INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 3, Month: June, Year: 2026
Article Number: 7526, https://doi.org/10.15837/ijccc.2026.3. 7526

CCC Publications

Interpolation and Approximation with Fuzzy Logic Systems and t-Logic Systems

Horia-Nicolai L. Teodorescu

Horia-Nicolai L. Teodorescu
"Gheorghe Asachi" Technical University of Iasi
Dept. ETTI
Iasi, Romania
and
Romanian Academy
Institute of Computer Science
Iasi, Romania
hteodor@etti.tuiasi.ro

Abstract

In many engineering, computer science and economic applications, the model, or the system design has to satisfy the condition that it precisely (or approximately) goes through a set of specified points. This means that the system has to perform as an (approximate) interpolator (interpolating function, also named interpolants). The first question asked is if there is a simple way to design good interpolators with FLSs. We answer this first question in a positive way by providing a systematic and simple method for interpolation with FLSs and proving that FLSs are universal exact interpolators. Fuzzy logic systems are known to be universal approximators; that guarantees that they can also perform as universal approximate interpolators. Finding a good approximator with fuzzy logic systems (FLSs) may be difficult and computationally demanding. A procedure is provided for automatically building guaranteed approximators for real valued functions defined on the line. Also, a method for building interpolators for functions with jump discontinuities is shown.

The next question asked is: Is it possible to define interpolators with logic systems under various other logics determined by t-norms and related s-co-norms? A method for interpolation and approximation in R^n is proposed, based on t-Logic Systems (t-LS, TLS). t-Logic Systems are systems from an n-dimensional space to the real line, based on a logic defined by t-norms and co-norms. They include a first stage of assigning to the real-valued inputs a t-distributions, a second stage where an inference is performed in the framework of the chosen logic, and an estimation stage, where the result of the inference is converted into a real number. Typical examples include fuzzy systems, Z-number based system, and probabilistic systems. Similar to neural networks (NN), fuzzy logic systems (FLSs) with center of gravity (c.o.g.) estimator (named defuzzifier for FLSs) are universal approximators. Even more, FLSs with a c.o.g. defuzzifier are interpolators for well-behaved functions. The general t-logic systems have similar properties. These general properties have wide applicability in economics, engineering, and decision making. In many cases, the type of t-logic, in particular the type of fuzzy logic used in the FLS does not play an important role in the universal approximation or interpolation properties, thus leaving much space for the tLS and FLSs optimization. The paper provides results about interpolation, and in subsidiary, for approximation with tLSs, in particular with FLSs, and discusses various applications. The results may find applications in various fields.

Keywords: interpolation, approximation, logic system, t-norm, fuzzy logic system, dense set, isopleth mapping.

1 Introduction

Applications of interpolating functions include domains such as image processing [29], cartography [10], modeling in economics [7], engineering [3], medicine, data recovery [4], [13], data analytics and visualization [11], [14], [16], [20], control, physics and material science [17], and numerical analysis.

We name interpolator any interpolating function; "interpolant" is another name used in the literature. The interpolators discussed have the form $f : R \to R^n$. For example, in cartography, the coordinates of the known points (x_k, y_k) are corresponding to known heights h_k and the problem of interpolation is to estimate the height at some point of coordinates (x, y) on the line segments between the known points (x_k, y_k) and (x_{k+1}, y_{k+1}). Interpolators are supposed to add some meaningful information to the one already contained in the known points, else, they are useless. For example, in cartography, in a planar representation, intepolators should allow us finding an estimation of the height of the terrain at a point between two neighbor points with known altitude. The simplest intepolators are line segments between two close (successive) known points and, in R^3, planar surfaces, e.g., triangles between three close points (nearest neighbors).

There are many classes of interpolating functions, beyond line segments and planar surfaces [10]. Some of the interpolating functions behave in an undesirable way, taking values far outside the interval delimited by the the interpolation points. Such a case is the class of polynomial interpolators.

Efficient interpolators have been continuously proposed and refined during the last two centuries, e.g. [28], [27], [5], [19]. This article shows that FLSs and, more generaly, t-LSs may offer an easy way of automatically building interpolators that have a set of desirable properties. Formally, the fact that FLSs can be interpolators is clear from the interpretation of a set of points in R^n, $\{(x_{k1}, \cdots, x_{k,n})\}_{k=1\cdots n} \in R^n$ as a Sugeno-type FLS assigning the singleton $(x_{kn}, 1)$ to the singletons $((x_{k1}, \cdots, x_{k,n-1}), 1)$. This construction is unsatisfactory and trivial because the function defined by the FLS with defuzzification is discontinuous at the interpolation points and is zero between these points.

Although the problems of interpolation and approximation are distinct, for practical reasons, the interpolation points are frequently points on the graph of a function; then, an interpolating function should also be a reasonable approximation of that function. Typical applications where that occurs are data visualization; in these cases, the measurement points serve for tracing surfaces and curves. The "good approximation" property of the interpolation function is observed in this article.

2 Types of functions considered for interpolation and desirable properties as approximators

The functions interpolated (and approximated) are considered to be real valued functions that are 'well behaved,' meaning that they are defined on a bounded closed interval $J \subset R^n$, are continuous except a finite number of discontinuities, all discontinuities being bounded jumps, and have a finite number of subintervals of monotony on J. Because they are continuous on a finite number of intervals, on those intervals the function is bounded. Moreover, we are ensured that the function is bounded on the definition interval, as it is on every subinterval and all jumps at the boundary of the subintervals are with bounded jumps. In other words, the function is defined on $\bigcup_{h=1}^{k} J_h$ where $J_h \subset J \subset R^n$ and J_h forms a partition of J, $J_h \cap J_{i \neq j} = \phi$, $\cup_{h=1}^{k} J_h = J$. On every J_h, except a finite number of points on their boundaries, the function has no discontinuities. The assumption of monotony on each subinterval is a technical one, allowing us to simplify the discussion; it can be removed without consequences for the results.

