

# MONOTONE APPROXIMATION OF AGGREGATION OPERATORS USING LEAST SQUARES SPLINES

G. BELIAKOV

*School of Information Technology  
Deakin University, 221 Burwood Hwy, Burwood 3125, Australia  
e-mail: gleb@deakin.edu.au*

Received

Revised

The need for monotone approximation of scattered data often arises in many problems of regression, when the monotonicity is semantically important. One such domain is fuzzy set theory, where membership functions and aggregation operators are order preserving. Least squares polynomial splines provide great flexibility when modeling non-linear functions, but may fail to be monotone. Linear restrictions on spline coefficients provide necessary and sufficient conditions for spline monotonicity. The basis for splines is selected in such a way that these restrictions take an especially simple form. The resulting non-negative least squares problem can be solved by a variety of standard proven techniques. Additional interpolation requirements can also be imposed in the same framework. The method is applied to fuzzy systems, where membership functions and aggregation operators are constructed from empirical data.

*Keywords:* Least squares spline; Monotone approximation; Fuzzy sets; Aggregation operators.

## 1. Introduction

In fuzzy set theory, estimation of membership functions and aggregation operators from empirical data is frequently required.<sup>1-3</sup> Parametrical techniques used for this purpose often do not provide semantical interpretation of the parameters, and are not as versatile as non-parametrical approximation. A non-parametrical technique which offers great flexibility in approximating functions of various shapes is that of polynomial splines. Splines are well studied, easy to compute, and they possess a range of other properties that make them most suitable for the problem of approximation of empirical data. They

have been applied to fuzzy logic by several authors,<sup>4-6</sup> because of their flexibility to model complex one- and multidimensional shapes, that arise in fuzzy systems. However, there is one major obstacle which prevents one from using standard spline algorithms in some problems in fuzzy logic. It is the fact that besides approximating well the data, splines are required to preserve certain important semantical properties of membership functions and aggregation operators, namely their monotonicity. Traditional B-splines fail to do so even in the case when the data is monotone, leaving along the case of noisy measurements.

A way of preserving monotonicity is to use the constrained splines. Constrained splines first appeared in the context of eliminating extraneous inflection points, and rapidly became popular specifically in computer-aided design. For scattered data approximation, which is the case in empirical measurements, especially in multidimensional case, least squares splines provide the adequate framework, but constrained least squares splines are not well studied. Our paper fills this gap and describes a general method of monotone least squares spline approximation in one and many variables.

Univariate monotone splines are suitable for approximation of membership functions, that have to be monotone or univariate. Other applications include monotone transformation, dose-response and growth curves.<sup>7,8</sup> For aggregation operators, multivariate monotone splines have to be used. However, certain classes of aggregation operators (such as Archimedian triangular norms and conorms) allow one to take advantage of their additive generators, which are monotone univariate functions, and this simplifies the problem significantly.

This paper details an algorithm for least squares monotone spline approximation, which involves very simple restrictions on spline coefficients. It is based on a particular representation of polynomial splines in a specially selected basis. Simple form of constraints allows one to use a range of proven techniques. Standard non-negative least squared algorithms<sup>9-11</sup> can be used without any modification. Some algorithms<sup>9</sup> allow one to specify additional interpolation conditions, and thus to incorporate geometrical restrictions on aggregation operators.

The next section introduces the problem of approximating membership functions and aggregation operators in fuzzy logic. Then we will give a brief overview of constrained splines, and then discuss the bases for polynomial splines. It will be shown why the restrictions on spline coefficients take such a simple form. Next, the method will be generalized to approximation of data in  $n$  dimensions using tensor-product splines. Finally the algorithm and results of its application in fuzzy logic will be presented.

## **2. Membership Functions and Aggregation Operators**

Fuzzy set theory (FST) and fuzzy logic are popular nowadays because of numerous engineering applications. FST is based on the notion of partial grades of membership of elements of the sets.<sup>12</sup> For convenience, the grades of membership are drawn from the

unit interval. Membership functions map the universe of discourse to the unit interval by associating each element with its membership value in a set; they are smooth analogues of characteristic functions of classical crisp sets.

Proper definition of membership functions of fuzzy sets is important. Traditionally, triangular, trapezoidal, bell-shape or S-shape functions are used in applications because of their simplicity, and usually they are defined *ad hoc*. Some methods build on the semantics of the membership values and their relation to probability theory,<sup>13, 14</sup> while others attempt to measure membership values experimentally.<sup>1-3, 15</sup> The latter approach requires fitting the empirical data to a selected model preserving the semantics of membership functions, which translates into monotonicity.

When two or more fuzzy sets are combined (as conjunction, disjunction, or their combination), the membership in the resulting fuzzy set has to be calculated from the membership values in the sets-components. This is done with the help of aggregation operators.

Initially, only two operators have been employed, *min* for conjunction and *max* for disjunction.<sup>12</sup> They possess nice theoretical properties,<sup>3, 16, 17</sup> and coincide with boolean operations in the limiting case of crisp sets. From the point of view of applications, *max* and *min* did not prove very helpful in modeling many problems, notably in mimicking human decision-making. These operators lack so called compensatory properties,<sup>2, 3</sup> which allow one to offset the low value of one attribute by high value of another.

Among other aggregation operators, triangular norms and conorms play a prominent role.<sup>17, 18</sup> These are real functions  $T$  and  $C$  that map the unit square into the unit interval, with the properties:

1.  $T(a,b)=T(b,a)$ ;  $C(a,b)=C(b,a)$  (commutativity);
2.  $T(a,T(b,c))=T(T(a,b),c)$ ;  $C(a,C(b,c))=C(C(a,b),c)$  (associativity);
3.  $T(a,b)\leq T(c,d)$ ;  $C(a,b)\leq C(c,d)$ , if  $a\leq c$  and  $b\leq d$  (monotonicity);
4.  $T(a,1)=a$ ;  $C(a,0)=a$  (boundary conditions).

