ファジィ推論システムの能力と学習法に関する研究

2016年 3月

宮島 洋文

目 次

第1章　序論

1.1　はじめに

近年における科学技術の発達は目覚ましいものがあり, その中でもコンピュータは単独またはネットワークとして日々その処理能力を向上させている. コンピュータが登場して以来, 科学技術上の多くの問題をコンピュータを用いて解こうとする研究がなされている. この場合, 二つの研究方向があり, 命題論理や述語論理などの論理システムにもとづくことやコンピュータの高度な計算能力を用いて問題を厳密に解いて最適解や準最適解求めることを目指すハードコンピューティングの分野と, 必ずしも厳密解にとらわれず, 比較的短い時間で良好な解を見つけることを目指すソフトコンピューティングの分野である [1, 2, 3, 4, 5]. ソフトコンピューティングの基本的な考え方は, 人間の脳や進化の過程に倣ったモデルを用いて, 問題を知識と数値を使って解こうとすることにある. 人間の脳はコンピュータに比べはるかに情報伝達速度が遅いにも関わらず情報処理能力はコンピュータを凌ぐことがある. これは, コンピュータが逐次型の情報処理を行うのに対して人間の脳が並列分散処理をおこなっていることによる. それゆえ, 人間の知的情報の並列分散処理の実現を目指す研究がさまざまな方向から盛んに行われている [5, 6, 7, 8]. さらに, この分野の研究は, 実際に人工知能, データマイニングや機械学習等の様々な分野への展開がはかられている. ソフトコンピューティングとは, 「取り扱い易さ, 頑健性, 低コストを達成するために, 不確実性をどこまで容認するかを探り, 高度な精確性を要求せずにシステムを解析・設計する計算方式」[5, 6, 7] と解釈され, 研究手法としては, 図 1.1 に示すように, ニューラルネットワーク, ファジィシステム, 確率システム, 進化的計算システム等が用いられる [1, 2, 8, 64]. このなかで, ニューラルネットワークや確率システムは, 問題解決への応用は容易であり, 理論的な解析も十分に行われているが, 構築されたシステムの中身はブラックボックスであり, その解釈は必ずしも容易ではない [3, 4, 9]. 進化的計算は様々な分野に容易に応用されるが, 理論的裏付けや結果の意味解釈が必ずしも十分とはいえない. ファジィシステムは, 人間の知識の定性的概念や推論のプロセスをモデル化できるが, 高い近似精度の実現や定量的な解析方法は知られていない. それゆえ, これらの方法は, 単独または複数の手法を組み合わせることが多く見られる [4, 26, 27, 50]. 今後の複雑化する社会へのソフトコンピューティングの応用を考える場合には, 安心かつ安全なシステム構築が欠かせない. そのために, 本研究ではシステムの中身の解釈と近似精度の向上の議論が比較的容易な方法論としてファジィ推論システ

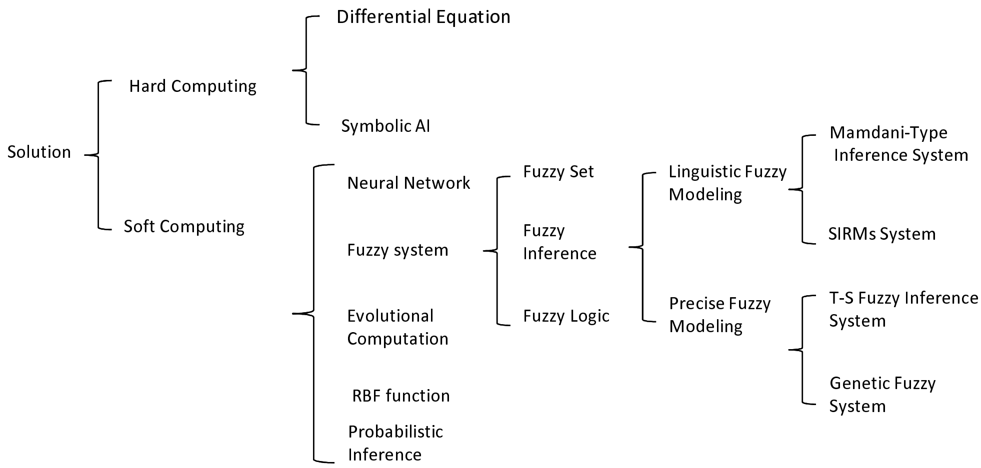


図 1.1: ファジィ推論システムの位置付け

ムを用いる [10, 11, 12, 18, 56]. このファジィ推論システムの能力解析, 近似性能と意味解釈を通して, 問題解決へのソフトコンピューティング的アプローチ法を提案する.

ファジィシステムの基礎をなすファジィ集合のアィデアは, 1965 年に Zadeh により提唱された [12]. 従来の集合論は, 集合に所属するまたは所属しない, という 2 値の論理にもとづくものである. 一方, ファジィ集合では, 集合への所属度を, 連続値により表すことで, 集合への所属の "あいまいさ" を連続的に扱うことが可能となる. この所属度を表す関数をメンバーシップ関数と呼ぶ. ファジィ集合はこのメンバーシップ関数により定義される. このような論理の一般化により, 単に集合の所属度だけでなく, "暑い" や "寒い" のようなあいまいさの処理が可能となる. このファジィ集合を if-then 形式の従来の論理に応用したものがファジィ推論であり, 結果としてあいまいさや不完全な知識を扱うことのできるファジィ推論システムが開発された. これにより, 人間が行うような曖昧さを含む情報処理がコンピュータ上で実装可能となった.

ファジィ理論の代表的な応用分野として, ファジィ推論は人間に理解可能な推論規則により構成されており, また, 非線形システムを扱うことができるという特徴がある. Mamdani は, 制御分野において, このことを初めて実現した [13]. すなわち, 従来は熟練した技術者が行っていた作業をファジィ推論に基づくアルゴリズムを実装したコンピュータに行わせるシステム (ファジィエキスパート システム) の誕生である. その後, この分野で多くの成功事例が発表された. 同時に, 人間がファジィ推論の規則の設定を行うためには, 推論規則の試行錯誤や調整を人間の手で行う必要があり, 多くの時間と労力を必要とした. 一方で, システムの近似精度は必ずしも十分とはいえないものであった. そこで, （学習用の）入出力データを用いて推論規則を自動的に構築する研究がおこなわれるようになった [15, 16]. このこ

とは, 人間の代わりにコンピュータが自動的に推論ルールの構成・調整を行うことが可能となり, 人間の労力が大幅に削減されただけでなく, 精度の高いファジィ推論システムの構築が可能となった. ただし, システムの精度改善を目指すことは, 必ずしもファジィ推論システムの説明能力の改善を意味しない. 精度 (Accuracy) と説明能力 (Interpretability) はトレードオフの関係にあることが明らかとなり, 以後どちらを目指すかにより異なる研究分野が存在することとなった [17, 18, 19, 54]. 前者は, モデルの構造やファジィルールについて制限を加えないモデルであり, 高木・菅野モデルの提案をはじめとして非常に多くのモデルが導入された [14]. これらは主に与えられた入出力データから最急降下法に基づいて自動的にファジィシステムの構築を目指す研究である [9]. 最急降下法に基づく局所探索によるシステム構築を行うので, 推論誤差や学習時間の増加等の問題があることが指摘されている. これを改善する試みとして, ファジィ推論ルールの学習則に注目した手法 [15,63] や推論モデルの構造に注目し, 推論ルールを逐次的に生成したり [20, 34], 不必要な推論ルールを削除する方法 [21, 22, 66], また, 一般化された目的関数を使って柔軟な推論を行う方法も提案されている [24]. また, ファジィ推論モデルと別の最適化アプローチを組み合わせ, 例えば, 遺伝アルゴリズムや PSO(Particle Swarm Optimization) による大域探索との組み合わせ [20, 26, 27, 33], 自己組織化やベクトル量子化など最急降下法により局所探索を行う方法と組み合わせた方法 [24,25,36] が提案されており, その有効性が示されている.

一方, 説明能力をもつファジィシステムの研究についても, 学習機能を導入する研究が行われた. ただし, この場合はファジィルールの意味解釈が可能な範囲での, すなわち, パラメータの領域が制限された範囲での学習を行うこととなる [54]. Mamdani モデルの前件部を固定し, 後件部を定数パラメータとして学習するモデルや Shi らの各変数の属性ごとに学習するモデルが知られている [28]. また, 遺伝的アルゴリズムのような進化的方法によりパラメータの領域を制限して学習する多くのモデルが提案されている.

このように, 高い近似精度をもつ, または説明能力の高いファジィ推論システムの構築法では, 主に以下の方法が用いられている. 1) 推論モデルの構造を更新する方法, 2) 遺伝的アルゴリズムや PSO のようにあらかじめ解の探索空間を制御して大域探索を実現する方法, 3) 自己組織化マップ法やベクトル量子化法を前処理として実効する場合のように, あらかじめソフトマッチングで大域探索を実現することにより初期パラメータを決定する方法が考えられている. しかしながら, このような学習システムの構築法だけでは, 説明能力の不足だけでなく、入力変数の増加に伴う学習の困難さを克服することが難しいことも知られている [19,26,27].

湯場崎らは、説明能力のあるファジィ推論システムとして 1 変数のルールモジュールからなる SIRMs (Single Input Rule Modules) モデルを提案し, その有効性を示した [30, 62]. このモデルはある種の非線形な問題にも有効なことも知られているが, EX-OR や複雑な制御問題では必ずしも有効でないことが知られている. さらに, このモデルを拡張するために関数機能を持たせたモデルも提案されているが,

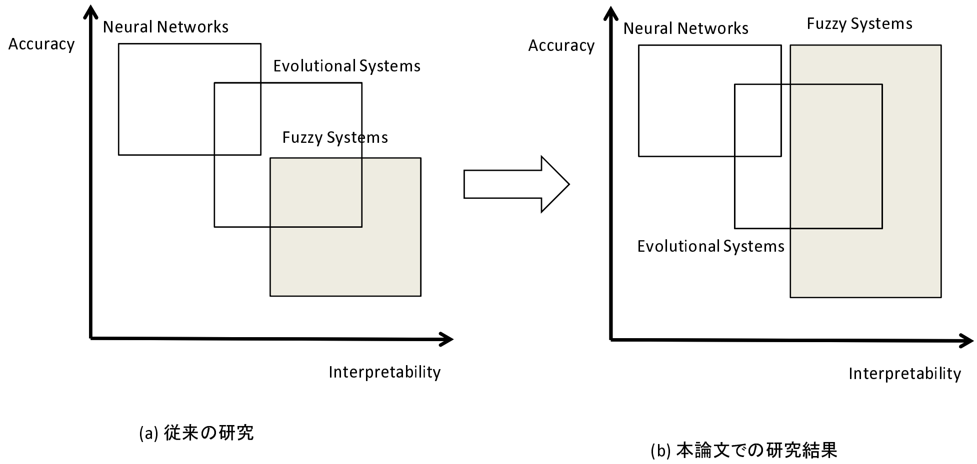


図 1.2: 研究の位置付け

必ずしも能力は十分ではない [31]. そこで, 少数の入力変数からなるルールモジュールからなる推論モデル SNIRMs (Small Number of Input Rule Modules) モデルが提案され, その有効性が示されている [32,59,68]. このモデルは各モジュールが少数個 (1,2 または 3 程度) の入力変数からモジュールを構成するものであり, SIRMs モデルの自然な一般化モデルとなっている. このモデルは SIRMs モデルと同様に単純な構造をもっているが, SIRMs モデルに比べてモジュールの個数が増加する欠点をもっている. モデルの汎化能力を上げるにはパラメータの数は少ない方がよいので, いかにしてモジュールやパラメータの個数を効率的に抑えるかが問題となる. その他, ニューラルネットワークと SIRMs モデルを組み合わせたモデルや, SIRMs モデルの後件部においてすべての変数を統合するモデルも提案されている. いずれの場合も, 十分な近似精度を実現できるが, 説明能力は十分とはいえない. またモデルの理論的な能力も示されていない.

1.2 本論文の構成

本論文では, はじめに従来研究において高い近似精度 (Accuracy) や説明能力 (Interpretability) をもつモデルやその汎化モデルについて, 理論的な能力, 学習システムとしての近似精度や説明能力, また入力変数の増加に伴う近似能力の柔軟性について検討する. この結果を踏まえて, 新しいモデルを提案し, 万能性 (Universal approximation capability), 学習システムとしての近似能力や説明能力, また入力変数の増加に伴う近似能力の柔軟性に関しての有効性を示す. 図 1.2 に本論文の研究の位置づけを示す.

以下, 本論文の構成を示す.

第 1 章では, ファジィ理論の歴史と工学的背景, およびファジィ推論システムにおける最近の研究と本論文の内容と構成について述べる.

第 2 章は, ファジィ集合とその応用であるファジィ推論モデルについて述べる. はじめに, ファジィ推論システムの従来モデルとして知られる TS 型, Mamdani 型と簡略型ファジィ推論システムを導入し, 学習による推論ルールの決定方法について述べる. さらに, 各モデルについてこれまでに得られている結果を与える.

第 3 章は, 高い説明能力をもつモデルとして知られているファジィ推論システムとその汎化モデルについて述べる. すなわち, 高い説明能力と近似精度を実現する新しいモデルとして属性型ファジィ推論モデルの提案を行い, このモデルの近似精度や万能性に関する理論的な解析を示す.

第 4 章は, ベクトル量子化を用いたファジィ推論システムの近似能力について述べている. すなわち, ニューラルガス等のベクトル量子化とファジィ推論システムを組み合わせた新しいモデルとその学習法を提案し, 数値シミュレーションにより近似精度や推論ルール数において有効性を示す。.

第 5 章は, メタヒューリスティクスを用いたハイブリッドなファジィ推論モデルとその学習法について述べている. すなわち, ファジィ推論システムに EM (Electromagnetism-like Mechanism) やランダムサーチのようなメタヒューリスティクを組み合わせたハイブリッドなモデルとその学習法を提案し, 数値シミュレーションによりその有効性を示す.

第 6 章は, 高い説明能力と近似精度をもつモデルとして提案された SNIRMs モデルの能力について述べる. すなわち, SNIRMs モデルの説明能力や近似精度について, 理論と数値シミュレーションにより明らかにする.

第 7 章は, 前章までの結果を踏まえて, 線形入力型 SIRMs モデルを提案し, その能力について述べる. すなわち, 第 1 段階で入力変数の線形変換を行い, 第 2 段階で SIRMs モデルによる出力導出を行う線形入力型 SIRMs ファジィ推論システムとその学習法を提案し, 理論と数値シミュレーションにより, 従来モデルやその汎化モデルと比べて高い近似能力をもつことを示す. また, 学習後に得られたファジィ推論ルールの意味解釈を与える方法を提案し, その有効性を示す.

第 8 章は, まとめである.

第2章　準備

本章では, 以下の章に必要なファジィ推論や数学的予備概念を与える.

2.1　ファジィ集合

広く用いられている集合であるクリスプ集合は, 集合の所属が客観的に明確に定義されている. 例えば, 「温度が 30°C 以上」はクリスプ集合である. しかしながら, 「温度が暑い」は人間の主観に依存しており, 集合への所属が客観的に明確に定義されていない. このようなあいまいな集合を定量的に解析するため, ファジィ集合を用いる. ファジィ集合は, 数学的には集合と以下に示すメンバーシップ関数により定義される [12]. 以下では, 集合が明らかな場合は, ファジィ集合とメンバーシップ関数は区別しない.

全体集合を X とする. ファジィ集合 A は, 次のようなメンバーシップ関数 μ_A により定義される.

$$\mu_A : X \rightarrow [0,1] \tag{2.1}$$

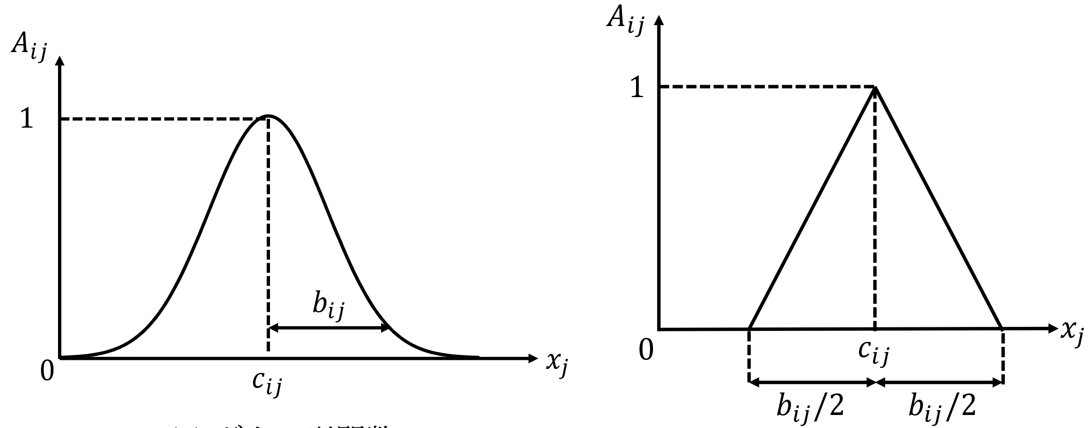
メンバーシップ関数としては, 以下のようなガウス型関数および三角型関数が用いられる (図 2.1 参照).
ガウス型関数

$$A_{ij}\left(x_j\right) = \exp\left(-\frac{1}{2}\left(\frac{x_j - c_{ij}}{b_{ij}}\right)^2\right) \tag{2.2}$$

三角型関数

$$A_{ki}(x_j) = \begin{cases} 0\left(x_j < c_{ij} - \frac{b_{ij}}{2}\right) \\ \frac{2}{b_{ij}}[x_j - \left(c_{ij} - \frac{b_{ij}}{2}\right)]\left(c_{ij} - \frac{b_{ij}}{2} \leq x_j \leq c_{ij}\right) \\ -\frac{2}{b_{ij}}[x_j - \left(c_{ij} + \frac{b_{ij}}{2}\right)]\left(c_{ij} < x_j \leq c_{ij} + \frac{b_{ij}}{2}\right) \\ 0\left(x_j > c_{ij} + \frac{b_{ij}}{2}\right) \end{cases} \tag{2.3}$$

ここに, c_{ij}, b_{ij} はそれぞれメンバーシップ関数 $A_{ij}(x_j)$ の中心と幅を表す.



(a) ガウス型関数

(b) 三角型関数

図 2.1: メンバーシップ関数の例

2.2 ファジィ推論システム

2.2.1 Mamdani 型ファジィ推論法

ファジィ集合を用いて, ファジィ推論システムを導入する. 自然数 k に対して $Z_k = \{1, 2, \cdots, k\}$, すべての実数の集合を $\boldsymbol{R}$ とする. 入力 $\boldsymbol{x} = (x_1, \cdots, x_m)$, 出力 y^* とする ($x_j \in \boldsymbol{R}$, $j \in Z_m$).

ファジィ集合を用いて, ファジィ推論は, 次のような if$\cdots$then$\sim$ 形式のルールで表される.

$$R_i : if\ x_1\ is\ A_{i1}\ and\ \cdots\ and\ x_m\ is\ A_{im}\ then\ y\ is\ B_i \tag{2.4}$$

ここで, $A_{ij}(j \in Z_m)$ は入力要素 x_j に関する前件部のメンバーシップ関数, B_i は出力要素 y に関する後件部のメンバーシップ関数, $i \in Z_r$ である.

推論の出力 y は, 次の式により導出される [13].

$$y = \frac{\int y \cdot \max_i \{\min\{A_{i1}(x_1), \cdots, A_{im}(x_m), B_i(y)\}\} dy}{\int \max_i \{\min\{A_{i1}(x_1), \cdots, A_{im}(x_m), B_i(y)\}\} dy} \tag{2.5}$$

2.2.2 簡略型ファジィ推論法

ファジィ推論法の一つである簡略型ファジィ推論法は, 次のような if$\cdots$then$\sim$ 形式のルールにより構成される.

$$R_i : if\ x_1\ is\ A_{i1}\ and\ \cdots\ and\ x_m\ is\ A_{im}\ then\ y\ is\ w_i \tag{2.6}$$

ここで, $A_{ij}(j{\in}Z_m)$ は入力要素 x_j に関するメンバーシップ関数, w_i は実数値, $i{\in}Z_r$ である.

推論規則 R_i に対する適合度 μ_i は, 次式により求めることができる.

$$\mu_i = \Pi_{j=1}^{m} A_{ij}(x_j) \tag{2.7}$$

推論の出力 y は, 次の式により導出される.

$$y = \frac{\sum_{i=1}^{r} \mu_i w_i}{\sum_{i=1}^{r} \mu_i} \tag{2.8}$$

2.2.3 TS ファジィ推論法

簡略型ファジィ推論法を一般化したものとして, TS(Takagi Sugeno) ファジィ推論法が知られている [14, 15].

TS ファジィ推論法の推論規則は, 以下のように与えられる.

$$R_i : if\ x_1\ is\ A_{i1}\ and \cdots and\ x_m\ is\ A_{im}\ then\ y\ is\ f_i(\boldsymbol{x}) \tag{2.9}$$

ここで, f_i は $x_1, \cdots, x_m$ を入力とする m 変数関数, $i{\in}Z_r$ である. f_i を定数に制限した場合が簡略型ファジィ推論法となる. 推論規則 R_i に対する適合度 μ_i は, 次式により求めることができる.

$$\mu_i = \Pi_{j=1}^{m} A_{ij}(x_j) \tag{2.10}$$

推論の出力 y は, 次の式により導出される.

$$y = \frac{\sum_{i=1}^{r} \mu_i f_i}{\sum_{i=1}^{r} \mu_i} \tag{2.11}$$

2.3 ファジィ推論モデルの学習法

$\boldsymbol{D} = \{(x_1^p, \cdots, x_m^p, y_p^*) | p{\in}Z_P\}$ を学習用データの集合とする. ここに, $\boldsymbol{x}^p = (x_1^p, \cdots, x_m^p)$ と y_p^* は p 番目の入力とその出力である. 以下, 本論文では, $\boldsymbol{D}$ を学習用データの集合として用いる. ファジィ推論モデルに入力 $\boldsymbol{x}^p$ を与えたときの推論出力を y_p とすると, 学習用データに対する平均二乗誤差 (Mean Square Error : MSE)E は, 次のように表される.

$$E = \frac{1}{P} \sum_{p=1}^{P} (y_p - y_p^*)^2 \tag{2.12}$$

E の最小化問題を解くことで, ファジィ推論モデルの適切なパラメータを求めることができる. 最小化問題を解く手法の一つとして, 以下の更新式を用いる最急降下法がある [9, 10, 11, 15].

$$\alpha(t+1) = \alpha(t) - K_\alpha \frac{\partial E}{\partial \alpha}|_{\alpha=\alpha(t)} \tag{2.13}$$

ここで, α は更新を行うパラメータ, t は学習回数, K_α は学習係数である.

以下では, 最急降下法に基づいたファジィ推論モデルの学習法を導入する.

2.3.1 Mamdani 型ファジィ推論モデルの学習

Mamdani の提案した Min-Max 重心法においては, 推論規則中の各ファジィ集合のメンバーシップ関数は人間の手により作成されていた. そのため, 自然言語による解釈が容易で説明能力の高いモデルであった. しかしながら, Min-Max 重心法は出力導出過程において不連続関数である min, max 関数が用いられており微分の計算が困難であるため, 最急降下法の適用が困難である. そこで, 以降では, 簡略型ファジィ推論法において, ルールの後件部パラメータのみを学習により決定する手法のことを Mamdani 型ファジィ推論法と呼ぶ. Mamdani 型ファジィ推論法においては学習の前後においてルールの前件部のメンバーシップ関数は変化しない. そのため, 学習開始時に人間の裁量により定義されたメンバーシップ関数が学習終了後も推論規則中に用いられており, 自然言語による解釈が容易で説明能力の高いモデルとなっている. 後件部パラメータ w_i の更新式は, 式 (2.16) のようになる.

Mamdani 型ファジィ推論法の学習アルゴリズム (A-M) は, 以下のようになる [15].

[**学習アルゴリズム A-M**]

Step A-M1 : しきい値 θ, 最大学習回数 T_{max} を与える. 推論規則の初期位置は等間隔に配置する. 整数 H に対して, 推論規則数 n を $n = H^m$ とおく. $t = 1$ とする.

Step A-M2 : パラメータ b_{ij}, c_{ij}, w_i を初期化する.

Step A-M3 : $p = 1$ とおく.

Step A-M4 : データ $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ を与える.

Step A-M5 : 式 (2.7) と (2.8) より, μ_i と y を求める.

Step A-M6 : 式 (2.16) より, パラメータ w_i を更新する.

Step A-M7: $p = P$ ならば Step A-M8 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A-M4 へ行く.

Step A-M8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A-M3 へ, $E(t) \leq \theta$ または $t > T_{max}$ ならば学習を終了する.

2.3.2 簡略型ファジィ推論モデルの学習

ガウス型関数を用いるとき, 簡略型ファジィ推論法の推論規則中の各パラメータの更新式は, 以下のようになる ($i{\in}Z_n$ and $j{\in}Z_m$)[9,10,11,29].

$$
\begin{aligned}
\mu_i &= \Pi_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{x_j - c_{ij}}{b_{ij}}\right)^2\right) && (2.14)\\
y &= \frac{\sum_{i=1}^{n} \mu_i w_i}{\sum_{i=1}^{n} \mu_i} && (2.15)
\end{aligned}
$$

として,

$$
\begin{aligned}
w_i(t+1) &= w_i(t) - K_w \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) && (2.16)\\
c_{ij}(t+1) &= c_{ij}(t) - K_c \frac{\mu_j}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (w_i - y) \cdot \frac{x_j - c_{ij}}{b_{ij}^2} && (2.17)\\
b_{ij}(t+1) &= b_{ij}(t) - K_b \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (w_i - y) \cdot \frac{(x_j - c_{ij})^2}{b_{ij}^3} && (2.18)
\end{aligned}
$$

また, 三角型関数を用いるとき, 簡略型ファジィ推論法の推論規則中の各パラメータの更新式は, 以下のようになる ($i{\in}Z_n$ and $j{\in}Z_m$).

$$
\begin{aligned}
A_{ij}(x_j) &= \begin{cases} 0\left(x_j < c_{ij} - \frac{b_{ij}}{2}\right) \\ \frac{2}{b_{ij}}[x_j - \left(c_{ij} - \frac{b_{ij}}{2}\right)]\left(c_{ij} - \frac{b_{ij}}{2}{\leq}x_j{\leq}c_{ij}\right) \\ -\frac{2}{b_{ij}}[x_j - \left(c_{ij} + \frac{b_{ij}}{2}\right)]\left(c_{ij} < x_j{\leq}c_{ij} + \frac{b_{ij}}{2}\right) \\ 0\left(x_j > a_{ij} + \frac{b_{ij}}{2}\right) \end{cases} && (2.19)\\
\mu_i &= \Pi_{j=1}^{m} A_{ij}(x_j) && (2.20)\\
y &= \frac{\sum_{i=1}^{n} \mu_i w_i}{\sum_{i=1}^{n} \mu_i} && (2.21)
\end{aligned}
$$

として,

$$
\begin{aligned}
w_i(t+1) &= w_i(t) - K_w \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) && (2.22)\\
c_{ij}(t+1) &= c_{ij}(t) - K_c \frac{\mu_j}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (w_i - y) \cdot \frac{\partial A_{ij}(x_j)}{\partial c_{ij}}|_{c_{ij}=c_{ij}(t)} && (2.23)\\
b_{ij}(t+1) &= b_{ij}(t) - K_b \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (w_i - y) \cdot \frac{\partial A_{ij}(x_j)}{\partial b_{ij}}|_{b_{ij}=b_{ij}(t)} && (2.24)
\end{aligned}
$$

ここで,

$$\frac{\partial A_{ij}(x_j)}{\partial c_{ij}} = \begin{cases} 0\left(x_j < c_{ij} - \frac{b_{ij}}{2}\right) \\ -\frac{2}{b_{ij}}\left(c_{ij} - \frac{b_{ij}}{2} \leq x_j \leq c_{ij}\right) \\ \frac{2}{b_{ij}}\left(c_{ij} < x_j \leq c_{ij} + \frac{b_{ij}}{2}\right) \\ 0\left(x_j > c_{ij} + \frac{b_{ij}}{2}\right) \end{cases} \tag{2.25}$$

$$\frac{\partial A_{ij}(x_j)}{\partial b_{ij}} = \begin{cases} 0\left(x_j < c_{ij} - \frac{b_{ij}}{2}\right) \\ \frac{2c_{ij}}{b_{ij}^2}\left(c_{ij} - \frac{b_{ij}}{2} \leq x_j \leq c_{ij}\right) \\ -\frac{2c_{ij}}{b_{ij}^2}\left(c_{ij} < x_j \leq c_{ij} + \frac{b_{ij}}{2}\right) \\ 0\left(x_j > c_{ij} + \frac{b_{ij}}{2}\right) \end{cases} \tag{2.26}$$

ガウス型関数を用いる学習アルゴリズムは, 以下のようになる.
[学習アルゴリズム **A**]
Step A1 : しきい値 θ, 最大学習回数 T_{max} を与える. 推論規則の初期位置は等間隔に配置する. 整数 H に対して, 推論規則数 n を $n = H^m$ とおく. $t = 1$ とおく.
Step A2 : パラメータ c_{ij}, b_{ij}, w_i を初期化する.
Step A3 : $p = 1$ とおく.
Step A4 : データ $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ を与える.
Step A5 : 式 (2.7) と (2.8) より, μ_i と y を求める.
Step A6 : 式 (2.17), (2.18), (2.16) より, パラメータ c_{ij}, b_{ij}, w_i を更新する.
Step A7: $p = P$ ならば Step A8 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A4 へ行く.
Step A8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A3 へ, $E(t) \leq \theta$ または $t > T_{max}$ ならば学習を終了する.

