技術論文

ファジイモデリングにおけるメンバシップ関数配置に関する一提案

A Proposal of Allocation of Membership Functions for Fuzzy Modeling

藤目葉子 Yoko Fujime　　山口昌志 Masashi Yamaguchi

●研究開発センター　制御技術室

要旨

筆者ら[1]は，既にファジイモデリングにおいてメンバシップ関数を自動的に配置する手法を提案しているが，ノイズに弱く，また，時には不必要なメンバシップ関数を生成してしまうという問題点があった．そこで，これらの問題点を解決するために，メンバシップ関数の生成位置判断を，局所的な誤差ではなく全体的な誤差分布で行うようにした．また，メンバシップ関数の生成と削除の終了基準となる関数の導入も行った．この新たな関数は，モデルの精度と簡潔さの双方を任意にバランスさせるものである．

以上の改良点効果を数値実験，およびエンジンのモデル化において確認した．

1 はじめに

ファジイモデリングは非線形対象の入出力関係を同定し，その関係に関する知識を抽出する手法である．ファジイニューラルネットワーク(FNN)[2]は，ファジイルールの同定とメンバシップ関数の調整をニューラルネットワークの学習能力により自動化し，ファジイモデリングの有力なツールとして用いられている．FNNを構成する際には，あらかじめメンバシップ関数の数や配置を設計者が決めておく必要があった．しかし，この初期設定がその後の学習精度に大きく影響するため，最適な初期値の決定には数多くの試行錯誤と設計者の熟練が必要とされた．

そこで，筆者らは，FNNの学習の過程で，自動的にメンバシップ関数を生成，削除する機能を備えたファジイモデリングのアルゴリズムを提案した．しかし，この手法では教師データ中のノイズを吸収するために不必要なメンバシップ関数を配置してしまったり，過剰学習によりメンバシップ関数の数が必要以上に増えたりするという問題点があった．

本論文では，新しいメンバシップ関数を配置する手法を提案し，メンバシップ関数の生成と削除の終了基準を導入する．さらに，本手法の効果を数値実験により確認するとともに，非線形性の強いエンジンのモデル化に応用した．

2 従来手法

FNNの学習の過程に，メンバシップ関数を生成と削除する機能を導入し，学習の進み具合によってメンバシップ関数の増減を行う．図1は，メンバシップ関数の生成と削除を行う流れ図を示す．

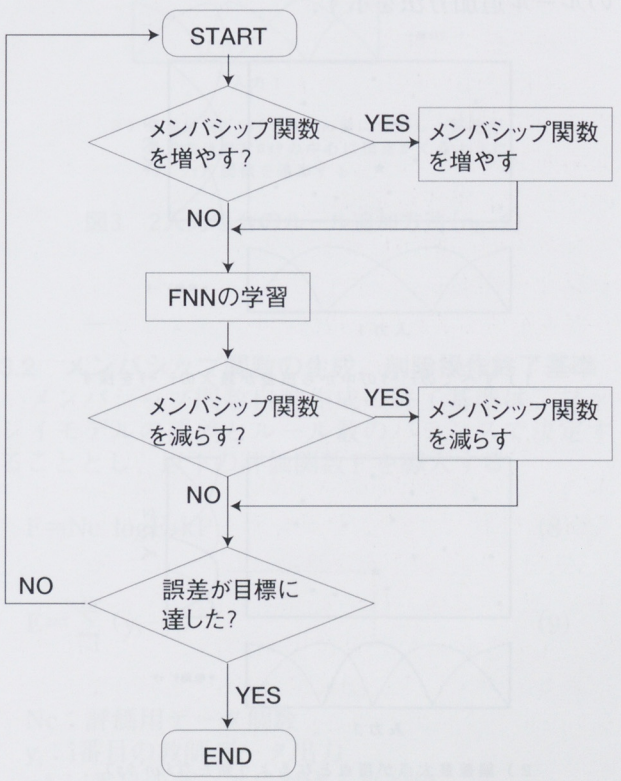


図1　メンバシップ関数生成と削除の流れ図

2.1 メンバシップ関数の生成

メンバシップ関数を増やすかどうかの判断基準は学習が順調に進んでいるか否かであり，学習における平均誤差の推移から判断する．例えばN個のデータを用いたt回目の学習で，平均誤差E(t)を