The associativity property allows one to extend the domain of  $T$  and  $C$  to  $[0,1]^n$ . Associativity of triangular norms and conorms allows one to define and use their additive generators. The additive generator of the (Archimedean) triangular norm  $T(x,y)$  or conorm  $C(x,y)$  is a monotone function  $g: [0,1]\rightarrow[0,\infty]$ ,  $g(1)=0$  ( $g(0)=0$  for conorms). The operators can be obtained as

$$T(a,b)=g^{[-1]}(g(a)+g(b)) \text{ or } C(a,b)=g^{[-1]}(g(a)+g(b)),$$

where  $g^{[-1]}$  is the pseudoinverse function.<sup>18</sup> The function  $g$  is defined up to an arbitrary positive multiplier. If  $g(0)=1$  (or  $g(1)=1$ ), the triangular norm (conorm) is strict.

Besides triangular norms and conorms, there are averaging operators, compensatory operators, ordered weighted aggregation<sup>17, 19, 20</sup> and many others. All aggregation operators serve the same purpose, to combine membership values in the set-components into one value, and all belong to the set of general aggregation operators,<sup>16</sup> the real functions  $f: [0,1]^n \rightarrow [0,1]$ ,  $f(\mathbf{0})=0$ ,  $f(\mathbf{1})=1$ , non-decreasing in all arguments. The number or character in bold denotes an  $n$ -vector. The monotonicity of aggregation operators is semantically important: it guarantees, for instance, that if applied to a

decision problem, the solution inferior in respect to one parameter (while other parameters are equal) is not selected (order preservation).

The task of selecting the aggregation operator appropriate for a given problem is not trivial. Given that semantically all operators are equivalent, and that theoretically none is better than the others,<sup>21</sup> there are no dominating criteria that would force one to choose one or another operator. The adaptability of the operators and their good fit to empirical data - human responses to the same situation, become the most important criteria.<sup>3</sup>

Spline functions possess both of these qualities. Given empirical data, the set of points in  $[0,1]^{n+1}$ , an approximation to the aggregation operator can be constructed using tensor-product splines. The degree of smoothness is not essential for aggregation operators (in fact many of them belong to  $C^0$ ), and the data may contain significant errors. What is important, though, is the monotonicity of the operators, and their compliance with the boundary conditions  $f(\mathbf{0})=0, f(\mathbf{1})=1$ . Besides, some other restrictions, such as  $f(0,x)=x$ , may also be imposed. The construction of the spline must allow for such restrictions.

Splines are also well suited for approximation of membership functions based on empirical data.<sup>1, 13</sup> They are flexible enough to capture peculiarities of membership functions while providing good data fit, and in fact have been used in FST for some time<sup>4-6</sup> (triangular, trapezoidal, some S-shape functions are nothing but polynomial splines). The semantics of grades of membership impose some restrictions on the approximating splines, in particular on their range (unit interval) and shape (monotonicity). Membership functions of simple sets are monotone or unimodal, which translates into monotonicity on two intervals. Therefore, monotone splines must be employed.

### 3. Constrained Splines

Interpolation and approximation with spline functions under monotonicity and convexity constraints have attracted substantial interest in the literature, specifically in computer aided design.<sup>8, 22-28</sup> For certain sets of data, interpolating polynomial splines introduce extraneous inflection points,<sup>29</sup> whereas in the problems of smoothing the data itself may not possess the characteristics required from the approximating function due to observation errors. The desired characteristics of monotonicity or convexity of the interpolating or approximating splines can be enforced by various methods. One such method, due to Schweikert and generalized by Spath,<sup>29</sup> is based on exponential piecewise approximation. Polynomial interpolating and smoothing constrained splines have been extensively studied.<sup>8, 22, 24-30</sup> They are based on introduction of additional interpolation knots or restrictions on smoothness of the spline. Several algorithms for monotone spline interpolation are available. The results have been extended for bi-variate interpolation as well.<sup>8, 22, 24-28, 31</sup>

For multivariate case the problem becomes more difficult because of the complexity of the formulas, and because the data is frequently scattered, rather than given on a rectangular grid. One way to approximate the data is to use Powell-Sabin splines,

possibly under monotonicity constraints.<sup>31, 32</sup> However its extension from bi-variate to multivariate case seems to be quite complicated. Variational approach allows one to approximate scattered data with thin plate splines in multidimensional case. It involves a quadratic programming problem with inequality and equality constraints using a dual-type iterative algorithm, with relatively high computational cost.

A somewhat different approach to spline approximation, advocated by P. Dierckx,<sup>32</sup> is to use least squares splines. The approximation knots do not coincide with the data in this case, and usually the number of spline segments is less than the number of data points. The coefficients of the spline are found as a solution to a linear least squares problem. When the knots of approximation coincide with the data, the least squares spline becomes the usual interpolating spline with the appropriate conditions imposed on the derivatives at the ends of the interpolation interval.

Least squares splines are easily extended for multivariate case using tensor products of univariate splines.<sup>7, 29, 31, 32</sup> The data can be given on a rectangular mesh or scattered, and the number of variables does not pose significant restrictions. It is also possible to impose monotonicity or convexity conditions on the least squares splines. In the univariate case these conditions were discussed in Dierckx.<sup>32</sup> Less attention has been given to shape preserving multivariate spline approximation. The next sections describe a general technique of univariate and multivariate monotone spline approximation based on specific basis functions.

#### 4. Basis for Splines

Initially, we consider the problem of monotone least squares splines in one dimension, suitable for approximating membership functions. Suppose, there is a given set of data points  $\{(x_i, y_i)\}_{i=1}^I$ , and a prescribed set of knots  $\{t_j\}_{j=-1}^{N+2}$ . The least squares linear spline  $S(x)$  is a piecewise linear function which minimizes the least squares criterion

$$\sum_{i=1}^I (S(x_i) - y_i)^2 .$$

The knots  $\{t_j\}$  are the abscissas of the points where the linear segments are joined together. The knots 0 and  $N+1$  correspond to the ends of the interval ( $[0,1]$  in our case), and the knots  $-1$  and  $N+2$ , possibly coinciding with  $t_0$  and  $t_{N+1}$ , form the extended partition. This indexing is taken from Dierckx.<sup>32</sup>

It is convenient to represent the spline  $S(x)$  as a linear combination of  $N+2$   $B$ -splines of order 2:

$$S(x) = \sum_{j=-1}^N a_j N_j^2(x) .$$

The functions  $N_j^2(x)$  are well known in the literature,<sup>29, 32, 33</sup> and they possess many useful properties, including linear independence, local support, partition of unity, numerical stability, etc. They form the basis in the  $(N+2)$ -dimensional space of polynomial splines of order 2 with the knots  $\{t_j\}$ . The problem of least squares

approximation is linear, and it involves solution of a linear system of equations in order to find the coefficients  $a_j$ .

