論文

個人条件メンバーシップ関数の推定を伴った ファジィ検索法の提案[†]

前田 隆宏^{*1} 大橋 一文^{*2} 小川 英明^{*2} 関根 正直^{*2}

検索法の研究分野において、ファジィデータ検索法の様々な手法が提案されている。本論文で我々は、ファジィ検索法の特徴を心理学的観点から考察し、問題点を指摘し、この解決のために新しいファジィ検索法を提案する。更に、提案した検索法に基づいた3つのシステムの動作例を挙げる(「料理メニュー検索システム」,「種牡馬検索システム」,「スキー場検索システム」)。

キーワード: 個人条件メンバーシップ関数, 再検索, 意思決定支援システム, ファジィ評価, ファジィ検索法

1. はじめに

我々は本論文において、ファジィデータベースシステムにおけるユーザー個人個人の検索条件のメンバーシップ関数の推定法、及びこれを利用した新しいファジィ検索法を提案する。

あらゆる人々にとって情報の収集と効率的な活用が重要であり、かつ必須である事は分野を問わず議論される事である。例えばトフラーの有名な著書[1]は世界中で広く読まれ、情報を効率よく利用する事は全ての人々にとって重要で、大きな力を持つ事に値する事を痛感させた。情報を効率よく活用するためには、大量の情報の中から自らの必要とする情報を拾い上げる事が前提条件となる。この様な働きを手助けしてくれるのがデータ検索システムである。更に、このシステムにファジィ理論を応用したものがファジィデータ検索システムと呼ばれるもので、このシステムの出現により、検索条件を明確に示さなくとも、あたかも人間に質問する様な、「言語」による検索が可能となった。ファジィ検索に使われる言語は、「高い」、「大きい」、「〜ぐらい」などである。この様な言語を使用できるように、ファジィデータ検索システムではこれらをメンバーシップ関数を用いて表現している。

ところで、従来のファジィ検索システムには、重大な問題点が存在する。それは、検索条件に使用される

あいまいな言語は、個々人によってその概念が異なっているという事にある。メンバーシップ関数は個々人の言語概念を反映する様に設定すべきものであろうから、これも個々人によって異なっている事は明かである。

従来のファジィデータ検索システムにおいて、内部に設定されている検索条件のメンバーシップ関数はその設定者自身の言語概念を反映しているに過ぎない。

しかし、人間の概念は個人個人異なっているので、ファジィデータ検索システムによってデータの検索を行う際、個々人の検索条件のメンバーシップ関数は異なるものを想定すべきである。つまり、個人個人の検索条件メンバーシップ関数の調整を行う必要が出てくる。ただし、この様な調整に際して、ユーザーの負担が増大する事があってはならない。この様な問題に対し、検索条件のメンバーシップ関数を個人個人の持つ概念に合わせて調整する試みもなされてきている。例えば、大木ら[2]はネットワーク上の検索システムにおいて、検索条件メンバーシップ関数を調整する方法を提案している。しかし、この様なネットワークを持たない、あるいは経済的に構築できない施設においても、データベースが必要となる状況はもちろんあるであろう。例えば、小規模な図書館や学校などにおいては、一台あるいは数台の検索システムを複数の人々が共有して使用する状況が考えられるが、この様な状況に対応する研究はあまりなされていないのが実状だろう。ネットワークの有無に関わらずデータベースシステムが必要となる状況はあるわけで、この様な全ての状況に対して適用可能な検索法を考察すべきである。また、ファジィデータ検索システムを利用して表示された検索結果から自らが最も良いと思われる物件を探し出す際、人間は、画面に表示されたデータを比較検討してよりよい物件を見つけだそうとする。この時、『あま

[†] A Proposal of Fuzzy-Data-Searching Method With Estimation of Personal Membership Functions
Takahiro MAEDA, Hihumi OOHASHI, Hideaki OGAWA and Masanao SEKINE

^{*1} 帝京平成大学大学院

Teikyo-Heisei Univ.

^{*2} 帝京平成大学情報学部情報システム学科

Teikyo-Heisei Univ.

『あ良い』物件を見つける事は比較的容易であるが、この『まあまあ良い』物件に近いけれど、これとはもう少し異なる『より良い』物件を探し出したい場合は、従来のファジィデータ検索システムでは十分に対応する事が出来なかった。

我々はこの様な状況に対応するために、個人個人の検索条件メンバーシップ関数の推定の際、データベースに登録されているデータを利用して、検索条件を調整し、再検索を行う方法と、個々人の検索条件のメンバーシップ関数をユーザーのもつ概念として徐々に推定していく方法を提案する。これによってユーザーは検索時に自分がどの物件を選択したか計算機に示すだけで、再検索時によりよい検索結果が得られるばかりか、次回からの検索のためのユーザー自身の検索条件のメンバーシップ関数を設定してゆく事が出来る。本論文では2章でファジィ検索システムについて議論し、3章で検索システムを意思決定支援システムとして見た場合について議論し、4章においてこれを受けて新しいファジィ検索法を提案する。また、5章では、提案した検索法に基づいてシステムを構築し、その動作例を挙げ、検討する。

2. ファジィ検索システムの出力について

ファジィ検索システムは、一致度によって検索が行われる。検索条件が複数の場合、これらの各条件とデータの一致度をまとめあげる手法が必要となる。その方法として、Min 法[2], Max 法, 重み付けられた平均法, ファジィ積分法[3], ファジィ推論法[4]などが用いられる。本論文では、単純なしかし、効果的な方法である重み付けられた平均法を用いる。

ファジィ検索システムの基本的思想は、この様な方法を使用して複数の検索条件との一致度を統合して、総合満足度を算出し、その値の大きいものから順に並び替える事にある。ファジィ検索法では、この様な満足度によって出力が決定されるため、通常の検索法と異なり、理論上、全ての検索物件が検索条件に当てはまり、その当てはまる程度が総合満足度として与えられている事になる。従ってその出力される検索物件の数は、全ての検索物件から唯一の検索物件までいろいろ考えられ、どこで出力・非出力の線を引きかが問題となる。出力数の決定をフレキシブルな方法によって決定する事も提案されているが[5]、この様な方法でも、あるところで出力するかしないか線引きをしなければならない。出力・非出力の線引きの結果、出力数が少なすぎる時はファジィ検索システムの長所が生かせないし、逆に、出力数が多いにも多いと人間は判断が困難になるであろう。

心理学における有名な Miller[6]の研究では、人間の記憶数は7個前後である事が明らかにされている。人間が全体を見渡して的確な判断を下せるのはその記憶項目数に依存し、これに基づいて判断がなされるとすれば、あまりにも多い出力数では意味がなくなるどころか、上手く判断が出来なくなり、意思決定過程上の不都合を生じる可能性がある。従って、出力結果は、多くとも7~10数個が妥当であろう。

この際に問題となる事は、この検索物件の表示数の決定によって、各ユーザーの持つ概念の僅かな違いで、物件の出力範囲にユーザーの概念に近い検索物件が入ってこなくなる事である。従って、この様な問題を解決するためには、各ユーザーのもつ概念を抽出する事が必要となる。逆に、ユーザーのもつ検索条件に関する概念が抽出できれば、この情報をもとにメンバーシップ関数を調整し、これによって再検索を行う事によりこの様な問題の解決法の一つとなる事が期待できる。

3. 意思決定支援と再検索

前章では、心理学的知見とファジィ検索法の特徴からファジィ検索システムの出力の問題点について述べた。また、その解決のために各ユーザーの概念の抽出が必要となる事を述べた。この事は、システムに対してユーザー側の情報をユーザー自らが与える過程が必要となる事を示している。従って、ユーザーに対してある種の手間の掛かる作業を要求する事になる。しかし、システムを使用するユーザー側の簡便性を考慮すると、この様な過程には出来るだけユーザーの負担を最小限に押さえるべきであると考えられる。それでは、これを可能とする方法について考察するために、人間が検索システムを利用して意思決定を行う過程について議論する。

