Why Triangular Membership Functions Work Well in F-Transform: A Theoretical Explanation

Jaime Nava and Vladik Kreinovich
Center for Theoretical Research and its
Applications in Computer Science (TRACS)
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
jenava@miners.utep.edu, vladik@utep.edu

Abstract

In many practical applications, it is useful to represent a signal or an image by its average values on several fuzzy sets. The corresponding *F-transform* technique has many useful applications in signal and image processing. In principle, we can use different membership functions. Somewhat surprisingly, in many applications, the best results occur when we use triangular membership functions. In this paper, we provide a possible theoretical explanation for this empirical phenomenon.

1 Triangular Membership Functions Work Well in F-Transform: An Empirical Fact

Need for approximating signals and images. In many practical situations, it is important to process signals and images. From the mathematical viewpoint, a signal is a function $x(t)$ that describe the recorded value of the corresponding quantity at different moments of time t. Similarly, an image is a function $I(x, y)$ that describe the intensity at a spatial point with coordinates (x, y). In principle, we can represent each signal by describing, for each moment t, the corresponding value $x(t)$; however, this representation often requires too much computer memory, more than the current device has – and/or too much computer time to process all these values, more than the time we need to make a decision. In such situations, it is desirable to come up with a smaller number of values representing the original signal – the values from which the signal can be reproduced with a good approximation accuracy.

Human experts can do it. We humans face the same problem when we need to make an urgent decision based on an observed signal or an observed image. In such situations, we do not use all the values, we usually base our decisions on our perception of the signal – perception which is usually described by using imprecise ("fuzzy") words from natural language. For example, when an investor uses a recorded performance of a certain stock to make a buy or sell decision, the investor usually explains his or her decision by using arguments like "the price of the stock was increasing rapidly last year, this year is somewhat slowed down". This explanation uses imprecise terms like "rapidly", "somewhat", etc. Such decisions are often very successful. It is therefore reasonable to try to teach computers how to make similar "fuzzy" arguments and decisions.

How to use imprecise expert knowledge: fuzzy techniques. A natural way to describe imprecise natural-language words in computer-understandable (precise) terms has been provided by *fuzzy logic*; see, e.g., [4, 5, 10]. In fuzzy logic, each imprecise term is characterized by a function $\mu(x)$ which assigns, to each possible value x of the corresponding quantity, a degree $\mu(x) \in [0,1]$ to which this value satisfies this property. For example, to describe a property "cheap", we assign, to each price x, the degree to which an expert considers this price to be cheap.

From crisp approximation to fuzzy approximation: the notion of F-transform. In order to come up with a reasonable way of using fuzzy techniques for approximation, let us first analyze how we can use crisp (= precise, non-fuzzy) ideas for such an approximation.

In the extreme case, when we can only store a single number to represent the whole signal $x(t)$ on a time interval $[\underline{t}, \overline{t}]$, it is reasonable to represent this signal by its average value $x \stackrel{\text{def}}{=} \dfrac{1}{\overline{t}-\underline{t}} \cdot \displaystyle\int_{\underline{t}}^{\overline{t}} x(t)\,dt$. Based on this average value, we can reconstruct the original signal as $x(t) \approx x_{\approx}(t) = x$ for all t.

If we have enough space and/or computation time to represent the original signal by $n > 1$ numbers, then it is reasonable to divide the corresponding time interval $[\underline{t}, \overline{t}]$ into n subintervals $[t_0, t_1)$, $[t_1, t_2)$, $\ldots$, $[t_{n-1}, t_n)$, with $t_0 = \underline{t}$ and $t_n = \overline{t}$, and store the averages over each subinterval $x_i \stackrel{\text{def}}{=} \dfrac{1}{t_i - t_{i-1}} \cdot \displaystyle\int_{t_{i-1}}^{t_i} x(t)\,dt$. Once we know these values, a natural way to reconstruct the original signal is to take $x_{\approx}(t) = x_i$ when $t \in [t_{i-1}, t_i]$.

This procedure can be equivalently represented in a more analytical form if we introduce the characteristic functions $A_i(t)$ of the corresponding intervals, i.e., functions for which $A_i(t) = 1$ when t belongs to the i-th interval and $A_i(t) = 0$ otherwise. These functions have the properties that for every t, only one of them is different from 0, and their sum is always equal to 1. (Alternatively, we can assign degree 1/2 to borderline points t_i, then the sum is still equal to 1, but we may have two functions different from 0 for some t.)

