

ファジィ概念を用いたスプライン関数の節点の決定

—知識処理支援数値計算の試み—

吉本 富士市†

本論文では、ファジィ概念を用いてスプライン関数の節点を決定する方法を提案する。この方法は、人間の知識、経験などに基づいて試行錯誤的に節点を決定する場合に用いられている単純な方法を、ファジィ概念を用いて改良することにより、スプライン関数の利用者により優しい（易しい）インタフェースを提供するものである。人間の知識、経験などに基づいて試行錯誤的に節点を定める場合には、一般に次のようなルールがよく用いられている：データのもとにある関数（信号）の形が大きく変化しているところには節点を密に入れ、ゆるやかに変化しているところには節点を疎に入れる。このルールの“大きく変化しているところ”および“ゆるやかに変化しているところ”は、ファジィ集合と見なすことができ、メンバーシップ関数を用いて表現できる。そのメンバーシップ関数の大きさを節点の密度と関連づけると、上記のルールにより適切な節点の配置を決めることができる。本論文では、あてはめの問題について、このような節点の決定方法を提案し、その有効性を示す例題をあげ、試行錯誤的な方法よりも優れていることを述べている。

Determination of the Knots of a Spline Function by the Use of Fuzzy Concept

—A New Approach of Numerical Computation Supported by Knowledge Processing—

FUJICHI YOSHIMOTO†

A method to determine good knots of a spline function is proposed by the use of fuzzy concept. A conventional method is often used: on the bases of knowledge and experience, one tries to determine good knots, and after a lot of trial and error, one obtains an acceptable number and positions of knots. We improve the method by introducing fuzzy concept, and presents more user friendly (easier) interface of a spline approximation. A rule is used in the conventional method: more knots are needed in regions where the behavior of the underlying function of data is changing rapidly and fewer where it is relatively smooth. Two phrases “where the behavior of the underlying function of data is changing rapidly” and “where it is relatively smooth” are considered to be fuzzy sets, and each of them can be represented by a membership function. If the magnitude of the membership function is related to the density of knots, we can determine good allocation of knots. In this paper, we propose such a method to determine good knots for curve fitting, give some examples to show the performance of our method, and urge that our method is superior than the conventional method.

1. はじめに

数値解析の研究成果は、それに基づいて作られる数学ソフトウェアの性能によって評価され、その数学ソフトウェアは、利用者にどれだけ有効に使われるかによって評価されることが多い。したがって利用者の立場から、すなわちユーザ・インタフェースの立場から、数値解析ないしは数学ソフトウェアの研究を行うことは、きわめて重要なことである。

この観点からスプライン関数とその応用の現状を眺めてみると、気になるのは節点の決定方法である。

よく知られているように、スプライン関数をうまく使う鍵は節点の決定にある^{1),2)}。このため、節点の決定方法について多くの研究が行われてきている³⁾⁻⁸⁾。しかし、これまで提案されている方法は、スプライン関数を専門としない一般の利用者にとっては、取扱いの容易さ、扱える問題の広さ、計算量などの点で、まだ十分満足できる状況にない。このため、現実にはスプライン関数の利用者が知識、経験などに基づいて試行錯誤的に決定している場合も少なくない⁹⁾。スプライ

† 和歌山大学教育学部
Faculty of Education, Wakayama University

ン関数を本当に使いやすくするためには、適切な節点を容易に決定できるような方法についての一層の研究が不可欠である^{9),10)}。

ところで、エキスパートの知識、経験などを計算機で処理するための方法論として、最近ファジィ概念の有効性が注目され、多くの成果をあげている。ファジィ理論とスプライン関数の関連についても、最近いくつかの研究が行われている¹¹⁾⁻¹³⁾。すでに述べたように、スプライン関数の節点を試行錯誤的に決定する場合、人間の知識、経験などを用いている。そこで、ファジィ概念をスプライン関数の節点の決定問題へ応用できるのではないかと考えられる。

本論文では、ファジィ概念を用いてスプライン関数の節点を決定する新しい方法を提案し、その有効性を示す。この方法は、手動的な節点の決定方法^{14),15)}をできるだけ自動化することにより、スプライン関数の利用者により優しい(易しい)インタフェースを提供しようとするものである。

2. スプライン関数によるあてはめ

節点の決定は補間問題でも重要である¹⁶⁾が、ここでは補間よりも節点の影響が大きいあてはめの問題を扱うことにする。また、簡単のため最小2乗法を用いた1次元データのあてはめ¹⁶⁾について考えることにする。