数理心理学的な意思決定モデルの分野において有名なものに EBA (Elimination By Aspect) [7]があるが、このモデルは人間が複数の選択肢の中から唯一の選択肢を選び出す時、それら選択肢がもつ属性に注目し、選択肢の絞り込みが行われる事を仮定している。実際に、人間が検索システムによって表示された物件の中から、最も適していると思われる物件を選択する際にも、表示された物件の属性を見ながら、それらを比較検討し、『まあまあ良い』と思われる物件を絞り込んで行くであろう。人間があまり多い表示物件の中から適切な選択が出来ないという事は前に議論したが、この時、表示された物件の属性を互いに比較できる様な表示を行っておけば、一回目の通常のファジィ検索によって一般的な人が『良い』と考える様な物件がすでに表示されているため、ユーザーはこの中から『まあま

あ良い』と思える物件を比較的容易に選択する事が出来る。この様にして選択された物件は、現時点で表示されている物件の中で『最も良い』と感じられるものであり、この様なユーザーが自ら選択した物件のもつ属性はユーザーの検索条件に関する概念を反映していると考えられる。従って、ユーザーのもつ検索条件に関する概念の抽出にこのデータの属性を利用する事は自然な考え方である。

また、多くのファジィ検索システムにおいては、ユーザーは言語による検索条件の提示を行う。この事は、この言語に対応するシステム内の検索条件のメンバーシップ関数を指定する事に相当する。つまりこれはユーザーの要求を、一般的な言語概念を表しているとシステムの設計者が考えているメンバーシップ関数によって、表している事になる。従って、これは言語によって喚起される、システムの設計者の持つ概念によって、システムに対して意思表示を行ったに過ぎない。これは、ユーザー自身の明らかな意思表示ではない。これに対して、現時点で『最も良い』とユーザーが感じる検索物件を示す事は、このユーザー自身の持つ概念の積極的提示を伴うユーザー自身の明らかな意思表示であると考えられる事が出来る。

そこで我々は、この様にしてユーザー自身が選択した物件の属性のメンバーシップ関数をユーザーのもつ検索条件に対する概念を反映する情報として抽出する事にした。我々はこの考え方に基づき、一回目にシステムの内部にあらかじめ設定されている検索条件のメンバーシップ関数を用いて検索を行い、ユーザー自身にこの検索結果の表示の中から『最も良い』と思われる物件を示してもらい、更に、これを用いて検索条件のメンバーシップ関数を調整し、再検索を行う方法を提案する。この時、ユーザーに課せられた作業は、自らがどの物件を選択したか示すのみで良い事になり、簡便性が実現される。

我々が後に提案する方法では、再検索の際、システム内に設定されている平均的あるいは一般的な検索条件メンバーシップ関数を個々のユーザーに対して用いるのではなく、ユーザーのもつ概念を推定し、再検索の際の新たなメンバーシップ関数とする。この事によって、ユーザーが自らの要求に最も適合すると考える物件が、より上位の順位となるばかりか、今までは検索結果の出力範囲に入っていなかった物件が出力される事によって、この様な問題の解決法の一つとなる事が、後のシミュレーションの結果明らかとなるであろう。

また、我々の提案する方法では再検索が行われるが、この様な過程は検索システムを意思決定支援システム

として見た場合も有効な過程であると考えられる。

これらの議論に関係して、実際の検索において人間が物件を決定する際についていまだ少し考察してみよう。

例えば、店頭などに設置されているデータベースシステムを使用して顧客が購入しようとしている物件を探す際、二通りのケースが考えられる。

人間がデータ検索システムの力を借りてある決定を行うとき、通常はいますぐに明確な決定を下さなければならないケースは少なく、処理結果として得られた複数の物件はプリントアウトして決定に至るまでの間、手元に置かれるだろう。これが第一のケースである。第二のケースは、検索作業をした後、比較的短時間のうちにどの物件を購入するか明確な決定をしなければならないケースである。

この様な二つのケースは、旅行に行く先を決定する際にも見られる。旅行に行く予定の日がまだ迫っていないときには明確に決定をせず(第一のケース)、様々な計画を考える時間があるが、予定の日が迫ってくると明確にどこ行くか決定しておかなければならない(第二のケース)。この様に、明確な意思決定をしなければならないのは、時が迫っている状況におかれている場合であると考えられる事が出来る。

ところで、今述べた第一のケースにも第二のケースの様な状況を作り出し、あえて積極的な判断をユーザーに行わせ、ユーザーからデータ検索システムに情報を与え、これを検討する事によって新たにユーザーに検索情報を与える事は有益であると考えられる。なぜなら、ただ単に処理結果として得られたプリントアウトを見ながら決定を行うより、この様なシステムとユーザーとの間の『対話』による相互作用によってよりよい決定がなされる可能性があるからである。第一のケースにおいて、ユーザーに表示されている物件からいずれかの物件を選択するように指示すれば、意思決定の主体(ユーザー)は、与えられている情報をもとに検討を重ね、明確な判断を行おうとするだろう。しかし、従来のファジィ検索法ではこの様な状況に充分対応する事は出来なかった。

本論文で提案する方法は、明確な意思決定を一度させて、データベースから再検索を行い、ユーザーは自らの意思を明確にする作業を行う。これによって、現時点でよりよい決定を行ってしまうとする積極的な意思決定が可能となる。もちろん初回の検索結果を含む各検索結果をプリントアウトして手元においておき、後に決定の修正をしても一向に不都合は生じないであろう。それどころか、再検索の結果による豊富な情報により、よりよい決定が可能となるだろう。

4. 個人条件メンバーシップ関数の推定法を利用したファジィ検索法

4.1 提案する検索法の概要

我々は、本章でメンバーシップ関数は個人個人異なっているものとして個々人の検索条件に関するメンバーシップ関数の違いを推定する方法を提案し、この考えに基づいた新しいファジィ検索法を提案する。しかし、この様な分野の研究はあまり行われていないので、ここでまず検索に用いられる検索条件のメンバーシップ関数に関する言葉の定義を行う。本論文では、データベース内にあらかじめ設定されている検索条件に関するメンバーシップ関数を『平均条件メンバーシップ関数』とし、それに対し、個人個人がもつ概念に基づいた検索条件に関するメンバーシップ関数を『個人条件メンバーシップ関数』と定義する。

それでは、我々が提案する新しいファジィ検索法についてその概要を簡単に述べる。提案する検索法の概要を図4.1.に示す。

(1) 検索条件の入力

まず検索条件を言語によって入力する。

この言語の入力はデータ検索システム内の検索条件のメンバーシップ関数を指定する事になる。本研究ではメンバーシップ関数はデータ、検索条件とも三角型ファジィ数を用いた。

(2) 平均条件メンバーシップ関数による検索

従来のファジィ検索法と同様に、(1)によって指定さ

れた検索条件のメンバーシップ関数とデータの属性のメンバーシップ関数の一致度により検索を行う。使用されるメンバーシップ関数はシステム内にすでに設定されているもの(平均条件メンバーシップ関数)を用いる。

(3) 検索結果の表示, 表示物件のうち最も望ましい物件を選択

検索結果は、その物件名だけではなく、各物件の属性、各条件の一致度、総合満足度等を同時に表示する。この表示内容を見て、ユーザーは自らの要求に一番適していると思われる物件を選択する。

(4) 個人条件メンバーシップ関数の推定

ユーザーが指定した現時点で最も適合すると考える物件のデータからこのユーザーの検索条件のメンバーシップ関数を推定する。推定は我々が後に本論文で提案する方法によって行われる。提案する推定法によって、メンバーシップ関数は一回目の検索でユーザーが自ら選択した物件と、一回目の検索で用いられた検索条件の中間的なパラメータをもつ事になる。