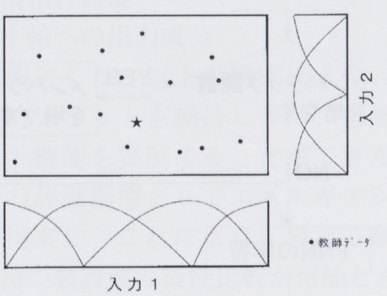
$$E(t)=\frac{1}{N}\sum_{n=1}^{N}e(n,t) \qquad (1)$$

e(n,t)：n個めのデータに対する誤差

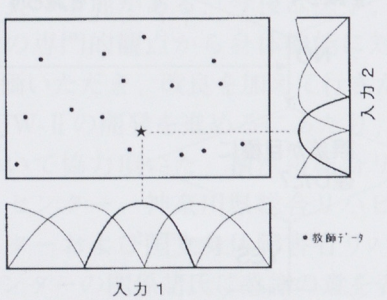
とすると，

$$E(t)\geqq\alpha E(t-1)$$
$$0\leqq\alpha\leqq 1 \qquad (2)$$

である場合にメンバシップ関数の追加を行う．メンバシップ関数の追加は，誤差が最大であったデータの入力を各入力変数に投影し，そこにメンバシップ関数が存在しなければそこを頂点とするメンバシップ関数を新たに加える．新たなメンバシップ関数の後件部の初期値には，誤差最大であったデータの出力値を与える．図2は2入力の場合のルール追加方法を示す．



1）すべてのデータの中から誤差が最大のデータを探す



2）誤差最大点が頂点となるよう新たなメンバシップ関数を挿入する

図2　2入力の場合のルール追加方法

2.2 メンバシップ関数の削除

メンバシップ関数を削除するかどうかの判断基準は後件部の線形性であり，3つ以上のルールで線形補完される場合，中央のルールは不要と判断し削除する．例えば2入力の場合，入力変数x_1，x_2に対してそれぞれM個，N個のメンバシップ関数を持っている場合を考える．x_1軸上m番目メンバシップ関数とx_2軸上n番目メンバシップ関数で表されるルールをルールmnと呼び，ルールmnの後件部の値をω_{fmn}とすると，

$$|\omega_{fmn}-\omega^{*}_{fmn}|\leqq\beta \qquad (3)$$

β：後件部の値に対して十分小さな実数値

である場合にルールmnが不要であると判断する．ここで，

$$\omega^{*}_{fmn}=\frac{p(m)-p(m-1)}{p(m+1)-p(m-1)}(\omega_{f(m+1)Xn}-\omega_{f(m-1)Xn}) \qquad (4)$$

p(m)：x_1軸上m番目メンバシップ関数の頂点位置

とし，ルールm1からルールmnまですべてのルールが不必要であると判断された場合に，x_1軸上m番目メンバシップ関数を削除する．

2.3 従来手法の問題点

1）メンバシップ関数が必要以上に生成されて，FNNの出力面がなめらかでない場合がある．
2）ノイズや局所的な誤差に過剰に反応して，不必要なルールを増やしてしまう場合がある．
3）ルール追加・削除の終了判断基準がない．

3 提案手法

3.1 メンバシップ関数の生成

従来手法でFNNの出力面がなめらかさを失ってしまったり，局所的な誤差やノイズに過敏に反応してしまう原因は，誤差最大の箇所にメンバシップ関数を生成する手法にある．そこで，入力空間を今あるメンバシップ関数の頂点で区切り，その中で誤差の2乗和の平均が最大であるブロックを