同様にして, 三角型メンバーシップ関数の場合も導入できる.

2.3.3 TS ファジィ推論モデルの学習

メンバーシップ関数としてガウス型関数を用いる. また, $f_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^m w_{ij} x_j (w_{i0}, w_{i1}, \cdots, w_{im} \in \boldsymbol{R})$ とするとき, 推論規則中の各パラメータの更新式は, 以下のようになる.

$$\mu_i = \Pi_{j=1}^m \exp\left(-\frac{1}{2}\left(\frac{x_j - c_{ij}}{b_{ij}}\right)^2\right) \tag{2.27}$$

$$y = \frac{\sum_{i=1}^{n} \mu_i \left(w_{i0} + \sum_{j=1}^{m} w_{ij} x_j \right)}{\sum_{i=1}^{n} \mu_i} \tag{2.28}$$

として,

$$\frac{\partial E}{\partial w_{i0}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \tag{2.29}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot x_j \tag{2.30}$$

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (f_i - y) \cdot \frac{x_j - c_{ij}}{b_{ij}^2} \tag{2.31}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^*) \cdot (f_i - y) \cdot \frac{(x_j - c_{ij})^2}{b_{ij}^3} \tag{2.32}$$

学習アルゴリズムは, 以下のようになる.

[**学習アルゴリズム A-TS**]

Step A-TS1 : しきい値 θ, 最大学習回数 T_{max} を与える. 推論規則の初期位置は等間隔に配置する. 整数 H に対して, 推論規則数 n を $n = H^m$ とおく. $t = 1$ とおく.

Step A-TS2 : パラメータ c_{ij}, b_{ij}, w_{ij} を初期化する.

Step A-TS3 : $p = 1$ とおく.

Step A-TS4 : データ $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ を与える.

Step A-TS5 : 式 (2.10) と (2.11) より, μ_i と y を求める.

Step A-TS6 : 式 (2.31), (2.32), (2.29), (2.30) より, パラメータ c_{ij}, b_{ij}, w_{ij} を更新する.

Step A-TS7: $p = P$ ならば Step A-TS8 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A-TS4 へ行く.

Step A-TS8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A-TS3 へ, $E(t) \leq \theta$ または $t > T_{max}$ ならば学習を終了する.

2.4 モデルの万能性

ソフトコンピューティングにおけるモデルの近似能力を示す重要な性質として, 万能性 (universal approximation capability) がある. この性質を満たすモデルは任意の連続関数を任意の精度で近似することが可能である. つまり, あらゆる連続関数を高い精度で近似できることが理論的に保証された, 能力の高いモデルとなる.

モデルの万能性を, 数学的なことばで定義する [9].

[定義 1] A を集合とする．A の閉包 (closure) $[A]$ とは, A のすべての集積点 (limit point) を含む集合である．
[定義 2] 集合 A が閉集合 (closed set) となるのは, $A=[A]$ が成り立つ場合である．
[定義 3] B を A の部分集合とする．B が A において稠密 (dense) であるのは, $[B]=A$ となる場合である．

稠密の定義は, 近似理論の立場からは, B が A において稠密であるなら, A の任意の要素は, B の要素によって任意に（いくらでも）近似できることを意味する．この場合, B は A において万能 (universal) 性を持つと言われる．本論文では, 集合族（モデルが実現する関数の集合）が全ての連続関数の集合において稠密かどうかを考える．

以下に示す定理は, 連続関数のすべての集合において, その部分集合が稠密であることを示す十分条件を与える．

[Stone-Weierstrass 定理][9, 34, 35]
S を m 次元上のコンパクト集合, $C(S)$ を S 上の実数値連続関数の全体集合とする．
Φ を, 以下を満たす実数値連続関数の集合とする:
(i) Identity function : 定数関数 $f(\boldsymbol{x})=1$ が Φ に含まれる．
(ii) Separability : S 上の任意の 2 点 $\boldsymbol{x}_1, \boldsymbol{x}_2$ $(\boldsymbol{x}_1 \neq \boldsymbol{x}_2)$ について, $f(\boldsymbol{x}_1) \neq f(\boldsymbol{x}_2)$ となるような f が Φ に含まれる．
(iii) Algebraic closure : Φ に含まれる任意の f, g と実数値 α, β について, 関数 $f \cdot g$ と $\alpha f+\beta g$ が Φ に含まれる．

このとき, Φ は $C(S)$ 内で稠密である．言い換えると, 任意の $\varepsilon>0$ と任意の関数 $g \in C(S)$ について, 任意の $\boldsymbol{x} \in S$ に関して以下のような性質を満たす関数 f が Φ 内に存在する．

$$|g(\boldsymbol{x})-f(\boldsymbol{x})|<\varepsilon \quad \square$$

この定理は, 集合族が稠密であることを, (i), (ii), (iii) の 3 つの条件の成立により示すことができることを述べている．この定理を使って, Wan は簡略型ファジィ推論法のモデル (正確に言うと実現される関数族) が, 連続関数のすべての集合において稠密であることを示した [34]. 結果は次の通りである．

[定理 2.1][34]
$\boldsymbol{R}^m$ の任意のコンパクトな有限集合 U と任意の $\varepsilon>0$ に対して, 次のような $f \in \Omega$ が存在する．

$$\sup_{\boldsymbol{x} \in U} |g(\boldsymbol{x})-f(\boldsymbol{x})|<\varepsilon \tag{2.33}$$

ここに,

$$\Omega_M=\{f(\boldsymbol{x})=\frac{\sum_{i=1}^M \mu_i(\boldsymbol{x}) w_i}{\sum_{i=1}^M \mu_i(\boldsymbol{x})} | w_i \in \boldsymbol{R}, \boldsymbol{x} \in S\} \tag{2.34}$$

$$\mu_i(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\sum_{j=1}^{m}(\frac{x_j - c_{ij}}{b_{ij}})^2\right) \tag{2.35}$$

かつ

$$\Omega = \bigcup_{m=1}^{\infty} \Omega_M \tag{2.36}$$

とする.

第3章　属性型ファジィ推論モデル

3.1　はじめに

本章では, 高い説明能力をもつモデルとして知られているファジィ推論モデルと属性型ファジィ推論モデルについて説明する. 従来型の簡略型ファジィ推論システムは学習後の推論ルールの言語的解釈が困難であるため, 得られたシステムを記述するファジィ推論ルールの説明能力が低いことが知られている. それゆえ, 高い説明能力と近似精度を実現する新しいモデルとして属性型ファジィ推論モデルの提案を行い, このモデルの能力の高さや万能性に関する理論的な解析を行う.

3.2　Mamdani 型ファジィ推論モデル

ここでは, ファジィ推論モデルの推論規則を以下のように与える.

$$\begin{aligned} R^{i_1\cdots i_m}\ :\ & \text{if } x_1 \text{ is } A_{i_1 1} \text{ and } \cdots \text{ and } x_m \text{ is } A_{i_m m} \\ & \qquad\qquad \text{then } y \text{ is } f_{i_1\cdots i_m}(x_1,\cdots,x_m) \end{aligned} \tag{3.1}$$

ここで, $1 \leq i_j \leq i_l (j \in Z_l)$, $A_{i_j j}(x_j)$ は入力 x_j に関するメンバーシップ関数である. このモデルは 2 章で導入した Mamdani 型モデルの特別な場合であるが, 同一の名前を使う.

入力 $\boldsymbol{x}$ に対する推論規則 $R^{i_1\cdots i_m}$ の適合度 $\mu_{i_1\cdots i_m}$ は, 以下のように与えられる.

$$\mu_{i_1\cdots i_m} = \prod_{j=1}^{m} A_{i_j j}(x_j) = A_{i_1 1}(x_1)\cdot\cdots\cdot A_{i_m m}(x_m) \tag{3.2}$$

出力 y は, 以下の式により求められる.

$$y^* = \frac{\sum_{i_1}\cdots\sum_{i_m} \mu_{i_1\cdots i_m} f_{i_1\cdots i_m}(x_1,\cdots,x_m)}{\sum_{i_1}\cdots\sum_{i_m} \mu_{i_1\cdots i_m}} \tag{3.3}$$

$f_{i_1\cdots i_m}(x_1,\cdots,x_m)$ が定数ならば簡略型, 関数であれば TS 型と呼ぶものとする.
[例題 3.1]

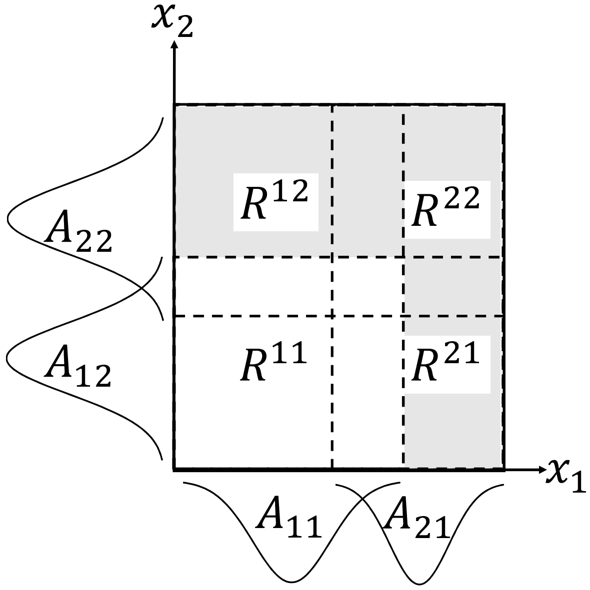


図 3.1: 学習前のメンバーシップ関数の中心の位置

$m=2$, $1 \leq i_1, i_2 \leq 2$ のときの推論規則の例を以下に示す (図 3.1 参照):

$$
\begin{aligned}
R^{11} &: \ if\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{12}\ then\ y\ is\ w_{11} \\
R^{12} &: \ if\ x_1\ is\ A_{11}\ and\ x_2\ is\ A_{22}\ then\ y\ is\ w_{12} \\
R^{21} &: \ if\ x_1\ is\ A_{21}\ and\ x_2\ is\ A_{12}\ then\ y\ is\ w_{21} \\
R^{22} &: \ if\ x_1\ is\ A_{21}\ and\ x_2\ is\ A_{22}\ then\ y\ is\ w_{22}
\end{aligned}
$$

式 (2.6) により与えられるファジィ推論モデルをアルゴリズム A により学習を行う場合, 学習後の各メンバーシップ関数の配置は図 3.2 のように, 中心位置が初期位置とずれた位置とする.

一方, Mamdani 型においては, 学習後の各メンバーシップ関数の配置は図 3.3 のように, 学習前の中心位置と幅が同じ値となる. それゆえ, それぞれのルールが自由に移動する簡略型モデルに比べて, Mamdani 型は近似精度があまり高くないという問題点がある. そこで, より説明能力の高い Mamdani 型の近似精度を改善する方法を提案する.

推論規則 (3.1) から成るファジィ推論において, $f_{i_1 \cdots i_m}(x_1, \cdots, x_m)$ を関数とする. 関数 $f_{i_1 \cdots i_m}$ のみを学習により決定する手法を, Mamdani 型 TS ファジィ推論法 (アルゴリズム A-MT) とする.

線形関数 $f_{i_1 \cdots i_m}(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{m} w_{ij} x_j (w_{i0},\ w_{i1}, \cdots, w_{im} \in \boldsymbol{R})$ を用いるとき, TS 型と組み合わせた Mamdani 型の簡略型ファジィ推論法の推論規則中の後件部パラ

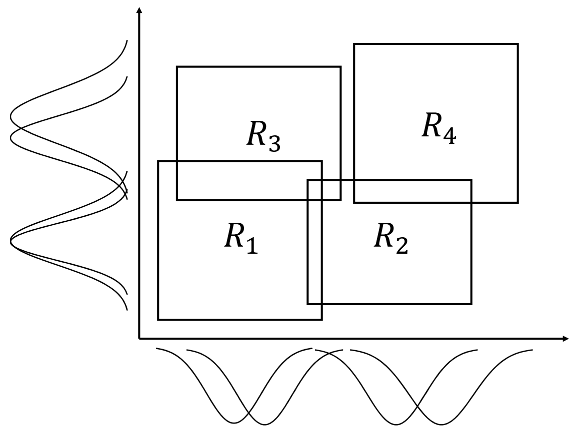


図 3.2: アルゴリズム A による学習後のメンバーシップ関数

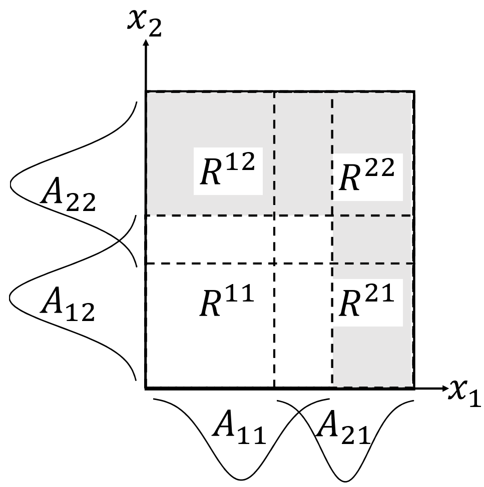


図 3.3: アルゴリズム A-M による学習後のメンバーシップ関数

メータの更新式は, 以下のようになる. ただし, $a_{ij} = 1(i{\in}Z_n \text{ and } j{\in}Z_m)$ とする.

$$\frac{\partial E}{\partial w_{i0}} = \frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y - y^*) \tag{3.4}$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n}\mu_i}\cdot(y - y^*)x_j \tag{3.5}$$

$f_{i_1\cdots i_m}(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{m} w_{ij}x_j(w_{i0},\, w_{i1},\cdots,w_{im}{\in}\boldsymbol{R})$ とするとき, 学習アルゴリズム A-MTS は, 以下のようになる.

[**学習アルゴリズム A-MTS**]

Step A-MTS1： しきい値 θ, 最大学習回数 T_{max} を与える. 整数 H に対して, 推論規則数 n を $n = H^m$ とおく. $t = 1$ とおく.

Step A-MTS2： $b_{ij} = (\max(x_j) - \min(x_j))/2(H-1)$, $c_{ij} = \min(x_j) + 2b_{ij}$ とする. パラメータ w_{ij} を初期化する.

Step A-MTS3： $p = 1$ とおく.

Step A-MTS4： データ $(x_1^p,\cdots,x_m^p,y_p^r){\in}\boldsymbol{D}$ を与える.

Step A-MTS5： 式 (2.10) と (2.11) より, μ_i と y を求める.

Step A-MTS6： 式 (3.4), (3.5) より, パラメータ w_{i0}, w_{ij} を更新する.

Step A-MTS7: $p = P$ ならば Step A-MTS8 へ, $p < P$ ならば $p{\leftarrow}p+1$ として Step A-MTS4 へ行く.

Step A-MTS8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t{\leftarrow}t+1$ として Step A-MTS3 へ, $E(t){\leq}\theta$ または $t > T_{max}$ ならば学習を終了する.

この手法は, 後件部が定数から 1 次関数に変更したため, Mamdani 型に比べて近似精度が高い. しかしながら, 後件部に自然言語による解釈が困難な関数が含ま

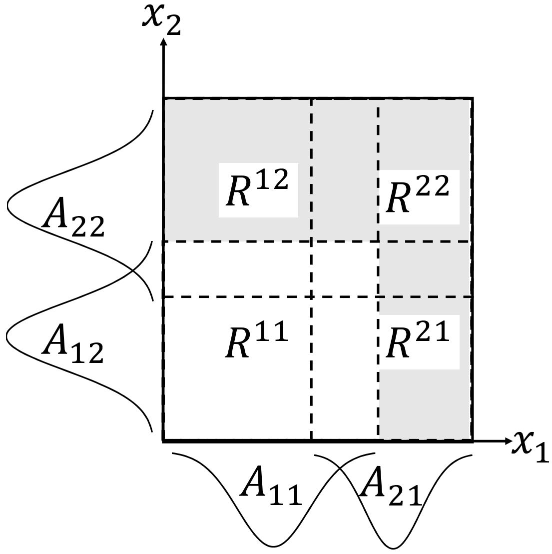


図 3.4: アルゴリズム A-E による学習前のメンバーシップ関数の中心の位置

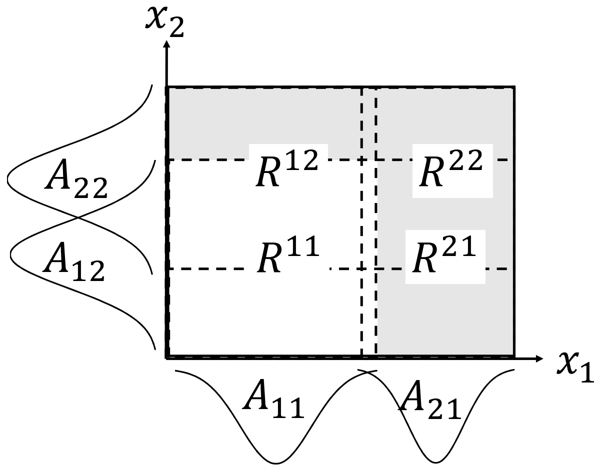


図 3.5: アルゴリズム A-E による学習後のメンバーシップ関数の中心の位置

れる. これにより, 推論規則の自然言語による解釈が困難となるため, この手法は Mamdani 型に比べて説明能力が低いといえる.

3.3 属性型ファジィ推論モデル

従来手法においては, 各推論規則中のメンバーシップ関数を個別に動かす学習手法では近似精度が高くなるが, 学習後にはファジィ集合が不規則に移動することにより言語的な解釈が困難である. また, メンバーシップ関数を移動させない学習手法では言語的解釈が容易である一方で, 近似精度が低いという問題点がある. Mamdani 型に TS 型を組み合わせた手法は, 推論規則の後件部に関数を含むため自然言語による解釈が難しくなる.

本章では, 学習前に同じファジィ集合として定められたものは, 異なる推論規則中に用いられていても, 学習時にはメンバーシップ関数のパラメータを同時に動かす手法を提案する. 以降, この提案手法を属性型ファジィ推論法と呼ぶ.

学習において, 規則 (3.1) の規則中のファジィ集合のメンバーシップ関数を

$$A_{i_j j}(x_j) = a_{i_j j} \exp\left(-\frac{1}{2}\sum_{j=1}^{m}\frac{(x_j - c_{i_j j})^2}{b_{i_j j}^2}\right) \tag{3.6}$$

とする. パラメータ $c_{i_j j}$, $b_{i_j j}$ および関数 $f_{i_1 \cdots i_m}$ の各パラメータを学習により決定するアルゴリズムを提案する. この属性型ファジィ推論モデルの最急降下法に基づく学習アルゴリズムをアルゴリズム A-E とする.

規則 (3.1) において $f_i(\boldsymbol{x}) = w_{i_1\cdots i_m}$ とするとき, 各パラメータは, 以下の関係が成立する.

$$\frac{\partial E}{\partial w_{i_1\cdots i_m}} = \frac{\mu_{i_1\cdots i_m}}{\sum_{i_1}\cdots\sum_{i_m}\mu_{i_1\cdots i_m}} \cdot (y - y^*) \tag{3.7}$$

$$\frac{\partial A_{i_j j}}{\partial a_{i_j j}} = \exp\left(-\frac{1}{2}\sum_{j=1}^{m}\frac{(x_j - c_{i_j j})^2}{b_{i_j j}^2}\right) \tag{3.8}$$

$$\frac{\partial A_{i_j j}}{\partial c_{i_j j}} = \frac{(x_j - c_{i_j j})}{b_{i_j j}^2}\exp\left(-\frac{1}{2}\sum_{j=1}^{m}\frac{(x_j - c_{i_j j})^2}{b_{i_j j}^2}\right) \tag{3.9}$$

$$\frac{\partial A_{i_j j}}{\partial b_{i_j j}} = \frac{(x_j - c_{i_j j})^2}{b_{i_j j}^3}\exp\left(-\frac{1}{2}\sum_{j=1}^{m}\frac{(x_j - c_{i_j j})^2}{b_{i_j j}^2}\right) \tag{3.10}$$

学習アルゴリズム A-E は, 以下のようになる.

[**学習アルゴリズム A-E**]

Step A-E1 : しきい値 θ, 最大学習回数 T_{max} を与える. 推論規則の初期位置は等間隔に配置する. 整数 H に対して, 推論規則数 n を $n = H^m$ とおく. $t = 1$ とおく.

Step A-E2 : パラメータ b_{ij}, c_{ij}, w_i を初期化する.

Step A-E3 : $p = 1$ とおく.

Step A-E4 : データ $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ を与える.

Step A-E5 : $\mu_{i_1\cdots i_m}$ と y を求める.

Step A-E6 : 式 (3.8), (3.9), (3.10), (3.7) より, パラメータ $c_{i_j j}$, $b_{i_j j}$, w_i を更新する.

Step A-E7: $p = P$ ならば Step A-E8 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A-E4 へ行く.

Step A-E8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A-E3 へ, $E(t) \leq \theta$ または $t > T_{max}$ ならば学習を終了する.

図 3.4 と 3.5 に示すように, 前件部メンバーシップ関数の中心と幅はルールごとに移動するのではなく, メンバーシップ関数 A_{ij} ごとに移動する. 例えば, R^{11} と R^{12} は A_{11} のメンバーシップ関数により同一の中心と幅を持つが, 学習後も同一の中心と幅を持つ (値は学習前と異なる).

このモデルに関して, メンバーシップ関数に三角型を用いるファジィ推論モデルの学習法については, すでに文献 [28] で提案されている.

3.4　属性型ファジィ推論モデルの万能性

メンバーシップ関数としてガウス型関数を用いた属性型ファジィ推論法のモデルについて, 万能性の証明を行う.

[定理 3.1]

Φ をコンパクト集合 $S\subset \boldsymbol{R}^m$ 上で定義された属性型ファジィ推論モデルの全ての集合とする. すなわち,

$$\Phi_{l_1\cdots l_m} = \left\{ f(\boldsymbol{x}) = \frac{\sum_{i_1}\cdots\sum_{i_m}\prod_j A_{i_j j}(x_j)w_{i_1\cdots i_m}}{\sum_{i_1}\cdots\sum_{i_m} A_{i_j j}(x_j)}, w_{i_1\cdots i_m}, a_{i_j j}, c_{i_j j}, b_{i_j j}\in \boldsymbol{R}, \boldsymbol{x}\in S\right\}$$

ここで, $A_{i_j j}(x_j) = a_{i_j j}\exp\left(-\frac{1}{2}\left(\frac{x_j - c_{i_j j}}{b_{i_j j}}\right)^2\right)$ かつ

$$\Phi = \bigcup_{l_1=1}^{\infty}\cdots\bigcup_{l_m=1}^{\infty}\Phi_{l_1\cdots l_m} \tag{3.11}$$

このとき, Φ は $C(S)$ において稠密である.

[証明]

Stone-Weierstrass 定理の 3 つの条件を満たすことを証明する.

(i)$w_{i_1\cdots i_m} = 1$ のとき $y = 1$ となることから, 関数 $f(\boldsymbol{x}) = 1(\boldsymbol{x}\in S)$ が Φ 内に存在する.

(ii) 指数関数の単調性より, $\boldsymbol{x}_1$, $\boldsymbol{x}_2(\boldsymbol{x}_1\neq\boldsymbol{x}_2)$ となるような f が Φ 内に存在する.

(iii)f, g を, 以下のような Φ 内の関数とする.

$$f(\boldsymbol{x}) = \frac{\sum_{i_1}\cdots\sum_{i_m}\prod_j A^f_{i_j j}(x_j)w^f_{i_1\cdots i_m}}{\sum_{i_1}\cdots\sum_{i_m}\prod_j A^f_{i_j j}(x_j)} \tag{3.12}$$

$$g(\boldsymbol{x}) = \frac{\sum_{l_1}\cdots\sum_{l_m}\prod_j A^g_{l_j j}(x_j)w^g_{l_1\cdots l_m}}{\sum_{l_1}\cdots\sum_{l_m}\prod_j A^g_{l_j j}(x_j)} \tag{3.13}$$

ここで,

$$A^f_{i_j j}(x_j) = a^f_{i_j j}\exp\left(-\frac{1}{2}\left(\frac{x_j - c^f_{i_j j}}{b^f_{i_j j}}\right)^2\right) \tag{3.14}$$

$$A^g_{l_j j}(x_j) = a^g_{l_j j}\exp\left(-\frac{1}{2}\left(\frac{x_j - c^g_{l_j j}}{b^g_{l_j j}}\right)^2\right) \tag{3.15}$$

とする.

これらを用いて, $\alpha f + \beta g\in\Phi$ と $f{\cdot}g\in\Phi(\alpha,\beta\in\boldsymbol{R})$ の証明を行う.

$$
\begin{aligned}
A^{fg}_{i_j l_j j}(x_j) &= A^{f}_{i_j j}(x_j)\cdot A^{g}_{l_j}(x_j) \\
&= a^{fg}_{i_j l_j j}\exp\left(-\frac{1}{2}\left(\frac{x_j - c^{fg}_{i_j l_j j}}{b^{fg}_{i_j l_j j}}\right)^2\right) \qquad (3.16)
\end{aligned}
$$

$$
\begin{aligned}
w^{fg1}_{i_1\cdots i_m l_1\cdots l_m} &= \alpha w^{f}_{i_1\cdots i_m} + \beta w^{g}_{l_1\cdots l_m} \qquad (3.17)\\
w^{fg2}_{i_1\cdots i_m l_1\cdots l_m} &= w^{f}_{i_1\cdots i_m}\cdot w^{g}_{l_1\cdot l_m} \qquad (3.18)
\end{aligned}
$$

とおく. $(a^{fg}_{i_j l_j j}, c^{fg}_{i_j l_j j}, b^{fg}_{i_j l_j j}, w^{fg1}_{i_1\cdots w_{i_m} l_1\cdots l_m}, w^{fg2}_{i_1\cdots i_m l_1\cdots l_m} \in \boldsymbol{R})$.

これらを用いると,

$$
\begin{aligned}
\alpha f(\boldsymbol{x}) + \beta g(\boldsymbol{x}) &= \frac{\sum_{i_1}\cdots\sum_{i_m}\sum_{l_1}\cdots\sum_{l_m}\prod_j A^{fg}_{i_j l_j j}(x_j) w^{fg1}_{i_1\cdots i_m l_1\cdots l_m}}{\sum_{i_1}\cdots\sum_{i_m}\sum_{l_1}\cdots\sum_{l_m}\prod_j A^{fg}_{i_j l_j j}(x_j)} \qquad (3.19)\\
f(\boldsymbol{x})\cdot g(\boldsymbol{x}) &= \frac{\sum_{i_1}\cdots\sum_{i_m}\sum_{l_1}\cdots\sum_{l_m}\prod_j A^{fg}_{i_j l_j j}(x_j) w^{fg2}_{i_1\cdots i_m l_1\cdots l_m}}{\sum_{i_1}\cdots\sum_{i_m}\sum_{l_1}\cdots\sum_{l_m}\prod_j A^{fg}_{i_j l_j j}(x_j)} \qquad (3.20)
\end{aligned}
$$

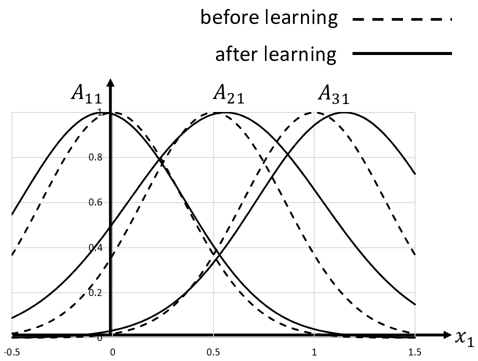
式 (3.19) と (3.20) より, 属性型ファジィ推論モデルについては $\alpha f + \beta g \in \Phi$ かつ $f\cdot g \in \Phi$ となる. □

以上のことから, 簡略型ファジィ推論モデルと属性型ファジィ推論モデルは万能性を満たすことが分かる. この場合どちらもガウス型メンバーシップ関数を用いたモデルである. 一方で, ガウス型メンバーシップ関数を用いた Mamdani 型と TS 型と組み合わせた Mamdani 型, および三角型関数をメンバーシップ関数として用いた各モデルについては, 条件 (iii) が必ずしも成り立つとは限らないため, Stone-Weierstrass 定理が成り立つとは限らない. また, この定理は任意の連続関数または任意のファジィ推論モデルを任意の精度で近似する属性型ファジィ推論モデルの存在を示す結果であり, 近似精度の高いモデルをいかに実現するかは示していない. そのため, このような学習法の 1 つとして, 最急降下法に基づくアルゴリズム A-E を導入した.

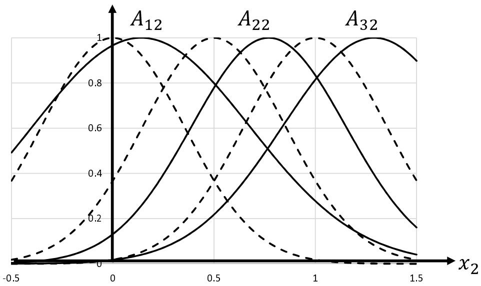
3.5 数値シミュレーション

ここでは, 関数近似およびパターン認識により提案手法の有効性を示す. ここで, $a_{ij} = 1$, $a_{i_j j} = 1 (i \in Z_n \text{ and } j \in Z_m)$ である.