いま、あてはめを行うべきデータは、 x 軸上の区間 $[a, b]$ 内で与えられ、

$$F_j = f(x_j) + \varepsilon_j \quad (j=1, 2, \dots, N) \tag{2.1}$$

と表されるものとする。ここで、 $f(x)$ はデータのもとにある関数(信号)であり、未知の関数であると仮定する。また、 ε_j は平均値0、分散 σ^2 の正規分布をする互いに独立な誤差であると仮定する。

必要な節点を $\xi_i (i=1-m, 2-m, \dots, n+m)$ と書くことにする。ここで、 n は区間 $[a, b]$ の内部に配置する節点 $\xi_i (i=1, 2, \dots, n)$ の数である。また、 m はスプライン関数を用いて $f(x)$ の近似関数 $S(x)$ を表すために使う B-スプライン $N_{m,i}(x)$ の階数(次数+1)である。両端の m 個の節点はそれぞれ端点 a, b に重ね、

$$\left. \begin{aligned} a &= \xi_{1-m} = \dots = \xi_0 \\ b &= \xi_{n+1} = \dots = \xi_{n+m} \end{aligned} \right\} \tag{2.2}$$

とする。

このとき、近似関数 $S(x)$ は

$$S(x) = \sum_{i=1}^{n+m} c_i N_{m,i}(x) \tag{2.3}$$

と表すことができる。ここで、 c_i は B-スプライン係数である。データ、節点、近似関数の関係は、たとえば図1のようになる。

最小2乗法を用いて式(2.3)を与えられたデータ(2.1)にあてはめるとき、残差の2乗和 G は

$$G = \sum_{j=1}^N w_j \{S(x_j) - F_j\}^2 \tag{2.4}$$

となる。ここで、 w_j はデータの重みであり、 $N > n+m$ とする。式(2.4)を最小にする条件から B-スプライン係数 $c_i (i=1, 2, \dots, n+m)$ を求めることができる¹⁶⁾。ただし、よい近似を得るためには内部の節点 $\xi_i (i=1, 2, \dots, n)$ の数と位置を適切に決める必要がある。このための新しい方法を提案することが、本論文の目的である。

3. ファジィ概念による節点の決定

3.1 メンバシップ関数と節点の配置

スプライン関数の利用者が知識、経験などに基づいて節点を決める場合には、一般に次のようなルールがよく用いられている^{14),15)}。

ルール: データのもとにある関数(信号)の形が大きく変化しているところには節点を密に入れ、ゆるやかに変化しているところには節点を疎に入れる。

このルールの下線部は、厳密に数量化することが困難な“あいまいさ”を含んでいるので、ファジィ集合¹⁷⁾と見なすことができる。したがって、節点の決定にファジィ概念を導入できる。いま、図2(a)のデータに対して上記のルールの下線部をメンバシップ関数を用いて表すと、たとえば図2(b)のようになる。

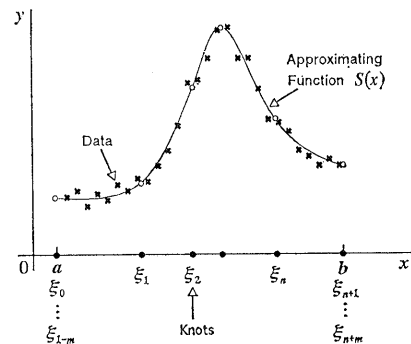


図1 データ、節点、近似関数の関係
Fig. 1 Relation among data, knots and an approximating function.

MF1 は“大きく変化しているところ”に対応し、MF2 は“ゆるやかに変化しているところ”に対応している。両者はちょうど裏返し（互いに補集合）の関係にあるので、どちらかの関数の大きさを節点の密度と関連づけると、上記のルールに基づいて節点の配置を決めることができる。

本論文では、MF1 を用いることにする。この場合には、メンバーシップ関数の大きさと節点の密度が比例するように節点を配置すればよい。なお、メンバーシップ関数の大きさと節点の密度を関連づけるという考え方は、有限要素法のための自動要素分割にも用いられている¹⁸⁾。しかし、本論文で扱っている問題は、あらかじめ与えられた離散的なデータをスプライン関数で近似するときの節点の配置であり、有限要素法のための節点の配置とは異なる問題である。

3.2 メンバーシップ関数の決定

メンバーシップ関数の形は、スプライン関数の利用者がデータを見ながら知識、経験などに基づいて決定する。すでに述べたように、メンバーシップ関数として MF1 を用いるときには、その大きさが節点の密度