For splines of a different order the results are similar. Let  $k+1$  denote the order of the spline, and let  $\{t_j\}_{j=-k}^{N+k+1}$  be the extended partition. The space of polynomial splines on this partition has dimension  $N+k+1$ , and each spline is represented as

$$S(x) = \sum_{j=-k}^N a_j N_j^{k+1}(x). \quad (1)$$

Even though the data to which the spline is adjusted is monotone, the splines themselves are not necessarily monotone. This is illustrated on Fig. 1.

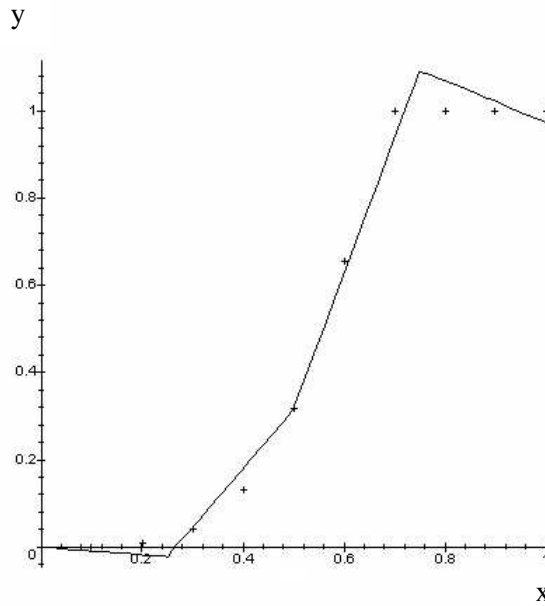


Fig. 1. Linear least squares spline fails to preserve the monotonicity of the data.

To impose the monotonicity, some restrictions have to be placed on the coefficients  $a_j$ . These restrictions are expressed in an especially simple form if we modify the basis. Consider the basis functions  $T_j(x)$  given by linear combinations of  $B$ -splines.

$$T_j^{k+1}(x) = \sum_{l=j}^N N_l^{k+1}(x). \quad (2)$$

There are  $N+k+1$  functions altogether and all are linearly independent; hence the set  $\{T_j(x)\}_{j=-k}^N$  forms a basis in the space of splines of order  $k+1$ . Note that the functions  $T_j^{k+1}(x)$  are closely related to the integrated  $I$ -splines,<sup>7</sup> also used for monotone approximation.



$$\mathbf{y}' = (\mathbf{T}^{k+1})' \mathbf{b} = \mathbf{N}^k (\mathbf{D}^k)^{-1} \mathbf{b}.$$

In other words, for any  $x \in [t_0, t_{N+1}]$ , the derivative of the spline is a linear combination of B-splines.

The advantage of (4) over (3) is that the matrix  $\mathbf{D}^k$  is diagonal (and positive), whereas  $\mathbf{A}^k$  is not. This allows us to use the special structure of the matrix  $\mathbf{N}^k$  for certain  $k$  to express the monotonicity restriction in a simple way. First, observe that because both  $\mathbf{N}^k$  and  $\mathbf{D}^k$  have non-negative entries,  $b_j \geq 0$  imply  $y'_i \geq 0$ . Thus, non-negativity of the coefficients of the spline is a sufficient condition for its monotonicity. Let us now establish the necessary condition.

**Proposition 1.** For  $k=1$  and  $k=2$  (linear and quadratic splines) (4), the necessary and sufficient condition for monotonicity is  $b_j \geq 0, j=-k+1, \dots, N$ .

**Proof.** Let one coefficient, say  $b_l < 0$ . The matrix  $\mathbf{N}^k$ :  $N_{ij}^k = N_j^k(t_i)$  ( $t_i$  are approximation knots) contains only one co-diagonal

$$(\mathbf{N}^1 = \mathbf{I}, \mathbf{N}^2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix}), \mathbf{0} \text{ is zero row-vector}.$$

$$\text{Then, for } k=1, y'_l = S'(t_l) = N_l^1(t_l) \frac{b_l}{D_{ll}^1} = \frac{b_l}{t_{l+1} - t_l}, \text{ and}$$

$$\text{for } k=2, y'_{l+1} = S'(t_{l+1}) = N_l^2(t_{l+1}) \frac{b_l}{D_{ll}^2} = \frac{2b_l}{t_{l+2} - t_l}.$$

Consequently,  $b_l < 0$  implies  $y'_l < 0$  ( $k=1$ ), or  $y'_{l+1} < 0$  ( $k=2$ ). Since we are interested in the derivatives at  $t_l, 0 \leq l \leq N+1$ , we require  $b_j \geq 0, j=-k+1, \dots, N$ .  $\square$

For splines of higher order, the non-negativity of the coefficients is only a sufficient condition.<sup>7</sup>

Thus, the problem of monotone least squares spline approximation takes the form

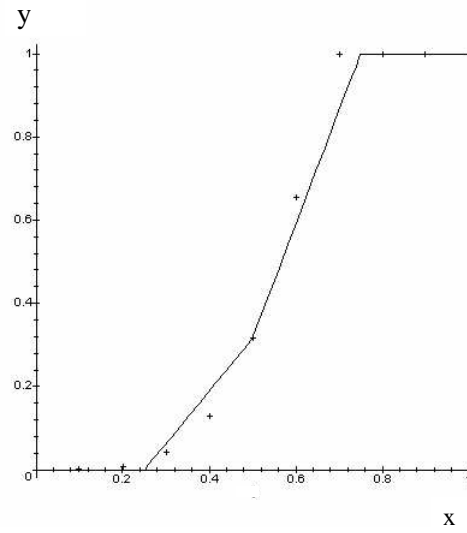
$$\text{Minimize } \sum_{i=1}^I \left[ \sum_{j=-k}^N b_j T_j(x_i) - y_i \right]^2, \text{ subject to } b_j \geq 0, j = -k+1, \dots, N.$$

This is the problem of non-negative least squares, described in ref.<sup>11</sup> There are standard solutions to it (e.g., NNLS and BVLS algorithms from,<sup>11</sup> LSEI algorithm from,<sup>9, 10</sup> both available in NETLIB<sup>34</sup>). They can be employed without any modification, and are quite fast and robust.