はじめに, 2 変数の関数近似問題を使って, 実際にアルゴリズム A-E により学習の前後でどのように前件部メンバーシップ関数の中心と幅が移動するかを示そう.



(a) 入力 x_1 に関するメンバーシップ関数の配置



(b) 入力 x_2 に関するメンバーシップ関数の配置

図 3.6: 式 (3.21) の関数近似問題における, 学習後のメンバーシップ関数の配置.

[例題 3.2]

例として, 以下の 2 次関数を用いる:

$$y = \frac{(2x_1 + 4x_2^2)(2x_1 + 4x_2^2 + 0.2)}{37.2} \tag{3.21}$$

ここに, $0 \leq x_1, x_2 \leq 1$.

式 (3.21) を近似するための属性型ファジィ推論モデルについて, $n = 2$, 属性数を $H = 3$ とする. 図 3.6 にメンバーシップ関数の配置を示す. メンバーシップ関数は若干の移動は見られるが, 説明能力 (属性の位置関係) は維持されているといえる.

以下において, 学習法 A, A-M, A-MTS, A-E を用いたものをそれぞれ Model A, Model A-M, Model A-MTS, Model A-E とする.

3.5.1 関数近似問題

ここでは, 以下に示す $[0, 1]\times[0, 1]$ 上の 4 つの関数を, 与えられた学習用データから同定する.

$$y = \sin(\pi x_1^3)\cdot x_2 \tag{3.22}$$

表 3.1: 関数近似における実験条件.

	Model A	Model A-M	Model A-MTS	Model A-E
T_{max}	50000	50000	50000	50000
K_c	0.01	0.0	0.0	0.001
K_b	0.01	0.0	0.0	0.001
K_w	0.1			
d	3	7	4	6
c_{ij} の初期値	等間隔			
b_{ij} の初期値	$\frac{1}{2(\text{分割数}-1)}\times$(入力値の範囲)			
w_{ij} の初期値	$[0,1]$ 内でランダムに選択			

$$y = \frac{\sin(2\pi x_1^3)\cdot\cos(\pi x_2)+1}{2} \tag{3.23}$$

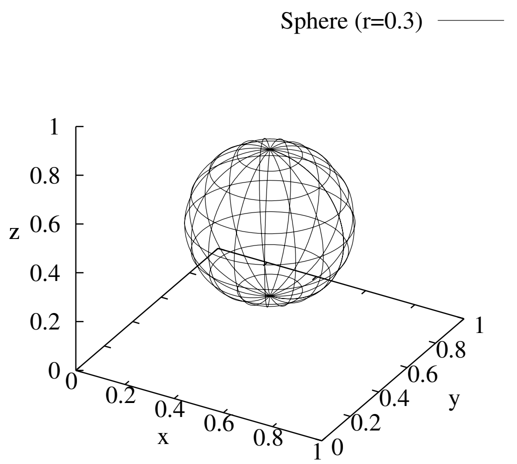
$$y = \frac{1.9(1.35+e^{x_1}\sin(13(x_1-0.6)^2)\cdot e^{-x_2}\sin(7x_2))}{7} \tag{3.24}$$

$$y = \frac{\sin(10(x_1-0.5)^2+10(x_2-0.5)^2)+1.0}{2} \tag{3.25}$$

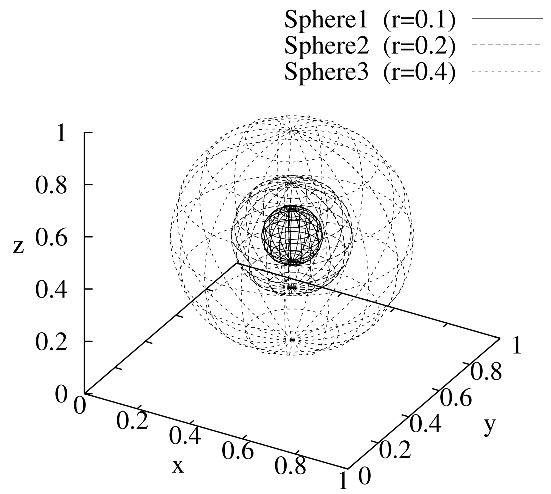
実験条件を表 3.1 に示す．しきい値 θ を 1.0×10^{-5}, 学習用データの数を 200, テスト用データの数を 2500 とする．表 3.2 はモデル A, A-M, A-MTS, A-E の結果を示す．表 3.2 の各値は, 学習用およびテスト用データの平均二乗誤差 (Mean Square Error:MSE) を示す．なお, 結果は 20 回試行の平均 ($\times10^{-4}$) である．表 3.2 においては, Model A と A-E は同等の精度を示している．ここに, ♯parameter はパラメータ数を示す.

3.5.2 パターン認識問題

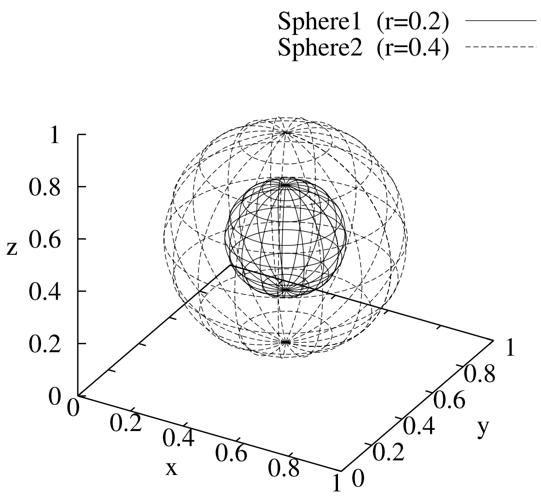
ここでは, 図 3.7 に示す 2 分類のパターン認識問題により, モデル A と A-E の能力の違いを示す．本分類問題においては, $[0,1]\times[0,1]\times[0,1]$ 空間内に存在する点を 2 種類のクラスに分類する (クラス 0 とクラス 1)．クラスの境界は $(0.5, 0.5, 0.5)$ を中心とする球として与えられる．Sphere では, 球の内側にある点をクラス 1, 外側にある点をクラス 0 とする．Double-Sphere では, Spheres 1 と 2 の間にある点をクラス 1, 他の位置にある点をクラス 0 とする．Triple-Sphere では, Sphere 1 の内側および Sphere 2 と Sphere 3 の間にある点をクラス 1, 他の位置にある点をクラス 0 とする．理想的な出力値 y_p^r は次のように与える: もし点 $\boldsymbol{x_p}$ がクラス 0 に属するなら $y_p^r = 0.0$, クラス 1 に属するなら $y_p^r = 1.0$ とする．学習用データの数は 512, テスト用データの数は 6400 である．実験条件を表 3.3 に示す．表 3.4 に, モデ



(a) Sphere



(c) Triple-Sphere



(b) Double-Sphere

図 3.7: 2 分類のパターン認識

表 3.2: 関数近似問題の結果.

		式 (3.22)	式 (3.23)	式 (3.24)	式 (3.25)
Model A	学習用データの MSE	0.10	0.61	0.93	0.22
	テスト用データの MSE	0.36	1.48	2.65	0.80
	パラメータ数	45	45	45	45
Model A-M	学習用データの MSE	1.17	12.37	1.28	4.44
	テスト用データの MSE	4.22	38.69	4.12	12.94
	パラメータ数	49	49	49	49
Model A-MTS	学習用データの MSE	0.24	2.49	0.70	3.01
	テスト用データの MSE	0.92	10.89	2.04	11.10
	パラメータ数	48	48	48	48
Model A-E	学習用データの MSE	0.10	0.70	0.11	1.17
	テスト用データの MSE	0.44	5.97	0.26	3.76
	パラメータ数	45	45	45	45

ル A および A-E における誤分類率を示す. ここに, 誤分類率とは学習用データ, テスト用データそれぞれに対して分類を行ったときに, 正しく分類することができなかったデータの割合を示す. 表中の値は 20 回試行の平均を示す. 表 3.4 においては, モデル A および A-E は同等の精度を示す.

表 3.3: パターン認識の実験条件.

	Model A	Model A-E
T_{max}	50000	50000
K_w	0.01	0.01
K_c	0.001	0.001
K_b	0.001	0.001
d	3	5
c_{ij} の初期値	等間隔	
b_{ij} の初期値	$\frac{1}{2(\text{分割数}-1)}\times$(入力値の範囲)	
w_{ij} の初期値	$[0, 1]$ からランダムに選択	

表 3.4: 2 分類のパターン認識の結果.

		Sphere	Double-Sphere	Triple-Sphere
Model A	誤分類率 (学習用)	0.007	0.033	0.037
	誤分類率 (テスト用)	0.027	0.078	0.078
	パラメータ数	189	189	189
Model A-E	誤分類率 (学習用)	0.009	0.018	0.020
	誤分類率 (テスト用)	0.036	0.057	0.061
	パラメータ数	155	155	155

3.6 まとめ

本章では, 説明能力の高いファジィ推論モデルについての議論を行った. ファジィ集合のメンバーシップ関数を学習により決定する簡略型ファジィ推論法は近似精度は高いが説明能力が高くはなく, また, メンバーシップ関数をヒューリスティスティクに決定する Mamdani 型ファジィ推論法は, 説明能力は高いが近似精度は必ずしも高くはない. Mamdani 型に TS 型を組み合わせた手法はメンバーシップ関数をヒューリスティクに決定するが, 後件部に関数を用いているため自然言語的な解釈が困難であり説明能力が低い. 提案手法である属性型ファジィ推論法は人間が決定したファジィ集合の意味を維持したままメンバーシップ関数を学習により決定することができ, 説明能力が高く, 近似精度も高い. さらに, 属性型ファジィ推論法が任意の連続関数を任意精度で近似できる手法で高い能力をもつことを理論的に示した. また, 数値シミュレーションにより, 属性型ファジィ推論法は他の手法に比べて少ないパラメータ数で高い近似精度を実現することを示した. 提案モデルに関する他の数値シミュレーションとしては, 障害物回避問題への適用を行っている. 結果は 2 つのシミュレーションと同様に, 提案モデルの有効性が示される [69].

一方, 問題点としては, 簡略型ファジィ推論モデルと同様に, ルール数やパラメータ数が多くなり, 学習時間や入力変数の多い問題への適用の難しさが挙げられる.

第4章　ベクトル量子化を用いた推論規則の決定方法

4.1　はじめに

ソフトコンピューティングの学習法の一つに, ベクトル量子化を組み合わせる方法が知られている. たとえば, RBF の学習では, k-means と最急降下法（または行列計算）によりパラメータを決定する方法が知られている [9]. この方法の特徴は, 従来法に比べて少ないパラメータ数をもつモデルを実現することにある. ファジィ推論システムの学習においても, ベクトル量子化を用いたいくつかの方法が提案されている [24, 25, 36]. この場合, 従来手法のベクトル量子化を用いた方法では, 推論規則の初期配置の決定には学習用のデータのうち, 入力データのみを用いている. しかしながら, 入力だけでなく, 入力と出力の両方を考慮した方が高い精度を実現できると考えられる. そこで, 出力も考慮した推論規則の初期配置の決定を行う, ファジィ推論システムの学習手法が提案されている [25, 36]. この方法は, 入力のみを考慮する場合に比べて良好な結果を示すが, 必ずしも満足できる近似精度は得られていない. この方法の問題点の 1 つは, ベクトル量子化が主に推論規則の初期配置の決定のみに用いられていることであると考えられる.

本章では, ベクトル量子化と最急降下法の組み合わせを山登り法にもとづいて繰り返す手法を提案する. 提案手法においては, はじめに, 各学習用データに対してそれぞれの入出力を考慮した頻度関数の計算を行い, これを用いたベクトル量子化によりファジィ推論システムの初期位置の決定を行う. 次に, 推論規則中のパラメータを最急降下法を用いて更新する. この過程を山登り法にもとづいて繰り返すことにより高い精度を実現する. 数値シミュレーションにより, この提案手法の有効性を示す.

4.2　ベクトル量子化

ベクトル量子化とは, 部分集合 $\boldsymbol{V}\subseteq\boldsymbol{R}^m$ 上のデータを有限個の参照ベクトル $\{\boldsymbol{c}_1,\cdots,\boldsymbol{c}_r\}$ $(\boldsymbol{c}_i\in\boldsymbol{R}^m)$ により特徴づける手法である. 参照ベクトル $\boldsymbol{c}_i$ により, クラス

$$C_i=\{\boldsymbol{x}\in\boldsymbol{V}\,|\,||\boldsymbol{x}-\boldsymbol{c}_i||\leq||\boldsymbol{x}-\boldsymbol{c}_l||\text{ for }l\in Z_r\} \tag{4.1}$$

を定義する. ここで, $\boldsymbol{x}\in\boldsymbol{V}$, $i\in Z_r$ である.

ベクトル量子化の手法の 1 つである k-means においては, 与えられたデータ $\boldsymbol{x}$ に対し, 最も近い参照ベクトルを次式で定義される $\triangle\boldsymbol{c}_j$ だけ変化させるステップを繰り返すことで最終的な参照ベクトルの位置を決定する

$$\triangle\boldsymbol{c}_j = \varepsilon\cdot(\boldsymbol{x} - \boldsymbol{c}(\boldsymbol{x})) \tag{4.2}$$

ここで, ε は定数, $c(\boldsymbol{x})$ はデータ $\boldsymbol{x}$ に最も近い参照ベクトルである.

k-means のアルゴリズムは, 以下のように与えられる.

[**学習アルゴリズム VQ***][40,41]

Step VQ*1 : 最大学習回数 T_{max} を与える. $t = 1$, $p = 1$ とおく. すべての参照ベクトルの初期値をランダムに設定する.

Step VQ*2 : データ $\boldsymbol{x} = (x_1^p, \cdots, x_m^p)$ を与える.

Step VQ*3 : $\boldsymbol{x}$ に最も近い参照ベクトル $\boldsymbol{C}_{min}$ を決定する.

Step VQ*4 : 式 (4.2) を使って $\boldsymbol{c}_{min} = \boldsymbol{c}_{min} + \triangle\boldsymbol{c}_{min}$ を計算する.

Step VQ*5 : $t = T_{max}$ ならばアルゴリズムを終了する. $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step VQ*2 へ行く.

ベクトル量子化の手法の 1 つであるニューラルガス (NG) においては, 各参照ベクトル $\boldsymbol{c}_i (i \in Z_r)$ に対し, 与えられたデータ $\boldsymbol{x}$ に対し, その近さに応じて次式で定義される $\triangle\boldsymbol{c}_i$ だけ変化させるステップを繰り返すことで最終的な参照ベクトルの位置を決定する.

$$\triangle\boldsymbol{c}_i = \varepsilon\cdot h_\lambda(k_i(\boldsymbol{x}))\cdot(\boldsymbol{x} - \boldsymbol{c}_i) \tag{4.3}$$

ここで, ε, λ は定数, h_λ は λ により定義される単調減少関数である. また, $k_i(\boldsymbol{x})$ は入力 $\boldsymbol{x}$ に対して c_i が何番目に近いかを与える関数であり, $0 \leq k_i(\boldsymbol{x}) \leq r - 1$ を満たす. 本論文では, h_λ として, 以下の関数を用いる.

$$h_\lambda(k_i(\boldsymbol{x})) = \exp\left(-\frac{k_i(\boldsymbol{x})}{\lambda}\right) \tag{4.4}$$

ニューラルガスのアルゴリズムは, 以下のように与えられる.

[**学習アルゴリズム VQ**][40,41]

Step VQ1 : 最大学習回数 T_{max} を与える. $t = 1$, $p = 1$ とおく. すべての参照ベクトルの初期値をランダムに設定する.

Step VQ2 : データ $\boldsymbol{x} = (x_1^p, \cdots, x_m^p)$ を与える.

Step VQ3 : $\boldsymbol{x}$ に対する $k_i(\boldsymbol{x})$ を決定する. ここに, $i \in Z_r$ とする.

Step VQ4 : 式 (4.3) を使って $\boldsymbol{c}_i = \boldsymbol{c}_i + \triangle\boldsymbol{c}_i (i \in Z_r)$ を計算する.

Step VQ5 : $t = T_{max}$ ならばアルゴリズムを終了する. $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step VQ2 へ行く.

4.3 ベクトル量子化によりルールの初期配置を決定する学習方法

ベクトル量子化を用いてファジイルールの初期配置を決定し, その後最急降下法により学習を行う方法を与える.

4.3.1 学習データの入力のみを考慮したベクトル量子化

前節で述べたように, ベクトル量子化法には, k-means 法や NG 法がある. その他にも, SOM や最大エントロピー法なども知られている. ここでは NG(k-means) 法を用いる場合を考える. はじめの方法は, 学習データの入力のみを使って, ベクトル量子化を行い, ルールの中心とそれを用いて幅を決定する. これをルールの初期配置 (w はランダム) として, 最急降下法により後件部実数値 w を決定する. この方法は, 基本的に RBF の学習法と同じである [9, 24]. これをアルゴリズム A-NG* と呼ぶ. 2 番目の方法は, 前半は, アルゴリズム A* と同じであるが, 後半は, すべてのパラメータ c, b, w を学習により決定する. これをアルゴリズム A-NG と呼ぶ. これらのアルゴリズムを以下に与える [24].

ここでは, 従来型の簡略ファジィ推論モデルの学習アルゴリズム A を以下のように, ルール数を自由に設定できるように一般化しておく. 学習アルゴリズム A においては, 推論規則の初期位置は入力空間上に等間隔に配置した. それゆえ, 等間隔に配置する推論規則数を自由に決定することが困難である. 一方で, 推論規則の初期位置を入力空間上にランダムに配置する, 以下に示す簡略型の学習アルゴリズム A* を用いることで, 推論規則数をより一般的に設定することができる.

[学習アルゴリズム **A***][29]

Step A*1 : しきい値 θ, 最大学習回数 T_{max}, 初期推論規則数 n_0 を与える. $n = n_0$ とする.

Step A*2 : パラメータ b_{ij}, c_{ij}, w_i の初期値をランダムに設定する. $t = 1$ とおく.

Step A*3 : $p = 1$ とおく.

Step A*4 : データ $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ を与える.

Step A*5 : 式 (2.7) と (2.8) より, μ_i と y を求める.

Step A*6 : 式 (2.17), (2.18), (2.16) より, パラメータ c_{ij}, b_{ij}, w_i を更新する.

Step A*7: $p = P$ ならば Step A*8 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A*4 へ行く.

Step A*8: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A*3 へ, $E(t) \leq \theta$ ならば学習を終了する. また, $E(t) \geq \theta$ かつ $t \geq T_{max}$ ならば $n \leftarrow n + 1$ として Step A*2 へ行く.

これを用いて, アルゴリズム A-NG* および A-NG は以下のように与えられる.

メンバーシップ関数の中心の位置を表すパラメータ $c_{ij}(i{\in}Z_n, j{\in}Z_m)$ をベクトル量子化により決定後に, 最急降下法を用いて各パラメータの更新を行う. ここで, パラメータ $b_{ij}(i{\in}Z_n, j{\in}Z_m)$ は, ベクトル量子化後の標準偏差

$$b_{ij} = \frac{1}{m_i} \sum_{\boldsymbol{x}\in C_i} (c_{ij} - x_j)^2 \tag{4.5}$$

とする. ここで, m_i はベクトル量子化により行われたクラス C_i の濃度である.

学習アルゴリズム A-NG は, 以下のようになる.

[**学習アルゴリズム A-NG**]

Step A-NG1 : しきい値 θ, 最大学習回数 T^0_{max}, T_{max}, 初期推論規則数 n_0 を与える. $n = n_0$ とおく.

Step A-NG2 : パラメータ b_{ij}, c_{ij}, w_i の初期値をランダムに設定する. $t = 1$ とおく.

Step A-NG3 : $p = 1$ とおく.

Step A-NG4 : データ $\boldsymbol{x}^p$ を与える.

Step A-NG5 : 式 (4.3) より, c_{ij} を更新する.

Step A-NG6 : $p = P$ ならば Step A-NG7 へ, $p < P$ ならば $p{\leftarrow}p + 1$ として Step A-NG4 へ行く.

Step A-NG7 : $t = T^0_{max}$ ならば $t = 1$ として Step A-NG8 へ, $t < T^0_{max}$ ならば $t{\leftarrow}t + 1$ として Step A-NG3 へ行く.

Step A-NG8 : 式 (4.5) により b_{ij} を決定する.

Step A-NG9 : $p = 1$ とおく.

Step A-NG10 : データ $(x^p_1, \cdots, x^p_m, y^r_p){\in}\boldsymbol{D}$ を与える.

Step A-NG11 : 式 (2.7) と (2.8) より, μ_i と y を求める.

Step A-NG12 : 式 (2.17), (2.18), (2.16) より, パラメータ c_{ij}, b_{ij}, w_i を更新する.

Step A-NG13: $p = P$ ならば Step A-NG14 へ, $p < P$ ならば $p{\leftarrow}p + 1$ として Step A-NG10 へ行く.

Step A-NG14: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t{\leftarrow}t + 1$ として Step A-NG9 へ, $E(t){\le}\theta$ ならば学習を終了する. $E(t){\ge}\theta$ かつ $t{\ge}T_{max}$ ならば $n{\leftarrow}n + 1$ として Step A-NG2 へ行く.

また, 学習アルゴリズム A-NG の基本的な流れは, 図 4.1 に示される. パラメータのうち c と b を固定した方法が, 学習アルゴリズム A-NG* である.

4.3.2 学習データの入出力を考慮したベクトル量子化

この場合は, 学習データの入出力を考慮したベクトル量子化を行い, ルールの中心を決定し, それを用いて幅を, 式 (4.5) と同様に決定する. また, w はランダムに設定する. さらに, パラメータ c, b, w を最急降下法により決定する. これをそれぞ

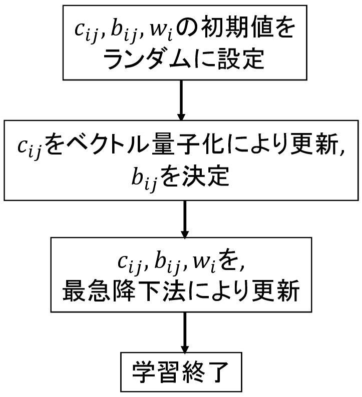


図 4.1: 学習アルゴリズム A-NG の流れ

れ, アルゴリズム P と学習アルゴリズム A-NGp と呼ぶ. このようなモデルはいくつか提案されているが, ここでは岸田の提案した方法を与える [36].

いま, 学習データ $\boldsymbol{D} = \{(\boldsymbol{x}^i, y^i)|i{\in}Z_p\}$ に対して, $\boldsymbol{D}^* = \{\boldsymbol{x}^i|i{\in}Z_p\}$ とする.

[アルゴリズム **P**]

Step P1 : 入力データ $\boldsymbol{x}^i{\in}\boldsymbol{D}^*$ を与え, $\boldsymbol{x}^i$ に対する近傍ランク $(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_1}, \cdots, \boldsymbol{x}^{i_k}, \cdots, \boldsymbol{x}^{i_{P-1}})$ を与える. ここで, $\boldsymbol{x}^{i_0} = \boldsymbol{x}^i$, $\boldsymbol{x}^{i_1}$ は $\boldsymbol{x}^i$ に最も近く, また, $\boldsymbol{x}^{i_k}(k = 0, \cdots, P-1)$ は $\boldsymbol{x}^j$ に対して $||\boldsymbol{x}^i - \boldsymbol{x}^j|| < ||\boldsymbol{x}^i - \boldsymbol{x}^{i_k}||$ となるような自然数 j が存在するような k 番目に近いベクトルである.

Step P2 : $H(\boldsymbol{x}^i)$ はデータ $\boldsymbol{x}^i$ に対する近傍の出力の変化の割合であり, 以下のように定義する:

$$H(\boldsymbol{x}^i) = \sum_{l=1}^{W} \left|\frac{y^i - y^{i_l}}{||\boldsymbol{x}^i - \boldsymbol{x}^{i_l}||}\right| \tag{4.6}$$

ここで, $\boldsymbol{x}^{i_l}(l{\in}Z_W)$ は, Step P1 で与えた $\boldsymbol{x}^i$ に対する l 番目の近傍ランク ($i{\in}Z_P$ かつ, y^i と y^{i_l} はそれぞれ入力 $\boldsymbol{x}^i$ と $\boldsymbol{x}^{i_l}$ に対する出力を意味する). また, 自然数 W は $H(\boldsymbol{x})$ に関する近傍ランクである.

Step P3 : $\boldsymbol{x}^i$ に対し, 正規化した $H(\boldsymbol{x}^i)$ を用いて頻度関数 $p_M(\boldsymbol{x}^i)$ を以下のように定義する.

$$p_W(\boldsymbol{x}^i) = \frac{H(\boldsymbol{x}^i)}{\sum_{j=1}^{P} H(\boldsymbol{x}^j)} \tag{4.7}$$

この頻度関数を用いて, 岸田は次の方法を提案した [36]. その基本的な流れを図 4.2 に示す.

[学習アルゴリズム **A-NGp**]

Step A-NGp1 : しきい値 θ, 最大学習回数 T^0_{max}, T_{max}, 初期推論規則数 n_0 を与える. $n = n_0$ とおく.

Step A-NGp2： $t = 1$ とおく.
Step A-NGp3： パラメータ b_{ij}, c_{ij}, w_i の初期値をランダムに設定する.
Step A-NGp4： 確率 $p_W(\boldsymbol{x}^1), \cdots, p_W(\boldsymbol{x}^P)$ に従い, $p(1 \leq p \leq P)$ を選択する.
Step A-NGp5： データ $\boldsymbol{x}^p$ を与える.
Step A-NGp6： 式 (4.3) より, c_{ij} を更新する.
Step A-NGp7： $t = T^0_{max}$ ならば $t = 1$ として Step A-NG'7 へ, $t < T^0_{max}$ ならば $t \leftarrow t + 1$ として Step A-NG'3 へ行く.
Step A-NGp8： 式 (4.5) により b_{ij} を決定する.
Step A-NGp9： $p = 1$ とおく.
Step A-NGp10： データ $(x^p_1, \cdots, x^p_m, y^r_p) \in \boldsymbol{D}$ を与える.
Step A-NGp11： 式 (2.7) と (2.8) より, μ_i と y を求める.
Step A-NGp12： 式 (2.17), (2.18), (2.16) より, パラメータ c_{ij}, b_{ij}, w_i を更新する.
Step A-NGp13: $p = P$ ならば Step A-NGp12 へ, $p < P$ ならば $p \leftarrow p + 1$ として Step A-NGp9 へ行く.
Step A-NGp14: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする. $E(t) > \theta$ かつ $t < T_{max}$ ならば $t \leftarrow t + 1$ として Step A-NGp11 へ, $E(t) \leq \theta$ ならば学習を終了する. また, $E(t) \geq \theta$ かつ $t \geq T_{max}$ ならば $n \leftarrow n + 1$ として Step A-NGp2 へ行く.