ルール追加候補とする．例えば領域sにおける重み付き平均2乗誤差Esは

$$E_S=\frac{1}{Ns}\sum_{i=1}^{Ns}\omega(x_i)\times(y_i-y_i^*)^2 \qquad (5)$$

Ns：領域sの評価用データ個数
y_i：i番目の教師データ出力
y_i^*：i番目のデータに対する推論値
$\omega(x_i)$：重み関数（挿入するメンバシップ関数）

として求める．

そして，このブロックを各入力方向に投影した領域の誤差の平均が大きい入力を，メンバシップ関数追加候補とする．

$$E_K=\sum_s\sum_{i=1}^{Ns}\omega(x_i)\times(y_i-y_i^*)^2/\sum_s Ns \qquad (6)$$

$\sum_s$：入力変数軸方向の総和

このブロックをさらにn_B個のブロックに分割して，誤差の最も大きなブロックにメンバシップ関数の追加を行う．新しくできたルールの後件部の初期値は以下の式で与える．

$$\sum_{i=1}^{Ns}\omega(x_i)\,y_i/\sum_{i=1}^{Ns}\omega(x_i) \qquad (7)$$

図3は，メンバシップ関数の頂点で区切った各部分空間のうち，誤差最大のブロックにメンバシップ関数を挿入する様子を示す．
これにより，局所的な誤差ではなくブロック内の誤差の分布を見てメンバシップ関数を配置できるようになり，不必要なルールが局所的に集中して生成されることを防いで，少ないルール数で誤差の小さなFNNを設計できるようになると期待される．また，結果的に出力面もなめらかにできる．

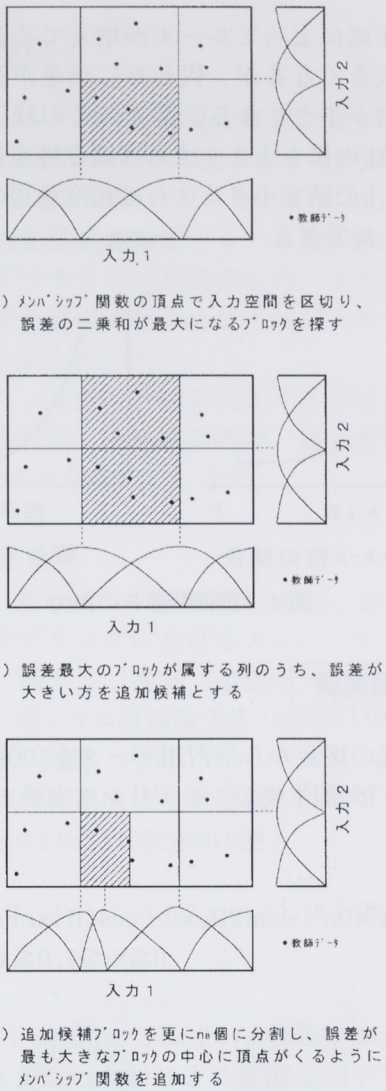


図3　2入力場合のルール追加方法（n_B=2）

3.2　メンバシップ関数の生成，削除操作終了基準

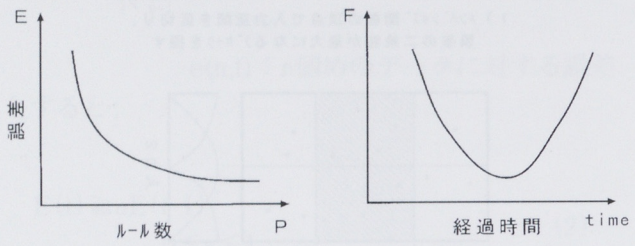
メンバシップ関数自動作成の終了基準は，ファジイモデルの誤差とルール数のバランスで決定することとし，以下の評価関数Fを導入する．

$$F=Nc\ \log E+kP \qquad (8)$$

$$E=\sum_{i=1}^{Nc}(y_i-y_i^*)^2 \qquad (9)$$

Nc：評価用データ個数
y_i：i番目の教師データ出力
y_i^*：i番目のデータに対する推論値
P：メンバシップ関数の数
k：重み関数

自動作成によってルールが増えてくると第2項の値が大きくなるが，代わりに誤差が減るため第1項の値が小さくなる．**図4**にこの様子を示す．重み関数kの値を大きくすれば簡潔性重視のモデルを，またkの値を小さくすれば精度重視のモデルを得ることができる．