(5) 平均条件メンバーシップ関数の調整

一回目の検索によって推定された個人条件メンバーシップ関数を使用して平均条件メンバーシップ関数を調整する。

(6) 再検索

一回目の検索によって推定された個人条件メンバーシップ関数を使用して再び検索を行う。これによって再検索は、一回目の検索でユーザーが自ら選択した物件と、一回目の検索で用いられた検索条件の中間的なパラメータをもつメンバーシップ関数によって行われる事になる。

(7) 必要ならば個人条件メンバーシップ関数を保存

推定された個人条件メンバーシップ関数は、ユーザーのもつ概念を反映していると考えられるため、これを次の検索に使用する事が考えられる。本検索法では、次の検索のために、今回の検索によって推定された個人条件メンバーシップ関数を保存する事が出来る。もし、保存しないならば次の検索は、平均条件メンバーシップ関数によって行われる。

4.2. 平均条件メンバーシップ関数による検索

はじめに検索を行う時、システムはユーザーの情報をもっていない。従って、この時使用される検索条件のメンバーシップ関数をシステム内にあらかじめ用意しておく必要がある。この様なメンバーシップ関数は、多くの人にとってほどほどに適合するものが望ましい。本研究では、初回の検索に用いられる検索条件のメンバーシップ関数に多数のユーザーにとって平均的なメ

図 4.1. 個人条件メンバーシップ関数の推定法を利用したファジィ検索法

ンバーシップ関数を採用する事にし、このメンバーシップ関数を平均条件メンバーシップ関数と呼ぶ事にした。初回の検索においては、これらのメンバーシップ関数とデータの一貫性に基づく検索を行う。本研究が取り扱う問題においては、データもメンバーシップ関数(三角型メンバーシップ関数)で表されていると仮定している。ファジィ集合論において、二つのメンバーシップ関数間の一致度は、様々定義可能であるが、我々はこれらの論理積集合の所属度最大値をとる方法であるMax-Min法を採用している(以下、Max-Min法と呼ぶ。)。また、この平均条件メンバーシップ関数は後に定義される方法により、調整がなされる。

4.2.1. 重要度

検索を行なう時、ユーザーはある属性に対して大きな関心を示し、結果についてもその属性を強く意識するという事がある。この様な時には、意識する属性以外を考慮に入れなくても、結果が同じという事になる。逆に、全く意識しない属性に関しては、考慮する必要がないとしても構わないといえる。この様にユーザーは全ての属性を同等に扱う事はない。また、全ての属性を必ずしも必要としていない。そこで、ユーザーが検索属性をどれだけ重視して検索を行なうか、その度合を示すものとして、本研究では、『重要度』というものを定義した。最も重視する属性に対しては“1.0”を、無視して構わない属性には“0.0”を振り分ける事により、ある時の検索における属性ごとの重要性の差を表現する。ある属性 K の重要度 ω_K は次の様な閉区間の間の値をとる。

$$\omega_K \in [0, 1] \quad (4.1)$$

検索システムをインターフェースの観点から考えた場合、重要度を数値で入力する事は便利であるとは言いがたい。したがって、検索条件の場合と同様に言語によって重要度を指定する事にした。それぞれの重要度は、重要性を示す言語によって、各々の値が設定される。例えば、我々が後に例として示すシステムでは、重要度は表4.1.の様に設定されている。

4.2.2. 最終満足度

条件と要素の一致の度合によって定められる一致度と、ユーザーが検索属性に対して考える重要度により、各属性に対する最終満足度を定義する。

ある属性 K に対する最終満足度 α_K^* は、ある属性 k における一致度 α_K と重要度 ω_K の積である。

$$\alpha_K^* = \omega_K \cdot \alpha_K \quad (4.2)$$

表 4.1. 重要度

重要性を示す言語	重要度
非常に重要	1
重要	5/6
やや重要	4/6
標準的	3/6
あまり重要でない	2/6
殆ど重要でない	1/6
全く重要でない	0

4.2.3. 総合満足度

次に物件の検索における総合的な評価値となる総合満足度を定義する。システムは検索結果として総合満足度の高い順に物件を並べ、その上位のものを出力する。総合満足度は、 ω_K による α_K の重みづけられた平均演算によって算出する。

検索属性数が n 個、それぞれの属性を $K_1, K_2, \dots, K_n$ 、重要度を $\omega_{K_1}, \omega_{K_2}, \dots, \omega_{K_n}$ とすると、ある物件の総合満足度は、式(4.3)(4.4)となる。

$$\alpha^* = \frac{\omega_{K_1} \cdot \alpha_{K_1} + \dots + \omega_{K_2} \cdot \alpha_{K_2} + \dots + \omega_{K_n} \cdot \alpha_{K_n}}{\omega_{K_1} + \dots + \omega_{K_2} + \dots + \omega_{K_n}} \quad (4.3)$$

$$= \frac{\sum_{i=1}^n \omega_{K_i} \cdot \alpha_{K_i}}{\sum_{i=1}^n \omega_{K_i}} \quad (4.4)$$

4.3. 検索結果の表示とユーザーによる物件の選択

検索による表示は、物件名のみではなくその物件がもつ属性をも同時に表示する。

これはユーザーの各物件の属性を検討させ、これらの表示物件の中から最も望ましい物件を選択させるために必要となる。例えば、表示は図4.2.の様になされる。これは本検索法の応用例として後に紹介するシステムの表示例である。この一回目の検索による表示から、ユーザーは、自らが最も『良い』と思われる、あるいは『マシ』であると考える物件を選択し、計算機に示す。これによって、システムはユーザーからユーザー自身の情報を与えられる事になる。

検索に使用された属性1(スピード)に対する検索条件のメンバーシップ関数(三角型メンバーシップ関数の中央値)

データの属性1(スピード)のメンバーシップ関数の中央値

属性1(スピード)の最終満足度(各属性ごとの満足度)

順位	種牡馬	スピード [400]	スタミナ [500]	底力 [300]	頑丈度 [400]	距離 [1600]	総合
1	ダンシング グレイブ	400 (83.3)	400 (33.3)	350 (23.8)	400 (66.7)	2000 (25.0)	81.9
2	シアトリカ ル	400 (83.3)	350 (16.7)	400 (25.0)	400 (66.7)	1850 (28.3)	73.3
3	ラッキー キャスト	300 (41.7)	400 (33.3)	300 (50.0)	400 (66.7)	1800 (26.7)	72.8
4	・	・	・	・	・	・	・

↑ 順位 ↑ 物件名 ↑ 属性1 ↑ 属性2 ↑ 属性3 ↑ 属性4 ↑ 属性5 ↑ 総合満足度

図 4.2. 表示例(種牡馬データベース[11])

4.4. 個人条件メンバーシップ関数の推定

4.4.1 検索における学習

一般的なファジィ検索システムにおいては、システム内にすでに設定されている『基準』で検索を行なう。ここでいう『基準』とは、あらかじめシステム内に設定した検索条件のメンバーシップ関数である。

ユーザーは人間であり、人間は、個々が違った考えを持って生きている。そのため、個々のユーザーが思い浮かべる条件にも多少の相違が出てくる事が考えられる。この事は我々が行った予備実験によっても支持されている。従って、システム内に予め設定された条件では、多くのユーザーを十分に満足させる事は難しいといえる。

これを打開するために、本論文において提案する方法では、検索の結果とユーザーの考えていた予想のずれを、検索システムにフィードバックする事によって再検索を行う。再検索は、検索条件メンバーシップ関数をデータベース内のデータのメンバーシップ関数を利用して学習させる事によってなされる。また、この操作を繰り返す事により、再検索によるフィードバックがあるたびに、検索システムはユーザーの条件設定の特徴を読み取り、ユーザー個人の知識を蓄えてゆく。この結果、検索システムの条件と、ユーザーの条件のずれは小さくなり、ユーザーの満足が満たされる。そしてこの知識を各ユーザーごとに保存する事により、次の検索に活用する事が可能となる。