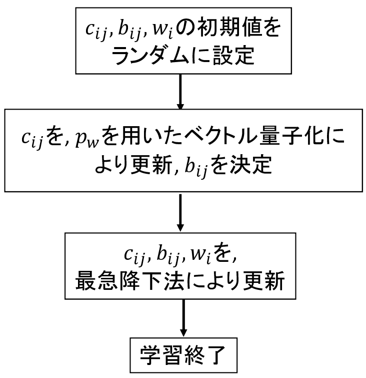


図 4.2: 学習アルゴリズム A-NGp の流れ

4.4 ベクトル量子化を用いたファジィ推論ルールの学習法

前節では, ベクトル量子化をルールの初期配置決定にのみ用いる方法を与えた. 本節では, この方法を山登り法により繰り返す学習法を提案する. 提案法は, 岸田

の方法の W の値 (関数の変化率を考慮する範囲) を自動的に決定する方法となっている．

ファジィルールは, 学習データの出力データの変化の激しい部分に多く配置することが必要である．このことを実現するために, 出力データの変化率を出現頻度関数として実現した．それでは, 出力変化を考慮するためにはどの近傍データまで考慮する必要があるか? 以下のアルゴリズムはこの領域 W を自動的に決定する方法を提案している．このアルゴリズムでは, 最急降下法により得られた最適なパラメータを維持しつつ, W の値を探索する方法となっている．すなわち, ベクトル量子化と最急降下法を山登り法により繰り返す方法となっている．

[**学習アルゴリズム A-HNG**]

Step A-HNG1 : しきい値 θ, 最大学習回数 T^0_{max}, T_{max}, 推論規則数 n, 自然数 $W_{max}, \beta(1{\le}W_{max}, \beta{\le}P)$, 初期推論規則数 n_0 を与える．パラメータ b^{best}_{ij}, c^{best}_{ij}, w^{best}_i の初期値をランダムに設定する．式 (2.12) より, パラメータ b^{best}_{ij}, b^{best}_{ij}, w^{best}_i を用いたときの学習用データの平均二乗誤差 E_{best} を求める．

Step A-HNG2 : $c_{ij}{\leftarrow}c^{best}_{ij}$, $b_{ij}{\leftarrow}b^{best}_{ij}$, $w_i{\leftarrow}w^{best}_i$ とする．$t = 1$ とおく．

Step A-HNG3 : 確率 $p_W(\boldsymbol{x}^1), \cdots, p_W(\boldsymbol{x}^P)$ に従い, $p(1{\le}p{\le}P)$ を選択する．

Step A-HNG4 : データ $\boldsymbol{x}^p$ を与える．

Step A-HNG5 : 式 (4.3) より, c_{ij} を更新する．

Step A-HNG6 : $t = T^0_{max}$ ならば $t = 1$ として Step A-HNG7 へ, $t < T^0_{max}$ ならば $t{\leftarrow}t + 1$ として Step A-HNG3 へ行く．

Step A-HNG7 : 式 (4.5) により b_{ij} を決定する．

Step A-HNG8 : $p = 1$ とおく．

Step A-HNG9 : データ $(x^p_1, \cdots, x^p_m, y^r_p){\in}\boldsymbol{D}$ を与える．

Step A-HNG10 : 式 (2.7) と (2.8) より, μ_i と y を求める．

Step A-HNG11 : 式 (2.17), (2.18), (2.16) より, パラメータ c_{ij}, b_{ij}, w_i を更新する．

Step A-HNG12: $p = P$ ならば Step A-HNG13 へ, $p < P$ ならば $p{\leftarrow}p + 1$ として Step A-HNG9 へ行く．

Step A-HNG13: $E(t)$ をステップ t での学習用データの平均二乗誤差 (式 (2.12)) とする．$E(t) > \theta$ かつ $t < T_{max}$ ならば $t{\leftarrow}t + 1$ として Step A-HNG10 へ, $E(t){\le}\theta$ または $t > T_{max}$ かつ $W < W_{max}$ ならば Step A-HNG14 へ行く．

Step A-HNG14: $E(t) < E_{best}$ ならば $c^{best}_{ij}{\leftarrow}c_{ij}$, $b^{best}_{ij}{\leftarrow}b_{ij}$, $w^{best}_i{\leftarrow}w_i$, $E_{best}{\leftarrow}E(t)$ とする．

Step A-HNG15: $E(t) > \theta$ かつ $W < W_{max}$ ならば $W{\leftarrow}W + \beta$ として Step A-HNG2 へ行く．$E(t){\le}\theta$ ならば学習を終了する．$E(t){\ge}\theta$ かつ $W{\ge}W_{max}$ ならば $n{\leftarrow}n + 1$ として Step A-HNG2 へ行く．

アルゴリズム A-HNG の大まかな流れを図 4.3 と 4.4 に示す．

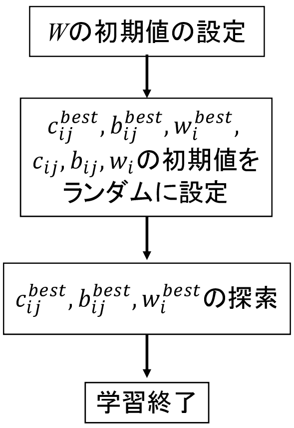


図 4.3: 学習アルゴリズム A-HNG の流れ

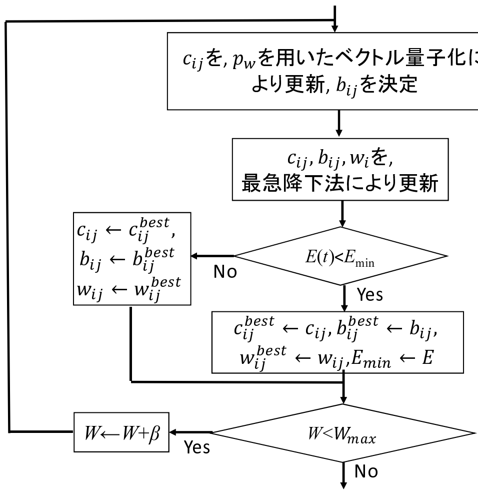


図 4.4: c_{ij}^{best}, b_{ij}^{best}, w_i^{best} の探索

4.5 数値シミュレーション

関数近似問題により, 提案手法の有効性を示す. 比較対象としては, アルゴリズム A*, A-NG, A-NGp, A-HNG を用いる. 以下に示す関数を用いて, 学習データに対する関数同定を行う.

$$y = \sin(\pi x_1^3)x_2 \tag{4.8}$$

$$y = \frac{\sin(10(x_1-0.5)^2+10(x_2-0.5)^2)+1}{2} \tag{4.9}$$

$$y = \frac{(2x_1+4x_2^2+0.1)^2}{37.21} \times \frac{(4\sin(\pi x_3)+2\cos(\pi x_4)+6)}{12} \tag{4.10}$$

$$y = \frac{(\sin(2\pi x_1)\times\cos(x_2)\times\sin(\pi x_3)\times x_4+1.0)}{2.0} \tag{4.11}$$

入力値の定義域は式 (4.8), (4.9) が $[0,1]^2$, 式 (4.10), (4.11) が $[-1,1]^4$ である. 学習用データおよびテスト用データは定義域からランダムに抽出する. 学習用データ数は式 (4.8), (4.9) が 200 個, 式 (4.10),(4.11) が 512 個である. テスト用データ数は式 (4.8), (4.9) が 2500 個, 式 (4.10), (4.11) が 6400 個である. 実験において, 推論ルール数の初期値を 2 として学習を行い, $E<\theta=1.0\times10^{-4}$ であれば学習終了, $E\geq\theta=1.0\times10^{-4}$ であれば推論ルールを逐次増やして同様の操作を繰り返す.

学習終了後, 得られた推論規則の数 n と学習終了時の評価値 E, およびテスト用データの平均二乗誤差 (Mean Square Error:MSE) を比較する. ここで, 実験条件は, T_{max} を最急降下法部分における更新回数, K_c を c_{ij} の学習係数, K_b を b_{ij} の学習係数, K_w を w_i の学習係数, W_0 をアルゴリズム A-HNG における W の初期値, β をアルゴリズム A-HNG における W の変化量, W_{max} をアルゴリズム A-HNG における W の最大値として, 表 4.1 に示す.

表 4.1: 関数近似における実験条件

	A*	A-NG	A-NGp	A-HNG
T_{max}	50000	50000	50000	5000
$K_{c_{ij}}$	0.01	0.01	0.01	0.01
$K_{b_{ij}}$	0.01	0.01	0.01	0.01
K_{w_i}	0.1	0.1	0.1	0.1
ε_{init}		0.1	0.1	0.1
ε_{fin}		0.01	0.01	0.01
λ		0.7	0.7	0.7

	Eq(4.8)	Eq(4.9)	Eq(4.10)	Eq(4.11)
θ	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-4}
W	100	100	200	200
W_0	100	100	200	200
β	10	10	50	50
W_{max}	140	140	400	400

表 4.2: 関数近似の結果

		Eq(4.8)	Eq(4.9)	Eq(4.10)	Eq(4.11)
A^*	推論規則数	7.2	19.2	4.5	21.8
	学習用データの MSE	0.42	8.77	0.29	9.59
	テスト用データの MSE	2.25	17.54	0.42	10.31
A-NG	推論規則数	9.0	17.0	5.1	14.0
	学習用データの MSE	0.44	0.36	0.23	0.75
	テスト用データの MSE	4.52	7.14	0.32	1.32
A-NGp	推論規則数	6.9	14.5	3.4	11.2
	学習用データの MSE	0.60	0.58	0.31	0.76
	テスト用データの MSE	1.64	3.71	0.36	1.60
A-HNG	推論規則数	5.5	8.0	3.2	8.0
	学習用データの MSE	0.30	0.41	0.34	0.74
	テスト用データの MSE	0.79	2.20	0.39	1.33

表 4.2 にシミュレーションの結果を示す．ここで, n は学習の結果得られた推論規則の数の平均値および学習用データ, テスト用データの平均二乗誤差 (1.0×10^{-4}) である．また, シミュレーション結果は 10 回試行の平均である．

学習アルゴリズム A-HNG は他の学習アルゴリズムに比べて推論規則数が少なく, また, テスト用データの平均二乗誤差も他の手法に比べ小さい．

4.6 まとめ

本章ではベクトル量子化と最急降下法を組み合わせた簡略型ファジィ推論システムの学習法について述べた．従来手法としては学習用データの入力のみを考慮したものや入力と出力をともに考慮したベクトル量子化を行った後に最急降下法を用いる方法がある．一方で, 提案手法であるベクトル量子化を用いた学習法では, 学習用データの入力と出力を考慮したベクトル量子化を行った後に最急降下法を用いる, というサイクルを山登り法の基づいて, 複数回繰り返すものである．数値シミュレーションにおいては, ベクトル量子化を用いる方法は他の手法に比べて少ない推論規則数で高い精度を実現できることを示した．さらに, 提案手法をパターン認識問題に応用し, その有効性を示している [72].

第5章　メタヒューリスティクを用いたファジィ推論法の学習

5.1　はじめに

メタヒューリスティクは, 問題に対するパラメータの作る解空間から, 大域探索や局所探索を用いて目的関数の最適解や準最適解を効率的に探索することを目的とする [37, 38]. ファジィ推論システムの学習にメタヒューリスティクを用いるアプローチ法は, 遺伝的アルゴリズムをはじめとして多くの研究が行われてきた. 近年の研究では, パラメータの定義域を制限することにより, 説明能力を保ちつつ精度の向上を実現する研究が行われている. これらには, 直接メタヒューリスティクを学習に用いる研究と, ファジィ推論システムの学習法と組み合わせる研究が行われている [56, 57, 58].

本章では, 後者についてのモデルの提案と, その近似能力について考察する. 前者についての考察は文献 [71] に結果を示している. 提案モデルとしては, 多点探索法である EM 法と標準的なランダムサーチを大域探索として用い, これにファジィ推論システムの最急降下法を組み合わせるモデルである. このような両者を組み合わせたハイブリッドアルゴリズムの有効性を, 数値シミュレーションにより示す.

5.2　予備概念

5.2.1　最適化問題

本章で扱う最適化 (最小化) 問題は, $\boldsymbol{L}$ や $\boldsymbol{U}$ により定義された領域内にある変数 $\boldsymbol{x}$ に対して関数 $f(\boldsymbol{x})$ が最小となる $\boldsymbol{x}$ を求めるものである. 数学的には, 以下のように定義される [38, 46];

$$\min_{\boldsymbol{L}\leq\boldsymbol{x}\leq\boldsymbol{U}} \; f(\boldsymbol{x}) \tag{5.1}$$

ここで, $f(\boldsymbol{x})$ を最小化する目的関数, $\boldsymbol{x}\in\boldsymbol{R}^N$ を目的関数の説明変数を集めたベクトル (パラメーター), $\boldsymbol{L} = (l_1, \cdots, l_N)$, $\boldsymbol{U} = (u_1, \cdots, u_N)$ をそれぞれ $\boldsymbol{x}$ の下限および上限とする.

例えば, 簡略型ファジィ推論システムの学習においては, $f(\boldsymbol{x})$ は学習用データの平均二乗誤差 E, $\boldsymbol{x}$ は推論規則のパラメータ群 $\{c_{ij}, b_{ij}, w_i | i{\in}Z_r, j{\in}Z_m\}$ として, 最小化問題 (5.1) を解くことで学習用データに適応する推論規則を得ることができる.

5.2.2 メタヒューリスティク

メタヒューリスティクとは, 最適化問題に対する解法のアルゴリズムについて, 特定の問題に依存しない汎用のヒューリスティックとして知られている. 解の精度はそれほど高くないが, 多くの問題に汎用的に対応できるという利点を持っている. 例としては, 進化的アルゴリズムとして GA(Genetic Algorithm) や GP(Genetic Program), 群知能として PSO(Particle Swarm Optimization), ACO(Ant Colony Optimization) や EM(Electromagnetism-like Mechanism), 近傍探索法としてランダムサーチ、タブーサーチや SA(Smulated Annealing), その他ニューラルネットワークやファジィ推論システム等も含まれる [37, 39, 42, 45, 46]. このうち, 進化と群知能アルゴリズムは, 多点探索を含み, 残りは単独の解を更新するアルゴリズムとなっている. 本章では, EM とランダムサーチを用いる.

(a)EM 法 [47, 48, 49, 50]

EM 法は, 相互作用, 引力, 斥力といった電荷を帯びた量子の振る舞いを模したものである. 通常の EM 法の流れを図 5.1 に示す [47]. EM 法は主に 3 つのステップから成る. はじめに (ステップ 1), 粒子群 (回の候補) の初期化を行っている. 次に (ステップ 4), 近傍での最適解を探す局所探索が行われている. さらに (ステップ 5 と 6), 粒子 (解の候補) 間に働く力と粒子間の距離に基づき移動する方法を示している. 局所探索によりすべてのサンプル点の更新を行った後, 評価関数の値が最も小さいサンプル点を $\boldsymbol{x}^{best}$ とする.

図 5.3 と 5.4 は大域探索のアルゴリズムを示す. 大域探索においては, 各サンプル点は他のサンプル点の位置および評価関数の値を基に, 位置の移動を行う [47].

図 5.2 の局所探索においては, 各パラメータの近傍内でより高い評価をもつパラメータを探索し, 最小のパラメータを求めるアルゴリズムである. また, 図 5.3 の大域探索では, はじめに各パラメータの相対的な評価値を電荷の強さとして定義し, これを用いて各電荷間の相互作用により受ける力を F として定義する. この力は自身より評価の高い電荷からは吸引力, そうでない場合は斥力として定義される. この F を用いて, 各電荷 (パラメータ) の移動距離を計算する (更新する).

(b) ランダムサーチ [51, 52, 53]

図 5.5 のランダムサーチでは, 1 つのパラメータから出発する. このパラメータの移動先を領域内のすべての範囲からランダムに決定する. ただし, この移動は (更新は) 移動先のパラメータの評価値が, 元のパラメータの評価値より高い場合のみ実行する. そうでない場合は, 元のパラメータから別の移動先を選択する. この探索は, 限られた回数 MAX だけ繰り返す.

アルゴリズム EM(n, MAX, LS, δ)

$\{\boldsymbol{x}^i | i \in Z_n\}$: サンプル点 (解の候補)
MAX: 最大探索回数
LS: 局所探索の最大探索回数
δ: 局所探索におけるパラメーター ($\delta \in J$)

```
Initialize(初期化)
iteration ←1
while iteration < MAX do
      Local(LS, δ)
      F←Calc F()
      Move(F)
      iteration ← iteration + 1
end while
```

図 5.1: EM 法の概略

アルゴリズム Local(LS, δ)

u_k : k 番目の入力要素の上限 ($k \in Z_N$)
l_k : k 番目の入力要素の下限 ($k \in Z_N$)
U$(0,1)$: $[0,1]$ 区間上の一様分布

```
counter ←1
Length ← δ (max_k{u_k − l_k})
for i = 1 to n do
 for k = 1 to N do
      λ_1 ← U (0, 1)
      while counter < LS do
         x' ← x^i
         λ_2 ← U (0, 1)
         if λ_1 > 0.5 then
            x'_k←x'_k + λ_2(Length)
         else
            x'_k←x'_k − λ_2(Length)
         end if
         if f(x') ¡ f(x^i) then
            x^i←x'
            counter ← LS - 1
          end if
          counter ← counter + 1
       end while
    end for
end for
x^best←arg min_{x^i}{f(x^i)}
```

図 5.2: EM 法における局所探索

アルゴリズム Calc F()

$||\boldsymbol{a}-\boldsymbol{b}||$: $\boldsymbol{a}$ と $\boldsymbol{b}$ の距離

$(||\boldsymbol{a}|| = \sqrt{\sum_{i=1}^{m} a_i^2}$, $\boldsymbol{a} = (a_1, \cdots, a_n))$

for i=1 to n
  $q_i \leftarrow \exp\left(-n\frac{(f(\boldsymbol{x}^i)-f(\boldsymbol{x}^{best}))}{\sum_{k=1}^{n}(f(\boldsymbol{x}^k)-f(\boldsymbol{x}^{best}))}\right)$
  $F^i \leftarrow 0$
end for
for i=1 to n
  for j=0 to n
    if $f(\boldsymbol{x}^i) < f(\boldsymbol{x}^j)$ then
      $F^i \leftarrow F^i + (\boldsymbol{x}^i - \boldsymbol{x}^j)\frac{q^i q^j}{||\boldsymbol{x}^i - \boldsymbol{x}^j||^2}$
    else
      $F^i \leftarrow F^i$ - $(\boldsymbol{x}^i - \boldsymbol{x}^j)\frac{q^i q^j}{||\boldsymbol{x}^i - \boldsymbol{x}^j||^2}$
    end if
  end for
end for

図 5.3: EM 法における粒子間に働く力の計算

アルゴリズム Move (F)

for i=1 to n
  if $i \neq best$ then
    $\alpha \leftarrow$ U(0,1)
    $F^i \leftarrow \frac{F^i}{||F^i||}$
    for k=1 to N
      if $F_k^i > 0$ then
        $x_i = x_i + \alpha F_k^i (u_k - x_k^i)$
      else
        $x_i = x_i + \alpha F_k^i (x_k^i - l_k)$
      end if
    end for
  end if
end for

図 5.4: EM 法における粒子の移動

アルゴリズム RS(MAX, LS, δ)

$\boldsymbol{x}$: サンプル点 (解の候補)

MAX: 最大探索回数

Initialize(初期化)
$iteration \leftarrow 1$
while $iteration < MAX$ do
  $\triangle\boldsymbol{x}$ をランダムに生成
  if $\boldsymbol{x} + \triangle\boldsymbol{x}$ が解の候補
    if $f(\boldsymbol{x} + \triangle\boldsymbol{x}) < f(\boldsymbol{x})$
      $\boldsymbol{x} \leftarrow \boldsymbol{x} + \triangle\boldsymbol{x}$
    end if
  end if
end while

図 5.5: ランダムサーチの概略

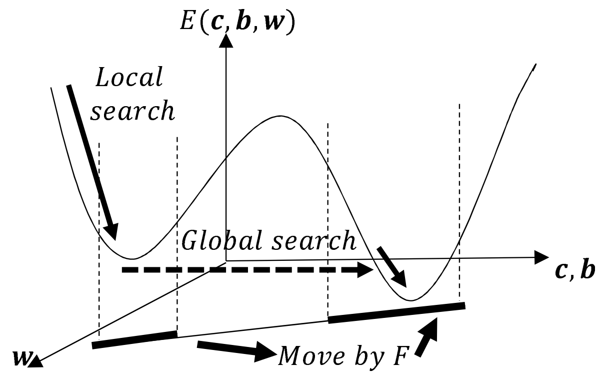


図 5.6: ハイブリッド EM 法による探索の概略

5.3　ハイブリッドファジィ推論モデル

最急降下法のみを用いる方法を除いて, ほとんどのメタヒューリスティクは, 大域と局所探索の機能を持っている. メタヒューリスティクは汎用の問題に適用される一方で, 十分な精度を得るには多くの時間を必要とする. そのために, 問題に応じて探索法を工夫することが望まれる. 本章では, メタヒューリスティクの大域探索とファジィ推論システムの学習法である最急降下法を組み合わせる学習法を提案する.

5.3.1　ハイブリッド EM 法とファジィ推論システム

EM 法 [47] は最適化問題を解くメタヒューリスティク手法の一つである. しかしながら, これは最適または準最適な解を得るまでに多くの時間を必要とする問題点がある. この問題点を解決する手法として, EM 法と最急降下法を組み合わせたハイブリッド EM 法が提案されている. この原理を 1 つのパラメータの動きを追跡することで説明する.

図.5.6 に示すように, 目的関数 E はパラメータ $\boldsymbol{c}$, $\boldsymbol{b}$, $\boldsymbol{w}$ から決定される. パラメータの初期位置がランダムに決定され, その後, 各パラメータは最急降下法により更新される. 最急降下法を用いているため, $E(\boldsymbol{c},\boldsymbol{b},\boldsymbol{w})$ はパラメータの更新が行われるたびに小さくなる. 一方, 通常の EM 法の局所探索においては各パラメータの近傍において $E(\boldsymbol{c},\boldsymbol{b},\boldsymbol{w})$ が小さくなる新たなパラメータの探索が行われるが, 長い時間を必要とする. さらに, $E(\boldsymbol{c},\boldsymbol{b},\boldsymbol{w})$ が停滞した場合には, 合力 F によるパラメータの更新 (大域探索) が行われる.

最急降下法においては, 局所的に良好なパラメータを得ることはできるが, その後解は停滞し, 大域的に良好なパラメータを得ることは困難である. そのため, EM 法における合力 F のようなパラメータの大域的な更新を必要とする.

アルゴリズム ハイブリッド EM(m, MAX, LS, δ)
$\{\boldsymbol{x}^i | i\in Z_m\}$: サンプル点 (解の候補)
MAX: 最大探索回数
LS: 局所探索の最大探索回数
δ: 局所探索におけるパラメーター ($\delta\in J$)
1: Initialize(初期化)
2: $iteration \leftarrow 1$
3: while $iteration < MAX$ do
4: 　　局所探索 (最大学習回数:LS, 学習係数:δ)
5: 　　$F\leftarrow$Calc F()
6: 　　Move(F)
7: 　　$iteration \leftarrow iteration + 1$
8: end while

図 5.7: ハイブリッド EM 法の概略

図 5.7 にハイブリッド EM 法を与える．この場合, ステップ 4 の局所探索では, ファジィ推論の学習アルゴリズム A のステップ A3 から A8 が実行される.

5.3.2 ハイブリッドランダムサーチ

ランダムサーチ [51] は広い解空間内をランダムに探索するため, 探索に時間がかかる．そこで, ハイブリッド EM と同様に最急降下法を組み合わせることで, ランダムサーチの大域探索と最急降下法の局所探索を行う．以降, これをハイブリッドランダムサーチ (Hybrid Random Search : HRS) と呼ぶ．その原理は図 5.6 の場合と同様である [53].

図 5.8 に, ハイブリッドランダムサーチのアルゴリズムを示す．この場合, ファジィ推論システムのハイブリッド法においては, ステップ 4 の局所探索がファジィ推論システムの学習アルゴリズム A のステップ A3∼A8 となる.

5.4 数値シミュレーション

本節では, 簡略型および Mamdani 型ファジィ推論法の推論規則の学習に最急降下法, ハイブリッドランダムサーチ, ハイブリッド EM を用いたときの能力の比較を行う．以下, Case 1:簡略型ファジィ推論法 (c, b, w の学習) と Case 2:Mamdani 型ファジィ推論法 (w のみの学習) とする.

アルゴリズム HRS(MAX, LS, RS, ε_0, ε_1)
$\boldsymbol{p}$: サンプル点 (解の候補)
MAX: 最大探索回数
Initialize(初期化)
$iteration \leftarrow 1$
while $iteration < MAX$ do
    局所探索 (最大学習回数:LS, 学習終了条件 ε_0)
    while $rs < RS$ and $f(\boldsymbol{\varepsilon_1})$ do
        $\triangle \boldsymbol{x}$ をランダムに生成
        if $\boldsymbol{x} + \triangle \boldsymbol{x}$ が解の候補
            if $f(\boldsymbol{x} + \triangle \boldsymbol{x}) < f(\boldsymbol{x})$
                $\boldsymbol{x} \leftarrow \boldsymbol{x} + \triangle \boldsymbol{x}$
            end if
        end if
    end if
end while

図 5.8: Outline of Random search

5.4.1 関数近似問題

以下の関数近似問題に対して従来手法と提案手法の比較を行う.

ここで, 入力空間は $[0,1]^4$(式 (5.2), (5.3)), $[-1,1]^4$(式 (5.4), (5.5)) である. 学習用データ数は 512, テスト用データ数は 6400 である. 推論規則数は, Case 1 が 81, Case 2 が 256 である.

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{37.21} \times \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6)}{12} \tag{5.2}$$

$$y = \frac{(\sin(2\pi x_1)\cos(\pi x_2)\sin(\pi x_3)x_4 + 1.0)}{2.0} \tag{5.3}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6)}{446.52} \tag{5.4}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(3e^{3x_3} + 2e^{-4x_4})^{-0.5} - 0.077}{4.68} \tag{5.5}$$

以下の手法により簡略型および Mamdani 型ファジィ推論法の推論規則の学習を行う.

(1) 最急降下法 (最大学習回数 20000 回, しきい値 $\theta = 10^{-5}$)
(2)HRS(20, 1000, 10000, 10^{-5}, 10^{-5})
(3)HEM(20, 1000, 10^{-5})

ここで, 最急降下法における学習係数は, $K_c = 0.01$, $K_w = 0.01$, $K_w = 0.1$ とする.

表 5.1: Case 1 における関数近似の結果.

		Eq.(5.2)	Eq.(5.3)	Eq.(5.4)	Eq.(5.5)
(1)	学習用データの MSE	0.10	0.10	0.10	0.10
	テスト用データの MSE	1.92	3.29	3.29	1.62
(2)	学習用データの MSE	0.08	0.09	0.09	0.10
	テスト用データの MSE	1.83	2.56	2.56	2.05
(3)	学習用データの MSE	0.09	0.09	0.09	0.09
	テスト用データの MSE	1.92	2.79	2.79	1.56

表 5.2: Case 2 における関数近似の結果.

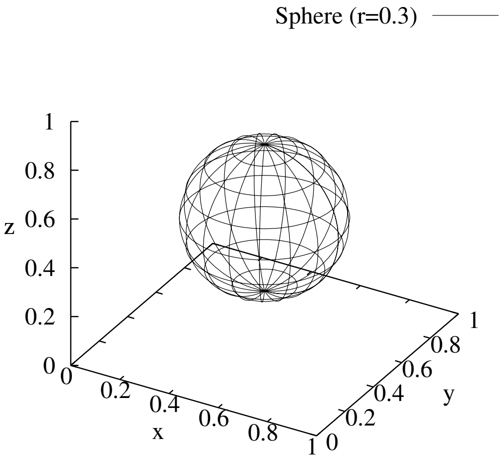
		Eq.(5.2)	Eq.(5.3)	Eq.(5.4)	Eq.(5.5)
(1)	学習用データの MSE	0.41	0.10	1.83	1.83
	テスト用データの MSE	7.74	14.47	21.16	23.89
(2)	学習用データの MSE	1.26	1.87	2.99	2.88
	テスト用データの MSE	21.76	23.96	33.85	31.63
(3)	学習用データの MSE	0.38	1.11	1.95	2.15
	テスト用データの MSE	3.27	7.69	13.81	14.32

表 5.1, 5.2 に Case 1(簡略型) および Case 2(Mamdani 型) における結果を示す. 表中の各値は, 上が学習用データに対する平均二乗誤差, 下がテスト用データに対する平均二乗誤差 ($\times 10^{-4}$) を意味する. いずれの問題においても, 簡略型および Mamdani 型ともにハイブリッド EM が他の探索手法を用いた場合よりも学習用およびテスト用データの平均二乗誤差が小さくなっており, 優れた探索結果を示している.

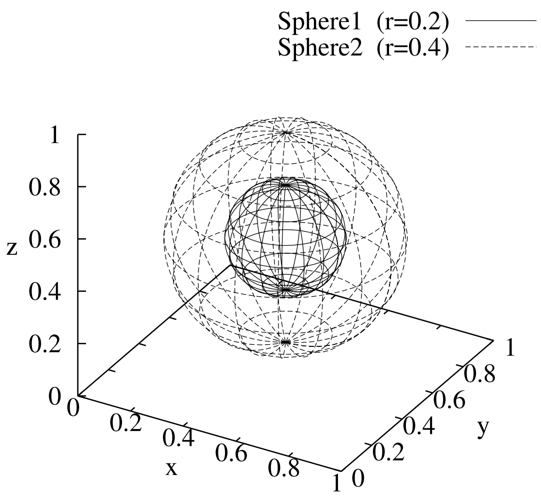
5.4.2 パターン認識問題

ここでは, 表 5.9 に示す 2 分類のパターン認識問題により, 各学習手法を用いた場合の能力の比較を行う.

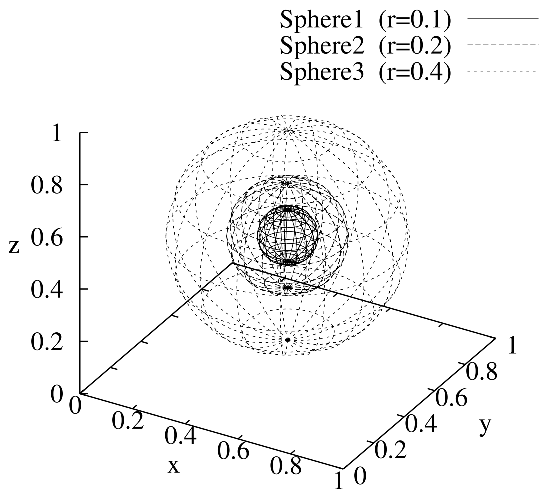
本分類問題においては, $[0,1] \times [0,1] \times [0,1]$ 空間内に存在する点を 2 種類のクラスに分類する (クラス 0 とクラス 1). クラスの境界は $(0.5, 0.5, 0.5)$ を中心とする球として与えられる. Sphere では, 球の内側にある点をクラス 1, 外側にある点をクラス 0 とする. Double-Sphere では, Spheres 1 と 2 の間にある点をクラス 1, 他の位置にある点をクラス 0 とする. Triple-Sphere では, Sphere 1 の内側および Sphere 2 と Sphere 3 の間にある点をクラス 1, 他の位置にある点をクラス 0 とす



(a) Sphere



(b) Double-Sphere



(c) Triple-Sphere

図 5.9: 2 分類のパターン認識

表 5.3: Case 1 における誤分類率

		Sphere	Double-Sphere	Triple-Sphere
(1)	学習用データ	0.007	0.033	0.037
	テスト用データ	0.027	0.078	0.078
(2)	学習用データ	0.033	0.105	0.087
	テスト用データ	0.041	0.117	0.099
(3)	学習用データ	0.013	0.065	0.065
	テスト用データ	0.026	0.089	0.087

表 5.4: Case 2 における誤分類率

		Sphere	Double-Sphere	Triple-Sphere
(1)	学習用データ	0.018	0.021	0.030
	テスト用データ	0.032	0.049	0.057
(2)	学習用データ	0.022	0.021	0.036
	テスト用データ	0.031	0.046	0.058
(3)	学習用データ	0.017	0.021	0.021
	テスト用データ	0.032	0.046	0.051

る．理想的な出力値 y_p^r は次のように与える: もし点 $\boldsymbol{x_p}$ がクラス 0 に属するなら $y_p^r = 0.0$, クラス 1 に属するなら $y_p^r = 1.0$ とする．学習用データの数は 512, テスト用データの数は 6400 である．

表 5.3, 5.4 に, 推論規則数が 27 のときの Case 1 と推論規則数が 125 のときの Case 2 の結果を示す．学習用データ数は 512, テスト用データ数は 6400 で, 入力データは空間 $[0,1]^3$ よりランダムに選ぶ. 表中の値は, 上が学習用データの誤分類率, 下がテスト用データの誤分類率である．

いずれの問題においてもハイブリッド EM における誤分類率は低く, 高い近似精度を実現している．

5.5 まとめ

本章では, メタヒューリスティクを用いた学習方法について述べた．推論規則の決定方法として広く用いられている最急降下法は局所探索であり, 学習に必要な時間は短いが推論精度は高くはない．一方, メタヒューリスティク (EM 法とランダムサーチ) は得られる推論精度は高いが, 局所探索の効率が悪く学習に必要な時間が長いという問題点がある．そこで, メタヒューリスティクを用いたファジィ推論

システムの学習法を実現することで効率的な探索を行い, 比較的短い時間で良好な解を得ることができた. 数値シミュレーションにおいては, 多点探索を行うハイブリッド EM を用いた方がハイブリッドランダムサーチを用いた場合に比べに比べて精度の高い簡略型ファジィ推論法の推論規則を得ることができた.