A desirable property of the interpolators is that, between the interpolating points, the interpolators have guaranteed "small", "constrained" variations, meaning that their variation is bounded by the values in the interpolation points. Recall that not all classes of interpolators have this property; for example, polynomial interpolants do not have it.

When discussing the interpolation problem, points of essential discontinuities of the interpolated function may become points of continuity of the interpolator, such that the interpolator is a continuous function on the entire definition interval. For example, polynomial interpolators behave in this way. However, there are cases when the interpolation points belong to a curve or surface with (bounded) jump-type discontinuities; in this situation, one may need interpolating functions with discontinuities. This issue is treated as a distinct case.

3 Why the FLS can interpolate

For an easy introduction of the topic, simple examples are shown for interpolators with FLSs in the next subsection.

3.1 Simple interpolators with fuzzy logic systems for points of continuous functions

We assume min-max fuzzy logic systems on an interval of R. The case of interpolated functions $f : J \subset R \to R$ and $f : J \subset R^2 \to R$, where J is a closed, bounded interval are the simplest; we also assume that f is continuous except a finite number of discontinuities, moreover that f does not have infinite oscillations on J; that means that for any value $a \in f(J \in R)$, there is a finite number of values $x \in J$ such that $f(x) = a$. For example, functions that behave similarly with $f(x) = \frac{sin(1/x)}{1/x}$ are not allowed. (As said, this assumption is only technical, for ease of discussion).

Solutions of interpolation with FLSs is graphically explained in Figure 1 . One of the possible solutions uses triangular membership functions (m.f.s), e.g., m.f.s overlapping two by two, with the condition that the points where the m.f.s defined on Ox (input) have value 1 are the abscissas of the points of interpolation; moreover, the m.f.s defined on Oy, including singletons, have the same property, having value 1 on the ordinates of the interpolation points. For Sugeno (also named Takagi, Sugeno, Kang, TSK) FLSs, the singletons are positioned at the ordinates of the interpolation points. The triangular m.f.s for the independent variable x were chosen for their simplicity; in fact, any m.f. type with a single point where it has value 1 is acceptable. The respective point will be named apex. A decisive demand for the simplicity of computations is that the m.f.s intersect at most two by two, such that, in the abscissas of the interpolation points, a single m.f. has non-zero value. This ensures that exactly a single m.f. is fired at any interpolation point. The condition is not, however, a necessary one. Similar considerations are used for Mamdani FLSs. (We write Mamdani, instead of Mandani; both variants are used and in some papers appear both variants, e.g., in [6].)

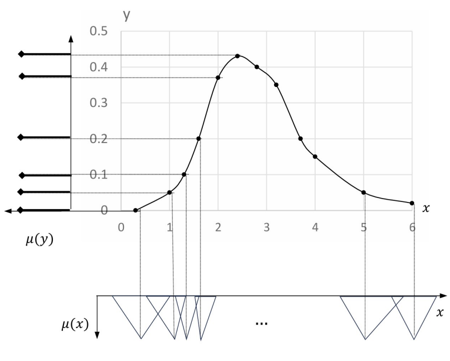

Figure 1: Sugeno type FLS interpolating an empirical distribution function (based on Fig. 1 in [9])

When x corresponds to an interpolation point, $(x_k, y_k) \in R^2$, as in Figure 1, only one singleton represents the output of the FLS; therefore, for $x = x_k$, $y = y_k$, and the system satisfies the interpolation condition. Importantly, the output value of the FLS with center of gravity (c.o.g.) defuzzification always remains in the interval $[y_k, y_{k+1}]$ when $x \in [x_k, x_{k+1}]$. This is due to the condition that the input m.f.s are overlapping at most two by two.

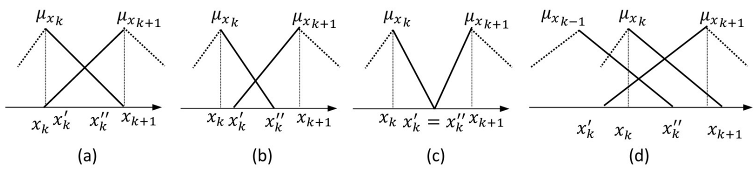

Figure 2: Details of the cases of input membership functions for Sugeno type FLS

We say that a set of m.f.s, each having an apex (a single point where they attain the value 1), $\{\mu_k : R \to R\}_k$, with apexes x_{v_k}, overlap (at most) two by two when (i) $\forall h \nexists j \neq h$ such that (s.t.) $\mu_j(x_{v_h}) \neq 0$; (ii) for any three different indices $h \neq j \neq k \neq h$, $\nexists x$ s.t. $\mu_h(x) \cdot \mu_j(x) \cdot \mu_k(x) \neq 0$. When $\forall h \exists j$ s.t. in any vicinity V of x_{v_h}, the condition $\mu_j(x_{v_h}) \neq 0$, we say that the m.f.s exactly (regularly) overlap (two by two). The definitions do not require a specific shape of the m.f.s. These definitions are directly expandable to m.f.s in R^n, but the condition (ii) becomes more laborious.