4.4.2. 内分点演算による満足度の上昇について

ファジィ検索法において重要な概念はメンバーシッ

プ関数の一致度である。ファジィ検索法は検索条件とデータとともにメンバーシップ関数で記述し、これらのメンバーシップ関数がお互いに『近い』ものを選び出す手法であると考えられる事が出来る。この『近い』度合いの指標としてメンバーシップ関数の一致度が使用されこれは多くの場合、検索上の満足度と呼ばれる。

あまり『近くない』二つのメンバーシップ関数を『近い』ものとするには双方のメンバーシップ関数のパラメータ間の差を小さくすればよい。この一つの方法として、一方のメンバーシップ関数のパラメータをそのパラメータと、他方のメンバーシップ関数のパラメータとの内分点をとるよう移動させる方法が考えられる。我々が提案する方法は、この様な内分点演算による一致度の上昇に基づいている。今、ある属性 K に属する条件集合があるとする。この条件集合とは、“高い” “低い” など、ユーザーが検索に際して任意に選択が可能な条件である。この条件集合の中からユーザーは条件を決める事になる。この中からユーザーが決定した条件を $k(K \supset k)$ とする。 k は、例えば“高い”である。

ここで、 n 個のパラメータをもつ検索回数 t での個人条件メンバーシップ関数 $S_k(t)$ とデータのメンバーシップ関数 D_K を次の様なものとし、学習について考えてみる。

$$S_k(t) = (s_{k1}(t), s_{k2}(t), \dots, s_{kn}(t)) \quad (4.5)$$

$$D_K = (d_{K1}, d_{K2}, \dots, d_{Kn}) \quad (4.6)$$

ファジィ検索法においてはこの $S_k(t)$ と D_K の各成分の一致度を計算する事によって、各物件を評価し、結果を出力する。もし、 $S_k(t)$ と D_K の各成分の一致度が低いものにも関わらず、ユーザーがその物件を選んだとすると、このユーザーの $S_k(t)$ は D_K に『近い』事がわかる。

従って、学習の方法は、個人条件メンバーシップ関数をデータのメンバーシップ関数の値に近づける事によって行う。

p を $p \in [0, 1]$ とした場合、内分点演算を使用した個人条件の学習を次の様に定義する。

$$S_k(t+1) = (1-p) \cdot S_k(t) + p \cdot D_K \quad (4.7)$$

パラメータ表記では式(4.8)となる。

$$s_{ki}(t+1) = (1-p) \cdot s_{ki}(t) + p \cdot d_{Ki} \quad (i=1, 2, \dots, n) \quad (4.8)$$

これにより、条件 k における個人条件の学習が可能となる。この時のメンバーシップ関数は変化し、デー

タのメンバーシップ関数に近づいた事になる。従って、満足度は上昇する事になる。

我々はこの様にユーザーが現時点で『良い』と考える物件のもつ各検索条件に近くなったメンバーシップ関数を用い再検索を行う事とする。

4.4.3. 個人条件メンバーシップ関数の学習の数理

4.4.2.において内分点を再検索時のメンバーシップ関数として使用する事について述べた。我々が本研究で用いたメンバーシップ関数は検索条件もデータも三角型ファジィ数である。ここではこの方法に基づいた三角型ファジィ数を用いた個人条件の推定法について提案する。

三角型ファジィ数 A は、次の様に表現される[8]。

$$\mu_A = \frac{x-a_1}{a_2-a_1} \cdot I(x|[a_1, a_2]) + \frac{a_2-x}{a_3-a_2} \cdot I(x|[a_2, a_3]) \quad (4.9)$$

ただし、

$$\text{if } A \ni x \quad \text{then} \quad I(x|A) = 1 \quad (4.10)$$

$$\text{otherwise} \quad I(x|A) = 0 \quad (4.11)$$

である。

我々はこの様な三角型ファジィ数をパラメータ表記(4.12)

$$A = (a_1, a_2, a_3) \quad (4.12)$$

によって記述する事にする。

4.4.2.において提案した学習則における問題はパラメータ p の設定である。

パラメータ p は、ある属性 K に対する重要度が大きく、かつ、その検索条件 $k(K \supset k)$ における一致度が大きくないとき大きくなるべきである。つまり、 p の大きさは、

(ルール4.1)：

$$\begin{aligned} & \text{『重要度が大きい』 かつ 『一致度が大きくない]} \\ & \rightarrow \text{『} p \text{ が大きい]} \end{aligned}$$

なるルールであらわされる。

また、検索回数 t における一致度が大きくない真理値を $\eta(t)$ 、重要度が大きい真理値を $w_K(t)$ とすれば、ルール4.1をあらわす最も簡単な演算は、“ $\wedge$ ”を合成命題演算子(『かつ』)として、

(ルール4.2)： $p = w_K(t) \wedge \eta(t)$

とあらわす事が出来る。

人間の判断における『かつ』(合成命題)モデルには二つの演算子が存在する。これらの演算子は $\text{Min}[\cdot]$

(Min 規則)と $\Pi[\cdot]$ (積規則)であるが、この二つの『かつ』真理値(合成命題真理値)の予測値は、その差が有意に大きいので、これらのうちのどちらが『かつ』モデルとして優れているか古くから議論されてきた。実際は、これらのモデルは考察する問題によってそれぞれがよくフィットしており、印象によって評価を与えよと指示された時は $\Pi[\cdot]$ (積規則)に近くなり、と論理的判断を強いられる時は $\text{Min}[\cdot]$ (Min 規則)に近くなる[9]。

多くのファジィ理論の応用においては、Min 規則が多く使われるが、我々は問題の性格から積規則を採用し、 $p = \eta(t) \cdot w_K(t)$ とおく事にする。従って、学習は式(4.7), (4.8)によって次の様に定義される。

検索条件 K において、条件 $k(K \supset k)$ に対する検索回数 t での個人条件メンバーシップ関数を

$$\mathbf{S}_k(t) = (s_{k1}(t), s_{k2}(t), s_{k3}(t)) \quad (4.13)$$

選択されたデータの条件 k に対するメンバーシップ関数を

$$\mathbf{D}_K = (d_{K1}, d_{K2}, d_{K3}) \quad (4.14)$$

として、個人条件メンバーシップ関数の学習を、全ての k に関して、

$$\begin{aligned} \mathbf{S}_k(t+1) &= (1 - \eta(t) \cdot w_K(t)) \cdot \mathbf{S}_k(t) + \eta(t) \cdot w_K(t) \cdot \mathbf{D}_K \end{aligned} \quad (4.15)$$

と定義する。

パラメータ表記では

$$\begin{aligned} s_{ki}(t+1) &= (1 - \eta(t) \cdot w_K(t)) \cdot s_{ki}(t) \\ &+ \eta(t) \cdot w_K(t) \cdot d_{Ki} \quad (i=1, 2, 3) \end{aligned} \quad (4.16)$$

となる。

また、 $w_K(t)$ は検索回数 t での条件 K に対する検索上の重み付け関数、 η は学習速度関数と呼ぶ事にする($w_K(t), \eta(t) \in [0, 1]$)。式(4.15)(4.16)は、個人条件メンバーシップ関数とデータのメンバーシップ関数の内分点を、学習後の個人条件メンバーシップ関数として、個人条件メンバーシップ関数をデータにどれだけ近づけるかは、重み付け関数と学習速度関数の値によって決定される。

図4.3.に例を示しているように、内分点は、 $(\eta(t) \cdot w_K(t))$ 対 $(1 - \eta(t) \cdot w_K(t))$ の割合を取る。

