will also be high, while if the input values are mostly low ones, the resulting probability will also be low. This behavior models human reasoning more than the behavior of either the t-norms or their corresponding pairs, the triangular conorms (or *t-conorms*), which is why we chose them to improve the recognition accuracy of our system.

The structure of this paper is as follows. First we define the speech recognition problem and the hypothesis space. Next, we introduce t-norms, t-conorms and uninorms, and show the connection between each. Then we describe the test environment and analyze the test results. Lastly we draw some conclusions about the methods used in this paper.

2 The Speech Recognition Problem

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

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

which, using Bayes' theorem, is equivalent to the maximization problem $\hat{w} = arg\max_{w \in W} \frac{P(A|w) \cdot P(w)}{P(A)}$. Further, noting the fact that $P(A)$ is the same for all $w \in W$, we have that

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

Speech recognition models can be divided into two types – the discriminative and generative ones – depending on whether they use Eq. (1) or Eq. (2). In the experiments we restricted our investigations to the generative approach [5].

2.1 Unified view

Both the generative and discriminative models exploit *frame-based* and/or *segment-based* features, and this fact allows us to have a unified framework of the frame- and segment-based recognition techniques. First we will provide a brief outline of this framework along with its hypothesis structure.

Let us define w as $o_1 \dots o_n$, where o_j is the jth phoneme of word w, and let $A_1, \dots, A_n$ be non-overlapping segments of $A = a_1 \dots a_t$, 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 into a vector $I_n = [t_0, t_1, \dots, t_n]$ $(1 = t_0 < t_1 < \dots < t_n = t)$. We assume that the phonemes in a word are independent so that $P(A|w)$ can be obtained from $P(A_1|o_1), \dots, P(A_n|o_n)$ in some way. To calculate $P(A|w)$, various aggregation operators can be used at two levels. In the first one the $P(A_j|o_j)$ probability values are supplied by a g_1 operator, i.e. $P(A_j|o_j) = g_1([t_{j-1}, t_j], o_j)$, which provides an overall value for measuring how well the A_j segment represents the o_j phoneme. In the second one, another operator (g_2) is used to construct $P(A|w)$ using the probability values $P(A_1|o_1), \dots, P(A_n|o_n)$.

The frame-based approach

The well-known *Hidden Markov Model* (*HMM*) is a frame-based approach, i.e. it handles the speech signal frame by frame [8]. Usually a *Gaussian Mixture Model* is applied to compute the $P(a_l|o_j)$ values (for delta and delta-delta features neighboring observations are also required) and for the A_j segment the $g_1([t_{j-1}, t_j], o_j)$ value is defined by

$$\prod_{l=t_{j-1}}^{t_j} c_{o_j} \cdot P(a_{l-k} \dots a_{l+k}|o_j), \tag{3}$$

where $0 \leq c_{o_j} \leq 1$. Thus g_1 includes all the information we have when we are in a particular state of a HMM model. As for the $P(A|w)$ value, the g_2 operator is usually defined by

$$P(A_n|o_n) \prod_{j=1}^{n-1} (1 - c_{o_j}) P(A_j|o_j). \tag{4}$$

2.2 The hypothesis space

The task of speech recognition is a selection problem over a Cartesian product space where the first dimension is a set of word prefixes, while the second is a set of segmentations. For a set of words W we denote the k-long prefixes of the words in W as $Pref_k(W)$. Let $I^k = \{[t_0, \dots, t_k] : 1 = t_0 < \dots < t_k \leq t\}$ be the set of sub-segmentations made of k segments. The hypotheses will be elements of $H = \bigcup_{k=0}^{\infty}(Pref_k(W) \times I^k)$, while the root of the tree will be $h_0 = (\emptyset, [t_0]) \in H$. $Pref_1(W) \times I^1$ contains the first-level nodes, and for a $(o_1 \dots o_j, [t_0, \dots, t_j])$ leaf we link all the nodes $(o_1 \dots o_j o_{j+1}, [t_0, \dots, t_j, t_{j+1}]) \in Pref_{j+1}(W) \times I^{j+1}$.