For computational reasons, it is desirable that, except the interpolation points, there is no point $x \in D$ where a single m.f. has non-zero values, as in Figure 2 (a), in $[x_k, x_{k+1}]$. Then, the intepolating function is a piecewise linear functions with line segments defined by the interpolation points. In this case, the use of a FLS for interpolation brings nothing but increased computation compared with the straightforward linear piece-wise interpolator. The case in Figure 2 (b) may be more interesting. On the intervals $[x'_k, x''_{k+1}]$ the defuzzified output has the form

$$y(x) = \frac{\beta_k \mu_k(x) + \beta_{k+1}\mu_{k+1}}{\mu_k(x) + \mu_{k+1}(x)}.$$

As $\mu_k(x), \mu_{k+1}$ are linear functions, the denominator and the nominator are linear expressions; therefore, on the interval $[x'_k, x''_{k+1}]$, the interpolator is a rational function. The respective rational function may be more suitable in certain applications than a linear segment; this may justify the use of this type of interpolator. However, on the intervals $[x_k, x'_k]$ and $[x''_k, x_{k+1}]$, the defuzzified output is a constant, equal respectively with β_k and β_{k+1}, which may be inconvenient in applications. The worst case is shown in Figure 2 (c), where the output is constant in the intervals $[x_k, x'_k)$ and $(x'_k, x_{k+1}]$ and is undefined at $x'_k = x''_{k+1}$. A more interesting case in applications is shown in Figure 2 (d), where three m.f.s intersect at every x_k except the first and the last. In this case, however, more computation is required. Specifically, excepting the first and the last interpolation points, a system of equations has to be solved to find the interpolator,

$$\begin{cases} \dots \\ \frac{\beta_{k-1}\mu_{k-1}(x_k)+\beta_k\mu_k(x_k)+\beta_{k+1}\mu_{k+1}(x_k)}{\mu_{k-1}(x_k)+\mu_k(x_k)+\mu_{k+1}(x_k)} = y_k \quad x = x_k \ . \\ \dots \end{cases}$$

One can solve the equation system by choosing a first m.f., μ_1, then determine μ_2 from the condition at x_1, $\dfrac{\mu_1(x_1)\beta_1 + \mu_2(x_1)\beta_2}{\mu_1(x_1) + \mu_2(x_1)} = y_1$. This condition determines the left vortex of μ_2. Then, one continues solving the equations from the system up to the last one.

In case of m.f.s with non-triangular shapes, between the interpolation points, the defuzzified output is a fraction with the nominator and the denominator represented by weighted sums of the respective of functions. When three or more m.f.s are allowed to overlap, the computations may become difficult, or may require approximate computations, ruining the aim of exact interpolation.

Property 1. A TSK interpolating FLS in the plane with input triangular m.f.s that overlap two by two is an interpolating function that has values in the interval $[y_k, y_{k+1}]$ on any interval between two successive interpolation points x_k, x_{k+1}.

The proof follows directly from the above description. More generally:

Property 2. Consider a TSK FLS on the line, with all m.f.s continuous, with values larger than 0 on an interval $J'' \subset J \subset R$ (with the closure of J'' denoted by $J' \subset J$), having a single apex, overlapping two by two. Then,

(i) the TSK with output singletons y_h corresponding to the m.f.s μ_h and with c.o.g. defuzzification interpolates through the set of points $\{(x_{vh}, y_h)\}_h$;

(ii) assume $x_{v_h} < x_{v_{h+1}}$; then, $\forall h, \forall x \in [x_{vh}, x_{vh+1}] : min(x_{vh}, x_{vh+1} \leq y(x) \leq max(x_{vh}, x_{vh+1})$; that is, the values of the defuzzified FLS remains between the interpolated points for two successive such points. Similarly for $x_{v_h} > x_{v_{h+1}}$

Proof. As a single m.f. is not null in a point representing the apex, (i) is true. Between two successive apexes only two m.f.s have non-null values; for a given $x \in [x_{vh}, x_{v(h+1)}]$ denote for easy notations $a = \mu_h(x), b = \mu_{h+1}(x)$. Then, the defuzzified output is $\frac{ay_h+by_{h+1}}{(a+b)}$. Assume $y_h \leq y_{h+1}$. Then, as $a, b \geq 0$, $\frac{ay_h+by_{h+1}}{(a+b)} \leq \frac{ay_{h+1}+by_{h+1}}{(a+b)} = y_{h+1}$. Similarly, $\frac{ay_h+by_{h+1}}{(a+b)} \geq y_h$.

Property 2 is useful in applications because it ensures us that TSK interpolators on the line behave also as approximators. Consider a function $f : J \to R$ with the stated properties (continuous, with a finite number of oscillations) and an error ϵ desired in approximation. Denote by $[m, M] = f(J)$. Build the subintervals $[m + k\epsilon, m + (k + 1)\epsilon]$, $k = 1, \cdots, \lceil \frac{(M-m)}{\epsilon} \rceil$. Then, use the set of points $\{(x_k = m + k\epsilon, y_k = y(m + k\epsilon))\}_k$ as interpolation points. Between to such points, we are guaranteed that the function remains at a distance less then ϵ from the respective points of interpolation; therefore, the approximation by the interpolating TSK is guaranteed, according to Property 2. As TSK FLSs are universal interpolators, Property 2 proves that TSK FLSs are universal approximators for functions $f : J \subset R \to R$, which is a proof of the theorem:

Property 3. TSK FLS are universal (exact) interpolators and uniform approximators for continuous functions $f : R \to R$ defined on bounded intervals.

The property #3 results directly from the first property and the above discussion. A similar proof shows that Mamdani FLSs are both universal (exact) interpolators and uniform approximators for continuous functions $f : R \to R$ defined on bounded intervals. Property #3 shows that FLSs are dense in the set of continuous functions on bounded intervals in R.

This property is meaningful in that, when the interpolation points are chosen such that the interpolated function (assuming the point are on an interpolated function) remains monotonic between successive interpolation points, the interpolator does not go outside the interval of values of the interpolated function. This is a desirable property for interpolators.