図 4.3. 個人条件メンバーシップ関数の学習

4.4.4. 重み付け関数について

検索回数 t でのある検索条件 K に対する重要度を $\omega_K(t)$ とすると、重み付け関数 $w_K(t)$ は次の様な関数となる。

$$w_K(t) = w(\omega_K(t)) \quad (4.17)$$

関数 $w(\omega_K(t))$ が満たすべき公理として、ルール4.1 から重要度 $\omega_K(t)$ に対して非減少関数の形を取るようにする。また、関数 $w(\omega_K(t))$ の値域は、 $[0, 1]$ とする。なお5章のシミュレーションにおいては、 $w(\omega_K(t)) = \omega_K(t)$ を用いた。

4.4.5. 学習速度関数について

学習速度関数は、検索によって得られた満足度に依りて、個人条件メンバーシップ関数の学習する割合を変化させるものである。つまり、ある属性 K で得られた満足度を α_K で表わすと、学習速度 $\eta(t)$ は次の様な関数となる。

$$\eta(t) = \eta(\alpha_K(t)) \quad (4.18)$$

関数 $\eta(\alpha_K(t))$ が満たすべき公理として、ルール4.1 から満足度 α_K に対して非増加関数の形を取るようにする。また、関数 $\eta(\alpha_K(t))$ の値域は、 $[0, 1]$ とする。具体的には、 $\eta(\alpha_K) = 1 - \alpha_K^2$, $\eta(\alpha_K) = 1 - \alpha_K^k$ あるいは、 $1/\{1 + \exp[\sigma(\alpha - 0.5)]\}$ ($\sigma \in \mathbb{R}^+$) などの減少関数をあげる事が出来るであろう。

なお、本研究のシミュレーションにおいては、 $\eta(\alpha_K) = 1 - \alpha_K^2$ を用いている。

学習速度関数及び、重み付け関数により、満足度が小さく、かつ、重要度が大きい場合、学習速度の効果はもっとも大きくなる。つまり、満足度が“0.0”で重要度が“1.0”であった時、メンバーシップ関数の移動距離は最大となる。逆に満足度が大きく、重要度が小

図 4.4. 学習速度関数と重み付け関数・1

さい場合、学習速度は小さい値をとり、移動距離も小さくなる。

学習速度関数が $\eta(\alpha) = 1 - \alpha^2$ 、重み付け関数が $w = \omega$ の時の $p(\alpha, \omega) = \eta \cdot w$ の変化を図4.4.に示す。

4.5. 平均条件メンバーシップ関数の調整

4.5.1. 『平均的な』検索条件と平均条件メンバーシップ関数

我々は、初回の検索に用いられるシステムにあらかじめ設定されている検索条件のメンバーシップ関数を平均条件メンバーシップ関数と呼んだ。これはその目的から文字どおりシステムを使用するユーザーの集団にとって『平均的な』メンバーシップ関数であるべきである。従ってシステムの設計者はこれらの事情及び利用可能な情報を考慮し、主体的に、しかし良識をもって『平均的な』メンバーシップ関数を設定する事が望まれる。しかし、この様に設定された『平均的な』メンバーシップ関数も、システムを使用するユーザーの集団が各検索条件に対してある種の固有の傾向をもっていた場合、十分にその目的を果たす事が出来ないだろう。我々の提案する検索法では、この平均条件メンバーシップ関数及び、データのメンバーシップ関数を利用して、それぞれのユーザーに対応する個人条件メンバーシップ関数を推定し、これを再検索に用いる。従って効率の良い推定のためには、この時利用される平均条件メンバーシップ関数は、ユーザーの集団の『平均的な』概念をより良く反映している事が望ましい。この様な事から、我々は再検索の際用いられた個人条件メンバーシップ関数によって、平均条件メンバーシップ関数を調整する。

我々はまず、『平均条件』とは、検索システム使用者の『個人条件』の平均を取ったものであると捉え直し、平均条件メンバーシップ関数の学習を定義する。更に、

システムの設計者が主体的に設定した平均条件メンバーシップ関数の情報も十分に活かす方法も与える。

それでは平均条件の学習を考える。もっとも初期の平均条件、つまりユーザーがまったくいない時は、システムは事前にユーザーの検索条件の方向性を知る事ができないので、システムの設計者が主体的に設定したメンバーシップ関数を平均条件メンバーシップ関数の初期値として採用する。

属性 K の条件 k における初期の平均条件メンバーシップ関数の初期値は次のようになる。

$$\mathbf{S}_k(0) = (s_{k1}(0), s_{k2}(0), s_{k3}(0)) \quad (4.19)$$

一回目の検索によって推定された個人条件メンバーシップ関数を式(4.20)とし、

$$\mathbf{S}_k(1)^* = (s_{k1}(1)^*, s_{k2}(1)^*, s_{k3}(1)^*) \quad (4.20)$$

1 回目の検索による平均条件の学習を考える。これを相加平均によって求めるとすると式(4.21)となる。

$$s_k(1) = \frac{s_k(0) + s_k(1)^*}{2} \quad (4.21)$$

検索が t 回行なわれた時の学習は次の様に定める。

$$s_k(t) = \frac{t \cdot s_k(t-1) + s_k(t)^*}{t+1} \quad (4.22)$$

各パラメータについては、

$$s_{ki}(t) = \frac{t \cdot s_{ki}(t-1) + s_{ki}(t)^*}{t+1} \quad (i=1, 2, 3) \quad (4.23)$$

となる。相加平均演算子は t -operator ではないので、結合性が無く、最も最近入力されたメンバーシップ関数に最も影響を受けてしまう。したがって式(4.22)の様に、1 回前の検索に使用した平均条件メンバーシップ関数と検索回数の積を取り、平均条件メンバーシップ関数を全てのメンバーシップ関数の相加平均としている。

4.5.2. t の積極的活用とシステムの設計者の主体性の保持

我々が定義した平均条件メンバーシップ関数の学習法によって学習を重ねると、ユーザーの集団によっては、システムの設計者があらかじめ設定したメンバーシップ関数から大きく変化してしまう可能性がある。我々は、 t はもともと回数として設定したが、もしシステムの設計者あるいは管理者が平均条件メンバーシップ関数をあまり動かしたくないと考えるならば、 t の初期値に大きな値を与えておけば良いだろう。これによって平均条件メンバーシップ関数はあまり改訂されなくなる。この事によって先に述べた平均条件の集団による傾向の抽出とシステムの設計者の主体性の保持の両立がなされるだろう。

4.6. 重要度の学習

我々は本検索法において、検索属性に対する重要度を導入した。先程定義した個人条件メンバーシップ関数の学習の際にもこの重要度は大きな役割を果たした。

ところで検索の目的が一定の場合、ユーザーがある属性をどれだけ重視するかが、あまり大きく変化はしない事がある。この様な場合、操作の簡便性からユーザーが重要度の入力を省略しても、システムがそれを各ユーザー毎に予測し検索を行えるよう、検索法に重要度の学習を導入する。

我々は重要度の学習を次の様に定義する。

$$\omega_K^*(t) = \frac{t \cdot \omega_K^*(t-1) + \omega_K(t)}{(t+1)} \quad (4.24)$$

ただし、属性 K における検索回数 t の時の重要度の予測値を $\omega_K^*(t)$ 、 $\omega_K^*(t-1)$ は $(t-1)$ 回目の検索までの学習によって得られた重要度の予測値、 $\omega_K(t)$ は今回(t 回目)の検索でユーザーが設定した重要度である。各属性について式(4.24)の学習式を用いる事によって、新たな重要度の予測値が得られる。