In terms of the characteristic functions, the above expression for x_i takes the form

$$x_i = \frac{\int A_i(t) \cdot x(t)\, dt}{\int A_i(t)\, dt}, \tag{1}$$

and the resulting approximation for the original signal takes the form

$$x_{\approx}(t) = \sum_{i=1}^{n} A_i(t) \cdot x_i. \tag{2}$$

In the crisp case, each value t either belongs to the i-th interval or not, so the value $A_i(t)$ is equal either to 1 or to 0. In the fuzzy case, instead of intervals, we have fuzzy sets with membership functions $A_i(t)$ which can take values from the whole interval $[0, 1]$. Similarly to the crisp case, it is reasonable to require that for each t, at most two functions $A_i(t)$ are different from 0 and that the sum $\sum\limits_i A_i(t)$ is always equal to 1. Once such membership functions $A_i(x)$ are given, we can use the same formula (1) and (2) to describe the corresponding approximation to the original signal.

The values x_i form a *fuzzy transform* (or *F-transform*, for short), and the resulting approximation $x_{\approx}(t)$ is known as the *inverse F-transform.*

F-transform is useful in many practical applications. In many problems related to signal and image processing, F-transform works very well; see, e.g., see, e.g., [3, 6, 7, 8, 9].

Triangular membership functions work well in F-transform: why? In principle, we can use many different membership functions $A_i(x)$ in F-transform. Interestingly, in many practical applications, triangular membership functions seem to work the best see, e.g., [3, 6, 7, 8, 9]. Until now, there was no theoretical explanation for this empirical fact. In this paper, we provide a possible theoretical explanation for this empirical phenomenon.

2 A Possible Theoretical Explanation

Numerical values of time depend on choice of a measuring unit and a starting point. Most applications of F-transform are to signal and image processing. A signal $x(t)$ describes how the value of the corresponding property x changes with time t.

To describe a signal in a computer, we need to measure time, i.e., we need to provide a numerical values for different moments of time. To describe such values, we need to select a starting point (corresponding to $t = 0$) and a measuring unit (corresponding to $t = 1$). If we change the starting point and/or measuring unit, we get different numerical values for time.

If we replace the original starting point with a new starting point which is s units before the previous one, then each original numerical value t will be

replaced by the new shifted value $t' = t + s$. Similarly, if we replace the original measuring unit by a new unit which is λ times smaller (e.g., replace a minute by a second which is 60 times smaller), then the resulting numerical values get multiplied by λ, i.e., instead of the original value t, we get a new re-scaled value $t' = \lambda \cdot t$.

It is reasonable to require that the class of approximations does not change when we change a measuring unit and/or a starting point. In general, an approximation (2) obtained by using F-transforms is a linear combination of the membership functions $A_i(x)$. On each interval, no more than two membership functions are different from 0. So, on the interval on which $A_i(x)$ and $A_{i+1}(x)$ are different from 0, an approximation takes the form

$$x_{\approx}(t) = x_i \cdot A_i(t) + x_{i+1} \cdot A_{i+1}(t). \tag{3}$$

On this interval, the formula $\sum\limits_i A_i(t) = 1$ takes the form $A_i(x) + A_{i+1}(t) = 1$, hence $A_{i+1}(x) = 1 - A_i(x)$, and the formula (3) takes the form

$$x_{\approx}(t) = a + b \cdot A_i(t), \tag{4}$$

where $a \stackrel{\text{def}}{=} a_{i+1}$ and $b \stackrel{\text{def}}{=} x_i - x_{i+1}$.

For different signals $x(t)$, we get all possible combinations of x_i and x_{i+1} and therefore, all possible combinations of the values a and b. Thus, the class of all approximations has the form $\{a + b \cdot A_i(t)\}_{a,b}$, for all possible values of a and b.

We are looking for techniques which work well for all possible signals, no matter what measuring unit and/or starting point we choose. It is therefore reasonable to require that the resulting class of approximating signals does not change if we simply change the measuring unit and/or the starting point.

Towards formalizing this requirement. If we change the starting point, then the numerical value of the time changes to $t' = t + s$. In the new units, we get family $\{a + b \cdot A_i(t')\}_{a,b}$, which, in the old units, takes the form $\{a + b \cdot A_i(t + s)\}_{a,b}$. We therefore require that these two classes coincide.