The above discussion shows why the case of Mamdani type FLS interpolators may be worth analyzing. That is because the form of the interpolators based on TSK systems, between the interpolation points, is always given by a ratio of sums of the input m.f.s., which may be a too simple function for the application treated.

An example of Mamdani-type interpolant in the plane is shown in Figure 3, where triangular m.f.s are used at the input and at the output. The interpolator is built in a way similar with that for TSK FLSs. The easiest construction is to choose the vertices of the input triangular m.f.s at the point x_k and the vertices of the output m.f.s at the values y_k, with the input m.f.s overlapping precisely two by two, and similarly for the output m.f.s. Easily extending the above Properties, one derives that Mamdani FLSs with triangular m.f.s are universal interpolators and approximators. The discussion in this Section was partly presented in [23], [24], [25], [26], under different frameworks.

Computational aspects. For the interpolation of a set of n points in the plane, $\{(x_k, y_k)\}_{k=1,\cdots,n}$, according to the method shown in this Section, one needs n m.f.s on Ox and n singletons, for the case of TSK FLSs, respectively a set of n m.f.s on Ox and n m.f.s on Oy for Mamdani FLSs. Similarly, for the interpolation of a set of n points in the R^m, one needs $n \times m$ m.f.s. The definition of $n \times m$ m.f.s overlapping exactly two by two can be done automatically, yet it is cumbersome. The number of m.f.s required for the interpolator can be reduced to $2n$ by defining $(n - 1)$-dimensional m.f.s for the input of the FLS and n one-dimensional m.f.s for the output of the FLS. An example for interpolating in R^3 is shown in Figure 4, where one replaces the one-dimensional (1D) m.f.s, $\mu : R \to [0, 1]$ with two-dimensional (2D) ones, $\mu : R^2 \to [0, 1]$.

While we discussed min-max logic FLSs, the extension to max-product fuzzy logic systems is immediate. All the above considerations, including the three properties, can easily be extended, with a few details changed, to Mamdani FLSs.

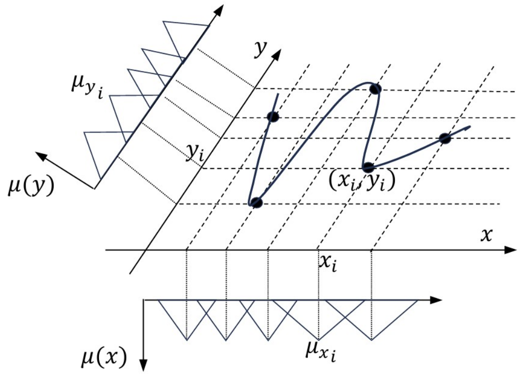

Figure 3: Mamdani type FLS interpolating through a random set of points in the plane.

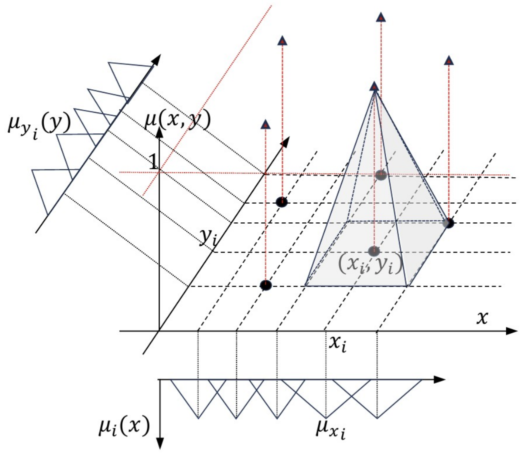

Figure 4: Mamdani type FLS interpolating a random set of points in the R^3. Only the projections of the points on the plane of the independent variables (inputs) and the related membership functions are shown. The vertical arrows show the interpolation points, where the input m.f.s must have value 1. Combining 1D triangular m.f.s into a pyramidal 2D m.f. simplifies the description of the FLS.

Being both universal interpolators and approximators, FLSs are powerful tools, as are the neural networks, various classes of polynomials, and trigonometric polynomials. Among others, they can model probability density functions, as in Fig. 1, which is a first step toward the use of FLSs in many AI applications.

3.2 Simple interpolators with FLSs for functions with jump discontinuity points

As an example of application where such functions may be of interest consider the cartography problem of representing a terrain with cliffs. Thereafter, we discuss the case of interpolation with FLSs jump discontinuities, in relation with Figure 5. The example in Figure 5 (a) uses Sugeno FLSs with c.o.g. defuzzification, but either Sugeno or Mamdani FLSs can be equally used. (In case of Mamdani FLSs, we assume that the output m.f.s are isosceles triangular, overlapping two by two.) In Figure 5 (a), the FLS output is a step function, with a jump discontinuity; for $x \leq x_i$, $y = \beta_i$; for $x > x_i$, $y = \beta_{i+1}$. Only two singletons and a single jump are shown. The discontinuity in Figure 5 (a) does not disappear if μ_k and μ_{k+1} overlap, but the output is no more a step. The example in Figure 5 (b) is more useful in applications, as a variety of shapes of the function can be obtained close to the discontinuity. Figure 5 (b) illustrates two cases, one TSK system and one Mamdani system (superposed). For the value at the right of the discontinuity is as specified in Figure 5 (b), μ_{k+2} with TSK system, the left vertex should be slightly larger than x_i at the discontinuity (the vertex coordinate should be $x_i + \epsilon$, which creates a small interval of length ϵ where the function has a constant value. The step of width ϵ can be avoided by replacing the triangular m.f. μ_{i+2} with a function that has at x_i a horizontal asymptote, for example $y(x) = 1 - \sqrt{1-(x-x_0)^2}$. In Figure 5 (b), for the case of the TSK system, the graph at left of x_i is given by an expression with the form $y(x) = \dfrac{\beta_{i-1}\cdot(a_{i-1}+b_{i-1}x)+\beta_i\cdot(a_i+b_ix)}{a_{i-1}+b_{i-1}x+a_i+b_ix}$, where $\mu_{i-1}(x) = a_{i-1}+b_{i-1}x$, with the constants a_{i-1}, b_{i-1} defining the triangular m.f.s at the left of x_0, and similarly for μ_i. The rational function $y(x)$ at the left of x_0 allows for an approximation of the interpolated function. A similar argument stands for the case at the right of x_0. Using Mamdani FLSs, see the doted triangular m.f.s in Figure 5 (b), may help improve the approximation of the interpolated function, as the input-output function of such a FLS is a fraction with a polynomial of degree 3 at the nominator and a polynomial of degree 2 at the denominator, when the input and output m.f.s are piecewise [23], [26].