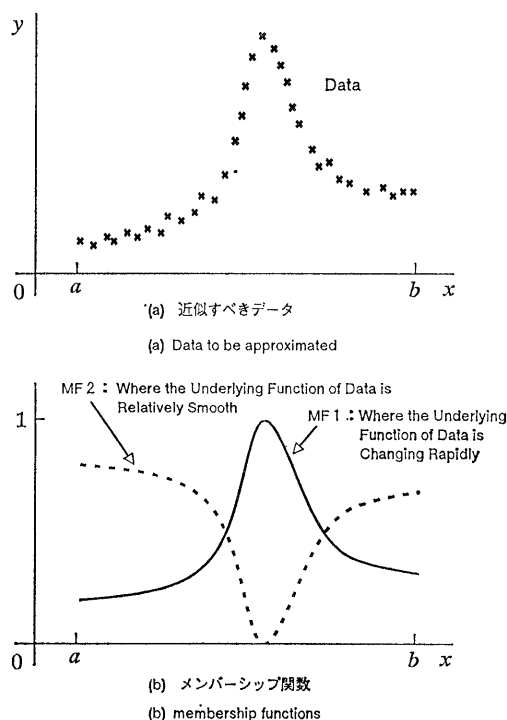


図 2 データとメンバーシップ関数の関係

Fig. 2 Relation between data and membership functions.

に比例するように節点の配置を決めなければならない。これは、区間 $[a, b]$ 上での MF1 の積分を等面積に分ける点を節点の位置とすることによって、近似的に実現できる。

ところが、MF1 は知識や経験などに基づいて決定されるものであるから、簡単な関数で表現できるとはかぎらない。むしろ、一般には自由曲線となることが多いと思われる。したがって、MF1 をそのまま用いると、等面積に分ける点（すなわち節点の位置）を求める計算が大変である。そこで MF1 を計算が簡単な関数で近似する必要がある。ここでは、MF1 を折れ線で近似する場合を扱う。その近似関数を“折れ線型メンバーシップ関数”と呼び、MF3 で表すことにする（図 3 参照）。これは、1 次スプライン関数を用いて元のメンバーシップ関数 MF1 を近似すると見なしでもよい。

以上のように考えると、メンバーシップ関数を決めるためには、結局 MF3 の各頂点の座標を決めればよいことになる。そのためのガイドラインは、現在のところ次の 3 つである：

- (1) データのもとにある関数 $f(x)$ の形が“大きく変化しているところ”に対応する MF3 の値が大きく、“ゆるやかに変化しているところ”に対応する MF3 の値が小さくなるようにする。
- (2) MF3 の頂点の数は少なくてもよく、その座標値はある程度大雑把に決めてもよい。それらを具体的にいくらにするかという判断を行うための知識は、蓄積されているあてはめの計算例データベース（4 章参照）を用いてあらかじめ獲得しておく。
- (3) 関数形に特異点がある場合には、そこで区間を分けて計算する。

このガイドラインは大雑把であるが、計算例データベースの内容が増えるにつれて逐次的に改良できるも

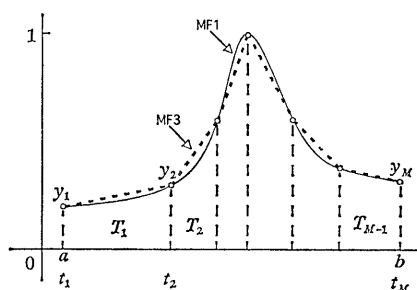


図 3 メンバーシップ関数の近似

Fig. 3 Approximation of a membership function.

のと考えている。そのための知識は、“どのようなデータのときにどのようなメンバーシップ関数がかかったか”という例を数多く分析することにより、経験則として獲得する。

なお、本論文ではメンバーシップ関数どうしの比較を容易にするため、正規性（高さが1）を仮定するとともに、全体集合（全体の区間）は $[a, b]=[0, 1]$ であるとす。このようにしても一般性を失うことはない。

3.3 節点の位置の計算

前節で述べたように、メンバーシップ関数として MF3 を用いるとき、内部の節点の位置はその積分を等面積に分ける点とする。いま、図3に示すように MF3 の各頂点を (t_i, y_i) ($i=1, 2, \dots, M$) と書くことにする。また、各区間 $[t_i, t_{i+1}]$ で MF3 を積分した値を T_i とし、全体の区間 $[a, b]$ で MF3 を積分した値を T とすると、

$$T = \sum_{i=1}^{M-1} T_i \tag{3.1}$$

である。ここで、 T_i の値は台形の面積であるから容易に計算できることに注意したい。内部の節点の数は n であるから、MF3 の積分 T を

$$U = T/(n+1) \tag{3.2}$$

で等分割する点を求めて、それらを節点 ξ_i ($i=1, 2, \dots, n$) の位置とする（図4参照）。このための主要な計算は2次方程式を n 回解くことであるので、その計算の手間はわずかである。

4. あてはめのアルゴリズム

式(2.1)で与えられる離散的なデータに対してスプライン関数(2.3)をあてはめるアルゴリズムの概要は、次のようになる。

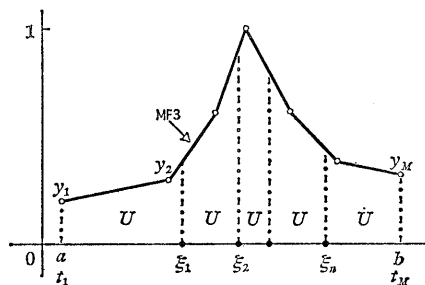


図4 節点の位置の計算
Fig. 4 Computation of the positions of knots.