Now we need to evaluate the nodes of the search tree. To this end let the g_1 and g_2 functions be defined by some aggregation operators. Then, for a node $(o_1 o_2 \dots o_j, [t_0, \dots, t_j])$, the value is usually defined by

$$g_2(g_1([t_0, t_1], o_1), \dots, g_1([t_{j-1}, t_j], o_j)). \tag{5}$$

Note that, in practice, it is worth calculating this expression recursively. In this paper we will use a frame-based framework, where g_1 is the traditional multiplication operator. As for g_2, we will test multiple aggregation operators to raise the recognition scores.

After defining the evaluation methodology, we will look for a leaf with the highest probability.

3 Full Reinforcement Aggregation Operators

In the fuzzy literature [6] a wide range of operators are described and studied. Hence, if we want to change the standard aggregation operator of the speech recognition problem, a straightforward idea is to look for one in the fuzzy domain. First we will define the triangular norms and triangular conorms, introduce the full reinforcement aggregation operators, then we will show how we can apply them in the speech recognition problem.

3.1 Triangular norms and conorms

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)

T-norms play the role of conjunction operators in fuzzy logic. One such t-norm is the product operator (T_P), which helps explain why it was useful for us to use various t-norms as we did with g_2 previously [4]. Moreover, it can be shown that for any t-norm T and $x, y \in [0, 1]$, $0 \leq T(x, y) \leq \min(x, y)$.

Definition 2 *A triangular conorm (t-conorm) is a binary operation S on the interval $[0, 1]$, i.e., a function $S : [0, 1]^2 \to [0, 1]$, which, for all $x, y, z \in [0, 1]$, satisfies $(T1) - (T3)$ and*

(S4)	$S(x, 0) = x$	(boundary condition)

T-conorms play the role of disjunction operators in fuzzy logic. One such t-conorm, of course, is the addition operator. Furthermore, as in the case of t-norms, it can be shown that for any $x, y \in [0, 1]$, $\max(x, y) \leq S(x, y) \leq 1$.

Lastly, let us introduce a notation. Associativity (T2) allows us to extend each t-norm T to an n-ary operation by induction, defining for each n-tuple $(x_1, x_2, \dots, x_n) \in [0, 1]^n$ as $T(x_1, x_2, \dots, x_n) = T(\dots T(T(x_1, x_2), x_3) \dots, x_n)$. If, in particular, we have $x_1 = x_2 = \dots = x_n$, we briefly write $x_T^{(n)} = T(x, x, \dots, x)$. Finally we put, by convention, for each $x \in [0, 1]$ $x_T^{(0)} = 1$ and $x_T^{(1)} = x$. Of course the same can be done with any t-conorm S. This extension makes it easier for us to use any t-norm T and t-conorm S in Eq. (5) both as g_1 or g_2, although here we will replace only the latter one.

3.2 Uninorms

Full reinforcement aggregation operators or uninorms are operators that are based on the common properties of both t-norms and t-conorms, i.e., commutativity, associativity and monotonicity. The difference is in the fourth axiom:

Definition 3 *A uninorm is a binary operation U on the unit interval $[0,1]$, i.e., a function $U : [0,1]^2 \to [0,1]$, which satisfies $(T1) - (T3)$ and*

(U4) U has a neutral element $e \in]0,1[$, i.e. $U(x,e) = x$

The construction of an n-ary operator above can be also done for a uninorm. Moreover, it can be shown that a uninorm behaves like a t-norm if $0 \leq x, y \leq e$ and like a t-conorm if $e \leq x, y \leq 1$. This property, if applied backwards, permits a way of uninorm construction such that for any uninorm U

$$U(x,y) = \begin{cases} eT(\frac{x}{e}, \frac{y}{e}) & \text{if } 0 \leq x, y \leq e, \\ e + (1-e)S(\frac{x-e}{1-e}, \frac{y-e}{1-e}) & \text{if } e < x, y \leq 1, \end{cases} \tag{6}$$

where T and S are any t-norm and t-conorm, respectively. There are remaining regions, however, where $0 \leq x \leq e$ and $e < y \leq 1$, and where $e < x \leq 1$ and $0 < y \leq e$. In these regions $U(x,y)$ can be any function as long as $\min(x,y) \leq U(x,y) \leq \max(x,y)$ is satisfied. In this paper we chose the minimum function, i.e. $U(x,y) = \min(x,y)$, based on the results of preliminary tests.