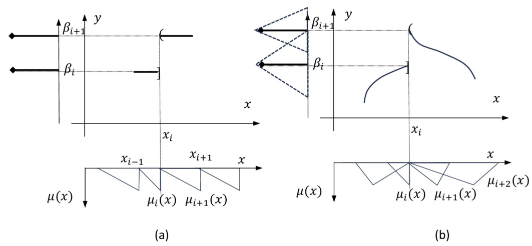

Figure 5: Discontinuities with FLSs at an interpolation point.

The generalization to the n-dimensional case of the interpolation with discontinuities is straightforward. Also, the algorithms for automatically building interpolators for a set of given points is straightforward.

Remark. As FLSs can produce step-wise interpolators; therefore, as stepwise functions are universal approximators (Weierstass approximation theorem, starting from stepwise functions), FLSs are universal approximators of continuous functions $C : R^n \to R$ defined on a closed bounded interval in R^n. Therefore, the set of FLSs is dense in the set C.

This is maybe the simplest proof of a result given in [22], based on Weierstrass theorem, and independently in [3], with further analysis in [12].

3.3 The n-dimensional case

The n-dimensional space construction is similar to the 1-dimensional construction, but it is more elaborate and there is no simple visualization.

Consider a set of interpolation points $Q_j = \{(x_j, y_j)\}_{j=1,...,m}$, $x_j = (x_{j1}, \ldots, x_{j(n-1)}) \in R^{n-1}$, and $y_j \in R$. For ease of explanation only, assume that no two interpolation points have the same coordinate values for any coordinate, that is, there are no $j_1 \neq j_2$ and no k such that $x_{j_1k} = x_{j_2k}$. (There is no consequence of this assumption for the proof, but the proof has to consider more special cases without this assumption.) Then, one can construct membership functions for each coordinate x_jk as discussed in relation with Figure 4 and choose either singletons or other type of m.f.s, with the output m.f.s symmetric and centered on y_j; the obtained FLS is guaranteed to interpolate the given points, without jumps.

For the case of R^2, if one accepts that the interpolator is not defined on polygonal lines and that a stepwise function is acceptable as interpolator, then instead of using the rectangles defined in Figure 4, one can use the polygonal regions defined by the Voronoi diagram [1], [15], [18] for those points and assign to each region an output singleton corresponding to the y coordinate value of the point defining that region in the Voronoi diagram. This method, however, makes more difficult the choice of the m.f.s, considering that the domains where they are not null should intersect such that in the interpolation points, representing the apexes of the m.f.s, and in the apexes a single m.f. should be not null.

Finally, as a matter of example, it is easy to see that, for the case in Figure 4, the interpolator is also an approximator in the sense of Property 2. Precisely, the height of the surface described by the interpolators remains between the maximal and minimal values of the neighboring singletons.

4 Interpolation with t-logic systems

In this Section, a direct generalization of the interpolation problem previously discussed is given using t-norms and co-norms instead of $min - max$ fuzzy logic. The structure of t-logic systems is the same as for FLSs. The difference is that instead of using min-max, or max-product logic one uses a general t-norm and its conorm; also, instead of m.f.s, we use the slightly more general concept of distribution function.

A function defined on R^n is named distribution function (DF) or, when there is no risk of confusion, simply distribution if (i) it has values in $[0, 1]$; (ii) it is continuous except a countable set of points of discontinuity, (iii) it is Lebesgue measurable, and (iv) its measure is finite. Distribution functions can be seen as probability density functions (pdf) in the context of probabilities, logic distributions functions when in the context of systems based on logic operators, membership functions in the context of fuzzy logic systems, or belief distributions in the context of belief theory. A convenient wrapping name for DF could be "credibility distributions". Only DFs with bounded support will be discussed in this paper, without loss of generality. In addition, with some loss of generality, the DF are considered continuous. This, in turn, makes them Riemann integrable.

For ease, we consider only density functions defined on the same bounded and closed domain $D \subset R^n$; possibly, some of these functions have non-null values only on an open compact of D. When, for ease of notation, it is preferable to write expressions on an infinite domain, the density functions are extended with 0 values outside D.

A DF $f(x)$ will be named unimodal iff there is a unique value of $x \in R^n$, denoted x_v, such that $f(x) = 1$. That value of x_v will be named apex (vertex, if no confusion arises) of $f(x)$; depending on the context, the point $(x_v, 1) \in R^n \times R$ will also be named apex of the DF.