ステップ 1: データ (x_j, F_j) ($j=1, 2, \dots, N$) を入力する。あてはめの区間が $[a, b]=[0, 1]$ となるように、必要であればデータの横座標 x を変数変換する。

ステップ 2: スプライン関数の階数 m (次数+1)、内部の節点数 n の最小値 n_{min} と最大値 n_{max} を入力する。

ステップ 3: 折れ線型メンバーシップ関数の座標値 (t_i, y_i) ($i=1, 2, \dots, M$) を入力する。

ステップ 4: $n \leftarrow n_{min}$ とする。

ステップ 5: メンバーシップ関数を用いて内部の節点 ξ_i ($i=1, 2, \dots, n$) の位置を計算する。

ステップ 6: Schoenberg-Whitney の条件¹⁶⁾を満たしているか? YES のとき次のステップ7へ行く。NO のときステップ11へ行く。

ステップ 7: 最小2乗法を用いて、近似関数 $S(x)$ の B-スプライン係数 c_i ($i=1, 2, \dots, n+m$) を求める。

ステップ 8: 近似関数の値、誤差分散の不偏推定量¹⁶⁾、AIC¹⁶⁾などを計算する。

ステップ 9: 許容できる結果が得られたか? YES のとき計算を終了する。NO のとき次のステップ10へ行く。

ステップ 10: $n \leftarrow n+1$ とする。 $n \leq n_{max}$ であればステップ5へ戻る。NO のとき次のステップ11へ行く。

ステップ 11: メンバーシップ関数の座標値を変更して再計算してみるか? YES のときステップ3へ戻る。NO のとき計算を終了する。

以上のアルゴリズムを考えるに際して、内部の節点数の変更とメンバーシップ関数の変更と、どちらを優先するかという問題があったが、上記のアルゴリズムでは前者を優先している。その理由は、節点数を連続的に変化させる方がメンバーシップ関数を連続的に変化させるよりも簡単であり、その部分のアルゴリズムを自動化しやすいからである。このことは、メンバーシップ関数の変更には知識や経験が必要であるが、節点数は全く自動的に変更できることから容易に納得できるであろう。

ステップ2の n_{min} および n_{max} の値は、次のように設定する。すなわち、良いあてはめを与える節点数 n がどの程度であるか、について全く見当がつかなければ $n_{min}=1, n_{max}=[N/2]$ とする。しかし、もしもそれがある程度予想できれば、 n_{min} を1より大きくした

り, n_{\max} を $[N/2]$ より小さくしたりして, n の値の探索区間 $[n_{\min}, n_{\max}]$ を狭めて計算量を節減する.

なお, よい結果を与えるメンバーシップ関数が求められた場合には, あてはめるべきデータとともにデータベース化しておく. これを“計算例データベース”と呼ぶことにする. このようにすると, そのデータベースを用いて学習することにより“ある新しいデータに対してどのようなメンバーシップ関数を用いればよいか”という判断を行うための知識を獲得することができる.

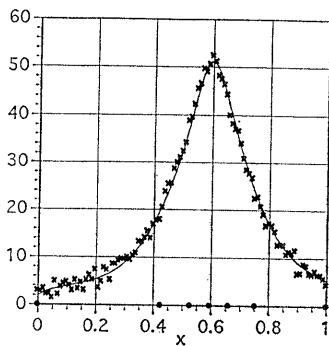
5. 計算例

前章で述べたアルゴリズムの有効性を示すため, ここでは3つの例題を取り上げる. いずれのデータも, 単一の多項式では近似することが困難なものである. スプライン関数の次数は, 一般に最もよく使われている3次の場合について計算した. 以下で述べる各図の

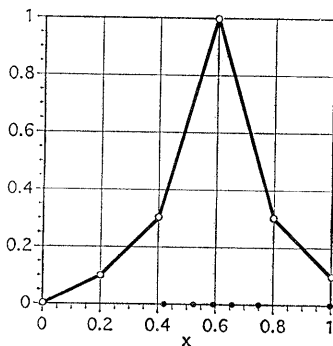
x 軸上の黒丸は, 節点の位置 (MF3 の積分 T を等面積に分割する点) を表している. また, 最小2乗近似を計算するときの重み w_j の値はすべて1とした.