第6章　少数入力モジュール型ファジィ推論システム

6.1　はじめに

ベクトル量子化やメタヒューリスティクを用いたファジィ推論システムの自動構築法に関する様々な研究が行われている．しかしながら, これらの方法は入力要素数の増加に伴って, ファジィ推論システムの自動構築が困難となることが知られている．この問題を克服するためにいくつかの方法が提案されているが, 有効な方法は知られていない．さらに, ファジィ推論システムの学習法については, 近似精度と説明能力のトレードオフがあることも知られている [54, 55, 56, 58]. SIRMs(Single Input Rule Modules) モデルは単一入力型の推論規則を用いるファジィ推論システムの一つであり, 入力変数が多い問題に対しても推論規則数が抑えられるという特徴を持つ [30, 62]. 一方, SIRMs モデルを複雑な問題に適用しても, 解の近似精度は必ずしも高くないことも知られている [32, 59]. そのため, SIRMs モデルをより一般化したものとして SNIRMs(Small Number Input Rule Modules) モデルが提案されている．これは, 1 つの推論規則中で扱う入力変数の数が少ないモジュールから構成されるファジィ推論システムである [32, 59, 68]. 特に, 2 変数のモジュールから成る DIRMs(Double Input Rule Modules) モデルは, SNIRMs の有効なモデルの 1 つとして提案されている．

本章では, SNIRMs モデルの導入並びに, 学習による近似精度の検討を行う．さらに, モデルの近似能力について, 理論的な考察を行う．

6.2　SNIRMs 推論モデル

2 章で説明した従来モデルの簡略型ファジィ推論モデルは, システムのすべての入力変数がルールの前件部に組み込まれるため, 入力変数の増加に伴い指数関数的にルール数が増加する．本章では, 少数の入力変数の組み合わせを用いたモジュールから構成される SNIRMs 推論モデルを導入する [32, 59]. はじめに, SIRMs 推論モデルを, 続いて, SNIRMs 推論モデルを導入し, これらのモデルの能力について結果を示す．

6.2.1 SIRMs 推論モデル

SIRMs 推論モデルは, 各入力変数に対してそれぞれのルール群を作り, 対応する入力だけを前件部変数とする 1 入力の if-then 形式のファジィルールを定義して, ルール群の推論結果の重み付き総和を最終結果とする. この場合, ファジィ推論モデルは, 次のようになる [30].

$$\text{SIRMs} - j\ \{R_j^i : \text{if } x_j \text{ is } A_{ij} \text{ then } y^* \text{ is } w_{ij}\}_{i=1}^n \tag{6.1}$$

ここで, $j{\in}Z_m$ である.

システムの入力 x_j が入力されたとき, ルール群 SIRMs モデルの i 番目ルールの適合度は, 式 (6.2) で与えられ, その推論結果は, 式 (6.3) で与えられる.

$$\mu_{ij} = A_{ij}(x_j) \tag{6.2}$$

$$y_j^0 = \frac{\sum_{i=1}^n \mu_{ij} w_i^j}{\sum_{i=1}^n \mu_{ij}} \tag{6.3}$$

さらに, SIRMs 推論モデルでは, 各モジュールの重視度 h_i を設定することでシステムのパフォーマンスを向上させる. 最終の出力 y^* を次のように定義する.

$$y^* = \sum_{j=1}^m h_j y_j^0 \tag{6.4}$$

6.2.2 SNIRMs 推論モデルの定式化

本節では, SNIRMs モデルと DIRMs モデルを導入する [59,68].

U_k^m を Z_m の順序つき k タッププルの以下の集合とする. ここに, $k < m$ である.

$$U_k^m = \{l_1, \cdots, l_k | l_i < l_j \text{for } i < j\} \tag{6.5}$$

U_k^m に対する (k 次) の SNIRMs 推論モデルは, 以下のように定義される.

SNIRMs-$l_1 \cdots l_k$

$$\{R_i^{l_1\cdots,l_k} : \text{if } x_{l_1}\ M_i^{l_1} \text{ and} \cdots \text{and } x_{l_k} \text{ is } M_i^{l_k} \text{ then } y_{l_1\cdots l_l} \text{ is } w_i^{l_1\cdots l_k}\}_{i=1}^n \tag{6.6}$$

[例題 6.1]

$U_1^3 = \{1, 2, 3\}$ とすると, 次のルールが得られる.

SIRMs-1 : $\{\text{if } x_1 \text{ is } A_i^1 \text{ then } y \text{ is } w_i^1\}_{i=1}^n$

SIRMs-2 : $\{\text{if } x_2 \text{ is } A_i^2 \text{ then } y \text{ is } w_i^2\}_{i=1}^n$
SIRMs-3 : $\{\text{if } x_3 \text{ is } A_i^3 \text{ then } y \text{ is } w_i^3\}_{i=1}^n$

[例題 6.2]
$U_2^3 = \{12, 13, 23\}$ のとき, 次のルールが得られる.
SNIRMs-12 : $\{\text{if } x_1 \text{ is } A_i^1 \text{ and } x_2 \text{ is } A_i^2 \text{ then } y \text{ is } w_i^{12}\}_{i=12}^n$
SNIRMs-13 : $\{\text{if } x_1 \text{ is } A_i^1 \text{ and } x_3 \text{ is } A_i^3 \text{ then } y \text{ is } w_i^{13}\}_{i=13}^n$
SNIRMs-23 : $\{\text{if } x_2 \text{ is } A_i^2 \text{ and } x_3 \text{ is } A_i^3 \text{ then } y \text{ is } w_i^{23}\}_{i=23}^n$

$\boldsymbol{x} = (x_1, \cdots, x_m)$ とする. SNIRMs-$l_1 \cdots l_k$ の i 番目のルールの適合度と出力は, 次のように表される.

$$\mu_i^{l_1 \cdots l_k} = A_i^{l_1}(x_{l_1}) \cdots A_i^{l_k}(x_{l_k}) \tag{6.7}$$

$$y_{l_1 \cdots l_k}^0 = \frac{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k} w_i^{l_1 \cdots l_k}}{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k}} \tag{6.8}$$

これまでのパラメータである c, b, w に加えて, モジュールの重視度 h が導入される. L 番目のモジュールに対する重視度を h_L とすると, システムの出力は次のようになる.

$$y^* = \sum_{L \in U_k^m} h_L y_L^0 \tag{6.9}$$

$k = 1$ と 2 の場合は, SIRMs と DIRMs 推論モデルと呼ばれる. 図 6.1 は, 従来モデルとこれらのモデルの関係を示している.

6.2.3 SNIRMs モデルの階層性

本節では, m 変数の EX-OR 問題を使って, SNIRMs 推論モデルの能力の階層性を説明する. EX-OR 問題は, 次のように定義される.

$$y = x_1 \oplus x_2 \oplus \cdots \oplus x_m, \tag{6.10}$$

このとき, 次の結果が得られる. ここに, $x_1, \cdots, x_m, y \in \{0, 1\}$ である.
[命題 6.1] $k+1$ 変数の EX-OR 問題は, k 次の SNIRMs 推論モデルでは実現できない.
[証明] 一般性を失うことなく, $k = 2$ の場合を証明する.
2 次の SNIRMs 推論モデルが表 6.1 に示した 3 変数の問題を実現できたと仮定する.

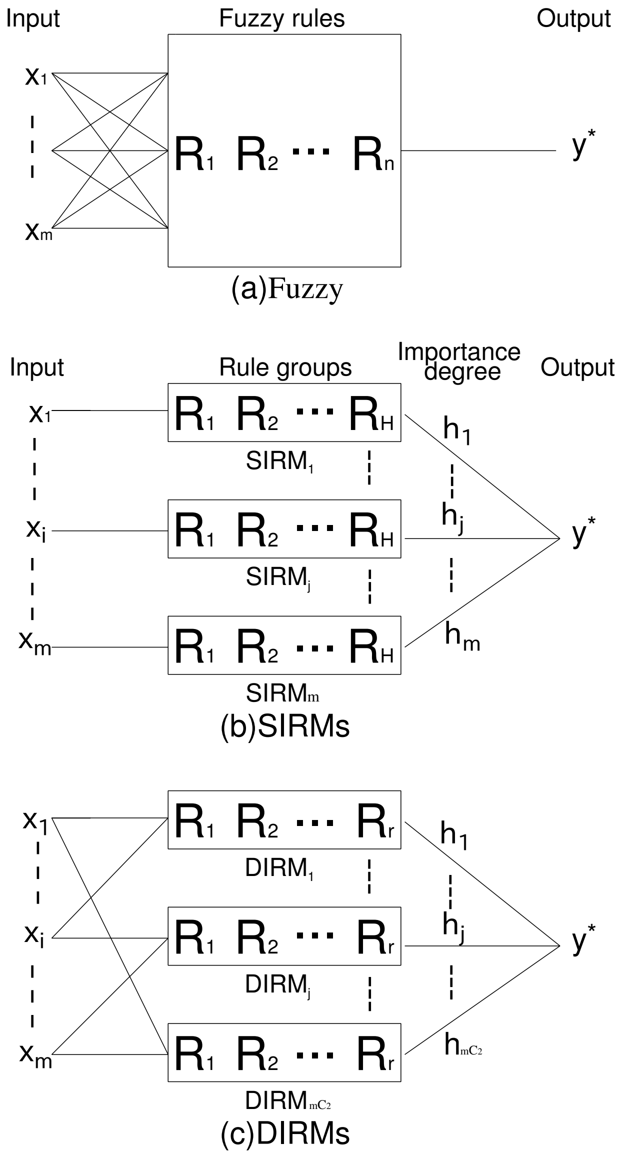


図 6.1: 簡略型, SIRMs, DIRMs の構造の概略

表 6.1: $m=3$ の場合の EX-OR 問題

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

モデルの出力と表 6.1 の関係から, 式 (6.11)~(6.18) が得られる.

$$\begin{aligned} & h_1 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^2(0) w_i^{12}}{\sum_{i=1}^{n} A_i^1(0) A_i^2(0)} + h_2 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^3(0) w_i^{13}}{\sum_{i=1}^{n} A_i^1(0) A_i^3(0)} \\ & \qquad + h_3 \frac{\sum_{i=1}^{n} A_i^2(0) A_i^3(0) w_i^{23}}{\sum_{i=1}^{n} A_i^2(0) A_i^3(0)} < 0.5 \end{aligned} \tag{6.11}$$

$$\begin{aligned} & h_1 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^2(0) w_i^{12}}{\sum_{i=1}^{n} A_i^1(0) A_i^2(0)} + h_2 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^3(1) w_i^{13}}{\sum_{i=1}^{n} A_i^1(0) A_i^3(1)} \\ & \qquad + h_3 \frac{\sum_{i=1}^{n} A_i^2(0) A_i^3(1) w_i^{23}}{\sum_{i=1}^{n} A_i^2(0) A_i^3(1)} \geq 0.5 \end{aligned} \tag{6.12}$$

$$\begin{aligned} & h_1 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^2(1) w_i^{12}}{\sum_{i=1}^{n} A_i^1(0) A_i^2(1)} + h_2 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^3(0) w_i^{13}}{\sum_{i=1}^{n} A_i^1(0) A_i^3(0)} \\ & \qquad + h_3 \frac{\sum_{i=1}^{n} A_i^2(1) A_i^3(0) w_i^{23}}{\sum_{i=1}^{n} A_i^2(1) A_i^3(0)} \geq 0.5 \end{aligned} \tag{6.13}$$

$$\begin{aligned} & h_1 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^2(1) w_i^{12}}{\sum_{i=1}^{n} A_i^1(0) A_i^2(1)} + h_2 \frac{\sum_{i=1}^{n} A_i^1(0) A_i^3(1) w_i^{13}}{\sum_{i=1}^{n} A_i^1(0) A_i^3(1)} \\ & \qquad + h_3 \frac{\sum_{i=1}^{n} A_i^2(1) A_i^3(1) w_i^{23}}{\sum_{i=1}^{n} A_i^2(1) A_i^3(1)} < 0.5 \end{aligned} \tag{6.14}$$

$$\begin{aligned} & h_1 \frac{\sum_{i=1}^{n} A_i^1(1) A_i^2(0) w_i^{12}}{\sum_{i=1}^{n} A_i^1(1) A_i^2(0)} + h_2 \frac{\sum_{i=1}^{n} A_i^1(1) A_i^3(0) w_i^{13}}{\sum_{i=1}^{n} A_i^1(1) A_i^3(0)} \\ & \qquad + h_3 \frac{\sum_{i=1}^{n} A_i^2(0) A_i^3(0) w_i^{23}}{\sum_{i=1}^{n} A_i^2(0) A_i^3(0)} \geq 0.5 \end{aligned}$$

$$h_1\frac{\sum_{i=1}^{n}A_i^1(1)A_i^2(0)w_i^{12}}{\sum_{i=1}^{n}A_i^1(1)A_i^2(0)}+h_2\frac{\sum_{i=1}^{n}A_i^1(1)A_i^3(1)w_i^{13}}{\sum_{i=1}^{n}A_i^1(1)A_i^3(1)}+h_3\frac{\sum_{i=1}^{n}A_i^2(0)A_i^3(1)w_i^{23}}{\sum_{i=1}^{n}A_i^2(0)A_i^3(1)}<0.5 \tag{6.15}$$

$$h_1\frac{\sum_{i=1}^{n}A_i^1(1)A_i^2(1)w_i^{12}}{\sum_{i=1}^{n}A_i^1(1)A_i^2(1)}+h_2\frac{\sum_{i=1}^{n}A_i^1(1)A_i^3(0)w_i^{13}}{\sum_{i=1}^{n}A_i^1(1)A_i^3(0)}+h_3\frac{\sum_{i=1}^{n}A_i^2(1)A_i^3(0)w_i^{23}}{\sum_{i=1}^{n}A_i^2(1)A_i^3(0)}<0.5 \tag{6.16}$$

$$h_1\frac{\sum_{i=1}^{n}A_i^1(1)A_i^2(1)w_i^{12}}{\sum_{i=1}^{n}A_i^1(1)A_i^2(1)}+h_2\frac{\sum_{i=1}^{n}A_i^1(1)A_i^3(1)w_i^{13}}{\sum_{i=1}^{n}A_i^1(1)A_i^3(1)}+h_3\frac{\sum_{i=1}^{n}A_i^2(1)A_i^3(1)w_i^{23}}{\sum_{i=1}^{n}A_i^2(1)A_i^3(1)}\geq 0.5 \tag{6.17}$$

(6.18)

式 (6.11), (6.15), (6.16), (6.17) を A, 式 (6.12), (6.13), (6.14), (6.18) を B とする. これに関して, A と B の左辺は互いに等しいが, A と B の右辺は等しくはない. これは矛盾である. 結果として, この問題に対するモデルは存在しない. □

[系 6.2] 2 変数の EX-OR 問題は, SIRMs 推論モデルで実現できない.

一般に, $k+1$ 次のモデルは k 次のモデルより高い能力を持っていると思われる. 一方, SIRMs モデルは AND や OR の論理関数を実現できる. 次の論理関数のシミュレーションを実行する.

$$y = x_1\vee\cdots\vee x_m \tag{6.19}$$

$$y = x_1\wedge\cdots\wedge x_m \tag{6.20}$$

ここに, $x_1, \cdots, x_m, y\in\{0,1\}$, $\vee$ と $\wedge$ は OR と AND 演算である.

表 6.2 はシミュレーションの条件を, 表 6.3 はその結果を示している.

さらに, 連続関数について考察を行う. 以下に示す関数は, 連続関数であり, その一部に排他的論理和の性質を持っていることに注意する.

$$f(x_1,x_2) = x_1+x_2-2x_1x_2 \tag{6.21}$$

関数 $f(x_1,x_2)$ をモデルで実現するとき, DIRMs 推論モデルはモジュール中のルールの数 n が増加するにつれて, 平均二乗誤差は減少するが, SIRMs では減少しない.

表 6.2: 論理関数問題における初期条件

	SIRMs		DIRMs	
	OR, AND	EX-OR	OR, AND	EX-OR
K_c	0.0001	0.001	0.0001	0.001
K_b	0.0001	0.001	0.0001	0.001
K_w	0.1	0.01	0.1	0.01
K_h	0.5	0.05	0.5	0.05
T_{max}	100	100	1000	2000
c の初期値	等間隔			
b の初期値	$\frac{1}{2(\text{分割数}-1)}\times$(入力値の範囲)			
w の初期値	[0,1] からランダムに選択			
h の初期値	[0,1] からランダムに選択			

表 6.3: 論理関数問題の結果

	SIRMs	DIRMs
OR($m = 2$)	0.00	0.00
AND($m = 2$)	0.00	0.00
EX-OR($m = 2$)	0.25	0.00

このことから, SIRMs モデルは万能性を持たないと思われる. 同様の考察から, k 次のモデル ($1 \leq k \leq m-1$) は万能性を持たないと思われる. m 次のモデル (簡略型ファジィ推論モデル) は, 万能性を持つので, k 次のモデル (SNIRMs ファジィ推論モデル) には, 階層性が存在すると思われる.

6.3 SNIRM 推論モデルの学習法

本節では, SNIRMs 推論モデルの学習法を与える. はじめに, ルール数を固定した標準的な学習法を, 次に, 生成と削除により最適なルール数もつ学習法を構築する.

6.3.1 モデルの学習法

各パラメータの 1 階微分は, 次のように与えられる.

$$\frac{\partial E}{\partial h_L} = (y^* - y^r) y_L^0, \tag{6.22}$$

$$\frac{\partial E}{\partial w_i^L} = h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^n \mu_i^L} (y^* - y^r), \tag{6.23}$$

$$\frac{\partial E}{\partial c_{rl_j}^L} = (y^* - y^r) h_L \frac{w_r^L - y_L^0}{\sum_{r=1}^n \mu_r^L} \frac{x_{rl_j} - c_{rl_j}^L}{(b_{rl_j}^L)^2} \tag{6.24}$$

$$\frac{\partial E}{\partial b_{rl_j}^L} = (y^* - y^r) h_L \frac{w_r^L - y_L^0}{\sum_{r=1}^n \mu_r^L} \frac{(x_{rl_j} - c_{rl_j}^L)^2}{(b_{rl_j}^L)^3} \tag{6.25}$$

式 (6.22), (6.23), (6.24), (6.25) から, 各パラメータの更新式は次のように与えられる.

$$h_r(t+1) = h_r(t) + (y^* - y^r) y_L^0, \tag{6.26}$$

$$w_i^L(t+1) = w_i^L(t) + h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^n \mu_i^L} (y^* - y^r), \tag{6.27}$$

$$c_{rl_j}^L(t+1) = c_{rl_j}^L(t) + (y^* - y^r) h_L \frac{w_r^L - y_L^0}{\sum_{r=1}^n \mu_r^L} \frac{x_{rl_j} - c_{rl_j}^L}{(b_{rl_j}^L)^2} \tag{6.28}$$

$$b_{rl_j}^L(t+1) = b_{rl_j}^L(t) + (y^* - y^r) h_L \frac{w_r^L - y_L^0}{\sum_{r=1}^n \mu_r^L} \frac{(x_{rl_j} - c_{rl_j}^L)^2}{(b_{rl_j}^L)^3} \tag{6.29}$$

$\boldsymbol{D}$ を学習データの集合とする. k 次 SNIRMs 推論モデルの学習式は, 次のように与えられる.
[学習アルゴリズム **B**(k)]
Step 1 [初期化]: パラメータ c_i^L, b_i^L, w_i^L の初期値, しきい値 θ_1, 最大学習回数 T_{max},

学習係数 K_c, K_b, K_w を与える.
Step 2 : $t = 1$ とする.
Step 3 : $p = 1$ とする.
Step 4 : 学習データ $(x_1^p, \cdots, x_m^p, y_p^r)$ を与える.
Step 5 : 式 (2.2) に従い, メンバーシップ関数値の計算を行う.
Step 6 [ファジィ推論の出力計算]: 式 (6.7), (6.8), (6.9) に従い, 出力 y_p の計算を行う.
Step 7 [パラメータの更新]: 勾配 $\frac{\partial E}{\partial h_L}$ を用いて, パラメータ h_L の更新を行う.
Step 8 : 勾配 $\frac{\partial E}{\partial w_i^L}$ を用いて, パラメータ w_i^L の更新を行う.
Step 9 : 勾配 $\frac{\partial E}{\partial c_i^L}$ および $\frac{\partial E}{\partial b_i^L}$ を用いて, パラメータ c_i^L および b_i^L の更新を行う.
Step 10 : もし $p = P$ ならば Step 11 へ, $p < P$ なら $p \leftarrow p + 1$ として Step 4 へ行く.
Step 11 : 式 (2.12) に従い平均二乗誤差 $E(t)$ の計算を行う. もし $E(t) < \theta_1$ なら学習を終了する.
Step 12 : もし $t \neq T_{max}$ なら, $t \leftarrow t + 1$ として Step 3 へ行く. そうでなければ学習を終了する.

簡略型, k 次の SNIRMs, SIRMs 推論モデルのルール数は, それぞれ $O(H^m)$, $O(m^k H^k)$, $O(mH)$ である. ここに, H はファジィ分割の数である. SNIRM 推論モデルも次数が高くなると, ルール数が多くなる. そこで, より最適なルール数を求める次節の学習方法が導入される.

6.3.2 生成と削除型学習法

本節は, DIRM 推論モデルに関する構造を決定する学習法を与える. 生成型の学習方法 GAL(Generative Algorithm of Learning) は, SIRMs 推論モデルに, DIRM 推論モデルのモジュールを逐次追加する方法である. q を目的のルール数とするとき, アルゴリズムは次のように与えられる.
[**学習アルゴリズム C**(q)](GAL for DIRMs model)
Step 1: SIRMs モデルを構築し, $k = 1$ の場合の学習アルゴリズム B を実行する. $i = 1$ かつ $j = 0$ とする.
Step 2: Step 1 において, i 番目に重視度の高い変数 x_0 を選び, x_0 を含む 2 変数の推論規則から成るモジュールを追加する ($1 \leq i \leq m$).
Step 3: アルゴリズム B により, 適切なパラメータの決定を行う. もし $j = q$ ならば学習を終了する. そうでない場合には, $i = i + 1$ かつ $j = j + (m - i)$ として Step 2 へ行く.

さらに, 削除的学習法 PAL(Pruning Algorithm of Learning) を提案する. このモデルは, 十分な数のルール数をもつ DIRMs 推論モデルから出発して, 目的のルー

ル数（またはしきい値）まで, 逐次ルール（モジュール）を減らす方法である.

アルゴリズムは, 以下のように与えられる.

学習アルゴリズム D(q)(PAL for DIRMs model)

Step 1: i をモジュールの数とし, $i = {}_m \mathrm{C}_2$ とする.

Step 2: モジュールの数が i の場合の DIRMs の学習アルゴリズム B を行う.

Step 3: $i = q$ またはいずれのモジュールにも含まれない入力要素が存在するとき, 学習を終了する. そうでなければ, $i = i - 1$ として Step 2 へ行く.

削除型の学習方法としては, 文献 [59] に基づいた忘却型の学習方法も考えられるが, シミュレーション実験ではほとんど能力差はなかった.

これらの方法は, 容易に任意の次数 $k(\geq 2)$ の SNIRMs モデルの学習方法に一般化できる.

6.4 数値シミュレーション

本節では, 関数近似問題およびパターン認識問題により, 各手法の能力比較を行う. 以下では, A, B, C, D はそれぞれ 簡略型, k-SNIRMs(SIRMs, DIRMs) の学習アルゴリズム, 学習アルゴリズム GAL, PAL を意味する.

6.4.1 関数近似問題

ここでは, 以下に示す 4 変数関数についての関数近似により簡略型 (A), SNIRMs(B(k)), GAL(C(q)),PAL(D(q)) の能力の比較を行う.

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{37.21} \times \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6}{12} \tag{6.30}$$

$$y = \frac{\sin(2\pi x_1) \times \cos(x_2) \times \sin(\pi x_3) \times x_4 + 1.0}{2.0} \tag{6.31}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{4\sin(\pi x_3) + 2\cos(\pi x_4) + 6}{446.52} \tag{6.32}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(3e^{3x_3} + 2e^{-4x_4})^{-0.5} - 0.077}{4.68} \tag{6.33}$$

実験条件を表 6.4 に示す. 学習用データ数を 512, テスト用データ数を 6400 とし, 式 (6.30), (6.31) は区間 $[0, 1]^4$, 式 (6.32), (6.33) は区間 $[-1, 1]^4$ からそれぞれランダ

表 6.4: 関数近似問題における実験条件

	A	B($k=1$)	B($k=2$)	C	D
K_c	0.001	0.001	0.001	0.001	0.001
K_b	0.001	0.001	0.001	0.001	0.001
K_w	0.01	0.01	0.01	0.01	0.01
K_h		0.05	0.05	0.05	0.05
T_{max}	50000	100	2000	10000	20000
c の初期値	等間隔				
b_{ij} の初期値	$\frac{1}{2(\text{分割数}-1)}\times$(入力空間の大きさ)				
w_{ij} の初期値	[0,1] からランダムに選択				
h_j の初期値	[0,1] からランダムに選択				

ムに選ぶ. 実験条件を表 6.4 に示す. 学習用データおよびテスト用データの数はそれぞれ 512 と 6400 であり, それぞれ入力空間からランダムに選ぶ. 表 6.5, 6.6 に結果を示す. ここで, 表中の学習, テスト欄は, それぞれ学習用データの平均二乗誤差, テスト用データの平均二乗誤差を意味する. 結果は 10 回試行の平均である.

実験結果より, 以下の知見が得られる:

(1) 表 6.5 より, 簡略型は SIRMs と DIRMs に比べて高い精度を実現しているが, パラメータ数が非常に多い.

(2) 表 6.5 より, DIRMs は SIRMs に比べて精度が高い.

(3) 表 6.6 より, PAL は GAL や SIRMs に比べて精度が高い. また, GAL と PAL はあまり多くないパラメータ数で高い精度を実現している.

6.4.2 パターン認識問題

SIRMs や DIRMs のような SNIRMs モデルはパラメータ数が少ないため, 入力要素数の多い問題に適用することができる. 表 6.7 に示すデータ [60] を用いたパターン認識問題により, GAL や PAL の有効性を示す. ここで, Wine と BCW データについては入力要素数が非常に多く, 推論ルール数が多くなるため, 簡略型では限られた時間内に結果が得られなかった. 実験結果を表 6.8 に示す.

2/3 のデータを学習用データとして用い, 残りの 1/3 のデータをテスト用データとして用いる. 表 6.9 は実験の初期条件を示す. 表 7.10 にパラメータ数とデータの誤分類率を示す. ここで, 表中の学習, データ欄はそれぞれ学習用データの誤分類率, テスト用データの誤分類率を意味する. 実験結果は 10 回試行の平均である.