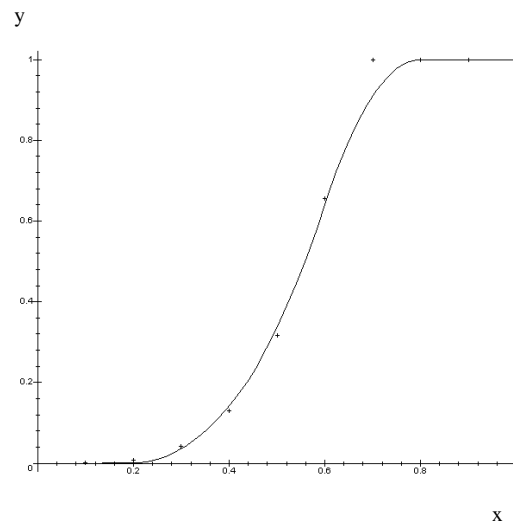
Fig. 2 illustrates approximation of monotone data using linear and quadratic T-splines. Calculation of the coefficients has been performed using LSEI algorithm. The additional restrictions on the function,  $f(0) = 0$  and  $f(1) = 1$ , can be specified as the additional equality constraints of LSEI algorithm.

T-splines basis is well conditioned for numerical calculations. Since the matrix  $\mathbf{L}$  is well conditioned (its condition number in  $l_1$ -norm is  $2(N+k+1)$ ), and  $\mathbf{T} = \mathbf{NL}$ , one can

obtain  $\mathbf{T}$  by using standard efficient algorithms to compute  $\mathbf{N}$ .<sup>29, 33</sup> Once the coefficients of the spline  $\{b_j\}$  are found, the value of  $S(x)$  can be calculated using  $\mathbf{y}=\mathbf{Tb}$ . One can then go back to the traditional  $B$ -spline representation using the conversion formula  $\mathbf{a}=\mathbf{Lb}$ , where  $\mathbf{a}$  are coefficients of  $B$ -splines in (1). The approximation conditions expressed in matrix form are  $\mathbf{Tb} \approx \mathbf{y}$  or  $\mathbf{NLb} \approx \mathbf{y}$ , where  $\approx$  stands for "approximately equal" in least squares sense.



(a)



(b)

Fig. 2. Monotone linear (a) and quadratic (b) least squares splines correctly approximate the monotone data from Fig.1.

## 5. Tensor Product Splines

In this section we extend the univariate monotone least squares splines to the bi-variate and multivariate cases. We will use tensor product linear and quadratic  $T$ -splines for this purpose.

The tensor product  $V$ -dimensional  $T$ -spline is the construction

$$T_{j_1 j_2 \dots j_V}(x_1, \dots, x_V) = \prod_{v=1}^V T_{j_v}(x_v).$$

The function of  $V$  arguments is approximated with

$$S(x_1, \dots, x_V) = \sum_{j_1=-k}^{N_1} \dots \sum_{j_V=-k}^{N_V} b_{j_1 \dots j_V} T_{j_1 \dots j_V}(x_1, \dots, x_V).$$

In the two-dimensional case these formulas take the form

$$S(x_1, x_2) = \sum_{m=-k}^M \sum_{n=-k}^N b_{mn} T_{mn}(x_1, x_2).$$

The monotonicity condition implies that all partial derivatives of the spline have to be non-negative at every point. Because functions  $T_{mn}(x_1, x_2)$  are tensor products, their partial derivatives are multiples of the derivatives of the univariate  $T$ -splines, which are non-negative in the region of approximation. We will express these conditions in matrix notation. Recall that the tensor (or Kronecker) product of two matrices  $\mathbf{A} \otimes \mathbf{B}$  is a matrix with block structure

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}[\mathbf{B}] & a_{12}[\mathbf{B}] & \dots & \cdot \\ a_{21}[\mathbf{B}] & & & \cdot \\ \dots & & & \cdot \\ \cdot & \cdot & \cdot & a_{mm}[\mathbf{B}] \end{pmatrix}$$

Tensor product is an associative operation and it is distributive with respect to matrix multiplication

$$(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D}).$$

The bi-variate spline can be represented as

$$S(x_1, x_2) = (\mathbf{T}_1 \otimes \mathbf{T}_2)(\mathbf{b}_1 \otimes \mathbf{b}_2) = (\mathbf{T}_1 \otimes \mathbf{T}_2)\mathbf{b}, \quad (6)$$

where  $\mathbf{T}_l$  is the row-vector of the values of  $T_j(x_l)$  (we omit the index  $k+1$ , denoting the order of the spline, for the moment), and matrix  $\mathbf{b}$  contains the components  $b_{mn}$  (it can also be reshaped to an array with components arranged in lexicographic order). Since

$$\mathbf{T}_l = \mathbf{N}_l \mathbf{L}_l, \quad l = 1, 2,$$

$$S = (\mathbf{T}_1 \otimes \mathbf{T}_2)\mathbf{b} = ((\mathbf{N}_1 \mathbf{L}_1) \otimes (\mathbf{N}_2 \mathbf{L}_2))\mathbf{b} = (\mathbf{N}_1 \otimes \mathbf{N}_2)(\mathbf{L}_1 \otimes \mathbf{L}_2)\mathbf{b}.$$

Therefore, the matrix of observation equations (consisting of  $I$  rows of the form (6)) can be obtained from the matrix of observation equations in B-spline representation by multiplying it by  $(\mathbf{L}_1 \otimes \mathbf{L}_2)$ .