メンバーシップ関数は, あてはめを行う者 (スプライン関数の利用者) が決める必要があるが, その方法は単純である. 3.2 節のガイドラインにしたがって, 節点を集めたいと思う部分のメンバーシップ関数の値が大きくなるように, メンバーシップ関数の座標値 (t_i, y_i) ($i=1, 2, \dots, M$) を決めるだけでよい. ただし, あてはめるべきデータを見ただけで直ちにより座標値を決められるとはかぎらない. 通常は, 若干の試行錯誤を必要とする. その程度は, あてはめを行う者の知識, 経験などに依存するが, ここで取り上げる例題のデータでは, 初心者でも数回~十数回の試行錯誤でよい結果が得られた.

なお, あてはめを行う者の知識や経験は, 計算例

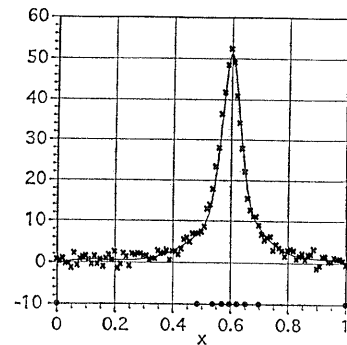


(a) あてはめの結果
(a) Result of fitting

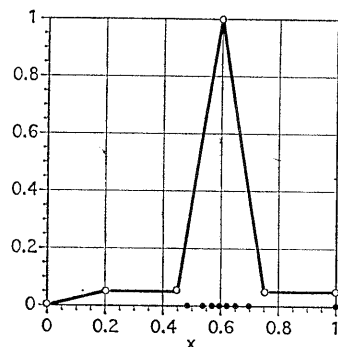


(b) メンバーシップ関数
(b) Membership function

図5 例1の結果 (折れ線型メンバーシップ関数, $\alpha=1$)
Fig. 5 Result of example 1.
(Polygonal membership function, $\alpha=1$)



(a) あてはめの結果
(a) Result of fitting



(b) メンバーシップ関数
(b) Membership function

図6 例1の結果 (折れ線型メンバーシップ関数, $\alpha=10$)
Fig. 6 Result of example 1.
(Polygonal membership function, $\alpha=10$)

データベースを用いて学習することにより容易に増やすことができる(4章参照).

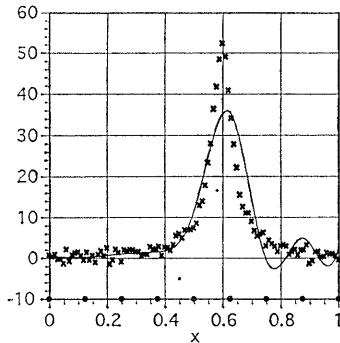
5.1 計算結果

例 1: 有理式を用いて作成したデータの場合
あてはめるべきデータを次の式

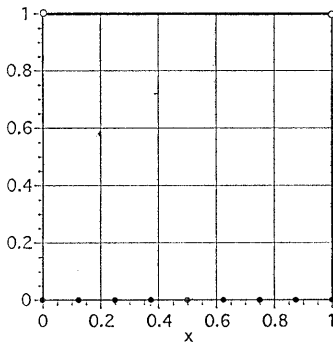
$$F_j = 1 / \{0.02 + \alpha(x_j - 0.6)^2\} + \varepsilon_j \quad (j=1, 2, \dots, N) \quad (5.1)$$

で作成した. ここで, ε_j は期待値 0, 分散 1 の正規分布をする誤差である. x_j の値は, 0.0, 0.01, ..., 1.0 の 101 個とした. またあてはめを行う区間は $[a, b] = [0, 1]$ にした. このデータの特徴は $x=0.6$ 付近でのスペクトル状の山である. 式(5.1)に含まれるパラメータ α を大きくすると山が鋭くなる.

図 5~図 6 に, 折れ線型メンバーシップ関数を用いてあてはめの計算を行った結果を示す. 節点の位置が, メンバーシップ関数 MF3 の積分 T を等面積に分ける点であることを注意したい. 図 7 は, 図 6 と同じ



(a) あてはめの結果
(a) Result of fitting



(b) メンバーシップ関数
(b) Membership function

図 7 例 1 の結果 (定数型メンバーシップ関数, $\alpha=10$)
Fig. 7 Result of example 1.
(Constant membership function, $\alpha=10$)

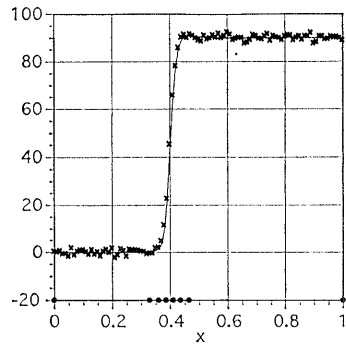
データに対して定数型のメンバーシップ関数 {MF 3 の値=1 | $x \in [0, 1]$ } を用いて計算した結果である. この場合には, 節点は等間隔になる.