実験結果より, 以下の知見が得られる:

(1) アルゴリズム B ($k=1$) は, Wine を除き, いずれの問題においても, いずれの

表 6.5: 関数近似の結果 1

学習手法		A	B ($k=1$)	B ($k=2$)
Eq.(7.44)	パラメータ数	729	40	276
	学習 ($\times 10^{-3}$)	0.0045	1.052	0.026
	テスト ($\times 10^{-3}$)	0.0189	1.199	0.050
Eq.(7.45)	パラメータ数	729	40	276
	学習 ($\times 10^{-3}$)	0.011	9.258	2.229
	テスト ($\times 10^{-3}$)	0.114	10.567	7.456
Eq.(7.46)	パラメータ数	729	40	276
	学習 ($\times 10^{-3}$)	0.027	6.027	0.234
	テスト ($\times 10^{-3}$)	0.234	6.682	0.615
Eq.(7.47)	パラメータ数	729	40	276
	学習 ($\times 10^{-3}$)	0.028	5.875	0.014
	テスト ($\times 10^{-3}$)	0.163	7.190	0.030

表 6.6: 関数近似の結果 2

学習手法		C($q=3$)	D ($q=3$)	D ($q=4$)
Eq.(7.44)	パラメータ数	178	138	184
	学習 ($\times 10^{-3}$)	0.443	0.431	0.154
	テスト ($\times 10^{-3}$)	0.686	0.835	0.275
Eq.(7.45)	パラメータ数	178	138	184
	学習 ($\times 10^{-3}$)	4.75	2.60	2.25
	テスト ($\times 10^{-3}$)	7.41	6.48	7.30
Eq.(7.46)	パラメータ数	178	138	184
	学習 ($\times 10^{-3}$)	1.37	0.014	0.010
	テスト ($\times 10^{-3}$)	1.98	0.020	0.016
Eq.(7.47)	パラメータ数	178	138	184
	学習 ($\times 10^{-3}$)	1.46	1.32	0.49
	テスト ($\times 10^{-3}$)	0.96	2.30	0.83

表 6.7: パターン認識で用いるデータ

	# input	# clusters	# data
Iris	4	3	150
Wine	13	3	178
BCW	9	2	683

表 6.8: パターン認識における初期条件

	B(k=1)	B($k=2$)	C	D
K_c	0.001	0.001	0.001	0.01
K_b	0.001	0.001	0.001	0.01
K_w	0.01	0.01	0.05	0.1
K_h	0.05	0.05	0.05	0.1
T_{max}	100	1000	500	20000
c の初期値	等間隔			
b の初期値	$\frac{1.5}{2(\text{分割数}-1)}\times$(入力空間の大きさ)			
w の初期値	[0,1] からランダムに選択			
h の初期値	[0,1] からランダムに選択			

表 6.9: パターン認識の結果

手法		B ($k=1$)	B ($k=2$)	C	D
Iris	パラメータ数	40	276	178($q=3$)	138($q=3$)
	学習 ($\times 10^{-2}$)	4.80	2.80	0.93	0.67
	テスト ($\times 10^{-2}$)	5.17	1.199	3.54	4.67
Wine	パラメータ数	130	3588	694($q=12$)	1196($q=26$)
	学習 ($\times 10^{-2}$)	6.00	0.00	2.22	3.92
	テスト ($\times 10^{-2}$)	27.8	12.2	12.8	9.04
BCW	パラメータ数	90	1656	458($q=8$)	368($q=8$)
	学習 ($\times 10^{-2}$)	1.28	0.67	2.16	3.22
	テスト ($\times 10^{-2}$)	3.90	4.50	3.87	5.14

手法も良好な結果を得た.
(2)GAL と PAL はいずれの問題においてもパラメータ数が少ない.
(3)GAL と PAL は入力要素数が多い問題に対して適用することができる.
(4) 簡略型を Iris に適用した場合, 誤分類率は 4.0 % である. これは, 表 6.9 中に示す各手法における結果とほぼ同等である.

6.5 まとめ

簡略型ファジィ推論モデルは高い近似精度を実現できるが, 推論規則数が多くなるという問題点がある. そこで, 本節では, 推論規則数の少ないファジィ推論モデルである SNIRMs モデルを導入し, その能力比較を行った. しかしながら, 近似精度の高い SNIRMs モデルは推論規則数が多くなる. そのため, 少ないモジュール数から成る SNIRMs モデルを実現する GAL と PAL の学習法を提案した. 数値シミュレーションにおいては, DIRMs モデルの GAL および PAL は少ない推論規則数で, 従来の DIRMs モデルと同等の精度となり, 有効な学習法であることを示した. さらに, これらの方法を用いた障害物回避問題の数値シミュレーションを行い, その有効性を示した [68].

第7章　線形入力型SIRMsモデル

7.1　はじめに

前章では, 入力変数の多い問題に対しても推論規則数を抑えることのできるモデルとして SNIRMs モデルについて述べた．しかしながら, SNIRMs は簡略型に比べて近似精度が十分でないという問題点がある．また, SIRMs モデルは推論規則数を抑えることができるが近似精度が低い．また, DIRMs モデルは SIRMs モデルと比べて近似精度は高いが推論規則数が多く, 簡略型モデルと比べて推論規則数は少なく近似精度が低いという問題が知られている．それゆえ, SIRMs モデルと同等の推論規則数で簡略型と同等の精度を実現する SNIRMs モデルの提案が望まれる．

本章では, 2 段階から成るファジィ推論法のモデルの提案を行う．第 1 段階で入力変数の線形変換を行い, 第 2 段階で SIRMs システムによる出力導出を行う．はじめに, このモデルが万能性の性質を満たすことを理論的に証明する．また, 数値実験によりこのモデルが従来手法と比べて少ない推論規則数 (パラメータ数) で高い精度を実現することを示す．さらに, この提案手法が入力要素数の多い問題に対しても有効であることを, パターン認識問題に応用することで示す．また, 制御問題への応用として, 障害物回避問題を用いて, その有効性を示す．その際に用いた推論規則の言語的解釈を行うアルゴリズムを示すことで, 提案手法が高い近似精度と説明能力を持つことを示す．

7.2　準備

はじめに, 入力 $\boldsymbol{x} = (x_1, \cdots, x_m)$ に対し, 関数 $\phi : \boldsymbol{R}^m \to \boldsymbol{R}^l$ (l は自然数) を用いて $\boldsymbol{z} = (z_1, \cdots, z_l)$ への変換を行う．

$$\boldsymbol{z} = \phi(\boldsymbol{x}) \tag{7.1}$$

ここに, $z_i = \phi_i(\boldsymbol{x})$, $i \in Z_m$, $\phi = (\phi_1, \cdots, \phi_m)$ とする．

さらに, $\boldsymbol{z}$ を入力とする SIRMs モデルを導入する．

$\boldsymbol{z}$ を入力とする SIRMs モデルの推論規則 $R_i^j (i \in Z_n, j \in Z_l)$ は,

$$R_i^j : \text{if } z_j \text{ is } A_i^j \text{ then } y_j \text{ is } w_i^j \tag{7.2}$$

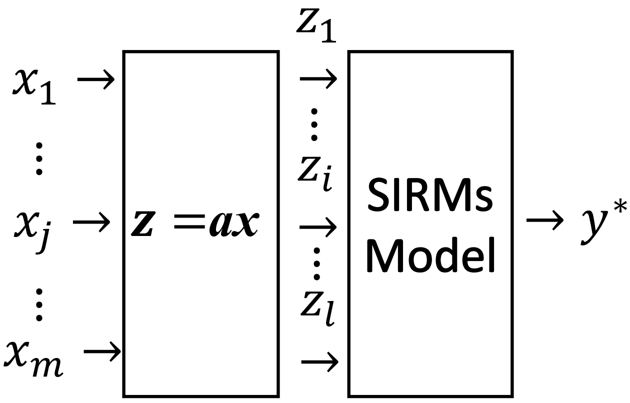


図 7.1: 線形入力型 SIRMs の概略図

出力 y は, 以下の式により導出される.

$$\mu_i^j = A_i^j(z_j) \tag{7.3}$$

$$y_j^0 = \frac{\sum_{i=1}^{n} \mu_i^j w_i^j}{\sum_{i=1}^{n} \mu_i^j} \tag{7.4}$$

$$y = \sum_{j=1}^{l} h_j y_j^0 \tag{7.5}$$

特に, 関数 ϕ が線形変換関数のとき, この推論システムを線形入力型 SIRMs モデルと呼ぶ. 以下に, このモデルとその学習法を提案する. (図 7.1 参照)

ϕ を線形変換関数とする. このとき, $z_j = a_{j0} + a_{j1}x_1 + \cdots + a_{jm}x_m (j \in Z_l)$ とおく.

$$\begin{pmatrix} z_1 \\ \vdots \\ z_j \\ \vdots \\ z_l \end{pmatrix} = \begin{pmatrix} a_{10} & \cdots & a_{1k} & \cdots & a_{1m} \\ & & \vdots & & \\ a_{j0} & \cdots & a_{jk} & \cdots & a_{jm} \\ & & \vdots & & \\ a_{l0} & \cdots & a_{lj} & \cdots & a_{lm} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_k \\ \vdots \\ \\ x_m \end{pmatrix} \tag{7.6}$$

ここで, $\boldsymbol{a} = (a_{jk})(j = 1, \cdots, l,\ k = 0, 1, \cdots, m,\ z_j = \sum_{k=0}^{m} a_{jk}x_k\ (x_0 = 1))$ とする. l は近似精度に応じた自然数を選ぶ.

規則 (7.2) に基づき, 式 (7.3), (7.4), (7.5) により出力 y を求める.

式 (2.2) に示すガウス型関数を用いた場合, 最急降下法による学習においては, 式 (2.13) の $\frac{\partial E}{\partial \alpha}$ として以下のような式が得られる.

$$\frac{\partial E}{\partial h_j} = (y - y^*) y_j^0, \tag{7.7}$$

$$\begin{aligned}
\frac{\partial E}{\partial w_i^j} &= h_j \frac{\mu_i^j}{\sum_{i=1}^n \mu_i^j}(y-y^*), & (7.8)\\
\frac{\partial E}{\partial c_i^j} &= (y-y^*)h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^n \mu_i^j}\frac{z_j - c_i^j}{(b_i^j)^2}, & (7.9)\\
\frac{\partial E}{\partial b_i^j} &= (y-y^*)h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^n \mu_i^j}\frac{(z_j - c_i^j)^2}{(b_i^j)^3}. & (7.10)
\end{aligned}$$

また, 線形変換の係数 a_{jk} については, 以下の関係が得られる:

$$\frac{\partial E}{\partial a_{jk}} = (y^* - y)\frac{h_j}{\sum_{i=1}^n \mu_i^j}\sum_{i=1}^{n}\mu_i^j(w_i^j - y_j^0)\frac{z_j - c_i^j}{\left(b_i^j\right)^2} \tag{7.11}$$

線形入力型 SIRMs モデルの学習アルゴリズムは, 以下のように与えられる :
[学習アルゴリズム **C**]
1 [Initialization]: パラメータ c_i^L, b_i^L, w_i^L, a_{jk} の初期値, しきい値 θ_1, 最大学習回数 T_{max}, 学習係数 K_c, K_b, K_w, K_a を与える.
2 : $t = 1$ とする.
3 : $p = 1$ とする.
4 : 学習データ $(x_1^p, \cdots, x_m^p, y_p^r)$ を与える.
5 : $\sum_{k=0}^m a_{jk}x_k$ および式 (2.2) に従い, メンバーシップ関数 $A_i^j(z_j)$ の値の計算を行う.
6 [Output of fuzzy inference]: 式 (7.3), (7.4), (7.5) に従い, 出力 y_p の計算を行う.
7 [Updating parameters]: 式 (7.7) を用いて, パラメータ h_j の更新を行う.
8 : 式 (7.8) を用いて, パラメータ w_i^j の更新を行う.
9 : 式 (7.9), (7.10), (7.11) を用いて, パラメータ c_i^j, b_i^j, a_{jk} の更新を行う.
10 [Termination]: もし $p = P$ ならば Step 11 へ, $p < P$ なら $p \leftarrow p + 1$ として Step 4 へ行く.
11 : 式 (2.12) に従い平均二乗誤差 $E(t)$ の計算を行う. もし $E(t) < \theta_1$ なら学習を終了する.
12 : もし $t \neq T_{max}$ なら, $t \leftarrow t + 1$ として Step 3 へ行く. そうでなければ学習を終了する.

同様に, 三角型メンバーシップ関数についても定義できる.

[例題 7.1]
ここでは, 例として 2 入力の EX-OR 問題を用いて, 線形入力型 SIRMs モデルのアイデアについて述べる.
アルゴリズム C により得られた線形入力型 SIRMs モデルの推論規則を示す:

$$z = 1.07 - 0.51x_1 - 0.64x_2, \tag{7.12}$$

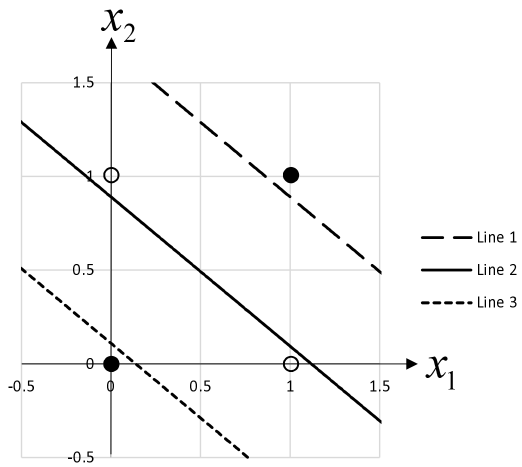


図 7.2: 線形入力型 SIRMs の説明のための図：3 本の直線はメンバーシップ関数の中心を示す．2 つの入力 $(0,1)$ と $(1,0)$ は Line 2 に近い (○ と ● は理想の出力値がそれぞれ 0, 1 であることを示す).

$$h_1 = 0.93,$$

$$SIRMs-1\begin{cases} R_1^1 \ : \ if \ z \ is \ A_1^1 \ then \ y \ is \ -0.06, \\ R_2^1 \ : \ if \ z \ is \ A_2^1 \ then \ y \ is \ 1.37, \\ R_3^1 \ : \ if \ z \ is \ A_3^1 \ then \ y \ is \ -0.06, \end{cases}$$

$$A_1^1 = \exp\left(-\frac{(z+0.01)^2}{0.12}\right), \tag{7.13}$$

$$A_2^1 = \exp\left(-\frac{(z-0.50)^2}{0.11}\right), \tag{7.14}$$

$$A_3^1 = \exp\left(-\frac{(z-1.00)^2}{0.12}\right). \tag{7.15}$$

式.(7.12) は入力 x_1 と x_2 の線形変換を表す．式 (7.13), (7.14), (7.15) は z に関するメンバーシップ関数である．図 7.2 中の斜線はこれらのメンバーシップ関数の中心を表す．ファジィ集合 A_2^1 のメンバーシップ関数の値は, 入力が $(0,0)$ と $(1,1)$ の場合はほぼ 0, 入力が $(0,1)$ と $(1,0)$ の場合はほぼ 1 である．同様に, ファジィ集合 A_1^1 と A_3^1 のメンバーシップ関数の値も計算される．結果として, 線形入力型 SIRMs モデルの出力値は, 入力 $(0,1)$ と $(1,0)$ に対してはほぼ 1, 入力 $(0,0)$ と $(1,1)$ に対してはほぼ 0(ファジィ集合 A_1^1, A_3^1 を含む推論規則においては出力 y の値が小さいため) となる．

一方で, 通常の SIRMs モデルにおいては, メンバーシップ関数の中心は x_1 軸または x_2 軸に平行な直線しか実現できないため, 入力 $(0,1)$ と $(0,1)$ に対して異なる出力とするような推論規則を表現することができない．□

7.3 線形入力型 SIRMs モデルの万能性

簡略型ファジィ推論法は Stone-Weierstrass 定理により万能性をもつことは示されている [35]. 一方, 線形入力型 SIRMs は Stone-Weierstrass 定理を使って直接証明することが難しい. それゆえ, ニューラルネットワークの万能性を示した関数解析的手法を用いて結果を示す. この関数解析の手法によるモデルの能力解析は, ファジィ推論モデルの研究においては筆者の知る限り行われていない. ここでは, Hornik らの手法 [67] にならって, 提案手法の万能性 (Universal approximation capability) についての理論的な証明を行う.

7.3.1 三角型メンバーシップ関数の場合

メンバーシップ関数が三角型関数の場合について, 関数解析的手法を用いて, 万能性を示す.

[補題 7.1][9, 67]

$\boldsymbol{\Omega}$ をコンパクト集合 $\boldsymbol{S}\subset\boldsymbol{R}^m$ 上の, cosig 関数から成る以下のような関数の集合とする:

$$\begin{aligned}\boldsymbol{\Omega}_Q &= \{g(\boldsymbol{x}) = \sum_{l=1}^{Q} u_l cosig\left(\sum_{k=0}^{m} d_{lk}x_k + \theta_l\right) \\ &\qquad |u_l, d_{lk}, \theta_l \in \boldsymbol{R}, \boldsymbol{x}\in\boldsymbol{S}\} \qquad (7.16)\\ \boldsymbol{\Omega} &= \bigcup_{Q=1}^{\infty} \boldsymbol{\Omega}_Q \qquad (7.17)\end{aligned}$$

ここで, $x_0 = 1$ かつ

$$cosig(x) = \begin{cases} 1 & (x > 0) \\ cos(2\pi x) & (-\frac{1}{2}\leq x\leq 0) \\ -1 & (x < -\frac{1}{2}) \end{cases} \qquad (7.18)$$

とする.

このとき, $\boldsymbol{\Omega}$ は $C[\boldsymbol{S}]$ において稠密である. □

[定理 7.1]

$\boldsymbol{\Phi}$ をコンパクト集合 $\boldsymbol{S}\subset\boldsymbol{R}^m$ 上の, 以下のような関数の集合とする.

$$\begin{aligned}\boldsymbol{\Phi}_H &= \{f(\boldsymbol{x}) = \sum_{j=1}^{H} h_j \frac{\sum_{i=1}^{n} A_i^j \left(\sum_{k=0}^{m} a_{jk}x_k\right) w_i^j}{\sum_{i=1}^{n} A_i^j \left(\sum_{k=0}^{m} a_k^j x_k\right)} \\ &\qquad |h_j, a_{jk}, b_i^j, c_i^j, w_i^j \in \boldsymbol{R}, \boldsymbol{x}\in\boldsymbol{S}\} \qquad (7.19)\\ \boldsymbol{\Phi} &= \bigcup_{H=1}^{\infty} \boldsymbol{\Phi}_H \qquad (7.20)\end{aligned}$$

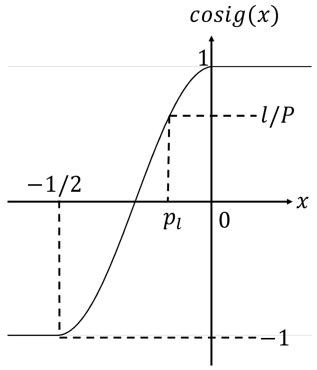


図 7.3: cosig 関数の図

ここで, A_i^j は式 (2.3) に示す三角型関数である.
　このとき, $\mathbf{\Phi}$ は $C[\boldsymbol{S}]$ において稠密である. □

[証明]
　補題 7.1 より, cosig 関数 $g\in\Omega$ と任意の実数 $\varepsilon > 0$ に対し, 以下のような関数 $f\in\Phi$ が存在することを示す.

$$|f(\boldsymbol{x}) - g(\boldsymbol{x})| < \varepsilon \quad \text{for } \boldsymbol{x}\in\boldsymbol{S} \tag{7.21}$$

　以下では, $m = 1$ の場合についてのみ証明する.
　図 7.3 にこの場合の cosig 関数 $g(\boldsymbol{x})$ の概形を記す. $[-L, L]$ を, $[-\frac{1}{2}, 0]$ を含む区間とする. このとき, 任意の実数 $\varepsilon > 0$ に対し, P を $\frac{2}{P} < \varepsilon$ と以下の性質を満たす正の自然数とする;

$$p_l = \sup_{x\in[-\frac{1}{2},0]} \{x | cosig(x) = \frac{l}{P}\} \tag{7.22}$$

ここで, $l = -P, \cdots, P$ とする.
　関数 $g(x)$ の近似を行うため, メンバーシップ関数が三角型関数とし, $\sum_{i=1}^{n} A_i^k(v_k) = 1(k\in\{-P, \cdots, P\})$ とする. このとき,

$$f(x) = \sum_{k=-P}^{P} h_k \sum_{i=1}^{n} A_i^k(v_k) w_i^k \tag{7.23}$$

である.
　ここで, 図 7.4 に示すような $2P + 1$ 個のモジュールから成る SIRMs モデルについて考える.
　$Q = 2P + 1$ をモジュールの数, $n = 3$ を各モジュールのファジィ集合の数とする. 三角型関数として, メンバーシップ関数の中心と幅を以下のようにおく:

$$c_1^k = p_{k-1},\ c_2^k \ = \ p_k,\ c_3^k = p_k + (p_k - p_{k-1})$$

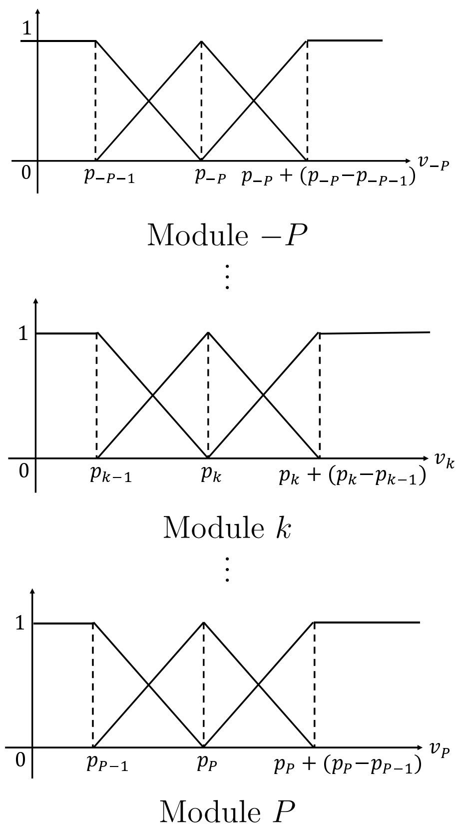


図 7.4: 各モジュール中のメンバーシップ関数

$$
\begin{aligned}
b_i^k &= 2(p_k - p_{k-1}) \\
h_k &= 1
\end{aligned}
$$

ここで, $k{\in}\{-P,\cdots,P\}$ かつ $p_{-P-1} = p_{-P+1} - p_{-P}$ とする (図 7.4).

さらに, モジュール中の実数 w_i^k を以下のようにおく:

$$
w_i^k = \begin{cases} \frac{1}{P} & \text{for } i = 2,3 \\ 0 & \text{for } i = 1 \end{cases} \quad k{\neq}-P \tag{7.24}
$$

$$
w_i^{-P} = -1 \quad \text{for } i = 1,2,3,\ k = -P \tag{7.25}
$$

$v_k = x\,(k{\in}\{-P,\cdots,P\})$ とする. $m = 1$ のときの $\mathbf{\Phi}$ の性質より, 以下の性質を得る.

$$
f(x) = \sum_{k=-P}^{P} \sum_{i=1}^{3} A_i^k(x) w_i^k \tag{7.26}
$$

($\sum_{i=1}^{3} M_i^k(x) = 1\ (k{\in}\{-P,\cdots,P\})$ かつ $h_j = 1$ より).

$f(x)$ と $g(x)$ の関係について考える.

(i)$x = p_k(k{\in}\{-P,\cdots,P\})$

$x = p_{-P}$ のとき

$$
f(x) = -1 = cosig(p_{-P}) \tag{7.27}
$$

$x = p_k(k{\neq}-P)$ のとき

$$
f(x) = -1 + \frac{P+k}{P} = \frac{k}{P} = cosig(p_l) \tag{7.28}
$$

式 (7.27) と (7.28) より,

$$
|f(x) - g(x)| = 0 < \varepsilon \tag{7.29}
$$

(ii)$p_{k-1} < x < p_k(k{\in}\{-P+1,\cdots,P\})$

$\frac{k-1}{P} < f(x) < \frac{k}{P}$ かつ $\frac{k-1}{P} < cosig(x) < \frac{k}{P}$. これより, 以下を得る.

$$
\begin{aligned}
|f(x) - g(x)| &= |\left(\frac{k-1}{P} + \frac{e_1}{P}\right) - \left(\frac{k-1}{P} + \frac{e_2}{P}\right)| \\
&< \frac{e_1 + e_2}{P} < \frac{2}{P} < \varepsilon
\end{aligned} \tag{7.30}
$$

(iii)$x < p_{-P}$, $x > p_P$

$$
x < p_{-P} \ :\ f(x) = cosig(x) = -1 \tag{7.31}
$$

$$
x > p_P \ :\ f(x) = cosig(x) = 1 \tag{7.32}
$$

式 (7.31) と (7.32) より,

$$|f(x)-g(x)|=0<\varepsilon \tag{7.33}$$

よって, $m=1$ のとき, 以下を得る:

$$|f(x)-g(x)|<\varepsilon \quad \text{for any } \boldsymbol{x}\in S \tag{7.34}$$

この結果を使って, 任意の m について証明することができる [78]. □

7.3.2　ガウス型メンバーシップ関数の場合

メンバーシップ関数としてガウス型を使う場合は, 以下のように得られる.
[補題 7.2][9]
$\boldsymbol{\Omega}$ をコンパクト集合 $\boldsymbol{S}\subset\boldsymbol{R}^m$ 上の, 以下のような関数の集合とする:

$$\boldsymbol{\Omega}_Q = \{g(\boldsymbol{x})=\sum_{l=1}^{Q} u_l \frac{1}{1+\exp\left(-\sum_{k=1}^{m} d_{lk}x_k-\theta_l\right)} | u_l, d_{lk}, \theta_l\in\boldsymbol{R}, \boldsymbol{x}\in\boldsymbol{S}\} \tag{7.35}$$

$$\boldsymbol{\Omega} = \bigcup_{Q=1}^{\infty} \boldsymbol{\Omega}_Q \tag{7.36}$$

ここで, $x_0=1$ とする.
このとき, $\boldsymbol{\Omega}$ は $C[\boldsymbol{S}]$ において稠密である. □
[定理 7.2]
$\boldsymbol{\Phi}$ をコンパクト集合 $\boldsymbol{S}\subset\boldsymbol{R}^m$ 上の, 以下のような関数とする.

$$\boldsymbol{\Phi}_H = \{f(\boldsymbol{x})=\sum_{j=1}^{H} h_j \frac{\sum_{i=1}^{n} A_i^j\left(\sum_{k=0}^{m} a_{jk}x_k\right) w_i^j}{\sum_{i=1}^{n} A_i^j\left(\sum_{k=0}^{m} a_k^j x_k\right)} | h_j, a_{jk}, b_i^j, c_i^j, w_i^j\in\boldsymbol{R}, \boldsymbol{x}\in\boldsymbol{S}\} \tag{7.37}$$

$$\boldsymbol{\Phi} = \bigcup_{H=1}^{\infty} \boldsymbol{\Phi}_H \tag{7.38}$$

ここで, A_i^j は式 (2.2) のようなガウス型関数である.
このとき, $\boldsymbol{\Phi}$ は $C[\boldsymbol{S}]$ 内で密である □

[証明]
補題 7.2 より, cosig 関数 $g\in\Omega$ と任意の実数 $\varepsilon>0$ に対し, 以下のような関数 $f\in\Phi$ が存在することを示す.

$$|f(\boldsymbol{x})-g(\boldsymbol{x})|<\varepsilon \quad \text{for } \boldsymbol{x}\in\boldsymbol{S} \tag{7.39}$$

$n=2,\ c_1^j=\frac{1}{4},\ c_2^j=-\frac{1}{4},\ b_1^j=b_2^j=1,\ w_1^j=1,\ w_2^j=0$ とおくと,

$$\begin{aligned}
\frac{\sum_{i=1}^{n} A_i^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)w_i^j}{\sum_{i=1}^{n} A_i^j\left(\sum_{k=0}^{m} a_k^j x_k\right)} &= \frac{A_1^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)}{A_1^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)+A_2^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)} \\
&= \frac{1}{1+\left(A_1^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)\right)^{-1}A_2^j\left(\sum_{k=0}^{m} a_{jk}x_k\right)} \\
&= \frac{1}{1+\exp\{-\sum_{k=0}^{m} a_{jk}x_k\}} \qquad (7.40)
\end{aligned}$$

ここで, $H=Q,\ a_{jk}=d_{jk}(k\in Z_m),\ a_{j0}=\theta_j,\ h_j=u_j$ とおくと,

$$\begin{aligned}
f(\boldsymbol{x}) &= \sum_{j=1}^{Q} u_j \frac{1}{1+\exp\left(-\sum_{k=1}^{m} d_{jk}x_k-\theta_j\right)} \\
&= g(\boldsymbol{x}) \qquad (7.41)
\end{aligned}$$

よって, 以下を得る:

$$\sup_{\boldsymbol{x}\in S}|f(\boldsymbol{x})-g(\boldsymbol{x})|=0<\varepsilon \qquad (7.42)$$

補題 7.2 および以上の結果より, 定理 7,2 を得る. □

以上の結果より, 線形入力型 SIRMs はメンバーシップ関数が三角型関数とガウス型関数いずれの場合においても万能性を満たすことが示される.

7.4 数値シミュレーション

簡略型, SIRMs, DIRMs, 線形入力型 SIRMs モデルを用いた数値シミュレーションを行い, 線形入力型 SIRMs モデルの近似能力を示す.

7.4.1 排他的論理和問題

ここでは, 6.2.3 で用いた, 入力数が m の排他的論理和 (exclusive or : EX-OR) 問題を扱う. EX-OR 問題は以下のように定義される:

$$y=x_1\oplus x_2\oplus\cdots\oplus x_m, \qquad (7.43)$$

ここで, $x_1,\cdots,x_m,y\in\{0,1\}$ であり, また, $\oplus$ は演算 Exclusive OR の演算子を表す.