誤差とルール数の関係　　関数Fの出力

図4　評価関数Fの出力

4 数値実験

(10)式の関数から学習用データを100個，評価用データを100個生成し，モデリング実験を行った．

$$y=\sin(2\Pi x_1^2)+\sin(2\Pi\sqrt{x_2})+\cos(\Pi(x_1+x_2)) \quad (10)$$
$$0\leqq x_1\leqq 1, 0\leqq x_2\leqq 1$$

x_1，x_2はランダムに生成し，出力yに$\sigma=0.1$のガウスノイズを加えたものについてモデリング実験を行った．各部分領域の分割数は$n_B=1$，ルール追加のしきい値は$\alpha=0.99999$，ルール削除のしきい値は$\beta=0.01$とした．

表1は，ノイズなしの場合のモデリング誤差である．データ生成から各手法によるモデリングまでを，3通りについて実行した結果の平均値を示している．重み係数kはそれぞれ0.002と2.0の2通りについて行った．k=0.002の時はモデル精度が重視され，k=2.0の時にはメンバシップ関数の少ない簡潔なモデルが重視されている．提案手法により，データにノイズが含まれていない場合にはモデル精度は少し低下するが簡潔なモデルが得られた．

表2は，データに$\sigma=0.1$のノイズが含まれている場合のモデリング結果である．それぞれ3通りの実験結果の平均値である．ノイズ存在下では，本手法は簡潔でしかも精度のよいモデルを得ることができた．

表1　実験結果（ノイズなし）

手法	重み係数k	モデル誤差	メンバシップ関数の数
提案	0.002	0.054	9.0
	2.0	0.090	8.3
従来	0.002	0.046	26.3
	2.0	0.046	23.7

表2　実験結果（ノイズあり，$\sigma=0.1$）

手法	重み係数k	モデル誤差	メンバシップ関数の数
提案	0.002	0.107	7.0
	2.0	0.113	5.7
従来	0.002	0.136	14.7
	2.0	0.136	14.7

5 エンジンモデリングへの応用

5.1 教師データ

4サイクルエンジンを使用し，エンジン回転数およびスロットル開度を一定にした定常運転時の体積効率の値を実機から収集した．入力1をエンジン回転数，入力2をスロットル開度，出力を体積効率とする．教師データを**図5**に示す．

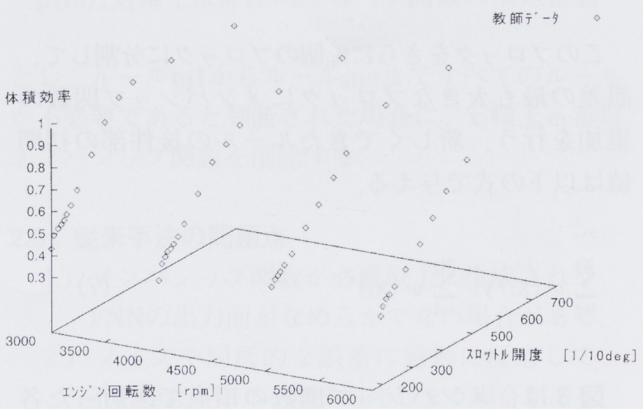
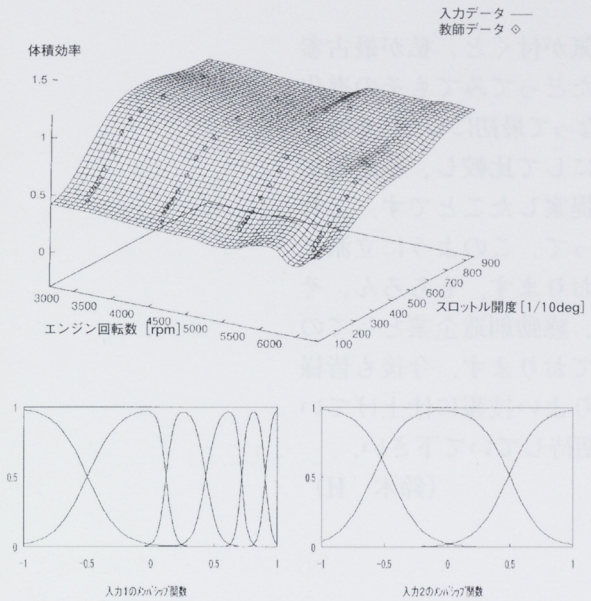