例 2: ロジスティック曲線を用いて作成したデータの場合
あてはめるべきデータを次の式

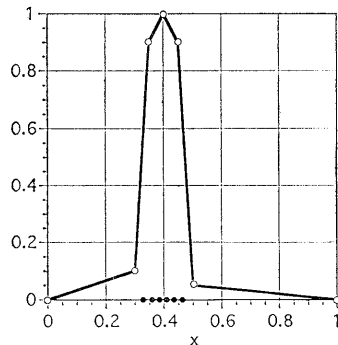
$$F_j = 90 / (1 + e^{-\alpha(x_j - 0.4)}) + \varepsilon_j \quad (j=1, 2, \dots, N) \quad (5.2)$$

で作成した. ここで, ε_j は期待値 0, 分散 1 の正規分布をする誤差である. x_j の値は, 0.0, 0.01, ..., 1.0 の 101 個とした. また, あてはめを行う区間は $[a, b] = [0, 1]$ にした. このデータの特徴は, $x=0.4$ 付近での階段状の鋭い変化である. 式(5.2)に含まれるパラメータ α を大きくすると階段状の変化が鋭くなる.

図 8 に, 折れ線型メンバーシップ関数を用いてあてはめの計算を行った結果の例を示す.



(a) あてはめの結果
(a) Result of fitting



(b) メンバーシップ関数
(b) Membership function

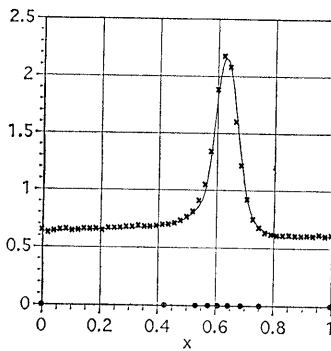
図 8 例 2 の結果 (折れ線型メンバーシップ関数, $\alpha=100$)
Fig. 8 Result of example 2.
(Polygonal membership function, $\alpha=100$)

例 3: チタンの熱特性データ^{2),3)}の場合

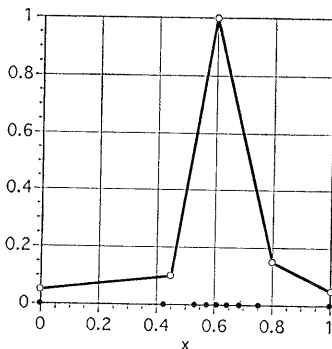
このデータは、スペクトル状の鋭い山を持つ実際のデータである。文献 3) では、データの x 座標は $x_j = 575 + jh$, ($j=1, 2, \dots, 49$; $h=10$) となっているが、ここではメンバーシップ関数を区間 $[0, 1]$ で考えたいので、それらを変数変換して $x_j = 0.0, 1/48, \dots, 1.0$ とした。図 9 に、折れ線型メンバーシップ関数を用いてあてはめの計算を行った結果の例を示す。

5.2 考察

例 1 の結果から、パラメータ α の値が大きくなり山が鋭くなるほど節点をそこに集中させる必要があることが分かる。例 2, 例 3 でもデータのもとにある関数の形が大きく変化している所に節点を集めると、よい結果が得られた。したがって、3 章で述べたルール「データのもとにある関数 (信号) の形が大きく変化しているところには節点を密に入れ、ゆるやかに変化しているところには節点を疎に入れる」が妥当である



(a) あてはめの結果
(a) Result of fitting



(b) メンバーシップ関数
(b) Membership function

図 9 例 3 の結果 (折れ線型メンバーシップ関数)
Fig. 9 Result of example 3.
(Polygonal membership function)

と思われる。

図 7 から分かるように、定数型のメンバーシップ関数のとき (節点が等間隔になるとき) には、あてはめの結果がよくない。図 6 と図 7 を比較すると、メンバーシップ関数の形が節点の配置に大きく影響を与えること、その結果としてあてはめの結果が大きく変わることが分かる。

本論文で提案する方法では、決定される節点の位置は通常“最適”にはなっていない。その理由としては、3.1 節で述べたルールが簡単なこと、メンバーシップ関数が大雑把に決められていること、などが考えられる。しかし、計算例を見て分かるように、節点の位置は必ずしも最適でなくてよく、準最適な位置が求まれば十分であることが多い^{2),16)}。