Differentiating,

$$\begin{aligned}\frac{\partial S(x_1, x_2)}{\partial x_1} &= (\mathbf{T}_1' \otimes \mathbf{T}_2) = \mathbf{N}_1' \mathbf{L}_1 \otimes \mathbf{T}_2 = \mathbf{N}_1^k \mathbf{\Lambda}_1^k \mathbf{L}_1 \otimes \mathbf{T}_2 = \mathbf{N}_1^k (\mathbf{D}_1^k)^{-1} \otimes \mathbf{T}_2 \\ &= (\mathbf{N}_1^k \otimes \mathbf{T}_2) ((\mathbf{D}_1^k)^{-1} \otimes \mathbf{I})\end{aligned}$$

and

$$\begin{aligned}\frac{\partial S(x_1, x_2)}{\partial x_2} &= (\mathbf{T}_1 \otimes \mathbf{T}_2') = \mathbf{T}_1 \otimes \mathbf{N}_2' \mathbf{L}_2 = \mathbf{T}_1 \otimes \mathbf{N}_2^k \mathbf{\Lambda}_2^k \mathbf{L}_2 = \mathbf{T}_1 \otimes \mathbf{N}_2^k (\mathbf{D}_2^k)^{-1} \\ &= (\mathbf{T}_1 \otimes \mathbf{N}_2^k) (\mathbf{I} \otimes (\mathbf{D}_2^k)^{-1})\end{aligned}$$

We require that the derivatives be positive (negative), and the diagonal factors  $((\mathbf{D}_1^k)^{-1} \otimes \mathbf{I})$  can be dropped. Then

$$\frac{\partial S(x_1, x_2)}{\partial x_1} \cong (\mathbf{N}_1^k \otimes \mathbf{T}_2) = (\mathbf{N}_1^k \otimes \mathbf{N}_2 \mathbf{L}_2) = (\mathbf{N}_1^k \otimes \mathbf{N}_2) (\mathbf{I} \otimes \mathbf{L}_2)$$

and

$$\frac{\partial S(x_1, x_2)}{\partial x_2} \cong (\mathbf{N}_1 \otimes \mathbf{N}_2^k) (\mathbf{L}_1 \otimes \mathbf{I}),$$

where  $\cong$  stands for "proportional" (with positive coefficient).

The first tensor product in the above expressions is a row-vector whose components are the values of  $(\mathbf{N}_1^k \otimes \mathbf{N}_2^{k+1})$  or  $(\mathbf{N}_1^{k+1} \otimes \mathbf{N}_2^k)$  taken at a particular point  $(x_1, x_2)$ . Depending on  $k$ , we need to evaluate the partial derivative at a certain number of points to guarantee that the derivatives do not change sign in the whole domain. For bilinear and bi-quadratic splines it is sufficient to evaluate the derivatives at the approximation knots  $(t_m, t_n)$ ,  $m, n=0, 1, \dots, N+1$ . The matrix of the system of linear restrictions on spline coefficients  $\mathbf{G}\mathbf{b} \geq \mathbf{0}$ , takes the form

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{pmatrix}, \quad \mathbf{G}_1 = (\mathbf{N}_1^k \otimes \mathbf{N}_2^{k+1}) (\mathbf{I} \otimes \mathbf{L}_2), \quad \mathbf{G}_2 = (\mathbf{N}_1^{k+1} \otimes \mathbf{N}_2^k) (\mathbf{L}_1 \otimes \mathbf{I}).$$

For splines in more than two variables the construction of  $\mathbf{G}$  is analogous. For multi-linear case  $\mathbf{G}$  has the form

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 \\ \dots \\ \mathbf{G}_V \end{pmatrix}, \quad \mathbf{G}_V = \mathbf{L}_1 \otimes \dots \otimes \mathbf{I}_V \dots \otimes \mathbf{L}_V.$$

Altogether there are at most  $V \times N_1 \times \dots \times N_V$  inequalities (some are redundant). As in the univariate case, we have the restricted least squares problem, and the LSEI<sup>9, 10</sup> algorithm can be used to find the solution. The entries of the matrices of observation equations are given by tensor products of the corresponding T-splines. Further details of the method are presented in ref<sup>35</sup>, and the software is available from the author.

## 6. Approximation of Membership Functions and Aggregation Operators

Given that calculation of spline coefficients is reduced to standard quadratic programming problem with linear inequality constraints, the implementation of the algorithm is straightforward. In matrix form it can be written as

$$\text{Solve } \mathbf{Tb} \approx \mathbf{y}, \text{ given that } \mathbf{Gb} \geq \mathbf{0} \text{ and } \mathbf{Eb} = \bar{\mathbf{y}}, \quad (7)$$

with  $\approx$  standing for ‘‘approximately equal’’.<sup>9, 10</sup>  $\mathbf{T}$  is  $I \times (N + k + 1)$  matrix with the entries given by the values of basis functions  $T_j(x_i)$  at data points  $x_i$ ,  $\mathbf{G}$  is the identity matrix  $\mathbf{I}$  (or a sparse square matrix consisting of 0s and 1s in multivariate case),  $\mathbf{E}$  is  $E \times (N + k + 1)$  matrix similar to  $\mathbf{T}$ , but the values of  $T_j$  are taken at those points where the spline is required to interpolate the data  $\bar{\mathbf{y}}$ , and  $E$  is the number of such points. Both matrices serve as the input to the algorithm LSEI. The matrix  $\mathbf{E}$  is used if additional restrictions, such as  $f(\mathbf{0})=0$  and  $f(\mathbf{1})=1$ , need to be imposed. As we mentioned earlier, computation of the entries of  $\mathbf{T}$  can be performed in a very effective way using  $B$ -splines basis  $N_j(x)$ .