Similarly, if we change the measuring unit, then the numerical value of the time changes to $t' = \lambda \cdot t$. In the new units, we get the family $\{a + b \cdot A_i(t')\}_{a,b}$, which, in the old units, takes the form $\{a + b \cdot A_i(\lambda \cdot t)\}_{a,b}$. We also require that these two classes coincide.

A membership function is usually piece-wise monotonic, i.e., its domain consists of finitely many intervals on each of which it is either non-decreasing or non-increasing. We are interested in a local behavior, so we can assume that the function $A(t)$ is monotonic.

So, we arrive at the following definitions.

Definition. *We say that a monotonic function $A(t)$* leads to shift- and scale-invariant approximations *if for every real number s and for every positive real number $\lambda > 0$, the classes $\{a + b \cdot A(t + s)\}_{a,b}$ and $\{a + b \cdot A(\lambda \cdot t)\}_{a,b}$ coincide with the class $\{a + b \cdot A(t)\}_{a,b}$.*

Proposition. *A function $A(t)$ leads to shift- and scale-invariant approximations if and only if this function is linear.*

Proof.

1°. If a function $A(t)$ is linear, then the class $\{a + b \cdot A(\lambda \cdot t)\}_{a,b}$ consists of all linear functions. It is easy to see that in this case, the classes $\{a+b\cdot A(t+s)\}_{a,b}$ and $\{a+b\cdot A(\lambda\cdot t)\}_{a,b}$ also consists of all linear functions and therefore, coincide with the class $\{a + b \cdot A(\lambda \cdot t)\}_{a,b}$.

2°. Let us now assume that the function $A(t)$ leads to shift- and scale-invariant approximations. This means, in particular, that for every s, the classes $\{a + b \cdot A(t + s)\}_{a,b}$ and $\{a + b \cdot A(t)\}_{a,b}$ coincide, so every function from the first class, including the function $A(t + s)$, belongs to the second class. In other words, for every s, there exist values $a(s)$ and $b(s)$ (depending on s) for which, for every t, we have

$$A(t + s) = a(s) + b(s) \cdot A(t) \tag{1}$$

2.1°. Let us prove that the function $A(t)$ is differentiable for all t.

It is known that a monotonic function is almost everywhere differentiable; see, e.g., [1, 2]. Let t_0 be a point at which the function $A(t)$ is differentiable, i.e., for which there exists a limit $A'(t_0) = \lim\limits_{h\to 0} \dfrac{A(t_0 + h) - A(t_0)}{h}$. Let t be any real number. We want to prove that the ratio $\dfrac{A(t + h) - A(t)}{h}$ also tends to a limit when h tends to 0. Indeed, for $s \stackrel{\text{def}}{=} t - t_0$, the formula (1) takes the form $A(t) = a(s) + b(s) \cdot A(t_0)$ and $A(t + h) = a(s) + b(s) \cdot A(t_0 + h)$. Substituting these expressions into the desired ratio, we conclude that

$$\frac{A(t + h) - A(t)}{h} = b(s) \cdot \frac{A(t_0 + h) - A(t_0)}{h}.$$

The right-hand side of this equality tends to $A'(t_0)$, thus, the left-hand side tends to the limit $b(s) \cdot A'(t_0)$. The statement is proven.

2.2°. Let us now prove that the functions $a(s)$ and $b(s)$ are also differentiable.

Indeed, for two different values $t_1 \neq t_2$, the formula (1) takes the form

$$A(t_1 + s) = a(s) + b(s) \cdot A(t_1); \tag{2}$$

$$A(t_2 + s) = a(s) + b(s) \cdot A(t_2). \tag{3}$$

Subtracting the equality (3) from the equality (2), we get $A(t_1+s)-A(t_2+s) = b(s)\cdot(A(t_1)-A(t_2))$, hence

$$d(s) = \frac{A(t_1+s)-A(t_2+s)}{A(t_1)-A(t_2)}. \tag{4}$$

Since the function $A(t)$ is differentiable, the right-hand side of the equality (4) is differentiable in s. Therefore, the function $b(s)$ is also differentiable.

From (2), we can now conclude that $a(s) = A(t_1+s) - b(s)\cdot A(t_1)$ is also differentiable. The statement is proven.