むしろ、必要であればこれらの節点を初期値として、たとえば文献 4) のような最適化を行って“仕上げ”をすることも可能である。また、スプライン関数によるあてはめの専門家や計算例データベースからより詳しいルールを引き出すことができるとともに、メンバーシップ関数を最適化する方法が見つければ、節点の位置をさらに正確に計算する方法を考えるとと思われる。

なお、誤差分散の不偏推定量および AIC の値から近似関数 $S(x)$ のパラメータの数 ($n+m$) がいくらのときにより結果が得られているかの見当を付けることができる¹⁶⁾。ただし、これらは統計量であるから、有効であるのは $N \gg n+m$ の場合であり、そうでない場合には、これらの統計量は利用できないことが多い。しかし、本論文で提案する節点の決定方法自体は、この条件には無関係であることに注意したい。すなわち、データの数と近似関数のパラメータの数かなり接近していても、 $N > n+m$ であり、Schoenberg-Whitney の条件¹⁶⁾を満たしておれば 4 章で述べたアルゴリズムは実行可能である。

6. 提案する方法の特徴

本論文で提案する方法は、メンバーシップ関数を導入することによって節点の位置を間接的に制御し、試行錯誤によって節点を決定する方法である。これと、メンバーシップ関数を用いないで節点の位置を直接制御し、試行錯誤によって節点を決める従来の方法を比べることによって、提案する方法の特徴を明らかにしたい。

まず、ここで提案する方法は、節点の位置を制御し

やすい特徴がある。その理由は、次の2つである。

(1) メンバシップ関数の頂点 y_i の値を1つ変えることによって、内部の節点の位置をバランスよく変えられる。 y_i の値を変えたことによる節点への影響は、一般に x_i 付近で大きく、それから離れた点では少ない。ただし、ある特定の節点だけを少し動かすといった微調整はできない。

(2) あてはめを行う者は、メンバシップ関数を決めるだけでよい。あとは自動的によい節点の“数と位置”を計算できる。一方、メンバシップ関数を用いない方法では、“各節点数ごと別々に”試験錯誤を行ってよい節点の位置を計算する必要がある。したがって、ここで提案する方法の方が簡単である。

ここで提案する方法のもう1つの特徴は、あてはめに関する知識、経験をメンバシップ関数という形で蓄積し、新しいデータに対してそれらを利用できる可能性である。すなわち、どのようなデータに対してどのようなメンバシップ関数を用いればよかったかをデータベースとして蓄積し、そのデータベースを用いて学習することによって、メンバシップ関数の決め方を習得できる。同様なことは、メンバシップ関数を用いない節点の位置を直接扱う方法でも不可能ではないが、メンバシップ関数を用いるの方が利用者にとって易しい(優しい)ように思われる。

なお、文献19)にすべてを自動的に行う方法が提案されているが、その方法は1次および2次微係数を必要とする。したがって、本論文の例題のように誤差を持つ離散的なデータを近似する場合に、それを適用することはむずかしい。

また、どのような近似関数がほしいかは、あてはめを行う者の好み(主観)にも左右される場合があり、既存の“すべてを自動的に決定する方法”では、うまくいかないことがある。ファジィ理論では、メンバシップ関数の決定は使いたい人が好きなように決めればよい(主観を許す)ことになってるので、本論文の方法はそのような場合にも柔軟に対応できると思われる。

7. む す び

本論文では、ファジィ概念を用いて節点を決定する方法を提案し、その有効性をテストデータで確認した。

この方法では、スプライン関数の利用者はメンバシップ関数を決めるだけでよい。あとは自動的によい

節点の“数と位置”を計算できる。したがって、従来の方法(各節点数ごと別々に試験錯誤を行うことによって、よい節点の位置を探索する方法)よりも簡単である。

また、どのようなデータに対してどのようなメンバシップ関数を用いればよかったかをデータベースとして蓄積し、そのデータベースを用いて学習することによって、ある新しいデータに対する適切なメンバシップ関数を容易に作成できるようになる。

なお、本論文の主たる目的は新しい方法論を提案することであり、ここで述べた方法自体にはまだ改善の余地がある。たとえば、

- (1) 節点決定のための、さらに詳しいルール(知識)の獲得と、そのメンバシップ関数の決定方法への応用、
 - (2) メンバシップ関数の自動的な改良方法(たとえば最適化の方法)、
- などは今後の課題である。