またユーザーが重要度を入力しなかった時、 $\omega_K(t)$ を以下の様に定める。

$$\omega_K(t) = \omega_K^*(t-1) \quad (4.25)$$

これを式(4.24)に代入し、学習後の重要度を求めてみると式(4.26)となる。

$$\omega_K^*(t) = \omega_K^*(t-1) \quad (4.26)$$

これによって、 t 回目の検索において重要度の入力がない場合、 $t-1$ 回目の検索における重要度と同じ値になる。もし、各検索において、ランダムに重要度が入力されたならば、全ての K に関して、 $\omega_K^* \rightarrow 0.5$ となる。この事によって、このユーザーの予想される重要度の入力は各条件で全て同じとなり、各条件が同程度考慮される事になる。

5. 「学習機能を持ったファジィ検索システム」の構築

4章において個人メンバーシップ関数の推定法及び、本推定法を利用したファジィ検索法についてを提案した。本検索法の応用例として、我々は「学習機能を持ったファジィ検索システム」を構築した。ここでいう学習機能とは、個人条件メンバーシップ関数の推定、平均条件メンバーシップ関数の調整、更に、ユーザーが検索の重要度を指定しなかった場合、そのユーザーの過去の検索から、重要度の分布を推定し、自動的に入力する機能の三つを指す[10], [11], [12]。作成したプログラムは「料理メニュー検索システム」[10], 「種牡馬検索システム」[11], 「スキー場検索システム」[12]の

三つである。

－処理結果例－

表 5.1. 検索条件(料理検索システム)

属性	予算	調理時間	熱量	塩分	難易度
条件	普通	普通	普通	普通	難しい
重要度	非常に重要	やや重要	あまり重要でない		非常に重要

表 5.2. 平均条件メンバーシップ関数による検索結果(料理検索システム)

順位	料理名	予算 [400]	時間 [30]	熱量 [400]	塩分 [30]	難易度 [65]	総合
1	豆腐と山芋	350 (85.7)	30 (66.7)	300 (23.8)	14 (9.1)	75 (85.7)	81.3
2	野菜のかき	500 (80.0)	30 (66.7)	300 (23.8)	35 (22.2)	60 (75.0)	80.3
3	福袋	400 (100)	40 (50.0)	280 (19.0)	28 (30.4)	55 (57.1)	77.0
4	新じゃがい	300 (71.4)	30 (66.7)	380 (31.0)	21 (19.0)	55 (57.1)	73.6
5	そばろのひ	500 (71.4)	40 (50.0)	500 (22.2)	63 (0.0)	70 (100)	73.1
6	豆腐入り鶏	500 (80.0)	30 (66.7)	320 (25.0)	40 (16.7)	50 (50.0)	71.5
7	なすの香り	300 (71.4)	35 (57.1)	230 (10.7)	18 (16.7)	60 (80.0)	70.8

表 5.3. 推定された個人条件メンバーシップ関数による再検索結果(料理検索システム)

順位	料理名	予算 [435]	時間 [30]	熱量 [383]	塩分 [30]	難易度 [65]	総合
1	野菜のかき	500 (86.0)	30 (66.7)	300 (25.2)	35 (22.2)	60 (87.5)	86.3
2	福袋	400 (91.3)	40 (50.0)	280 (20.7)	28 (30.2)	55 (71.4)	79.1
3	豆腐入り鶏	500 (86.5)	30 (66.7)	320 (26.6)	40 (16.7)	50 (62.5)	77.5
4	新じゃがい	300 (61.4)	30 (66.7)	380 (33.0)	21 (17.5)	55 (71.4)	75.0
5	豆腐と山芋	350 (75.7)	30 (66.7)	300 (25.2)	14 (6.7)	50 (73.0)	74.2
6	豚肉の赤み	550 (81.3)	30 (66.9)	250 (15.0)	18 (14.5)	60 (62.5)	72.0
7	なすの香り	300 (61.4)	35 (57.1)	230 (12.3)	18 (5.2)	60 (90.0)	70.8

表 5.4. 検索条件(種牡馬検索システム)

属性	スピード	スタミナ	底力	頑丈度	距離
条件	やや速い	タフ	普通	ややある	やや短い
重要度	重要	やや重要	標準的	やや重要	あまり重要でない

「料理メニュー検索システム」, 「種牡馬検索システム」, 「スキー場検索システム」の平均条件メンバーシップ関数による検索結果を図に示す。

さらに一回目の検索結果を受けて、ユーザーが自らが要求する検索物件に一番近いと思われる物件としてそれぞれ、『野菜のかき』(2位), 『グリーングラス』(4位), 『八方尾根』(2位)を選択し、再検索を行った時の処理結果を示す。それぞれのシステムにおいて、選択した検索物件の総合満足度及び順位が上昇し、料

表 5.5. 平均条件メンバーシップ関数による検索結果(種牡馬検索システム)

順位	種牡馬	スピード [400]	スタミナ [500]	底力 [300]	頑丈度 [400]	距離 [1600]	総合
1	ダンシング グレーブ	400 (83.3)	400 (33.3)	350 (23.8)	400 (66.7)	2000 (25.0)	81.9
2	シアトリカ ル	400 (83.3)	350 (16.7)	400 (25.0)	400 (66.7)	1850 (28.3)	73.3
3	ラッキー キャスト	300 (41.7)	400 (33.3)	300 (50.0)	400 (66.7)	1800 (26.7)	72.8
4	グリーン ラス	400 (41.7)	400 (33.3)	300 (50.0)	500 (66.7)	2600 (18.2)	71.6
5	アンバー シャダイ	300 (83.3)	400 (33.3)	300 (50.0)	400 (33.3)	2100 (14.8)	69.9
6	シアトル スルー	400 (83.3)	350 (16.7)	300 (50.0)	300 (33.3)	2000 (23.8)	69.0
7	ルシオン	350 (62.5)	350 (16.7)	250 (37.5)	400 (66.7)	2000 (23.8)	69.0

表 5.6. 推定された個人条件メンバーシップ関数による再検索結果(種牡馬検索システム)

順位	種牡馬	スピード [400]	スタミナ [500]	底力 [300]	頑丈度 [400]	距離 [1600]	総合
1	ダンシング グレーブ	400 (83.3)	400 (50.3)	350 (37.5)	400 (50.3)	2000 (30.7)	84.1
2	グリーン ラス	400 (83.3)	400 (50.3)	300 (50.0)	500 (47.2)	2600 (20.7)	83.9
3	シアトリカ ル	400 (83.3)	350 (33.7)	400 (25.0)	400 (50.3)	1850 (33.1)	75.2
4	ラッキー キャスト	300 (41.7)	400 (50.3)	300 (50.0)	400 (50.3)	1800 (32.5)	75.0
5	アンバー シャダイ	300 (47.1)	400 (50.3)	300 (50.0)	400 (50.3)	2100 (27.0)	73.1
6	レインボー クエスト	400 (83.3)	400 (50.3)	300 (50.0)	200 (0.0)	2000 (31.0)	71.5
7	ルシオン	350 (62.5)	350 (33.7)	250 (37.5)	400 (50.3)	2000 (30.3)	71.4

表 5.7. 検索条件(スキー場検索システム)

属性	難度	雪質	距離	規模	人気
条件	難しい	良い	やや速い	大きい	ある
重要度	やや重要	非常に重要	標準的	標準的	殆ど重要でない

理検索システムにおいては、『野菜のかき』は第1位となった(『グリーングラス』(4位→2位),『八方尾根』(2位→2位))。また、検索物件のうち今までは出力の範囲内に入っていなかった物件がそれぞれのシステムにおいて見られる事がわかる(『そばろのひ』→『豚肉の赤み』(料理検索システム),『シアトルスルー』→『レインボークエスト』(種牡馬検索システム),『安比高原』→『野沢温泉』(スキー場検索システム))。

6. 考察 一問題点及び今後の展望一

それぞれのシステムの処理結果例において、再検索の結果、ユーザーが一回目の検索で選択した物件がより上位に位置するようになったばかりか、今までは結

表 5.8. 平均条件メンバーシップ関数による検索結果(スキー場検索システム)