3.3 Tested Uninorms

Klement, Mesiar and Pap [6] list the most important t-norm and t-conorm families. Based on their work, and using the experience gained from our previous tests [4], we will use the following triangular norm and conorm families:

Schweizer-Sklar t-norms and t-conorms ($\lambda \in R, \lambda \neq 0$):

$$T_\lambda^{SS}(x,y) = (\max((x^\lambda + y^\lambda - 1), 0))^{1/\lambda} \tag{7}$$

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

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

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

$$S_\lambda^{H}(x,y) = \frac{x + y - xy - (1-\lambda)xy}{1 - (1-\lambda)xy} \tag{10}$$

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

$$T_\lambda^{D}(x,y) = \frac{1}{1 + ((\frac{1-x}{x})^\lambda + (\frac{1-y}{y})^\lambda)^{1/\lambda}} \tag{11}$$

$$S_\lambda^{D}(x,y) = 1 - \frac{1}{1 + ((\frac{x}{1-x})^\lambda + (\frac{y}{1-y})^\lambda)^{1/\lambda}} \tag{12}$$

Aczél-Alsina t-norms and t-conorms ($\lambda > 0$):

$$T_\lambda^{AA}(x,y) = e^{-((-\log x)^\lambda + (-\log y)^\lambda)^{1/\lambda}} \tag{13}$$

$$S_\lambda^{AA}(x,y) = 1 - e^{-((-\log(1-x))^\lambda + (-\log(1-y))^\lambda)^{1/\lambda}} \tag{14}$$

3.4 Uninorms in Speech Recognition

In speech recognition the norms we described can be used as the g_1 or g_2 function. As the default value of these functions is multiplication in both cases, it is straightforward to use triangular norms with these problems [4]. T-norms, however, have the property that one low value drastically reduces the resulting value. As it happens, the human mind works in a different way for a t-norm: one low value can be compensated by the other high ones and vice versa. And this is precisely what uninorms give us: if all – or almost all – the input values (in the case of g_2 phoneme probabilities) are high, the result will be high. But if most input values are low, the result – in our case the hypothesis probability – will also be low.

The application of uninorms is not limited to g_2. They can be used to define g_1, provided the speech recognition framework is a frame-based one. In this paper, however, we only discuss g_2 due to lack of space.

4 Experiments and Results

The tests were made in the framework of the OASIS Speech Laboratory [7]. The train database consisted of 500 speakers, each uttering 10 sentences and 4 words via telephone. The test database consisted of all these speakers uttering the name of a town or city. Some of these utterances were unrecognizable even to humans, so in the end the test database contained 431 different words. The *HTK* system [9] used for reference produced 92.11% under these circumstances. Unfortunately, the Hungarian language is an agglutinative one which makes it more difficult to construct a language model than it is for English, but in the near future we are planning to recognize whole sentences instead of isolated words.ab

In the further tests the problem was that there were far too many possible uninorms to test, so we had to somehow reduce the ways of uninorm-construction. First we decided to pair only t-norms and t-conorms from the same family. But still we had three possible parameters from now on: λ_T, the λ parameter of the t-norm; λ_S, that is of the t-conorm; and e, the neutral element of the uninorm. First we decided to assign λ_T to a value where T produces the best results using it as the aggregation function g_2. (This is equivalent as having e fixed to 1.) Next, λ_S was determined in the same way by fixing e to 0, although it produced worse recognition rates due to the fact that a t-conorm is not really suitable for being used as a g_2 function. Lastly, the optimal value of e was determined, where we could now use the values λ_T and λ_S. Figure 1 shows the recognition rates we obtained for each t-norms, t-conorms and uninorms.

It can be seen, then, that the t-norms alone outperformed the original product operator. T-conorms, as expected, did not perform so well, although, surprisingly, the Dombi and the Aczél-Alsina t-conorm families attained a result over 10%. The performance of the uninorms was even better than that of the t-norms in the case of U^D and U^{AA}, too. In each case on the lower e values naturally the t-conorms dominated, while on the higher ones the recognition percentage corresponded to the one for the t-norm alone. In the end we found that the uninorms constructed from the Dombi [2] and the Aczél-Alsina [1] family produced the best rates.