An estimator E is a function from the set of density functions, $\{\eta(x) : R^n \to [0, 1]\}$ to R_+. Here, x is considered an element in R^n. When the estimator is applied to a specified density function, one says that it estimates the variable having that density. For ease of treatment, we use only DF defined on $D \subset R$, D compact, in the remaining part of the article. An estimator E of a variable x with density $\eta(x)$, $\eta : R \to [0, 1]$ is said to be an average (or center of gravity, c.o.g.) estimator when it is defined by $E[x] = x_e = \dfrac{\int_D x\eta(x)dx}{\int_D \eta(x)dx}$. For generality, when D is extended to R, $E[x] = \int_{-\infty}^{\infty} x\eta(x)dx$. If η is unimodal with apex x_v and $x_v = x_e$, η is said to have the center at x_v.

Examples. For η defined on R, Gaussian densities $\eta(x) = 1/(\sigma\sqrt{2}\pi)e^{-(x-a)^2}/(2\sigma^2)$ are centered at a. The Gauss-like density $\eta(x) = \begin{cases} Ae^{-\frac{(x-a)^2}{2\sigma^2}} & x \in [a - b, a + b] \\ 0 & else \end{cases}$ is centered. Also, the 'triangular' density $\eta(x) = \begin{cases} (x - a)/b & if x \in [a, a + b] \\ 1 - (x - a - b)/b & if x \in [a + b, a + 2b] \\ 0 & else \end{cases}$ is centered.

Recall that a triangular norm, or t-norm is an operation $t : [0, 1]^2 \to [0, 1]$ satisfying the conditions: (i) $t(x, y) = t(y, x)\forall x, y$, commutativity; (ii) $t(x, t(y, z)) = t(t(x, y), z)$ (associativity); (iii) if $x \leq y$ then $t(x, z) \leq t(y, z)\forall z$ (non-strict monotony); (iv) $t(x, 1) = x\forall x$ (1 is neutral element for the operation represented by the t norm.) For details, see [8]. Idempotent elements satisfy the condition $t(x, x) = x$ [8]. From (iv), $t(0, 1) = 0$ and because as $0 \leq 1$, applying (iii), it results that $t(0, 0) = 0$. Also, from (iv), $t(1, 1) = 1$; thus, 0 and 1 are idempotent. The s-conorm of the t-norm is defined by $s(x, y) = 1 - t(1 - x, 1 - y)$. Conorms have are commutative, associative, monotonic, and have 0 as neutral element, $s(x, 0) = x$. Clearly, $u < 1$ implies that $s(u, v) < 1$, which is important in developing interpolators with t-logic systems (t-LSs). Subsequently, only continuous t-norms and corresponding co-norms are considered.

As said, t logic systems (t-LSs) are built with rules in a similar way with the construction of FLSs. A rule h with m premises $x_1, \ldots x_m$ and a single conclusion (outcome), y, with densities η_1,η_m of the premise variables, and with the density of the outcome of the rule denoted by $\eta_{oh}(y)$, is written as the resultant density $\eta'_{oh}(y)$ of the y variable,

$$\eta'_{oh}(y) = t(\eta_1(x_1), \ldots, \eta_m(x_m), \eta_{oh}(y)).$$

Several rules that are connected by the co-norm of t constitute a t-LS with conclusion y. A t-LS with q rules is described by the overall resultant density $\eta_o(y)$ of the y variable, at the given point $(x_1, x_2, \ldots, x_m) \in D$,

$$\eta_o(y(x_1, x_2, \ldots, x_m)) = s(\eta'_{o1}(y), \cdots, \eta'_{oh}(y), \cdots, \eta'_{oq}(y)) \tag{1}$$

where η_{oh} is the density function of the h^{th} density function in the consequence and h is the number of the rule. The construction closely follows the one of fuzzy logic systems. The estimator of $\eta_o(y)$ is the output of the system. For a specified input value $x = (x_{10}, x_{20}, \ldots, x_{m0})$, the outcome density is computed accordingly, $\eta'_o(y((x_{10}, x_{20}, \ldots, x_{m0})) = t(\eta_1(x_{10}), \ldots, \eta_m(x_{m0}), \eta_o(y))$.

When x corresponds to an apex, denoted $x = x_{v_h}$, $\eta'_{oh}(y(x)) = t(\eta_1(x_v), \ldots, \eta_m(x_v), \eta_{oh}(y)) = t(0, 0, \cdots, 1, \cdots, 0, \cdots 0, \eta_{oh}(y)) = \eta_{oh}(y)$. Then, if $\eta_{oh}(y)$ is centered (symmetrical) and the input DFs overlap precisely (i.e., when at the vertex of a DF all other DFs have value 0), the c.o.g. estimated value is y_{oh_v}. Therefore:

Remark. When the density η_o is centered, moreover the densities are regularly overlapping,

$$E[\eta'_o(y)] = E[t(\eta_1(x_{1v}), \ldots, \eta_m(x_{mv}), \eta_o(y)] = y_v \tag{2}$$

We assume that densities in equation (1) guarantee that for any $(x_1, x_2, \ldots, x_m) \in D$ the density values of the combined premises is not null, except for the contour of D. This requires in turn that for any point in D there is at least one density function with non-null value; that is, there are at least two densities functions that meaningfully overlap.

