USING TRIANGULAR NORMS IN A SEGMENT-BASED ASR SYSTEM

Gábor GOSZTOLYA¹, András KOCSOR²

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

²MTA-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 \( A = a_1 a_2 \ldots a_t \), 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

\[
\hat{w} = \arg \max_{w \in W} P(w|A),
\]

which, using Bayes’ theorem, is equivalent to

\[
\hat{w} = \arg \max_{w \in W} \frac{P(A|w) \cdot P(w)}{P(A)}.
\]

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

\[
\hat{w} = \arg \max_{w \in W} P(A|w)P(w).
\]

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 \( o_1 \ldots o_n \), where \( o_j \) is the \( j \)th phoneme of a word (or word sequence) \( w \). Furthermore, let \( A_1, \ldots, A_n \) be non-overlapping segments of the observation series \( A = a_1 \ldots a_t \), where \( A_j = a_{t_{j-1}} \ldots a_{t_j}, \ j \in \{1, \ldots, n\} \). 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 \( A = A_1, \ldots, A_n \) we put the values of the time indices belonging to each segment into a vector \( S_n = [t_0, \ldots, t_n] \) (\( 1 = t_0 < \ldots < t_n = t \)). We make the conventional assumption that the phonemes in a word are independent, thus \( P(A|w) \) can be obtained from \( P(A_1|o_1), \ldots, P(A_n|o_n) \) 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 $\text{Pref}_k(W)$ 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 $a_1 \cdots a_t$. The hypotheses will be the elements of $H = \bigcup_{k=0}^{\infty} (\text{Pref}_k(W) \times S^k)$. We will denote the root of the tree – the initial hypothesis – by $h_0 = (\emptyset, [t_0])$, $h_0 \in H$. $\text{Pref}_1(W) \times S^1$ will contain the first-level nodes. For a $(o_1 \ldots o_j, [t_0, \ldots, t_j])$ leaf we link all $(o_1 \ldots o_j o_{j+1}, [t_0, \ldots, t_j, t_{j+1}]) \in \text{Pref}_{j+1}(W) \times S^{j+1}$ 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

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

(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 probabil-
ity 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 $T : [0, 1]^2 \rightarrow [0, 1]$, 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

$$T_L(x, y) = \max(x + y - 1, 0).$$  \hfill (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 $\lambda$ values corresponding to some basic t-norms, but here we omit the full listing due to lack of space. Schweizer-Sklar t-norms ($\lambda \in \mathbb{R}, \lambda \neq 0$):

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

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

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

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

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

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

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

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

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

Aczél-Alsina t-norms ($\lambda > 0$):

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

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

$$T_{\lambda}^{MT}(x, y) = \begin{cases} \max(x + y - \lambda, 0) & \text{if } \lambda \in [0, 1] \text{ and } (x, y) \in [0, 1]^2, \\ \min(x, y) & \text{otherwise.} \end{cases}$$  \hspace{1cm} (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 $T_{\lambda}^{MT}$ 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
Table 1. The intervals where the triangular norm families we deal with are continuous, Archimedean, and both continuous and Archimedean.
Table 2. The tested interval and the step sizes of the triangular norm families.

<table>
<thead>
<tr>
<th></th>
<th>$T^S_3$</th>
<th>$T^H_1$</th>
<th>$T^Y_1$</th>
<th>$T^D_1$</th>
<th>$T^{AA}_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word interval</td>
<td>[−3, 1]</td>
<td>[0, 15]</td>
<td>[180, 320]</td>
<td>[0, 5]</td>
<td>[0, 20]</td>
</tr>
<tr>
<td>step</td>
<td>0.01</td>
<td>0.02</td>
<td>0.5</td>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>Sentence interval</td>
<td>[−2, 0.5]</td>
<td>[0, 2]</td>
<td>—</td>
<td>[0.1, 0.5]</td>
<td>[0.9, 1.4]</td>
</tr>
<tr>
<td>step</td>
<td>0.01</td>
<td>0.01</td>
<td>—</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

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 $\lambda$ 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 $\lambda$ 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 $T_L = T^{MT}_1$, 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

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

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

<table>
<thead>
<tr>
<th>t-norm family</th>
<th>$T_P$</th>
<th>$T_L$</th>
<th>$T^{SS}_L$</th>
<th>$T^H_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best performance</td>
<td>92.17%</td>
<td>0.26%</td>
<td>93.47%</td>
<td>92.60%</td>
</tr>
<tr>
<td>Rel. error reduction</td>
<td>—</td>
<td>—</td>
<td>16.60%</td>
<td>5.49%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t-norm family</th>
<th>$T_P$</th>
<th>$T^{Y}_L$</th>
<th>$T^{D}_L$</th>
<th>$T^{AA}_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best performance</td>
<td>92.17%</td>
<td>89.45%</td>
<td>93.47%</td>
<td>93.04%</td>
</tr>
<tr>
<td>Rel. error reduction</td>
<td>—</td>
<td>—</td>
<td>16.60%</td>
<td>11.11%</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>t-norm family</th>
<th>$T_P$</th>
<th>$T_L$</th>
<th>$T^SS$</th>
<th>$T^SH$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best accuracy</td>
<td>91.69%</td>
<td>0.69%</td>
<td>92.61%</td>
<td>92.50%</td>
</tr>
<tr>
<td>Relative error reduction</td>
<td>—</td>
<td>—</td>
<td>11.07%</td>
<td>9.74%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t-norm family</th>
<th>$T_P$</th>
<th>$T_Y$</th>
<th>$T^H$</th>
<th>$T^AA$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best accuracy</td>
<td>91.69%</td>
<td>—</td>
<td>92.25%</td>
<td>92.03%</td>
</tr>
<tr>
<td>Relative error reduction</td>
<td>—</td>
<td>—</td>
<td>6.73%</td>
<td>9.74%</td>
</tr>
</tbody>
</table>

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 prob-
ability 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, result-
ing 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 per-
formance, although the Yager t-norm family could not achieve the initial per-
centage value in the word recognition task. Also, it performed poorly on sen-
tence 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 ef-
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 rela-
tive 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 $\lambda$ can be tuned to suit the particular prob-
lem. 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 re-
inforcement aggregation operators (or uninorms) because their full reinforce-
ment property makes them closer to human decision making. This is what we intend to do in the future.

References