2.3°. Now that we know that the functions $A(t)$, $a(s)$, and $b(s)$ are all differentiable, we can differentiate both sides of the formula (1) with respect to s. As a result, we get the equality

$$A'(t+s) = a'(s) + b'(s)\cdot A(t).$$

Substituting $s=0$ into this formula, we get

$$A'(t) = a + b\cdot A(t), \tag{5}$$

where we denoted $a \stackrel{\text{def}}{=} a'(0)$ and $b \stackrel{\text{def}}{=} b'(0)$. Thus, $\frac{dA}{dt} = a + b\cdot A$. Moving all the terms containing A to the left-hand side and all the other terms to the right-hand side, we get

$$\frac{dA}{a+b\cdot A} = dt. \tag{6}$$

To integrate this equation, we consider two possible cases: $b=0$ and $b\neq 0$.

2.4°. If $b=0$, then $\frac{dA}{a} = dt$, so integrating both sides, we get $\frac{A}{a} = t + C$ for an integration constant C, i.e., we get $A(t) = a\cdot t + a\cdot C$. In this case, $A(t)$ is a linear function.

2.5°. If $b\neq 0$, then we can introduce a new variable $B \stackrel{\text{def}}{=} A + \frac{a}{b}$. In terms of this new variable, $a = b\cdot A = b\cdot B$, so the formula (6) takes the form $\frac{dB}{b\cdot B} = dt$. Integrating both sides, we get $\frac{1}{b}\cdot\ln(B) = t + C$, hence $\ln(B) = b\cdot t + C_1$, where we denoted $C_1 \stackrel{\text{def}}{=} b\cdot C$. Exponentiating both sides, we get $B(t) = C_2\cdot\exp(b\cdot t)$, where $C_2 \stackrel{\text{def}}{=} \exp(C_1)$, and thus,

$$A(t) = B(t) - \frac{a}{b} = C_2\cdot\exp(b\cdot t) - \frac{a}{b}. \tag{7}$$

Let us show that such functions are not scale-invariant and thus, this case is impossible.

Indeed, scale-invariance means, in particular, that for every $\lambda > 0$, the function $A(\lambda\cdot t)$ has the form $a(\lambda) + b(\lambda)\cdot A(t)$ for appropriate $a(\lambda)$ and $b(\lambda)$. For the function (7), this means that

$$C_2\cdot\exp(b\cdot\lambda\cdot t) - \frac{a}{b} = a(\lambda)\cdot\left(\exp(b\cdot t) - \frac{a}{b}\right) + b(\lambda). \tag{8}$$

For $\lambda = 2$, if $b > 0$, the left-hand side of the formula (8) grow faster than the right-hand side for $t \to +\infty$; for $b < 0$, the same is true when $t \to -\infty$. In both cases, it is not possible to have the equality (8) for all t.

Thus, the case $b \neq 0$ is indeed impossible, so the only shift- and scale-invariant function $A(t)$ is indeed linear. The proposition is proven.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

The authors are thankful to Irina Perfilieva for her encouragement and helpful discussions.

References

[1] R. G. Bartle, *The Elements of Real Analysis*, Wiley, New York, 1976.

[2] M. Hazewinkel, "Monotone function", In: *Encyclopedia of Mathematics*, Springer, 1994.

[3] M. Holčapek and T. Tichý, "A smoothing filter based on fuzzy transform", *Fuzzy Sets and Systems*, 2011, Vol. 180, pp. 69–97.

[4] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[5] H. T. Nguyen and E. A. Walker, *First Course In Fuzzy Logic*, CRC Press, Boca Raton, Florida, 2006.

[6] V. Novák, M. Štěpnička, A. Dvořák, I. Perfilieva, V. Pavliska, and L. Vavřícková, "Analysis of seasonal time series using fuzzy approach", *International Journal of General Systems*, 2010, Vol. 39, No. 3, pp. 305–328.

[7] V. Novák, M. Štěpnička, I. Perfilieva, and V. Pavliska, "Analysis of periodical time series using soft computing techniques", *Proceedings of the 8th International FLINS Conference on Computational Intelligence in Decision and Control FLINS'2008*, Madrid, Spain, September 21–24, 2008.

[8] I. Perfilieva, "Fuzzy transforms: theory and applications", *Fuzzy Sets and Systems*, 2006, Vol. 157, pp. 993–1023.

[9] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, "Analysis and prediction of time series using fuzzy transform", *Proceedings of IEEE World Congress on Computational Intelligence WCCI'2008*, 2008, pp. 3875–3879.

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