Remark. A system defined by equation (1) is an interpolator for the set of points

$$P = \{(x_{1v_{11}}, x_{2v_{21}}, \ldots, x_{mv_{m1}}; y_{1v}), \ldots, (x_{1v_{q1}}, x_{2v_{q1}}, \ldots, x_{mv_{m1}}; y_{qv})\}.$$

Monotony property - case on R. The following property is useful in the analysis of the approximation power of the t-interpolators. Consider two unimodal density functions with apexes v_1 and v_2, η_1 and η_2 on R with $x_{v_1} < x_{v_2}$, both being continuous and strictly monotonic in the interval $[v_1, v_2]$. Consider that the two density functions are regularly and meaningfully overlapping. Let x_{12} be the value of x where $\eta_1(x_{12}) = \eta_2(x_{12})$. Clearly, η_1 should be decreasing for $x > x_{v_1}$ and η_2 should be increasing for $x < x_{v_2}$. Then, $x_{12} \in (x_{v_1}, x_{v_2})$ and, because of the continuity, there is at least a vicinity of x_{12} where both densities are not null. In addition, there must be a point x_{2-} where $\eta_2(x_{2-}) = 0$ with $lim_{x \to x_{2-}, x > x_{2-}} \eta_2(x) = 0$ (due to continuity), and values of η_2 at right being positive. Similarly, there is a point x_{1+} where $\eta_1(x_{1+}) = 0$ and $\eta_1(x) > 0$ at left of it. Due to continuity and strict monotony, on the interval (x_{2-}, x_{1+}) both densities are positive, $\eta_1(x) > 0, \eta_2(x) > 0 \quad \forall x \in (x_{2-}, x_{1+}) \subset (x_{v_1}, x_{v_2})$. No other density function is non-null on the (x_{v_1}, x_{v_2}) interval.

The expression $t(\eta_1(x), \eta_2(x))$ is decreasing on the interval (x_{v_1}, x_{2-}) as $t(\eta_1(x), \eta_2(x)) = \eta_1(x)$ and $\eta_1(x)$ is decreasing. On the interval (x_{2-}, x_{12}) clearly $\eta_1(x) > \eta_2(x)$ and vice versa on (x_{12}, x_{1+}). On the interval (x_{2-}, x_{12}), $\eta_1(x)$ is decreasing and $\eta_2(x)$ increases; also, on the interval (x_{12}, x_{1+}), $\eta_1(x)$ is decreasing and $\eta_2(x)$ increases. To simplify the discussion, assume that the two densities are derivable on (x_{2-}, x_{1+}). Then,

Proposition (Interpolation). A TSK-type t-LS with average (c.o.g.) estimator with input and output logical functions $\eta_{i,o}(x) : R \to [0, 1]$ is an interpolator when the following conditions are satisfied: (i) the functions $\eta_i(x) : R \to [0, 1]$ have a single point where they attain the value 1 and that point is the abscissa of an interpolation point; (ii) the functions $\eta_o(x) : R \to [0, 1]$ have a single point where they attain the value 1 and that point is the ordinate of the corresponding interpolation point.

Then, for any finite number n of interpolation points, there is a t-LS system that is an exact interpolation and the complexity of building the interpolator is $O(n)$. Finally, one could extend the above considerations to t-logic systems based on non-commutative logics, as described in [21].

5 Conclusions

The properties of t-logic systems, including FLSs, derive from a combination of the properties of the t-norms and of the estimator operator (defuzzification), combined with the choice of the distribution (membership) functions and of their mapping. Choosing distributions that have a single mode (have an apex) and overlap without covering other apexes localize the properties of the system to the regions between neighboring apexes and greatly simplify the analysis of t-LSs. Further simplification is obtained by choosing output distributions that have their c.o.g. at the apex position. Then, it is easy to derive properties related to interpolation and approximation with t-LS systems on bounded domains.

Interpolation is essential in many applications in engineering, finance, economics, medicine and other fields. We proved that (i) standard min-max and max-product fuzzy logic systems, both Sugeno and Mandani type FLSs are universal interpolators; moreover, the interpolators are universal approximators in the sense of Property 1; (ii) the case of interpolators for points on graphs of discontinuous functions can also be covered by FLSs; (iii) the n-dimensional case was discussed; (iv) t-logic systems (TLSs), that generalize FLSs, are universal interpolators as well as universal approximators.

Finding an interpolating function is useful when the interpolator is supposed to bring information between the interpolating points, that is, to somehow extend the information contained in the interpolation points to the intermediate regions between those points. That is not guaranteed for general classes of interpolating functions, but it was shown that the property is respected for FLSs.

A general method to build interpolators with t-logic systems has been presented. The method covers as a particular case the interpolation with FLSs, with some details present in case of FLS missing in the more general case.

A discussion of the use of the presented results in data representation and visualization will be presented in another study.

Acknowledgments

Prof. Mike H.M. Teodorescu helped with writing about data representation and economic applications and on data visualization, provided references, and proposed the interpolating among various fairness criteria as an application.

Funding

No funding was received.

Conflict of interest

The author declares no conflict of interest.

References

[1] Ahuja, N. (1982). Dot pattern processing using Voronoi neighborhoods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3, 336-343.

[2] Buckley J., Hayashi Y, (1993). Fuzzy input-output controllers are universal approximators. *Fuzzy Sets and Systems* Vol 58, 3, 1993, 273-278.

[3] Buckley, J. J., Hayashi, Y. (1994). Fuzzy genetic algorithm and applications. *Fuzzy Sets and Systems*, Vol. 61, 2, 129-136, Elsevier North-Holland.

[4] Ferreira, P. J. S. G. (1994). Interpolation and the discrete Papoulis-Gerchberg algorithm, *IEEE Transactions on Signal Processing*, vol. 42, no. 10, 2596-2606, doi: 10.1109/78.324726.

[5] Franke R., Nielson G. (1990). Scattered data interpolation and applications, in *Geometric Modelling: Methods and Their Applications*, H. Hagen, D. Roller (eds.), Springer, 1990, 131-160.

[6] Jassbi, J. J., Serra P.J.A., Ribeiro R.A., Donati A. (2006). A Comparison of Mandani and Sugeno Inference Systems for a Space Fault Detection Application, *Proc. World Automation Congress*, Budapest, Hungary, 2006, 1-8, doi: 10.1109/WAC.2006.376033.