順位	スキー場	難易度 [900]	雪質 [900]	距離 [800]	規模 [700]	人気 [900]	総合
1	キロリソート	800 (55.5)	900 (100)	999 (0.0)	700 (50.0)	900 (16.7)	78.4
2	八方尾根	900 (66.7)	999 (100)	380 (8.0)	999 (25.0)	999 (16.7)	76.4
3	安比高原	500 (28.6)	800 (100)	800 (50.0)	900 (33.3)	900 (16.7)	74.8
4	山形蔵王	700 (47.6)	999 (83.3)	520 (22.0)	999 (25.0)	999 (16.7)	74.6
5	八海山	900 (66.7)	900 (100)	280 (0.0)	500 (35.7)	500 (7.1)	73.9
6	五竜とおみ	700 (47.6)	900 (100)	360 (6.0)	800 (41.7)	600 (8.3)	71.9
7	白馬岩岳	700 (47.6)	900 (100)	380 (8.0)	900 (33.3)	700 (11.1)	70.6

表 5.9. 推定された個人条件メンバーシップ関数による再検索結果(スキー場検索システム)

順位	スキー場	難易度 [900]	雪質 [900]	距離 [595]	規模 [812]	人気 [900]	総合
1	山形蔵王	700 (47.6)	999 (100)	520 (34.4)	999 (30.8)	999 (16.7)	81.3
2	八方尾根	900 (66.7)	999 (100)	380 (8.2)	999 (30.8)	999 (16.7)	78.4
3	キロリソート	800 (55.5)	900 (100)	999 (0.0)	700 (41.5)	900 (16.7)	75.4
4	五竜とおみ	700 (47.6)	900 (100)	360 (4.3)	800 (48.9)	600 (8.3)	73.8
5	白馬岩岳	700 (47.6)	900 (100)	380 (8.2)	900 (41.0)	700 (11.1)	73.4
6	八海山	900 (66.7)	900 (100)	280 (0.0)	500 (26.4)	500 (7.1)	70.7
7	野沢温泉	800 (55.5)	900 (100)	320 (0.0)	999 (30.8)	800 (13.9)	70.7

果の表示範囲に入っていなかった検索物件が出力されているのがわかる。選択した物件の総合満足度が上昇している事から考えても(野菜のかき:80.3→86.3, グリーングラス:71.6→83.9, 八方尾根:76.4→78.4)、再検索に用いられた検索条件のメンバーシップ関数は、選択された物件の属性のメンバーシップ関数に近くなっている事がわかり、新たに出力範囲に加わった物件は表示されなくなった物件よりも選択された物件に近い事が示唆される。

実際の検索においては、ユーザーは1回目の検索で最も良いと思われた検索結果と、この様な個人条件メンバーシップ関数の推定後の再検索によって新たに結果の出力に現れた物件のどちらがより良いか決定する事になるが、これは、1回目の検索結果を検討するよりも更に容易であり(二つあるいは三つか、そのくらいの物件を比較する事に他ならないから)、ユーザーは新たに現れた物件が最も良いと感じられた時は、その物件を選んだという事をシステムに告げ、再び再検索を行うであろうし、もし、その物件があまり良くないと感じられた時にはこれ以上の良い結果は得られないと判断し、前回の物件を選択するか、あるいは物件を得る事をあきらめるだろう。

また、本検索システムにおいては、再検索に使用した推定された個人条件メンバーシップ関数を保存する事が出来、次回の検索に使用する事が出来る。ユーザーが検索結果が実際良いものであったか、良くないものであったかを判断するのは、システムの出力結果に基づいて判断を下し、行動を行った後である。もし、ユーザーが出力結果に基づいて行動し、満足いく結果を得たならば、ユーザーは前回の検索によって得られた『推定された個人条件メンバーシップ関数』を使用して検索を行う事が考えられる。

最後に提案した検索法に関する今後の課題として以下の(1)~(3)をあげておく。

- (1) ある検索条件を推定し保存して次回の個人条件メンバーシップ関数として使用する場合、これを繰り返す事によってその検索条件のメンバーシップ関数が隣り合ったメンバーシップ関数を追い越してしまう事がある:

例えば、種牡馬検索システムにおいて、あるユーザーが検索属性「スピード」に関して「普通」ばかりを使用し、かつ、このユーザーの「普通」が平均的なユーザーの「速い」に近かった場合、個人条件メンバーシップ関数を推定するために学習を重ねると、「普通」のメンバーシップ関数が「やや速い」のメンバーシップ関数に極めて近くなってしまう事がある。

これはユーザーが毎回の検索である特定の条件のみ

を検索条件として選択する場合に起こる。このような場合、このユーザーは他の条件は使用していないと考えられるので、検索上の不都合は生じないかもしれない。しかし、より完全な検索システムを構築するためには各条件間の相互作用を推定法の理論の中に組み入れる必要がある。

(2) 学習速度・重み付け関数の検討：

本検索法における個人条件メンバーシップ関数の学習は、学習速度・重み付け関数によって様々変化する可能性がある。我々は学習速度関数を α に関する非増加関数 ($\eta \in [0, 1]$) としたがこの最も簡単な例として、

$$\eta(\alpha) = \delta_\alpha \cdot (1 - \alpha^r) \quad (\delta_\alpha \in [0, 1], r \in R^+) \quad (6.1)$$

をあげる事が出来る。5章のシステム例ではこのうち、 $\eta = 1 - \alpha^2$ を採用したが、これは個人条件メンバーシップ関数を比較的大きくデータのメンバーシップ関数に近づける。しかし、学習速度に、式(6.1)において $r < 1$ なる r を採用すると、個人条件メンバーシップ関数はあまり大きくデータのメンバーシップ関数に近づかなくなる。我々は同じ個人条件メンバーシップ関数の推定に関しても、再検索に用いる場合は、 $\eta = 1 - \alpha^2$ などの $r > 1$ なる急速に学習が起る学習速度を設定し、個人条件メンバーシップ関数の設定用に用いる場合は、 $\eta = 0.5 \cdot (1 - \alpha^{1/2})$ などの $r < 1$ なる学習速度を設定すべきであると考えている。

重み付け関数に関しては、各検索条件に関する重要度に対する非減少関数であるとした ($w(\omega_K(t)) \in [0, 1]$)。この様な最も簡単な関数としては、

$$w(\omega_K(t)) = \delta_w \cdot (\omega_K(t))^r \quad (\delta_w \in [0, 1], r \in R^+) \quad (6.2)$$

をあげる事が出来る。我々はシステム例において、これらのうち、 $w(\omega_K(t)) = \omega_K(t)$ を用いたが、あるいは $r > 1$ なる重み付け関数を採用する事が必要となるかもしれない。

また、先に述べたように、学習速度を『一致度が大きい』真理値と考え、重み付け関数を『重要度が大きい』真理値と考えると、それぞれ、

$$\eta(\alpha | \sigma_\alpha, \theta_\alpha) = 1 / \{1 + \exp[\sigma_\alpha \cdot (\alpha - \theta_\alpha)]\} \quad (6.3)$$

$$w(\omega_K(t) | \sigma_w, \theta_w) = 1 / \{1 + \exp[-\sigma_w \cdot (\omega_K(t) - \theta_w)]\} \quad (6.4)$$

(ただし、 $\sigma_\alpha \in R^+, \theta_\alpha \in [0, 1]; \sigma_w \in R^+, \theta_w \in [0, 1]$)