Let us now consider using the proposed method of monotone splines to approximate membership functions and aggregation operators in FST. Figure 3 illustrates approximation of the membership function of the fuzzy set ‘‘tall people’’ using the empirical measurements from ref<sup>1</sup>. Figure 4 illustrates monotone approximation on a model problem: the aggregation operator is given and is used to randomly generate 20 data points. These points are subsequently used as the empirical data to reconstruct the aggregation operator.

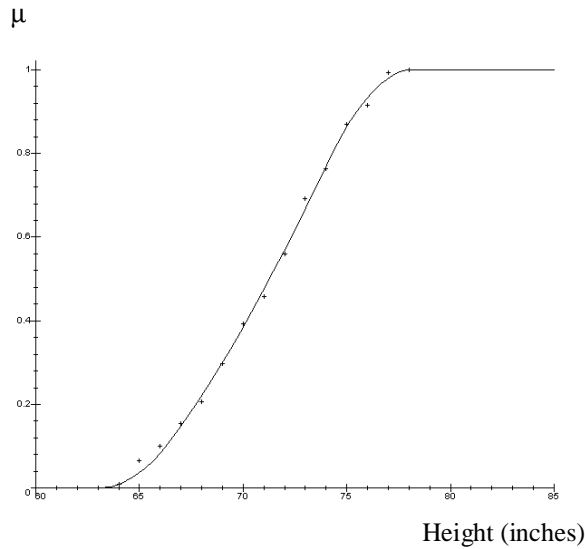


Fig. 3. Approximation of the membership function of the fuzzy set ‘‘tall people’’ using monotone quadratic spline with 5 knots. The empirical data are taken from ref<sup>1</sup>.

Figures 5 and 6 present the results of bi-variate monotone spline approximation of the empirical data given in ref <sup>2</sup>. These data represent subjects' estimates of the membership values of various objects in the fuzzy sets "metallic object", "container" and then in the compound set "metallic container". The task is to find a model for such aggregation. The empirical data is shown with circles. Fig. 6 assesses the quality of the operator by plotting predicted vs. observed membership values in the compound set. The closer data points to the diagonal, the better.

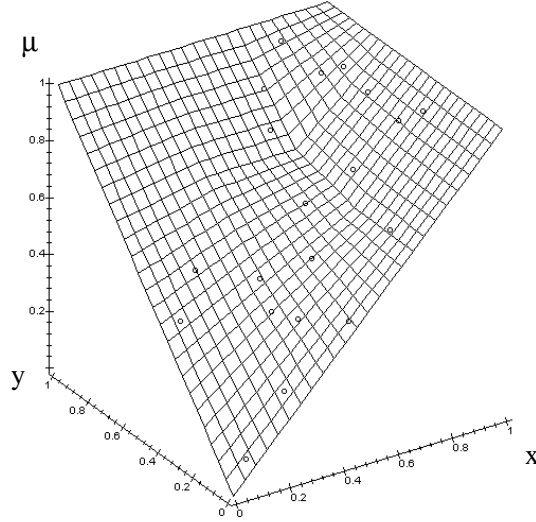


Fig. 4. Approximation of a triangular co-norm using tensor-product linear spline.

The Hamacher sum operator  $f(x, y) = \frac{x + y - 2xy}{1 - xy}$  is used as the model.

The 20 data points are randomly generated and are marked with circles.

The data contains random noise uniformly distributed in [-0.1,0.1].

## 7. Approximation of Aggregation Operators with Specific Properties

Additional constraints  $f(\mathbf{0}) = 0$  and  $f(\mathbf{1}) = 1$ , important for aggregation operators, can be imposed in a straightforward manner within LSEI algorithm. Any interpolation condition can be specified in  $\mathbf{E}\mathbf{b} = \bar{\mathbf{y}}$  system in (7). The above equality constraints will automatically form part of the algorithm.

Another interpolation condition important for triangular conorms is  $f(0,0,\dots,x_i,\dots,0) = x_i$ . It can be enforced by making the spline to interpolate the values  $f(0,0,\dots,t_j,\dots,0) = t_j$ , where  $t_j$  are the knots of approximation in respect to coordinate  $x_i$ . The restriction  $f(1,1,\dots,x_i,\dots,1) = 1$  is imposed by forcing the spline to interpolate the

values  $f(1,1,\dots,t_j,\dots,1)=1$ . Similar restrictions are applied to triangular norms. The idempotency of averaging operators ( $S(x,x)=x$ ) is dealt with in the same way, and is translated into the interpolation conditions  $S(t_l, t_l) = t_l$ .

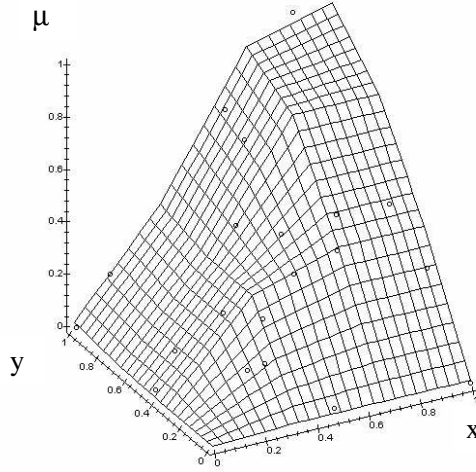


Fig. 5. Approximation of an aggregation operator using empirical data from ref. 2.