表 7.1: m 入力の EX-OR 問題における初期条件

	簡略型	SIRMs	DIRMs	線形入力型 SIRMs
K_c	0.001	0.001	0.001	0.001
K_b	0.001	0.001	0.001	0.001
K_w	0.05	0.05	0.05	0.05
K_h		0.05	0.05	0.05
K_a				0.01
H	3	3	3	3
T_{max}	50000	100	2000	50000
c_{ij} の初期値	Equal intervals			
b_{ij} の初期値	$\frac{1}{2(H-1)}\times$(入力値の範囲の大きさ)			
w_{ij} の初期値	[0,1] 内でランダムに選択			
h_j の初期値		[0,1] 内でランダムに選択		
a_{ik} の初期値				[-1,1] 内でランダムに選択

表 7.2: m 入力の EX-OR 問題の実験結果

	$m=2$	$m=3$	$m=4$	$m=10$
簡略型	0.00	0.00	0.00	
SIRMs	0.25			
DIRMs	0.00	0.25		
線形入力型 SIRMs ($l=m$)	0.00	0.00	0.00	0.00

表 7.3: 関数近似における初期条件

	簡略型	SIRMs	DIRMs	線形入力型 SIRMs
K_c	0.001	0.001	0.001	0.001
K_b	0.001	0.001	0.001	0.001
K_w	0.05	0.05	0.05	0.05
K_h		0.05	0.05	0.05
K_a				0.01
H	3	3	3	3
T_{max}	50000	50000	50000	50000
c_{ij} の初期値	等間隔			
b_{ij} の初期値	$\frac{1}{2(H-1)}\times$(入力値の範囲の大きさ)			
w_{ij} の初期値	[-1,1] よりランダムに選択			
h_j の初期値		[-1,1] よりランダムに選択		
a_{ik} の初期値				[-1,1] よりランダムに選択

表 7.1 に実験の初期条件を表す．表 7.2 に各モデルを用いた際の MSE(平均二乗誤差) を表す．なお, 表中の MSE の値は 10 回試行の平均である．表より, $m = 3$ のときの EX-OR は DIRMs モデルでは近似精度が低く, また, $m = 2$ のときの EX-OR は SIRMs モデルでは近似精度が低い．さらに, $m = 10$ の場合など, 入力 m の数が大きい EX-OR 問題は, 通常の簡略型ではパラメータ数が膨大となり学習による推論規則の決定が困難となる．一方, 線形入力型 SIRMs モデルではパラメータ数が少ないため, 適用が容易である．

7.4.2 関数近似問題

簡略型, SIRMs, DIRMs, 線形入力型 SIRMs モデルの学習による, 関数近似能力についての数値シミュレーションを行う．

入力が 4 変数で入力空間が式 (7.44), (7.45) に対しては $[0,1]^4$, 式 (7.46), (7.47) に対しては $[-1,1]^4$ とする．簡略型, SIRMs, 提案手法を用いて関数近似を行う．

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{37.21} \times \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6)}{12} \tag{7.44}$$

$$y = \frac{(\sin(2\pi x_1) \times \cos(x_2) \times \sin(\pi x_3) \times x_4 + 1.0)}{2.0} \tag{7.45}$$

表 7.4: 関数近似の結果

		式 (7.44)	式 (7.45)	式 (7.46)	式 (7.47)
簡略型	MSE(学習用データ)	0.01	0.07	0.08	0.07
	MSE(テスト用データ)	0.03	0.14	0.14	0.14
	推論規則数	81	81	81	81
SIRMs	MSE(学習用データ)	1.60	10.69	1.40	3.34
	MSE(テスト用データ)	1.80	10.71	1.62	3.75
	推論規則数	12	12	12	12
DIRMs	MSE(学習用データ)	0.02	2.29	0.01	0.01
	MSE(テスト用データ)	0.06	7.80	0.01	0.02
	推論規則数	54	54	54	54
線形入力型	MSE(学習用データ)	0.05	1.23	0.01	0.05
SIRMs	MSE(テスト用データ)	0.06	1.69	0.02	0.06
($M = 7$)	推論規則数	21	21	21	21

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6)}{446.52} \tag{7.46}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(3e^{3x_3} + 2e^{-4x_4})^{-0.5} - 0.077}{4.68} \tag{7.47}$$

入力データは入力空間内でランダムに生成し, 学習用データ 512 個, テスト用データ 6400 個を用いた. 実験の初期条件を表 7.3 に示す.

結果は表 7.4 のように得られた. ここで, 各欄の上段は学習用データの平均二乗誤差 ($\times 10^{-3}$), 中段はテストデータの平均二乗誤差 ($\times 10^{-3}$), 下段がルール数である.

線形入力型 SIRMs モデルの推論規則数は簡略型および DIRMs モデルに比べて少ないが, 学習用データおよびテスト用データの平均二乗誤差はともに SIRMs モデルや DIRMs モデルよりも良好で, 簡略型に近い結果となった.

7.4.3 モデル選択によるシステム評価

与えられた問題のデータを確率モデルから生成されたと考えて, 複数のモデル (ここではファジィシステム) の中からある尺度に従ってモデルを選択する. AIC や BIC は, 情報量基準と呼ばれるモデル選択を行う基準として知られている [61].

AIC(Akaike's Information Criterion) や BIC(Bayesian Information Criterion) は, モデルの複雑さとデータへの適合の高さのバランスを考慮するために用いられる. 評価値が低いモデルほどよいモデルとみなすことができる.

L を最大尤度, n をデータ数, m をパラメータ数とすると, AIC, BIC の評価値はそれぞれ

表 7.5: 初期条件

<table>
<tr><th></th><th>簡略型</th><th>SIRMs</th><th>DIRMs</th><th>線形入力型 SIRMs</th></tr>
<tr><td>K_c</td><td>0.001</td><td>0.001</td><td>0.001</td><td>0.001</td></tr>
<tr><td>K_b</td><td>0.001</td><td>0.001</td><td>0.001</td><td>0.001</td></tr>
<tr><td>K_w</td><td>0.05</td><td>0.05</td><td>0.05</td><td>0.05</td></tr>
<tr><td>K_h</td><td></td><td>0.05</td><td>0.05</td><td>0.05</td></tr>
<tr><td>K_a</td><td></td><td></td><td></td><td>0.01</td></tr>
<tr><td>H</td><td>3</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>T_{max}</td><td>50000</td><td>50000</td><td>50000</td><td>50000</td></tr>
<tr><td>c_{ij} の初期値</td><td colspan="4">等間隔</td></tr>
<tr><td>b_{ij} の初期値</td><td colspan="4">$\frac{1}{2(H-1)}\times$(入力値の範囲の大きさ)</td></tr>
<tr><td>w_{ij} の初期値</td><td colspan="4">[-1,1] よりランダムに選択</td></tr>
<tr><td>h_j の初期値</td><td></td><td colspan="3">[-1,1] よりランダムに選択</td></tr>
<tr><td>a_{ik} の初期値</td><td colspan="3"></td><td>[-1,1] よりランダムに選択</td></tr>
</table>

$$AIC = -2\mathrm{ln}L + 2m \tag{7.48}$$
$$BIC = -2\mathrm{ln}L + m\log(n) \tag{7.49}$$

で与えられる．確率モデルが同一の正規分布からのデータ生成とすると，

$$AIC = n\mathrm{ln}\left(\sigma^2\right) + 2m \tag{7.50}$$
$$BIC = n\mathrm{ln}\left(\sigma^2\right) + m\log(n) \tag{7.51}$$

ここで, σ^2 は分散, すなわち残差平方和である．

ここでは, 前述の 4 次元の関数 (7.44), (7.45), (7.46), (7.47) に関する関数近似問題に対し, AIC, BIC を用いたモデルの能力の比較を行う．学習用データは, ε を平均 0, 分散 0.0001 の正規分布 $N(0, 0.0001)$ に従ってランダムに生成した実数値を付加したものを学習データとして用いる．実験の初期条件を表 7.5 に示す．モデル選択には, 簡略型, SIRMs, DIRMs, 提案手法を用いる．各手法における AIC, BIC の値が低くなるルール数 r を求め, そのモデルに対するテスト用データの平均二乗誤差 ($\times 10^{-4}$), AIC, BIC の値の比較を行う．

図 7.5 に, 関数 (7.44) に対するルール数と AIC, テスト用データに対する平均二乗誤差 (Mean Square Error：MSE) の関係の一例を示す．この図から, AIC の低さと MSE の精度がほぼ一致していることが分かる．これらの結果から, ファジィ推

表 7.6: 関数近似に対するファジィ推論法の各手法の結果

		式 (7.44)	式 (7.45)	式 (7.46)	式 (7.47)
簡略型	AIC	-5730	-5771	-5447	-5501
	BIC	-2176	-1906	-1851	-2135
	パラメータ数	729	729	729	729
	MSE(テスト用データ)	0.44	0.52	0.28	0.34
SIRMs	AIC	-3494	-1668	-3721	-2790
	BIC	-1471	-1294	-3524	-2475
	パラメータ数	52	76	40	64
	MSE(テスト用データ)	16.62	98.08	13.34	32.77
DIRMs	AIC	-5919	-2463	-6059	-6072
	BIC	-4558	-1102	-4698	-4711
	パラメータ数	276	276	276	276
	MSE(テスト用データ)	0.44	30.29	0.12	0.13
線形入力型 SIRMs	AIC	-6168	-5578	-6147	-6147
	BIC	-5650	-4468	-5112	-5185
	パラメータ数	105	225	210	195
	MSE(テスト用データ)	0.27	0.86	0.12	0.16

論システムのルール数 (パラメータ数) の決定に AIC や BIC を用いることが有効といえる. 表 7.6 にこの方法により得られた結果を示す. 表中の 4 つの項目は, 上から AIC, BIC, パラメータ数とテストデータに対する平均二乗誤差 ($\times 10^{-4}$) である. 提案手法は簡略型, SIRMs や DIRMs モデルに比べて AIC, BIC の値が低く, 良好なモデルとみなすことができる.

ここで, 従来のファジィ推論システムの場合, パラメータ数 (ルール数) を増やして近似精度を上げる方法としてはルール中の属性の数を増やす方法がある. 線形入力型 SIRMs モデルの場合, パラメータ数を増やす方法としては, 変換後の入力要素数を増やす方法とルール中の属性の数を増やす方法が考えられる. 線形入力型 SIRMs モデルに対する両者の結果を表 7.7 に示す. 結果として, 変換後の入力要素数を増やす方が AIC, BIC, テスト用データの平均二乗誤差が小さくなり, 有効と考えられる. それゆえ, 上述のシミュレーションでは, この方法を用いている.

7.4.4 パターン認識問題

UCI データベース より, 表 7.8 に示すデータを用いた数値シミュレーションを行う [60]. ここでは, 5 分割交差検証 (5-fold cross varidation) を用いる: 5 分割交差検

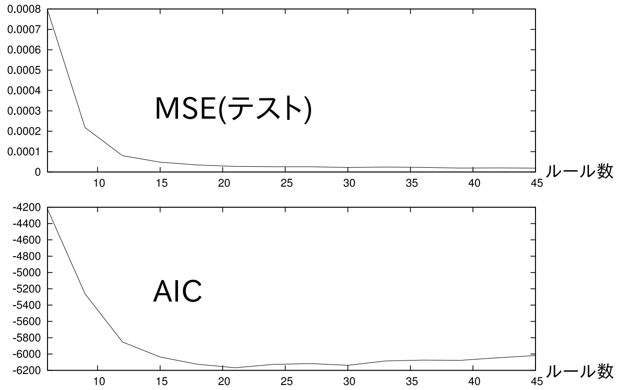


図 7.5: 線形入力型 SIRMs の関数 (7.44) に対するルール数とテスト用データの平均二乗誤差 (上) および AIC(下) の関係

表 7.7: 関数近似に対する提案手法の各モデルの結果

		式 (7.44)	式 (7.45)	式 (7.46)	式 (7.47)
$(r=3)$	M	7	15	14	13
	AIC	-6168	-5578	-6147	-6147
	BIC	-5650	-4468	-5112	-5185
	MSE(テスト用データ)	0.27	0.86	0.12	0.16
$(M=4)$	r	6	5	4	4
	AIC	-5893	-3186	-6122	-5061
	BIC	-5419	-2772	-5767	-4706
	MSE(テスト用データ)	0.64	21.12	0.37	2.69

表 7.8: パターン認識で用いるデータ

	Iris	Wine	Sonar	BCW
データ数	150	178	208	683
入力要素数	4	13	60	9
分類数	3	3	2	2

表 7.9: パターン認識における初期条件

<table>
<tr><th></th><th>簡略型</th><th>SIRMs</th><th>DIRMs</th><th>線形入力型 SIRMs</th></tr>
<tr><td>K_c</td><td>0.001</td><td>0.001</td><td>0.001</td><td>0.001</td></tr>
<tr><td>K_b</td><td>0.001</td><td>0.001</td><td>0.001</td><td>0.001</td></tr>
<tr><td>K_w</td><td>0.05</td><td>0.05</td><td>0.05</td><td>0.05</td></tr>
<tr><td>K_h</td><td></td><td>0.05</td><td>0.05</td><td>0.05</td></tr>
<tr><td>K_a</td><td></td><td></td><td></td><td>0.01</td></tr>
<tr><td>H</td><td>3</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>T_{max}</td><td>50000</td><td>50000</td><td>50000</td><td>50000</td></tr>
<tr><td>c_{ij} の初期値</td><td colspan="4">等間隔</td></tr>
<tr><td>b_{ij} の初期値</td><td colspan="4">$\frac{1}{2(H-1)}\times$(入力値の範囲の大きさ)</td></tr>
<tr><td>w_{ij} の初期値</td><td colspan="4">[0,1] よりランダムに選択</td></tr>
<tr><td>h_j の初期値</td><td></td><td colspan="3">[0,1] よりランダムに選択</td></tr>
<tr><td>a_{ik} の初期値</td><td colspan="3"></td><td>[-1,1] よりランダムに選択</td></tr>
</table>

証においては, 全データが 5 等分され, 5 個のデータの部分集合に分けられる. 5 個のデータ集合について, そのうちの 1 個をテスト用データとし, 残りの 4 個の集合のデータを用いてモデルの学習を行う. 同様の操作を, すべてのデータがテスト用データとして用いられるまで (5 回) 行う. この操作 5 回の平均をとり, モデルの評価を行う.

表 7.9 は実験の初期条件を示し, 表 7.10 は各手法での分類結果を示す. 表 7.10 中の各値について, 上から順に学習用データの誤分類率, テスト用データの誤分類率, モデルのパラメータ数を表す. なお, 実験結果は 10 回試行の平均値である.

表 7.10 中の斜線は, パラメータ数が非常に多く, 学習によるパラメータの値の決定が困難であることを表す. 表 7.10 より, 線形入力型 SIRMs モデルは SIRMs や DIRMs モデルよりも近似精度が高く, また, 簡略型よりも少ないパラメータ数で同

表 7.10: パターン認識の結果

	Iris	Wine	Sonar	BCW
簡略型	0.004 0.055 (729)			
SIRMs	0.021 0.052 (40)	0.022 0.102 (130)	0.024 0.301 (600)	0.055 0.063 (90)
DIRMs	0.001 0.057 (276)	0.011 0.092 (3588)		0.001 0.065 (1656)
線形入力型 SIRMs	0.029 0.031 (45)	0.001 0.037 (72)	0.001 0.205 (213)	0.016 0.036 (60)

等の近似精度が得られた.

7.4.5 障害物回避問題

ここでは, 制御問題への応用として, 障害物回避問題を扱う. 障害物回避問題に線形入力型 SIRMs モデルを適用し, その有効性を示す.

障害物回避問題では, 図 7.6 に示すような, 移動物体 (mobile object) と障害物 (obstacle) の距離 r_1 となす角 θ_1 および移動物体とゴール (designated place) の距離 r_2 となす角 θ_2 をもとに, 障害物を回避しゴール地点に到達するような移動物体の動かし方を求める. 移動物体は速度 $\boldsymbol{A} = (A_x, A_y)$ で動いている. ここで, 速度 $\boldsymbol{A}$ の要素 A_x は一定であり, $\boldsymbol{A}$ の要素 A_y を変化させる. ここでは, r_1, θ_1, r_2, θ_2 を入力とするファジィ推論システムにより, 障害物を回避しゴール地点に到達するような A_y を求める. あらかじめ図 7.7 に示すような障害物の軌跡が与えられており, 軌跡上の各点における r_1, θ_1, r_2, θ_2 および A_y を学習用データとして用いる. 学習用データ点の数は 200 個である. これらの学習用データを用いてファジィ推論システムを構築する. 表 7.11 に実験条件を示す. 各モデルの入力空間の分割数は 3 とする. SIRMs モデルのパラメータ数は 40, 線形入力型 SIRMs モデルのパラメータ数は 30 である.

学習終了後, 構築されたファジィ推論システムを用いて以下のようなテストを行う:

(1) Test 1 では, 学習用データと同じ位置に障害物とゴールが設置されている状態で, 異なる地点からスタートしたいくつかの移動物体の軌跡を見る. (図 7.8). 図

表 7.11: 障害物回避問題における実験条件

	SIRMs	線形入力型 SIRMs
K_c	0.001	0.001
K_b	0.001	0.001
K_w	0.05	0.05
K_h	0.05	0.05
K_a		0.05
H	3	3
T_{max}	50000	50000
c_{ij} の初期値	等間隔	
b_{ij} の初期値	$\frac{1}{2(H-1)}$×(入力値の範囲の大きさ)	
w_{ij} の初期値	[0,1] 内でランダムに選択	
h_j の初期値	[0,1] 内でランダムに選択	
a_{ik} の初期値		[-1,1] 内でランダムに選択

7.8 の横軸と縦軸は, 図 7.7 の x-軸と y-軸に対応する．図 7.8 は, y-軸上のいくつかの地点からスタートし, $(1.0, 0.5)$ に設置されたゴールに向かう移動物体の軌跡を示す．学習終了後, 図 7.8 に示す地点 $(0.0, 0.1), (0.0, 0.2), \cdots, (0.0, 0.8), (0.0, 0.9)$ から移動物体をスタートさせる．図 7.8 に示すように, SIRMs モデルを用いた場合にはスタート地点によっては移動物体はゴールに到達できないが, 線形入力型 SIRMs モデルを用いた場合にはどの地点からスタートした場合でも移動物体は障害物を回避しゴール地点に到達することもできる．

(2) Test 2 では, 障害物の位置が学習用データと異なり, ゴールの位置も学習用データと異なる．障害物の位置は (0.4, 0.4) であり, ゴールの位置は (1.0, 0.6) である (図 7.9)．図 7.9 に示すように, 線形入力型 SIRMs モデルを用いることで, 移動物体は障害物を回避しゴールに到達することができる．

(3) Test 3 では, 障害物が一定の速度で移動している場合について扱う．ここでは, 障害物が速度 (0.01, 0.02) で (0.3, 0.0) から (0.8, 1.0) に移動する (図 7.10)．図 7.10 に示すように, 線形入力型 SIRMs モデルを用いることで, 移動物体は障害物を回避しゴールに到達することができる．

(4) Test 4 では, 図 7.11 に示すように障害物がランダムに移動する場合について扱う．ここで, 障害物の移動速度の大きさ $|\boldsymbol{B}|$ は一定であり, 角度 θ_b がランダムに変化する．ここで, 障害物のスタート地点は $(0.5, 0.0)$ とする．図 7.12 に示すように, 線形入力型 SIRMs モデルを用いることで, 移動物体は障害物を回避しゴールに到達することができる．

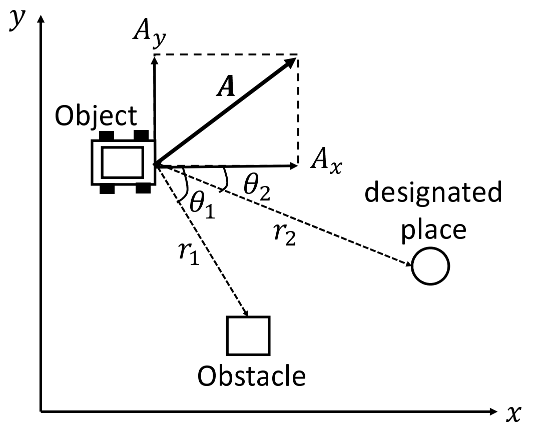


図 7.6: 障害物回避問題における入力要素.

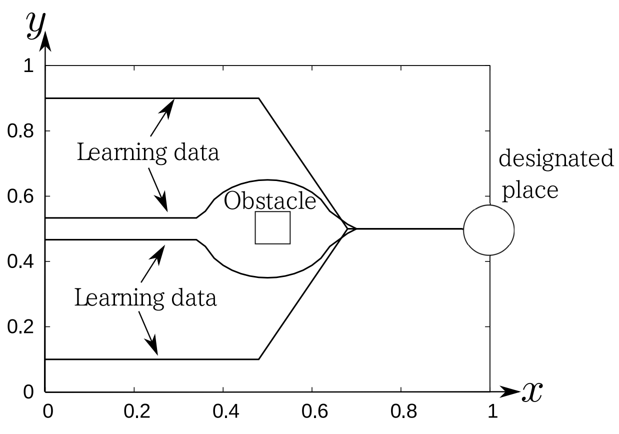
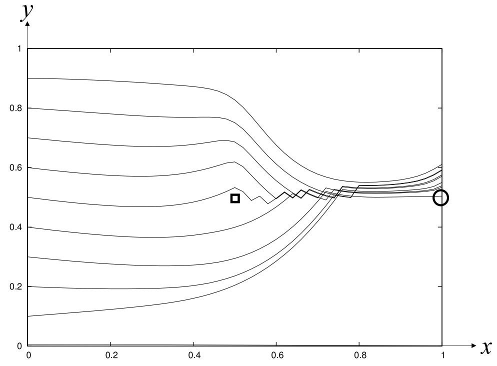
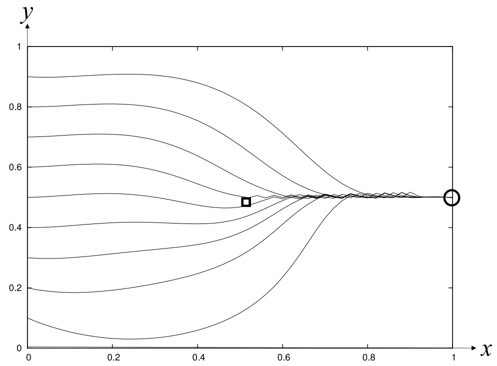


図 7.7: 障害物回避問題における学習用データ:(0.5, 0.5) に設置された障害物を回避し, (1.0, 0.5) に設置されたゴールを目指す.



(a) SIRMs model



(b) Proposed model

図 7.8: Test 1 の結果:(0.5, 0.5) に設置された障害物を回避し, (1.0, 0.5) に設置されたゴールを目指す.

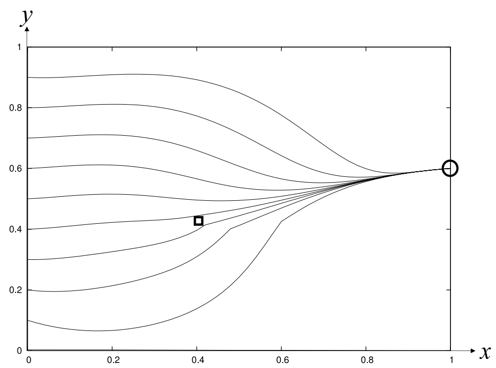


図 7.9: Test 2 の結果:(0.4, 0.4) に設置された障害物を回避し, (1.0, 0.6) に設置されたゴールを目指す.

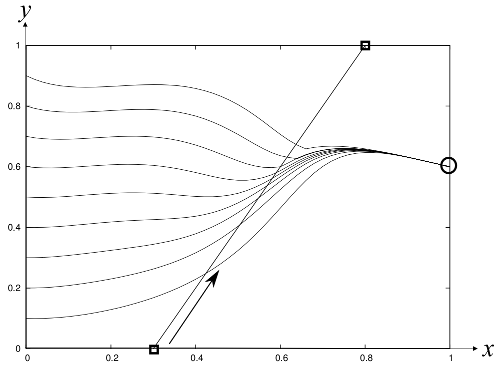


図 7.10: Test 3 の結果:一定速度で移動する障害物を回避し, (1.0, 0.6) に設置されたゴールを目指す.

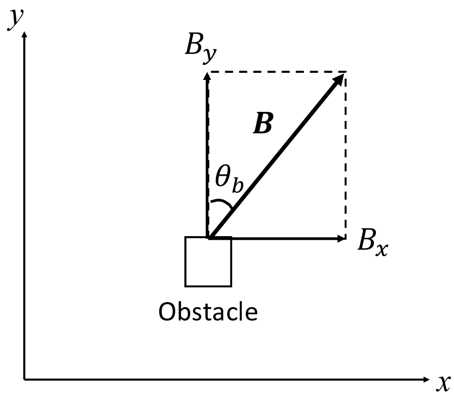


図 7.11: 速度 $\boldsymbol{B}$ でランダムに動く障害物 (速度の大きさ $|\boldsymbol{B}|$ は一定, 角度 θ_b はランダムに変化する.)

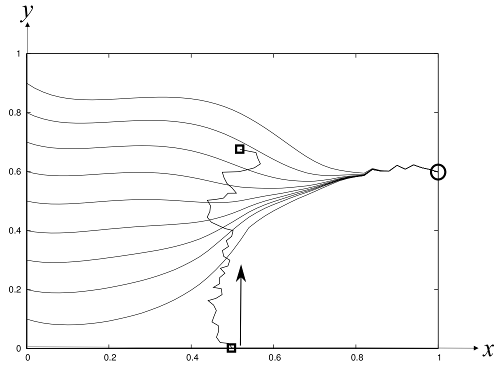


図 7.12: Test 4 の結果:ランダムに移動する障害物を回避し, (1.0, 0.6) に設置されたゴールを目指す.

表 7.12: A fuzzy inference system for obstacle avoidance

$z_1 = 0.432 + 0.751x_1 + 0.432x_2 + 0.818x_3 - 1.033x_4$

$z_2 = 0.190 + 0.357x_1 + 0.439x_2 + 0.706x_3 + 0.051x_4$

$h_1 = 0.168$

$A_{11} = \exp\left(-\frac{(z_1+1.001)^2}{2\cdot 0.248^2}\right) \rightarrow -0.660$

$A_{12} = \exp\left(-\frac{(z_1+0.060)^2}{2\cdot 0.201^2}\right) \rightarrow -0.528$

$A_{13} = \exp\left(-\frac{(z_1-0.980)^2}{2\cdot 0.257^2}\right) \rightarrow 0.640$

$h_2 = 0.179$

$A_{21} = \exp\left(-\frac{(z_2+1.008)^2}{2\cdot 0.240^2}\right) \rightarrow 0.208$

$A_{22} = \exp\left(-\frac{(z_2+0.064)^2}{2\cdot 0.232^2}\right) \rightarrow 0.901$

$A_{23} = \exp\left(-\frac{(z_2-0.992)^2}{2\cdot 0.192^2}\right) \rightarrow -0.627$

以上のように, 障害物回避問題において線形入力型 SIRMs モデルは少ないパラメータ数でありながら, 各テストにおいて障害物の回避とゴールへの到達を実現している.

ここで, 障害物回避問題において用いた線形入力型 SIRMs モデルの推論規則の意味解釈について検討する. ここでは, 表 7.12 のような障害物回避問題において, 学習により得られた推論規則を扱う.

システムの入力 r_1, r_2, θ_1, θ_2 については" 小さい" と" 大きい" という 2 種類の属性, システムの出力 A_y には移動物体の移動方向に関して" 左に移動する"($A_y > 0$) と" 右に移動する"($A_y < 0$) という属性を与えるとする. このとき, 後に示す理論を使って, 以下のような推論規則を得ることができる :

ファジィ集合 Z_1 : (r_1 が" 小さい") or (θ_1 が" 大きい") or (r_2 が" 大きい") or (θ_2 が" 小さい")

A_{11} : もしファジィ集合 Z_1 のメンバーシップ関数の値が小さいならば, 移動物体を右に移動する

A_{12} : もしファジィ集合 Z_1 のメンバーシップ関数の値が中くらいならば, 移動物体を右に移動する

A_{13} : もしファジィ集合 Z_1 のメンバーシップ関数の値が大きいならば, 移動物体を左に移動する

ファジィ集合 Z_2 : (r_1 が" 大きい") or (θ_1 が" 小さい") or (r_2 が" 小さい") or (θ_2 が" 大きい")

A_{21} : もしファジィ集合 Z_2 のメンバーシップ関数の値が小さいならば, 移動物体を左に移動する

A_{22} : もしファジィ集合 Z_2 のメンバーシップ関数の値が中くらいならば, 移動物体を左に移動する

A_{23}：もしファジィ集合 Z_2 のメンバーシップ関数の値が大きいならば, 移動物体を右に移動する

ここで, 以下に示す (I), (II) の場合における移動物体の振る舞いについて考えてみる.

(I) 移動物体が障害物に近づいているとき:

この場合, "r_1 が小さい" と "r_2 が大きい" が有効である.

(i)θ_1 が大きいならば, 移動物体は A_{13} に従い左に移動する.

(ii)θ_1 が大きくないならば, 移動物体は A_{11} または A_{12} に従い右に移動する.

Z_2 についても同様の言語的解釈を行うことができる.

(II) 移動物体がゴールに近づいているとき:

この場合, "r_1 が大きい" と "r_2 が小さい" が有効である.

(i)θ_2 が大きいならば, 移動物体は A_{23} に従い右に移動する.

(ii) If θ_2 が大きくないならば, 移動物体は A_{21} または A_{22} に従い左に移動する.

Z_1 についても同様の言語的解釈を行うことができる.

これらの言語的解釈に従うと, 移動物体は障害物に近づくと障害物から離れる動きをし, ゴールに近づくとゴールに向かって動く結果が得られる.

以下では, なぜこのような意味解釈が可能になるかについて, その導出方法と上記の問題に適用した結果を示す.