5 Conclusions

In this paper we investigated the usefulness of fuzzy operator types in speech recognition. We tested triangular norm and triangular conorm families for the purpose of hypothesis probability calculations, then with their combinations we constructed full reinforcement aggregation operators for the same task. These operators, having the advantages of both the t-norms and the t-conorms, were able to improve the recognition accuracy from 94.20% to 95.12%, which was

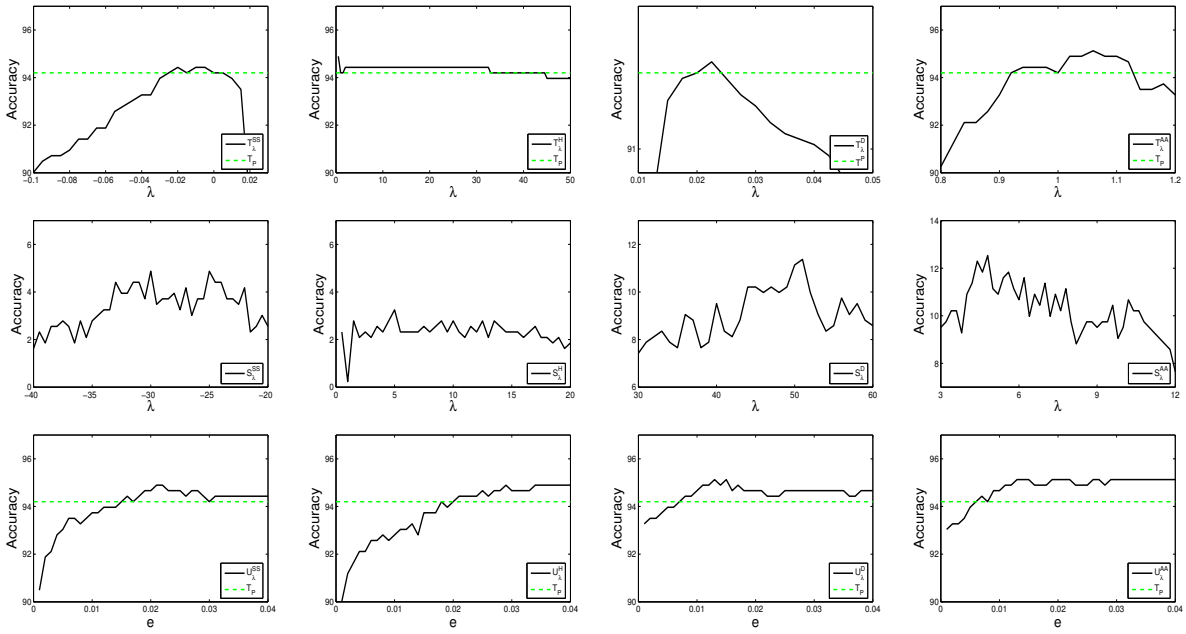

Figure 1: Recognition accuracy using the Schweizer-Sklar, Hamacher, Dombi and Aczél-Alsina t-norm and t-conorm families, and their constructed uninorms.

equivalent to a 16% increase in terms of the relative error reduction.

References

[1] J. Aczél and C. Alsina (1984). Characterizations of some classes of quasilinear functions with applications to triangular norms and to synthesizing judgements. In *Methods Oper. Res. 48*, pages 3–22, 1984.

[2] J. Dombi (1982). A general class of fuzzy operators, the de morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. In *Fuzzy Sets and Systems 8*, pages 149–163, 1982.

[3] J. Glass, J. Chang, and M. McCandless (1996). A probabilistic framework for features-based speech recognition. In *Proceedings of ICSLP '99*, pages 2277–2280, Philadelphia, PA, 1996.

[4] G. Gosztolya and A. Kocsor (2006). Using triangular norms in a segment-based asr system. In *Submitted to ICAISC*, Zakopane, Poland, 2006.

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

[6] E. Klement, R. Mesiar, and E. Pap (2000). *Triangular Norms*. Kluwer Academic Publisher, 2000.

[7] A. Kocsor, L. Tóth, and A. Kuba Jr (1999). An overview of the oasis speech recognition project. In *Proceedings of ICAI '99*, Eger-Noszvaj, Hungary, 1999.

[8] L. Rabiner and B.-H. Juang (1993). *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

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