The associativity property of triangular norms and conorms does not have a simple geometrical illustration. However, it can be used to obtain triangular norms and conorms in an alternative way. As we mentioned, all continuous Archimedean triangular norms ( $T(x,x)<x$ ) and conorms ( $C(x,x)>x$ ),  $x \in (0,1)$ , possess additive generators.<sup>18</sup> Given the additive generator, the operator itself can be obtained from

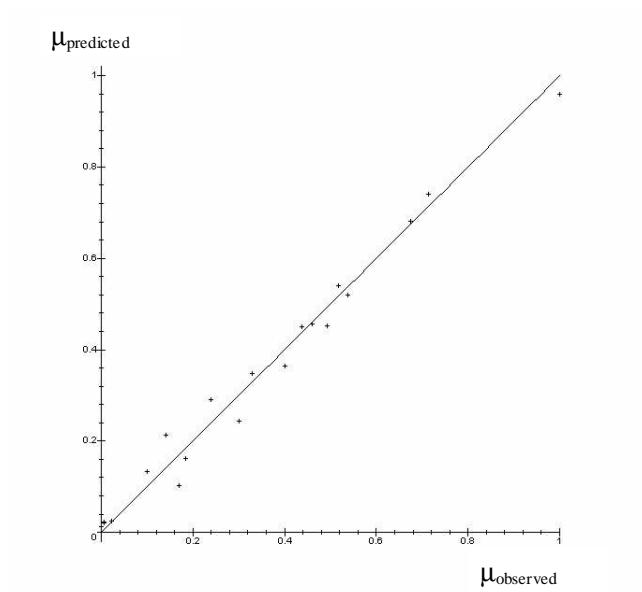
$$T(a,b)=g^{[-1]}(g(a)+g(b)) \text{ or } C(a,b)=g^{[-1]}(g(a)+g(b)), \quad (8)$$

where  $g^{[-1]}$  is the pseudoinverse function.

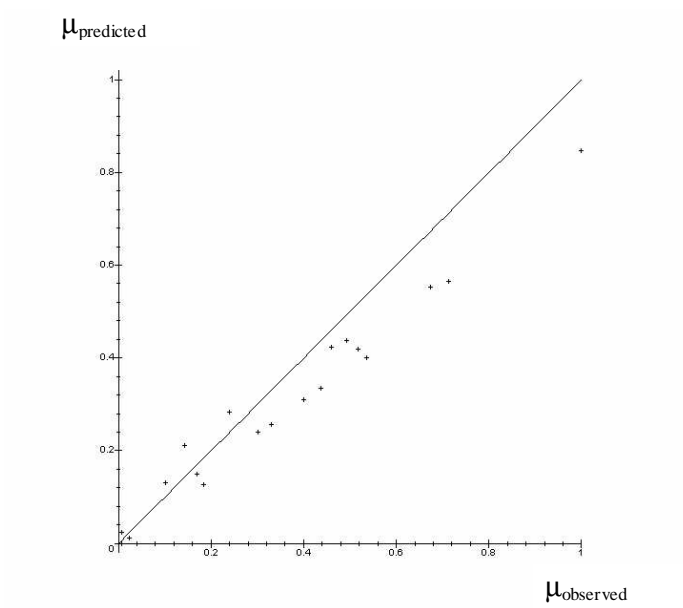
Since additive generators define the triangular norms and conorms uniquely, they can be used to approximate them. Thus, rather than approximating the aggregation operator directly using tensor product splines, we can approximate its additive generator using monotone univariate spline. This will guarantee that the operator belongs to the class of triangular norms (conorms).

Consider triangular conorms. The Eq.(8) translates into  $g(C(x_1,x_2)) = g(x_1) + g(x_2)$ , or  $g(x_1) + g(x_2) - g(C(x_1,x_2)) = 0$ . As the empirical data we have  $\{(x_{1i}, x_{2i}, y_i)\}_{i=1}^I$ . Since  $g$  is represented by the spline, we can write

$$\sum_j b_j (T_j(x_1) + T_j(x_2) - T_j(y)) \approx 0, \text{ or } \sum_j b_j \bar{T}(x_1, x_2, y) \approx 0. \quad (9)$$



(a)



(b)

Fig. 6. Comparison of the quality of the proposed approximation of the aggregation operator with that of  $\min$  operator: observed vs. computed membership values. The closer the data to the diagonal, the better. (a) Proposed least squares monotone spline. (b)  $\min$  operator.

Next we solve the non-negative linear least squares problem (  $g$  is monotone ), in which the basis functions are given as  $\bar{T}(x_1, x_2, y)$ . Once the coefficients  $b_j$  are found, the additive generator is defined, and the value of the operator  $C$  can be numerically determined from  $g$ . In a similar way, triangular norms and other operators that possess additive generators, like uninorms,<sup>20</sup> can be found from empirical data.

Quasi-arithmetic means is another family of operators defined via generating functions

$$M(a, b) = g^{-1}\left(\frac{g(a) + g(b)}{2}\right),$$

or in the multivariate case,

$$M(x_1, \dots, x_n) = g^{-1}\left(\frac{g(x_1) + \dots + g(x_n)}{n}\right).$$

Instead of (9) one uses

$$\sum_j b_j \left( \frac{1}{n} \{T_j(x_1) + \dots + T_j(x_n)\} - T_j(y) \right) \approx 0$$

and solves the corresponding non-negative least-squares problem.

## 8. Conclusion

Monotonicity of membership functions and aggregation operators is semantically important in fuzzy set theory. These functions frequently have to be built using empirical data, which is a) noisy and b) scattered. Simple yet powerful tools are required to approximate the data while preserving monotonicity of the functions. Least squares splines are versatile in fitting the data but they are rarely monotone. Representing splines in a modified basis allows one to express the monotonicity conditions in a very simple way, as non-negativity of the coefficients.

The non-negative least squares problem, resulting from such a representation, has been thoroughly studied, and effective algorithms are available. Besides, both approximation and interpolation conditions can be specified as the input of some algorithm. This helps to include additional boundary conditions that are important in FST. Finally, the proposed  $T$ -splines are easily computed from the traditional  $B$ -splines with the help of one matrix multiplication. Therefore, all the calculations are well-conditioned and the existing spline algorithms can be used with only minor modifications. The coefficients of the monotone spline are also transformed into the coefficients of  $B$ -spline representation, which makes computation of spline values very efficient.

Approximation of membership functions with splines is relatively straightforward: linear or quadratic monotone splines provide an appropriate solution. Aggregation operators provide room for several methods of approximation.