[7] Judd KL, Maliar L, Maliar S, Valero R. Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *National Bureau of Economic Research*; 2013 Aug 16. https://www.nber.org/system/files/working_papers/w19326/w19326.pdf

[8] Klement E.P., Mesiar R., Pap E. (2004). Triangular norms. Position paper III: continuous t-norms. *Fuzzy Sets and Systems* 145, 439–454.

[9] Kuhn, J., Teodorescu, M.H.M. (2021). The track one pilot program: Who benefits from prioritized patent examination? *Strategic Entrepreneurship Journal*, vol. 15, 2, 185-208.

[10] Lam, N. S. N. (1983). Spatial Interpolation Methods: A Review. *The American Cartographer*, 10, 2, 129–150. https://doi.org/10.1559/152304083783914958.

[11] Lee M. A., Esbensen, H. (1997). Fuzzy multi-objective genetic systems for intelligent systems design tools and components. In *Fuzzy Evolutionary Computation*,57-80. Kluwer, Boston, MA, 1997. http://neo.lcc.uma.es/emoo/lee97.ps.gz (accesed Aug. 2007).

[12] Liu, P. (2001). Universal approximations of continuous fuzzy-valued functions by multi-layer regular fuzzy neural networks. Fuzzy Sets and Systems, Vol 119, 2, 2001, 313-320.

[13] Martin-Bautista, M.J., Vila, M.-A., Larsen, H. (1999). A fuzzy genetic algorithm approach to an adaptive information retrieval agent. *Journal of the American Society for Information Science*, vol. 50, 9, 760-71.

[14] Mishra, S.R., Mohapatra, H., Saxena, S. (2024). Leveraging Data Analytics and a Deep Learning Framework for Advancements in Image Super-Resolution Techniques: From Classic Interpolation to Cutting-Edge Approaches. In *Data Analytics and Machine Learning. Studies in Big Data*, vol 145. Springer, Singapore. https://doi.org/10.1007/978-981-97-0448-4_6.

[15] Mumm, M. (2004). Voronoi diagrams. *The Mathematics Enthusiast*, 1, 2, 44-55.

[16] Oliveira Bastosde, A., et al. (2023). Visual analytics for diagnosing spatial interpolation models. *Transactions in GIS*, 27, 364–387. https://doi.org/10.1111/tgis.13028.

[17] Pesenson, M.Z., Suram, S.K., Gregoire, J.M. (2015). Statistical Analysis and Interpolation of Compositional Data in Materials Science. *ACS Combinatorial Science. ACS Comb. Sci.* 17, 2, 130–136, doi: 10.1021/co5001458.

[18] Polianskii, V. and Pokorny, F.T. (2020). Voronoi graph traversal in high dimensions with applications to topological data analysis and piecewise linear interpolation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2154-2164.

[19] Scheuerer, M., Schaback R., Schlather M. (2013). Interpolation of spatial data – A stochastic or a deterministic problem?. *European Journal of Applied Mathematics*, 24, 601-629. doi:10.1017/S0956792513000016.

[20] Shu, H. (2016). Big data analytics: six techniques. *Geo-Spatial Information Science*, 19, 2, 119–128. https://doi.org/10.1080/10095020.2016.1182307.

[21] Teodorescu, M.H.M., Teodorescu H.N.L. (2021). Noncommutative logic systems with applications in management and engineering. 2021 *International Journal of Computers Communications & Control*, Vol. 16, 1.

[22] Teodorescu, H. N. (1991). Fuzzy systems with defuzzification: the equivalent crisp systems. In: E.P. Klement and H.N. Teodorescu (Eds.), Fuzzy Systems in Engineering, 95-104. Polytechnic Institute of Iasi, Romania, and Johanes Kepler University of Linz, Austria. Iasi, 1991.

[23] Teodorescu, H. N. (2007). Sisteme Nuantate (Fuzzy) si Soft Computing – Compendium, Editura Politehnium, Iasi, 2007.

[24] Teodorescu, H. -N. (2010). Fuzzy logic systems as approximators - where do we stand today?, 4th *International Workshop on Soft Computing Applications*, Arad, Romania, 2010, 25-25, doi: 10.1109/SOFA.2010.5565629.

[25] Teodorescu, H. -N. L. (2017). Fuzzy logic system linearization for sensors, 2017 *International Symposium on Signals, Circuits and Systems (ISSCS)*, Iasi, Romania, 2017, 1-4, doi: 10.1109/ISSCS.2017.8034892.

[26] Teodorescu, H.N. (2022). Why fuzzy logic systems (FLS) are difficult to tame: General FLS are not expressed by algebraic operations. *Workshop on Intelligent Information Systems*, 8-13, https://ibn.idsi.md/vizualizare_articol/165834.

[27] Tong, W., Li, L., Zhou, X. et al. (2019). Efficient spatiotemporal interpolation with spark machine learning. *Earth Sci Inform* 12, 87–96 (2019). https://doi.org/10.1007/s12145-018-0364-4.

[28] J. Van Deun and A. Bultheel. (2004). An interpolation algorithm for orthogonal rational functions. *J. Comput. App Math.* 164-165, 749-762, 2004.

[29] Zhang L., Wu, X. (2006). An edge-guided image interpolation algorithm via directional filtering and data fusion, *IEEE Trans. Image Processing*, vol. 15, 8, 2226-2238, 2006.

Teodorescu H.N. (2026). Interpolation and Approximation with Fuzzy Logic and t-Logic Systems, *International Journal of Computers Communications & Control*, 21(3), 7526, 2026.
https://doi.org/10.15837/ijccc.2026.3. 7526