式 (7.52) の線形変換について考える:

$$z_k = d_{k0} + \sum_{j=1}^{m} d_{kj} x_j \tag{7.52}$$

ここで, $0 \leq x_j \leq 1$ かつ $k \in Z_l$ である.

$$d_{kj}^{+} = \begin{cases} d_{kj} \ (d_{kj} \geq 0) \\ 0 \ (d_{kj} < 0) \end{cases} \tag{7.53}$$

$$d_{kj}^{-} = \begin{cases} 0 \ (d_{kj} \geq 0) \\ -d_{kj} \ (d_{kj} < 0) \end{cases} \tag{7.54}$$

とおくと ($j \in Z_m$ かつ $k \in Z_l$), 以下に示す式 7.55 が得られる:

$$\begin{aligned} &\frac{1}{\sum_{j=1}^{m} d_{kj}^{+} + \sum_{j=1}^{m} d_{kj}^{-}} \{z_k + \sum_{j=1}^{m} d_{kj}^{-} - d_{k0}\} \\ &= \frac{1}{\sum_{j=1}^{m} d_{kj}^{+} + \sum_{j=1}^{m} d_{kj}^{-}} \{\sum_{j=1}^{m} d_{kj}^{+} x_j + \sum_{j=1}^{m} d_{kj}^{-} (1 - x_j)\} \end{aligned} \tag{7.55}$$

両辺の最小値は 0, 最大値は 1 である.

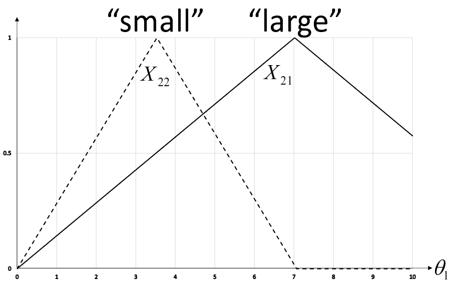


図 7.13: $x_2(\theta_1)$ に関する変数 X_{21} と X_{22}

$$\tilde{Z}_k(z_k) = \frac{1}{\sum_{j=1}^{m} d_{kj}^{+} + \sum_{j=1}^{m} d_{kj}^{-}}\{z_k + \sum_{j=1}^{m} d_{kj}^{-} - d_{k0}\} \tag{7.56}$$

$$X_{jk}(x_j) = \begin{cases} \frac{d_{kj}^{+}}{\sum_{j=1}^{m} d_{kj}^{+}+\sum_{j=1}^{m} d_{kj}^{-}} x_j \ (d_{kj} \geq 0) \\ \frac{d_{kj}^{-}}{\sum_{j=1}^{m} d_{kj}^{+}+\sum_{j=1}^{m} d_{kj}^{-}} (1 - x_j) \ (d_{kj} < 0) \end{cases} \tag{7.57}$$

($j \in Z_m$ かつ $k \in Z_l$) とすると,

$$\tilde{Z}_k(z_k) = \sum_{j=1}^{m} X_{jk}(x_j) \tag{7.58}$$

$$= \min\left(\sum_{j=1}^{m} X_{jk}(x_j), 1\right) \tag{7.59}$$

$\tilde{Z}_k$, X_{jk} をそれぞれ $\tilde{Z}_k(z_k)$, $X_{jk}(x_j)$ をメンバーシップ関数とするファジィ集合とすると, 以下のような解釈を得ることができる.

$$Z_k = X_{1k} \text{ or } X_{2k} \text{ or } \cdots \text{ or} X_{mk} \tag{7.60}$$

ここで, この結果を障害物回避問題に適用する.
式 7.59 より, 以下に示す式が得られる.

$$\tilde{Z}_1 = 0.248X_{11} + 0.142X_{21} + 0.269X_{31} + 0.340X_{41} \tag{7.61}$$

$$\tilde{Z}_2 = 0.230X_{12} + 0.283X_{22} + 0.454X_{32} + 0.033X_{42} \tag{7.62}$$

同様にして, r_1, θ_1, r_2 and θ_2 についても, 同様の式を得る.
結果として, Z_1 と Z_2 について以下のような解釈を与えることができる:
$\tilde{Z}_1$= (r_1 が小さい) or (θ_1 が大きい) or (r_2 が大きい) or (θ_2 が小さい)
$\tilde{Z}_2$= (r_1 が大きい) or (θ_1 が小さい) or (r_2 が小さい) or (θ_2 が大きい)

さらに, $\tilde{Z}_1$ および $\tilde{Z}_2$ の属性の数が 3 のとき, 以下のような推論規則を得る:
A_{11} : $\tilde{Z}_1$ が小さいならば右へ移動する
A_{12} : $\tilde{Z}_1$ が中くらいならば右へ移動する
A_{13} : $\tilde{Z}_1$ が大きいならば左へ移動する
A_{21} : $\tilde{Z}_2$ が小さいならば左へ移動する
A_{22} : $\tilde{Z}_2$ が中くらいならば左へ移動する
A_{23} : $\tilde{Z}_2$ が大きいならば右へ移動する

7.5 まとめ

本章では, 入力要素の線形変換と SIRMs モデルを組み合わせた線形入力型 SIRMs モデルの提案を行った. 従来の推論規則数の少ない手法は近似精度が低いが, 線形入力型 SIRMs モデルは任意の連続関数を任意精度で近似することができ, 高い近似能力をもつことを理論的に示した. そして, 数値シミュレーションにおいて, 線形入力型 SIRMs モデルと簡略型, SIRMs モデル, DIRMs モデルとの比較を行った. その結果, 線形入力型 SIRMs モデルは他の手法に比べて少ないパラメータ数 (推論規則数) で高い精度を実現することができた. AIC や BIC によるモデル選択においても, 線形入力型 SIRMs モデルは他のモデルに比べて少ないパラメータ数で高い精度を実現する効率的なモデルという結果であった. さらに, 障害物回避問題における推論規則の自然言語的な解釈を行い, 線形入力型 SIRMs モデルが自然言語による解釈が容易な説明能力の高いモデルであることを示した.

第8章　結論

8.1　まとめ

本論文では, はじめに, 従来研究において高い近似精度や説明能力をもつモデルやその汎化モデルについて, 理論的な能力 (万能性), 学習システムとしての近似精度や説明能力, また入力変数の増加に伴う近似能力の柔軟性について検討した. これらの結果を踏まえて, 新しいモデルを提案し, 万能性, 学習モデルとしての近似能力や説明能力, また入力変数の増加に伴う近似能力の高さについて, その有効性を示した. 本論文の 1, 2 章は準備であり, 3 から 6 章ではファジィ推論システムの従来モデルとその汎化モデルを提案し, その有効性を示した. さらに, 7 章では, これらの結果を踏まえた新しいモデルとその学習法を提案し, 理論と数値シミュレーションにより有効性を示した. その詳細は, 以下の通りである.

第 1 章では, ファジィ理論の歴史と工学的背景, およびファジィ推論システムにおける最近の研究と本論文の内容と構成について述べている.

第 2 章は, ファジィ集合とその応用であるファジィ推論とその学習法について述べている. はじめに, ファジィ推論システムの従来モデルとして知られる TS 型, Mamdani 型と簡略型ファジィ推論システムを導入し, 学習による推論ルールの決定方法について述べている. さらに, 各モデルについてこれまでに得られている結果を与えている.

第 3 章は, 高い説明能力をもつモデルとして知られているファジィ推論システムとその汎化モデルについて説明している. 従来型の簡略型ファジィ推論システムは学習後の推論ルールの言語的解釈が困難であるため, 得られたシステムの問題に対する解の説明能力が低いことが知られている. 本章では, 高い説明能力と近似精度を実現する属性型ファジィ推論モデルの提案を行い, このモデルの近似精度の高さや万能性に関する理論的な解析を行っている.

第 4 章は, ベクトル量子化を用いたファジィ推論システムの近似能力について述べている. 従来のファジィ推論システムの学習においては, 入力要素数が多い問題に対して計算量が膨大となるという問題があることが知られている. ソフトコンピューティングの分野では, ベクトル量子化を用いた学習モデルは, 入力要素数が多い問題に対してもパラメータ数を低く抑えることができることが知られている. ファジィ推論の分野においても, ベクトル量子化を用いた従来モデルの学習法が提案されているが, 学習後のモデルの近似精度が低いという問題点があった. 本章では, ニューラルガス (k-means) 等のベクトル量子化を用いた新しいモデルとその学

習法を提案し, 近似精度の高さや推論ルール数において有効となることを示している.

第 5 章は, メタヒューリスティクスを用いたハイブリッドなファジィ推論モデルとその学習法について述べている. ファジィ推論の学習に用いられる最急降下法の学習時間は短いが, 局所解に陥りやすく近似精度の高いモデルを得ることは難しい. 一方, 階層型ニューラルネットワークの学習においてはメタヒューリスティックスを組み合わせたモデルの学習を行うことで, 局所解に陥ることを避け近似精度の高いモデルが得られることが知られている. 本章では, ファジィ推論システムに EM やランダムサーチのようなメタヒューリスティクスを用いたハイブリッドモデルと学習法を提案し, その有効性を示している.

第 6 章は, 高い説明能力と近似精度をもつモデルとして提案された SNIRMs モデルの能力について考察している. 高い説明能力をもつモデルとして提案された SIRMs モデルは単一入力型の推論ルールを用いるファジィ推論モデルの一つであり, 入力要素数が多い問題に対しても計算量が少ないという特徴を持つ. しかしながら, SIRMs モデルは複雑な問題に対しては解の精度は十分でないことが知られており, SIRMs モデルをより一般化した少数入力ルール群型ファジィ推論モデルである SNIRMs モデルが提案されている. 本章では, SNIRMs モデルの説明能力や近似精度について, 理論と数値シミュレーションにより明らかにしている.

第 7 章は, 前章までの結果を踏まえて, 線形入力型 SIRMs モデルを提案し, その能力について詳述している. SNIRMs モデルは, 高い説明能力と変数増加に対する計算量を抑えることができるが, モデルの近似精度は十分とはいえない. 第 7 章では, 第 1 段階で入力変数の線形変換を行い, 第 2 段階で SIRMs モデルによる出力導出を行う線形入力型 SIRMs ファジィ推論システムとその学習法を提案し, 理論と数値シミュレーションにより, 従来モデルやその汎化モデルと比べて高い能力をもつことを示している. また, 学習後に得られたファジィ推論ルールの意味解釈を与える方法を提案し, 障害物回避問題に適用し, その有効性を示している.

8.2 今後の課題

本研究では, ファジィ推論システムの推論規則の学習による自動構築について, 推論システムの精度の高さ, 入力要素数の多い問題に対する推論規則数の増加の抑制, 学習後の推論規則の言語的な説明能力という観点から, いくつかの有効なモデルの提案を行った. 特に, 線形入力型 SIRMs モデルはこれらの性質をすべて満たす有効なモデルである. 入力要素の線形変換は OR 演算として解釈が可能である. しかしながら, ファジィ理論においては, AND, OR 演算にはいくつか種類が存在するため, 言語的な解釈が可能な入力要素の変換は線形変換以外にも存在する. そのため, 入力要素の変換に線形変換以外の非線形な変換を用いるモデルについての検討が必要である. また, 入力要素の変換後に SIRMs モデルを用いているが, SIRMs モデルは SNIRMs モデルの中で最も精度の低いモデルである. そのため, DIRMs

モデルなど, SIRMs モデルよりも精度の高い SNIRMs を用いるモデルについての検討が必要である. さらに, 最急降下法に代わる, 学習が速くて近似精度の高い技法の導入が望まれる.

謝辞

本研究を行うにあたり, 多忙な中, ご指導, ご助言を頂きました, 鹿児島大学理工学研究科の重井徳貴准教授に深く感謝いたします. また, 本論文をまとめるにあたり, 数々のご助言を頂きました, 鹿児島大学大学院理工学研究科の八野知博教授, 鹿児島大学理工学研究科の福島誠治教授, 鹿児島大学大学院理工学研究科の湯ノ口万友教授に感謝いたします.

また, 常日頃からご指導, ご助言を頂きました鹿児島大学理工学研究科の宮島廣美教授に深く感謝いたします.

最後に, 研究生活を送るうえでお世話になりました研究室の皆様, 教職員の皆様に感謝の意を記します.

参考文献

[1] L. A. Zadeh, Outline if a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Transactions on Systems, Man and Cybernetics, 3, pp.28-44, 1973.

[2] L. A. Zadeh, Some Reflections on Soft Computing, Granular Computing and Their Roles in the Conception, Design and Utilization of Information/Intelligent Systems, Soft Computing, 2, pp.23-25, 1998.

[3] システム制御情報学会, システム/制御/情報：ソフトコンピューティング特集号, 43, pp.163-208, 1999.

[4] L.A. Zadeh, Fuzzy logic, neural networks, and soft computing, Comm. ACM, Volume 37 Issue 3, March, pp.77-84, 1994.

[5] ファジィとソフトコンピューティングハンドブック, 日本ファジィ学会編, 共立出版, 2000.

[6] 馬場他, ソフトコンピューティングの基礎と応用, 共立出版, 2012.

[7] OR 事典, 日本オペレーションリサーチ学会編, 日科技連出版, 1975.

[8] N. Walia, H. Singh, A. Sharma, ANFIS, Adaptive Neuro-Fuzzy Inference System-A Survey, Journal of Computer Applications (0975-8887) Volume 123, No.13, pp.32-38, 2015.

[9] M.M. Gupta, L. Jin and N. Homma, Static and Dynamic Neural Networks, IEEE Press, 2003.

[10] B. Kosko, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ, 1992.

[11] C. Lin and C. Lee, Neural Fuzzy Systems, Prentice Hall, PTR, 1996.

[12] L.A.Zadeh, Fuzzy Sets, Information and Control, Vol.8, pp. 338-353, 1965.

[13] E.H.Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, Proceedings of IEEE, Vol.121, pp.1585-1588, 1974.

[14] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-15, No.1, pp.116-132, 1985.

[15] 市橋秀友, 渡辺俊彦, 簡略ファジィ推論を用いたファジィモデルによる学習型制御, 日本ファジィ学会誌, Vol. 2, No. 3, pp. 429-437, 1990.

[16] 堀川慎一, 古橋武, 内川嘉樹, ファジィニューラルネットワークの構成法と学習法, 日本ファジィ学会誌, Vol. 4, No. 5, pp. 906-928, 1992.

[17] A. Kaur, Amrit Kaur, Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System, Journal of Soft Computing and Engineering, Vol.2, No.2, pp.323-325, 2012.

[18] J. Casillas, O. Cordon and F. Herrera, L. Magdalena, Accuracy Improvements in Linguistic Fuzzy Modeling, Studies in Fuzziness and Soft Computing, Vol. 129, Springer, 2003.

[19] S. M. Zhoua, J. Q. Ganb, Low-level Interpretability and High-Level Interpretability: A Unified View of Data-driven Interpretable Fuzzy System Modeling, Fuzzy Sets and Systems 159, pp.3091-3131, 2008.

[20] 井上博行, 亀井且有, 井上和夫, 遺伝的アルゴリズムと超円錐形メンバーシップ関数によるファジィルール自動生成手法の提案, 日本ファジィ学会誌, Vol.8, No.6, pp.1104-1115, 1996.

[21] H. Miyajima, S. Fukumoto, Y. Ishizuka, Fuzzy Modeling Based on Deleting Rules, Instutute of Electronics, Information, and Communication Engineers, J77-A(11), pp.1555-1562, 1994.

[22] S. Fukumoto, H. Miyajima, K. Kishida and Y. Nagasawa, A Destructive Learning Method of Fuzzy Inference Rules, Proc. of IEEE on Fuzzy Systems, pp.687-694, 1995.

[23] S. Fukumoto and H. Miyajima, Learning Algorithms with Regularization Criteria for Fuzzy Reasoning Model, Journal of Innovative Computing Information and Control, 1,1, pp.249-263, 2006.

[24] S. Fukumoto, H. Miyajima, N. Shigei, K. Uchikoba, Decision Procedure of the Initial Values of Fuzzy Inference System Using Counterpropagation Networks, Journal of Signal Processing, Vol.9, No.4, pp.335-342, 2005.

[25] W. Pedrycz, H. Izakian, Cluster-Centric Fuzzy Modeling, IEEE Trans. on Fuzzy Systems, Vol. 22, Issue 6, pp. 1585-1597, 2014.

[26] O. Cordon, A Historical Review of Evolutionary Learning Methods for Mamdani-type Fuzzy Rule-based Systems, Designing interpretable genetic fuzzy systems, Journal of Approximate Reasoning, 52, pp.894-913, 2011.

[27] M. Fazzolari, R Alcala, Y Nojima, H Ishibuchi, F Herrera, A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions, Fuzzy Systems, IEEE Transactions on Fuzzy Systems, Vol.21, No.1, pp.45-65, 2012.

[28] Y. Shi, M. Mizumoto, N. Yubazaki and M. Otani, A Self-Tuning Method of Fuzzy Rules Based on Gradient Descent Method, Japan Journal of Fuzzy Set and System, Vol.8, No.4, pp.757-747, 1996.

[29] S. Araki, H. Nomura, I. Hayashi and N. Wakami, A Fuzzy Modeling with Iterative Generation Mechanism of Fuzzy Inference Rules, Journal of Japan Society for Fuzzy Theory &Systems, vol.4, no.4, pp.722-732, 1992.

[30] N. Yubazaki, J. Yi and K. Hirota, SIRMS(Single Input Rule Modules) Connected Fuzzy Inference Model, J. Advanced Computational Intelligence, Vol.1, No.1, pp.23-30, 1997.

[31] H. Seki, H. Ishii and M. Mizumoto, On the Nonlinear Identification by Functional Type SIRMs Connected Type Fuzzy Reasoning Method, Proc. of Int. Conf. on Industrial Eng. Theory, Applications and Practice, pp.1441-1446, 2006.

[32] N. Shigei, H. Miyajima and S. Nagamine, A Proposal of Fuzzy Inference Model Composed of Small-Number-of-Input Rule Modules, Proc. of Int. Symp. on Neural Networks: Advances in Neural Networks - Part II, pp.118-126, 2009.

[33] H. Nomura, I. Hayashi and N. Wakami, A Learning Method of Simplified Fuzzy Reasoning by Genetic Algorithm, Proc. of the Int. Fuzzy Systems and Intelligent Control Conference, pp.236-245, 1992.

[34] L. X. Wang and J. M. Mendel, Fuzzy Basis Functions, Universal Approximation and Orthogonal Least Square Learning, IEEE Trans. Neural Networks, Vol.3, No.3, pp.807-814, 1992.

[35] D. Tikk, L. T. Koczy and T. D. Gedeon, A Survey on Universal Approximation and its Limits in Soft Computing Techniques, Journal of Approximate Reasoning, Vol.33, pp.185-202, 2003.

[36] K. Kishida, H. Miyajima, A Learning Method of Fuzzy Inference Rules using Vector Quantization, Proc. of the Int. Conf. on Articial Neural Networks, Vol.2, pp.827-832, 1998.

[37] 相吉英太郎, 安田恵一郎, メタヒューリスティクスと応用, 電気学会, 2007.

[38] I. Boussaid, J. Lepagnot, P. Siarry, A Survey on Optimization Metaheuristics, Information & Sciences, 237, pp.82-117, 2013.

[39] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[40] D. I. Choi and S. H. Park, Self-creating and organizing neural networks, IEEE Trans. Neural Networks, Vol.5, No.4, pp.561-575, 1994.

[41] T. Martinetz, S. Berkovich and K. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Networks, Vol.4, No.4, pp.558-569, 1993.

[42] H. Yuan, J. Ahi, J. Liu, Application of Particle Swarm Optimization Algorithms based Fuzzy BP Neural Network for Target Damage Accesment, Scientific Research and Essays, Vol.6, No.15, pp.3109-3121, 2011.

[43] C.H.Lee, C.T.Li, F.Y.Chang, A Species-based improved Electromagnetism-like Mechanism Algorithm for TSK-type integral-valued Neural Fuzzy System Optimization, Fuzzy Set and Systems, Vol.171, pp.22-43, 2011.

[44] C.H. Lee, F.Y. Chang, C.T. Lee, A hybrid of electromagnetism-like mechanism and back-propagation algorithms for recurrent neural fuzzy systems design, Journal of Systems Science, 43, 2, pp.231-247, 2012.

[45] C.H. Lee, F.K. Chang, Y.C. Lee, Nonlinear systems design by a novel fuzzy neural system via hybridization of electromagnetism-like mechanism and particle swarm optimization algorithms, Information Sciences, 186, 1, pp.59-72, 2012.

[46] 茨木俊秀, 最適化の数学, 共立出版, 2011.

[47] S. I. Birbil, S. C. Fang, An Electromagnetism-like Mechanism for Global Optimization, Journal of Global Optimization, 25, pp.263-282, 2003.

[48] J. Jiang, H. Shang, K. Liu, Q. Su, L. Zhang, A Clustering Method Using Electromagnetism-like Mechanism Algorithm, Journal of Computational Information Systems, Vol.9, No.10, pp.3985-3191, 2013.

[49] J. L. Lin, C.H. Wa, H.Y. Chung, Performance Comparison of Electromagnetism-like Algorithms for Global Optimization, Applied Mathematics, 3, pp.1265-1275, 2012.

[50] H.H Chang, T.Y. Huang, Mixture Experiment Design Using Artificial Neural Networks and Electromagnetism-like Mechanism Algorithm, in Proc. Second International Conference on Innovative Computing, Information and Control, pp.397-397, 2007.

[51] J. Matyas, Random Optimization, Automation & Remote Contr., 26, pp.246-253, 1965.

[52] N. Baba : A new Approach for Finding the Global Minimum of Error Function of Neural Networks, Neural Networks, Vol.2, pp. 367-373, 1989.

[53] N. Baba, A Hybrid Algorithm for Finding the Global Minimum of Error Function of Neural Networks and its applications, Neural Networks, Vol.7, No.8, pp.1253-1265, 1994.

[54] S. Guillaume, Designing Fuzzy Inference Systems from Data, An Interpretability-Oriented Review, IEEE Trans. Fuzzy Systems, Vol. 9, No. 3, pp.426-443, 2001.

[55] A. P. Jacquin, A. Y. Shamseldin, Review of the application of fuzzy inference systems in river flow forecasting, Journal of Hydroinformatics, Vol. 11, No. 3-4, pp.202-210, 2009.

[56] S. Chaudhari et al., American International, S. Chaudhari1, M. Patil, Study and Review of Fuzzy Inference Systems for Decision Making and Control, Journal of Research in Science, Technology, Engineering & Mathematics, Vol.5, No.1, pp. 88-92, 2014.

[57] J. M. Alonso, L. Magdalena and G. Gonza, Looking for a good fuzzy system interpretability index: An experimental approach, Journal of Approximate Reasoning, Vol.51, pp.115-134, 2009.

[58] M. J. Gacto, R. Alcala and F. Herrera, Interpretability of Linguistic Fuzzy Rule-based Systems:An Overview of Interpretability Measures, Inf. Sciences, Vol.181, pp.4340-4360, 2011.

[59] S. Miike, H. Miyajima, N. Shigei and K. Noo, Fuzzy Reasoning Model with Deletion of Rules Consisting of Small-Number-of-Input Rule Modules, Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, Vol.22, No.5, pp.621-629, 2010 (in Japanese).

[60] UCI Repository of Machine Learning Databases and Domain Theories, ftp://ftp.ics.uci.edu/pub/machinelearning-Databases

[61] 鈴木義一朗, 情報量規準による統計解析入門, 講談社サイエンティフィク, 1995.

[62] J. Yi, N. Yubazaki and K. Hirota, A Proposal of SIRMs Dynamically Connected Fuzzy Inference Model for Plural Input Fuzzy Control, Fuzzy Sets and Systems, 125, pp.79-92, 2002.

[63] H. Nomura, I. Hayashi and N. Wakami, A Self-Tuning Method of Fuzzy Reasoning by Delta Rule and Its Application to a Moving Obstacle Avoidance, Journal of Japan Society for Fuzzy Theory & Systems, vol.4, no.2, pp. 379-388, 1992.

[64] L.A.Zadeh, Probability Theory and Fuzzy Logic, http://kmh-lanl.hansonhub.com/uncertainty/meetings/zadeh03vgr.pdf, April 24, 2003.

[65] 石川真澄, 忘却を用いたコネクショニストモデルの構造学習アルゴリズム, 人工知能学会誌, Vol.5, No.5, pp.595-603, 1990.

[66] M. Umano, S. Fukunaka, I. Hatano, H. Tamura, Extraction of Fuzzy Rules Using Fuzzy Neural Networks with Forgetting, Transactions of the Society of Instrument and Control Engineers, Vol.32, No.3, pp.409-416, 1996.

[67] K. Hornik, M. Stinchcombe and H. White, Multilayer Feedforward Networks and Universal Approximators, Neural Networks, Vol.2, pp.359-366, 1989.

発表文献

論文

[68] H. Miyajima, N. Shigei and H. Miyajima, Fuzzy Inference Systems Composed of Double-Input Rule Modules for Obstacle Avoidance Problems, IAENG Int. Journal of Computer Science, Vol. 41, No. 4, pp.222-230, 2014.

[69] H. Miyajima, N. Shigei and H. Miyajima, Approximation Capabilities of Interpretable Fuzzy Inference Systems, IAENG International Journal of Computer Science, Vol.42, No.2, pp.117-124, 2015.

[70] H. Miyajima, N. Shigei and H. Miyajima, Some Properties on Fuzzy Inference Systems Composed of Small Number of Input Rule Modules, Advances in Fuzzy Sets and Systems, Vol.20, pp.155-175, 2015.

[71] H. Miyajima, N. Shigei and H. Miyajima, Performance Comparison of Hybrid Electromagnetism-like Mechanism Algorithms with Descent Method, Journal of Artificial Intelligence and Soft Computing Research, Vol.5, No.4, pp.271-282, 2015.

[72] H. Miyajima, N. Shigei, K. Kishida, Y. Akiyoshi, H. Miyajima, An Improved Learning Algorithm of Fuzzy Inference Systems using Vector Quantization, Advanced in Fuzzy Sets and Systems (in print).

国際学会

[73] H. Miyajima, N. Shigei and H. Miyajima, A Proposal of Hybrid Electromagnetism-like Mechanism Method, Proc. 2014 RISP Int. Workshop on Nonlinear Circuits, Communications and Signal Processing pp. 281-284, Hawaii, 2014.

[74] H. Miyajima, F. Kawai, N. Shigei and H. Miyajima, An Application of Fuzzy Inference System Composed of Double-Input Rule Modules to Control Problems, Proc. Int. MultiConference of Engineers and Computer Scientists 2014, Vol.I, pp. 23-28, Hong Kong, 2014.

[75] H. Miyajima, N. Shigei, H. Taketatsu and H. Miyajima, Numerical Simulations for Hybrid Electromagnetism-like Mechanism Optimization Algorithms with Descent Methods, Proc. The 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Vol.2, pp. 293-306, Singapore, 2014.

[76] H. Miyajima, N. Shigei and H. Miyajima, A Proposal of SIRMs Model with Linear Transformation of Input Variables, Proc. Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems, pp. 311-316, Kiyakyusyu, 2014.

[77] H. Miyajima, N. Shigei and H. Miyajima, On the Capability of a Fuzzy Inference System with Improved Interpretability, Proc. Int. MultiConference of Engineers and Computer Scientists 2015, Vol.I, pp. 51-56, Hong Kong, 2015.

[78] H. Miyajima, N. Shigei and H. Miyajima, SIRMs Fuzzy Inference Model with Linear Transformation of Input Variables and Universal Approximation, Advances in Computational Intelligence: Proc. 13th Int. Work-Conference on Artificial Neural Networks, Part I, pp.561-575, Spain, 2015.

口頭発表

[79] 宮島 洋文, 蓮池 隆, 森田 浩, ガウス型メンバシップ関数における重み付き TS ファジィ逆推論, 2011 IEEE SMC Hiroshima Chapter 若手研究会, 広島, 7 月, 2011 年.

[80] 宮島 洋文, 蓮池 隆, 森田 浩, ガウス型メンバシップ関数における重み付き TS ファジィ逆推論, 日本オペレーションズ・リサーチ学会 平成 23 年秋季研究発表会, 神戸, 9 月, 2011 年.

[81] 宮島 洋文, 重井 重貴, 宮島 廣美, ハイブリッド EM 法に関する考察, 第 15 回 SOFT 九州支部学術講演会, 下関, 12 月, 2013 年.

[82] 宮島 洋文, 重井 重貴, 宮島 廣美, 線形入力型 SIRMs ファジィ推論法の提案, 第 67 回電気・情報関係学会九州支部連合大会, 鹿児島, 9 月, 2014 年.

[83] 竹田津宏樹, 宮島廣美, 重井徳貴, 宮島洋文, ハイブリッド EM 法の高速化, 第 67 回電気・情報関係学会九州支部連合大会, 鹿児島, 9 月, 2014 年

[84] 宮島 洋文, 重井 重貴, 宮島 廣美, ベクトル量子化法を用いたファジィ推論システムの学習法, 第 16 回 SOFT 九州支部学術講演会, 北九州, 3 月, 2015 年.

[85] 宮島 洋文, 重井 重貴, 宮島 廣美, 線形入力型 SIRMs ファジィ推論法の提案, 第 25 回ソフトサイエンスワークショップ, 下関, 3 月, 2015 年.

[86] 宮島 洋文, 重井 重貴, 宮島 廣美, メタヒューリスティクスを用いた SNIRMs ファジィ推論法の学習に関する考察, 第 68 回電気・情報関係学会九州支部連合大会, 福岡, 9 月, 2015 年.

[87] 秋吉勇佑, 宮島廣美, 重井徳貴, 宮島洋文, ベクトル量子化法を用いたファジィ推論システムの学習法, 第 68 回電気・情報関係学会九州支部連合大会, 福岡, 9 月, 2015 年.