- General aggregation operators can be approximated using monotone tensor product spline. This method takes into account only the data.
- Additional boundary conditions and idempotency can be specified as interpolation conditions.
- Commutativity can be achieved by making the matrix of coefficients symmetric.
- Triangular norms and co-norms can be approximated using their additive generators. It ensures their associativity.
- Quasi-arithmetic means can be approximated using their generator functions.

Thus, the advantage of using monotone splines is their generality on the one hand, and a range of tools they provide to select a particular class or property on the other. Linearity of the approximation problem is also a bonus.

Besides being a method for building membership functions and aggregation operators from empirical data, monotone splines are also an effective tool for representation and coding of known operators. Indeed, rather than representing an operator/membership function in algebraic form, one can specify its values at certain points, and then build a spline that accurately approximates it. The numerical computation of such approximation is practically as effective as using the original algebraic representation (one should pass to B-spline representation (1), which is numerically very stable and efficient). This way one can provide a generic code (for fuzzy controllers, or for decision or expert systems), while the actual operators and membership functions used by the system will be determined by the supplied array of coefficients.

## References

1. A. M. Norwiche and I. B. Turksen, "A model for the measurement of membership and the consequences of its empirical implementation," *Fuzzy Sets and Systems* **12** (1984), 1-25.
2. H.-J. Zimmermann and P. Zysno, "Latent connectives in human decision making," *Fuzzy Sets and Systems* **4** (1980), 37-51.
3. H.-J. Zimmermann, *Fuzzy set theory - and its applications* (Kluwer, Boston, 1996).
4. L. Yen, R. Langari, and L. Wang, "Principal components, B-splines, and fuzzy system reduction," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* **4** (1996), 561-572.
5. K. Shimojima, T. Fukuda, and F. Arai, "Self-tuning fuzzy inference based on spline function," *Proc. of the 3rd Intl. Conf. on Fuzzy Systems*, IEEE, 1994.
6. M. F. Kawaguchi and M. Miyakoshi, "Fuzzy rule interpolation technique based on bi-splines in multiple input systems," *Proc. of the 9th Intl. Conf. on Fuzzy Systems*, IEEE, 2000.
7. J. O. Ramsay, "Monotone regression splines in action," *Stat. Science* **3** (1988), 425-461.
8. X. He and P. Shi, "Monotone B-spline smoothing," *J. Am. Stat. Ass.* **93** (1998), 643-650.
9. R. J. Hanson and K. H. Haskell, "Algorithm 587. Two algorithms for the linearly constrained least squares problem," *ACM Trans. Math. Software* **8** (1982), 323-333.
10. K. H. Haskell and R. Hanson, "An algorithm for linear least squares problems with equality and nonnegativity constraints," *Math. Progr.* **21** (1981), 98-118.
11. C. Lawson and R. Hanson, *Solving least squares problems* (SIAM, Philadelphia, 1995).
12. L. Zadeh, "Fuzzy sets," *Information and Control* **8** (1965), 338-353.
13. G. P. Amaya Cruz and G. Beliakov, "Approximate reasoning and interpretation of laboratory tests in medical diagnostics," *Cybernetics and Systems* **26** (1995), 713-729.
14. G. Beliakov, "Fuzzy sets and membership functions based on probabilities," *Information Sciences* **91** (1996), 95-111.
15. R. Giles, "The concept of grade of membership," *Fuzzy Sets and Systems* **25** (1988), 297-323.

16. G. Klir and T. Folger, *Fuzzy sets, uncertainty, and information* (Prentice Hall, Singapore, 1992).
17. D. Dubois and H. Prade, "A review of fuzzy set aggregation connectives," *Information Sciences* **36** (1985), 85-121.
18. E. P. Klement, R. Mesiar, and E. Pap, *Triangular norms* (Kluwer, Dordrecht, 2000).
19. R. Yager and J. Kacprzyk, eds., *The ordered weighted averaging operators. Theory and Applications*. (Kluwer, Boston, 1997).
20. R. Yager and A. Rybalov, "Uninorm aggregation operators," *Fuzzy Sets and Systems* **80** (1996), 111-120.
21. G. Beliakov, "Definition of general aggregation operators through similarity relations," *Fuzzy Sets and Systems* **114** (2000), 437-453.
22. L.-E. Andersson and T. Elfving, "An algorithm for constrained interpolation," *SIAM J. Scient. Stat. Comput* **9** (1987), 1012-1025.
23. L.-E. Andersson and P.-A. Ivert, "Constrained interpolants with minimal  $W_{k,p}$  norm," *J. Approx. Theory* **49** (1987), 283-288.
24. L.-E. Andersson and T. Elfving, "Interpolation and approximation by monotone cubic splines," *J. Approx. Theory* **66** (1991), 302-333.
25. R. K. Beatson and Z. Ziegler, "Monotonicity preserving surface interpolation," *SIAM J. Numer. Anal.* **22** (1985), 401-411.
26. R. E. Carlson and F. N. Fritsch, "Monotone piecewise bicubic interpolation," *SIAM J. Numer. Anal.* **22** (1985), 386-400.
27. P. Constantini and F. Fontanella, "Shape-preserving bivariate interpolation," *SIAM J. Numer. Anal.* **22** (1985), 488-506.
28. B. Juttler, "Surface fitting using convex tensor-product splines," *J. Comp. Appl. Math.* **84** (1997), 23-44.
29. C. De Boor, *A practical guide to splines* (Springer, Berlin, New York, 1978).
30. H. Greiner, "A survey on univariate data interpolation and approximation by splines of given shape," *Math. Comp. Modelling* **15** (1991), 97-106.
31. R. L. Eubank, *Spline smoothing and nonparametric regression* (Marcel Dekker Inc, New York, 1988).
32. P. Dierckx, *Curve and surface fitting with splines* (Clarendon press, Oxford, 1995).
33. L. L. Schumaker, *Spline functions: Basic theory* (Wiley, New York, 1981).
34. J. J. Dongarra and E. Grosse, "Distribution of mathematical software via electronic mail," *Comm. ACM* **30** (1987), 403-440.
35. G. Beliakov, "Shape preserving approximation using least squares splines," *Approximation theory and applications* **16** (2000), 80-98.