図5　教師データ

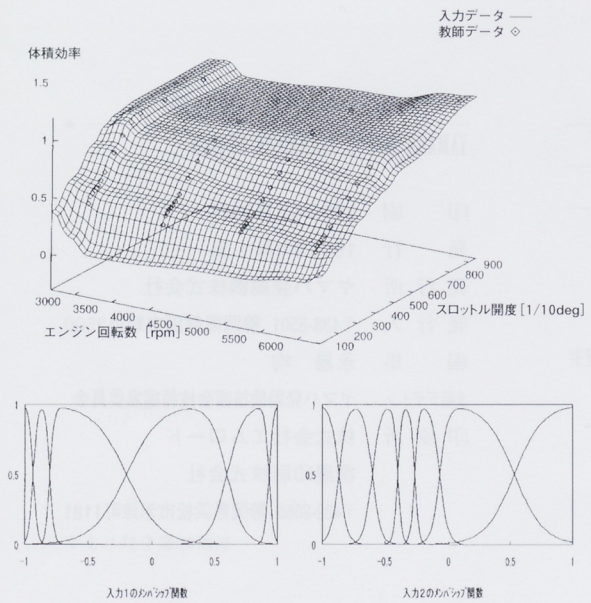
5.2 モデリング結果の比較

モデリングに使用した式(2)はα=0.99999，式(3)はβ=0.01，式(8)はk=0.002とする．提案手法および従来手法における出力面の様子を**図6**に，モデリング誤差を**表3**に示す．

提案手法を使用すると，少ないルール数で従来手法と同等の誤差を実現でき，FNNの出力面も滑らかになった．ルール数が半分になったということは演算時間を半分にできるということであり，将来実応用を考えた場合にメリットのある結果である．



a） 提案手法



b） 従来手法

図6 モデリング結果

表3 誤差比較

手法	重み係数k	モデル誤差	ルール数
提案	0.002	0.023	18(6×3)
従来	0.002	0.019	42(6×7)

6 おわりに

メンバシップ関数を自動で配置するアルゴリズムを改良し，最終的な出力面のなめらかさの改善，必要なルール数の削減，対ノイズ性の向上を行い，さらに，自動作成を誤差とルール数の最適な時点で自動終了できる手法を提案した．さらに，本手法の有効性を数値実験およびエンジンモデリングへの応用により示した．

おわりに，本件の共同研究先である名古屋大学の内川嘉樹教授，古橋武助教授，橘完太氏に，紙面をお借りして深謝いたします．

■参考文献

（1）橘，長谷川，古橋，内川，藤目，山口：ファジイモデリングにおけるメンバーシップ関数配置に関する一提案，第13回ファジイシステムシンポジウム講演論文集 (1997)87-90

（2）堀川，古橋，内川：ファジイニューラルネットワークの構成法と学習法，日本ファジイ学会誌 Vol.4,No.5,(1992)906-928

● 著者

藤目葉子

山口昌志