の様な関数を設定する事が理にかなっているかもしれない。

学習速度関数が

$$\eta(\alpha) = 1 / \{1 + \exp[12 \cdot (\alpha - 0.5)]\},$$

重み付け関数が

$w = 1 / \{1 + \exp[-12 \cdot (\omega - 0.5)]\}$ の時の $p(\alpha, \omega) = \eta \cdot w$ の変化を図6.1.に示す。式(6.3)及び(6.4)を使用すると、学習は $\omega_K(t) > \theta_w$ かつ $\alpha < \theta_\alpha$ に関しては急激に起こり、 $\omega_K(t) < \theta_w$ かつ $\alpha > \theta_\alpha$ に関してはほとんど行われないう様にする事が出来る。

本研究の応用例として構築した検索システムのプログラミングにおいて、 η の変更は容易に出来るようになっているので、システムの管理者がそのシステムに適合した関数を打ち込んで使用しても良いであろう。しかし、これらの関数の決定はその目的に応じて決定されるべきものであり、心理学的な研究と関連づけながら検討すべき課題であると我々は考えている。学習速度・重み付け関数の導入は本検索法の大きな特徴の一つであり、これは今後の興味ある課題となるだろう。

(3) GUI (Graphical User Interface) の採用：

本提案の応用例として構築した「料理メニュー検索システム」、「種牡馬検索システム」、「スキー場検索システム」は全て MS-DOS 上で動作するプログラムである。そのためその表示が図4.2.に示した様なもので、ユーザーは数値で表されたデータを検討する事によって、物件の決定を行わなければならなかった。しかし、この方式ではユーザーはデータのメンバーシップ関数を明確にイメージする事が困難であるため、システムの実用にあたっては GUI 化が是非とも必要である。GUI 化により、個人条件やデータのメンバーシップ関数をグラフで表示でき、ユーザーは物件の選択をより簡単に行なえるようになり、より効率よくユーザーの持つ概念を抽出する事が出来るようになるだろう。

図 6.1. 学習速度関数と重み付け関数・2

参考文献

- [1] アルビン・トフラー(徳山次郎 訳): パワーシフト上巻・下巻, 中公文庫(1993)
- [2] 大木直人, 野村恭彦, 田代秀夫, 亀倉龍, 横山光男, 松下温: 複数のユーザーが共有するファジィデータベースにおけるメンバーシップ関数のチューニング, 日本ファジィ学会誌 Vol.6, No.2, pp.351-362 (1994)
- [3] 杉山孝男, 椎塚久雄: ファジィ積分による総合評価法とその応用, 日本ファジィ学会誌 Vol.5, No.4, pp.772-782(1993)
- [4] 中島信之, 般若裕子: ファジィ推論による総合評価法とその応用, 日本ファジィ学会誌 Vol.5, No.4, pp.761-771 (1993)
- [5] 醍醐元正, 中島信之, 般若裕子: 検索条件の自動変更を組み込んだ商品購入支援システムの構築, 日本ファジィ学会誌 Vol.7, No.2, pp.380-389 (1995)
- [6] G.A.Miller, The magical number seven, plus or minus two. Some limits on our capacity for processing information, Psychological Review, 63, pp81-97, 1956
- [7] A.Tversky: Elimination by aspects: A theory of choice, Psychological Review, 79, pp281-299, (1972)
- [8] Arnold Kaufmann, Madan M.Gupta: FUZZY MATHEMATICAL MODELS IN ENGINEERING AND MANAGEMENT SCIENCE, ELSEVIER SCIENCE PUBLISHERS B.V.(1988)
- [9] 山下利之, 山下清美: ファジィ論理判断における情報の統合過程, 心理学評論, vol.32, No.4., pp351-369(1989)
- [10] 大橋一文: 学習機能を持ったファジィ検索システムの構築-料理検索システムへの応用-, 帝京平成大学卒業研究論文(1996)
- [11] 小川英明: 学習機能を持ったファジィ検索システムの構築-種牡馬検索システムへの応用-, 帝京平成大学卒業研究論文(1996)
- [12] 関根正直: 学習機能を持ったファジィ検索システムの構築-スキー場検索システムへの応用-, 帝京平成大学卒業研究論文(1996)

(1997年1月16日 受付)

(1997年12月11日 再受付)

[問い合わせ先]

〒290-0193

千葉県市原市潤井戸大谷2289-23

帝京平成大学大学院応用数理研究室

前田 隆宏

TEL : 0436-74-5511(代表)

E-mail : maeda@cohen.is.thu.ac.jp

著者紹介

前田 隆宏 (まえだ たかひろ)

帝京平成大学大学院(旧 帝京技術科学大学大学院)応用数理研究室

東邦大学(理学部生物学科)、早稲田大学(大学院人間科学研究科)、多摩大学(大学院経営情報学研究科)で生物学、情報科学、経営科学を学ぶ。1995～1997年度、帝京平成大学大学院情報学研究科博士後期課程にて研究・教育に従事する。学問に関しては、生物物理学、人間科学の諸分野、特に脳機能論(美的感覚の進化論、認識論、数理神経システム論)、スポーツトレーニング理論に興味を持つ。また教育にも大変興味があり、大学研究室の教育的重要性を説き、共同研究を通じて学部生・大学院生・社会人の教育・再教育に貢献したいと考えている。

大橋 一文 (おおはし ひふみ)

帝京平成大学(旧 帝京技術科学大学)応用数理研究室

1995年度 帝京平成大学(旧 帝京技術科学大学)応用数理研究室・学士卒業研究生。1996年3月、帝京平成大学(旧帝京技術科学大学)情報学部情報システム学科卒業(現在、新潟リコー(株)勤務)。在学中は、データ検索法の研究および、多変量解析、人間の集団間の概念の違いを考慮したシステムの研究に従事。お菓子づくりを趣味の一つとし、料理や栄養の観点から人間をサポートするシステムに興味を持つ。

小川 英明 (おがわ ひであき)

帝京平成大学(旧 帝京技術科学大学)応用数理研究室

1995年度 帝京平成大学(旧 帝京技術科学大学)応用数理研究室・学士卒業研究生。1996年3月、帝京平成大学(旧帝京技術科学大学)情報学部情報システム学科卒業(現在、関東管区警察局千葉県通信部勤務)。在学中は、データ検索法の研究および、人間と機械とのインターフェースを利用した学習機能の研究に従事。CAI(Computer Aided Instruction)をはじめとする教育システムに関する研究にも関心がある。また、競走馬(サラブレッド)の芸術的、あるいは生態学的側面に興味を持つ。

関根 正直 (せきね まさなお)

帝京平成大学(旧 帝京技術科学大学)応用数理研究室

1995年度 帝京平成大学(旧 帝京技術科学大学)応用数理研究室・学士卒業研究生。1996年3月、帝京平成大学(旧帝京技術科学大学)情報学部情報システム学科卒業(現在、日本電気通信システム(株)勤務)。在学中は、データ検索法の研究および、Cプログラミングを代表とするプログラミング言語の研究に従事。塾講師(中学生対象)としても長い経歴を持つ。趣味はスキー、その他スポーツ全般。

A Proposal of Fuzzy-Data-Searching Method With Estimation of Personal Membership Functions

by

Takahiro MAEDA, Hihumi OOHASHI, Hideaki OGAWA and Masanao SEKINE

Abstract :

Various fuzzy-data-searching method in the study of data-searching are proposed up to now. In this paper , we review the characteristics of fuzzy-data-searching method from psychological point of view. And the problems is discussed and new fuzzy-data-searching method is proposed. Also , to the end of this paper , three systems based on our proposed method are proposed. (Cooking Menu Searching System , Stallion Data-base System and Ski Resort Seaching System)

Keywords : Personal Membership Function , re-Searching (retrial of searching) , Decision-Supporting-System , Fuzzy Evaluation , Fuzzy-Data-Searching

Contact Address : **Takahiro MAEDA**

Teikyo-Heisei Univ.

2289-23, Ootani, Uruido, Ichihara-shi, Chiba, 290-01 Japan

TEL : 0436-74-5511

E-mail : maeda@cohen.is.thu.ac.jp