最後に、本研究は数値計算に知識処理を応用する研究の1つと見なすことができるが、このようなアプローチは数値計算の研究において今後重要になってくるものと思われる。

謝辞 計算例の作成には、玉野佐知子さん(和歌山大学教育学部情報科学コース学生、現在K&Uソフトウェア(株))の協力を得た。建設的なコメントをくださった査読者に御礼申し上げる。文献19)は査読者にご紹介いただいたものである。

参 考 文 献

- 1) Rice, J. R.: *The Approximations of Functions*, Vol. II, p. 334, Addison-Wesley, Reading, Mass, (1969).
- 2) Rice, J. R.: *Numerical Methods, Software, and Analysis: IMSL Reference Edition*, p. 661, McGraw-Hill, New York (1983).
- 3) de Boor, C. and Rice, J. R.: *Least Squares Cubic Spline Approximation II—Variable Knots*, *Purdue University Report*, CSD TR 21 (1968).
- 4) Ichida, K., Yoshimoto, F. and Kiyono, T.: *Curve Fitting by a Piecewise Cubic Polynomial*, *Computing*, Vol. 16, pp. 329-338 (1976).
- 5) 吉本富士市, 市田浩三, 清野 武: 区分的3次関数を用いたデータ平滑化—節点の決定について—, *情報処理*, Vol. 17, pp. 200-206 (1976).
- 6) 馬渡鎮夫, 隆 雅久, 豊田吉顯: スプライン平滑化における節点の自動決定に関する一考察, *電子情報通信学会論文誌 D-II*, Vol. J 72-D-II,

- pp. 1816-1823 (1989).
- 7) Cox, M. G., Harris, P. M. and Jones, H. M.: Strategies for Knot Placement in Least Squares Data Fitting by Splines, *NPL Report*, DITC 101/87 (1987).
 - 8) Anthony, H. M., Cox, M. G. and Harris, P. M.: The Use of Local Polynomial Approximations in a Knot-Placement Strategy for Least-Squares Spline Fitting, *NPL Report*, DITC 148/89 (1989).
 - 9) Cox, M. G.: Data Approximation by Splines in One and Two Independent Variables, *The State of the Art in Numerical Analysis*, Iserles, A. and Powell, M. J. D. (eds.), pp. 111-138, Clarendon Press, Oxford (1987).
 - 10) Cox, M. G. et al.: Mathematical Software, Particularly Software for the Approximation of Functions and Data, A Panel Session, *Algorithms for Approximation*, Cox, M. G. and Mason, J. C. (eds.), pp. 689-694, Clarendon Press, Oxford (1987).
 - 11) Chang, R. L. and Pavlidis, T.: Applications of Fuzzy Sets in Curve Fitting, *Fuzzy Sets and Systems*, Vol. 2, pp. 67-74 (1979).
 - 12) 藪本 剛, 宮本定明, 村上征文: スプライン関数による確立密度推定法を用いたメンバーシップ関数の同定, 第8回「ファジィシステムシンポジウム」講演論文集, pp. 261-264 (1992).
 - 13) 下島康嗣, 福田敏男, 新井史人: スプライン関数を応用したファジィ推論, 第9回「ファジィシステムシンポジウム」講演論文集, pp. 845-848 (1993).
 - 14) Cox, M. G.: A Survey of Numerical Methods for Data and Function Approximation, *The State of the Art in Numerical Analysis*, Jacobs, D. A. H. (ed.), pp. 627-668, Academic Press, London (1977).
 - 15) Cox, M. G.: Algorithms for Spline Curves and Surfaces, *NPL Report*, DITC 166/90 (1990).
 - 16) 市田浩三, 吉本富士市: スプライン関数とその応用, p. 220, 教育出版, 東京 (1979).
 - 17) 水本雅晴: ファジィ理論とその応用, p. 359, サイエンス社, 東京 (1988).
 - 18) 矢川元基: ファジィ推論—計算力学・応用力学への応用—, p. 278, 培風館, 東京 (1991).
 - 19) Kusters, M.: Curvature-Dependent Parameterization of Curves and Surfaces, *Computer-Aided Design*, Vol. 23, pp. 569-578 (1991).
- (平成6年1月17日受付)
(平成6年6月20日採録)



吉本富士市 (正会員)

昭和18年生。昭和41年岡山大学工学部電気工学科卒業。明石工業高等専門学校助手、講師、助教授、教授を経て、現在和歌山大学教育学部情報科学教室教授。工学博士（京都大学）。昭和59年11月～60年8月まで文部省在外研究員として米国パデュー大学計算機科学科などに出張。現在の研究テーマは、多次元形状モデリング、知的図形処理、知識処理支援数値計算、並列数値計算など。著書に「スプライン関数とその応用」（共著、教育出版）がある。電子情報通信学会、日本応用数理学会、日本ファジィ学会、日本認知科学会各会員。