

知能情報処理 不確実性推論(1)

不確実な情報に基づいて行動するエージェント

ファジィ推論

- ファジィ論理
- ファジィ集合
- ファジィ推論
- ファジィ制御



ファジィ理論とは

Fuzzy theory

- **あいまい性**についての数学的な理論

例:「非常に背が高い」

- **主観**の科学的利用 \longleftrightarrow **客観**

- **言語**で表現された知識の利用 \longleftrightarrow **数値**

- 提唱者: **ザデー**(1965, カリフォルニア大)

- 応用: **日本**で開花(1987, 家電, 地下鉄)

ファジィ制御

ファジィ・エキスパートシステム

人工知能技術で実現されるソフトウェア(知的エージェント)は、情報の一部が欠けていたり、あいまいだったりするような不確実性のある環境においても、最善の努力をして意思決定して行動することが望まれる。今回はそのような目的で研究されている分野として**ファジィ理論**、特にその中でも**ファジィ推論**を中心に学ぶ。

ファジィ理論の **fuzzy** という単語は「あいまいな」という形容詞で、その名の示すとおり、**あいまい性**を扱うための数学的な理論である。理論自体は、あいまいではなく厳密である。扱う対象が「あいまい性をもつ知識表現」ということである。

その特徴を見ておこう。

たとえば、「非常に背が高い」という知識は、あいまい性をもつ。ある特定の人を連れてきたときに、「非常に背が高い」といえるかどうかを確実に決めることはできない。そのあいまい性の原因は何だろうか。

第一に、それは**主観的**な知識だからである。ある人を見たときに、その人が「非常に背が高い」と思うかどうかは、見る人の感覚次第で、主観的である。こういうものを科学で扱おうというのは、客観性を絶対視し、主観性を排除してきた近代科学の基本哲学に真っ向から立ち向かう革命的な考え方である。

第二に、この知識は人間の使う言語(**自然言語**)で書かれているということである。人間の使う言語は、そもそも、あいまい性を含むものである。これは、あいまい性を排除するために、近代科学がいろいろな知識を何でも数値や数式で表してきたことに対立する考えで、これまた革命的である。

ファジィ理論は、1960年代にザデーという研究者によって創始された。(いまもまだ生きていようだ。)その理論は長い間無視されてきたが、1980年代後半に、日本の研究者が、家電や地下鉄の制御に応用して注目を浴びることとなった。そのような制御技術は**ファジィ制御**と呼ばれている。専門家が言葉で述べたあいまいな知識をコンピュータに移植して、専門家に近い性能をもつ**ファジィ・エキスパートシステム**というものも開発された。

ファジィ論理

Fuzzy logic

P	Q	$P \text{ and } Q$	$P \text{ or } Q$	$\text{not } P$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

2値論理

ファジィ論理	P	Q	$P \text{ and } Q$	$P \text{ or } Q$	$\text{not } P$
	0.6	0.8	0.6	0.8	0.4

$\min(P, Q)$ $\max(P, Q)$ $1 - P$

ファジィ理論の基礎にあるのは、ファジィ集合やファジィ論理という概念である。先に後者から見ていこう。

ふつうの論理は2値論理と呼ばれていて、論理値は「真」か「偽」、あるいは「1」か「0」の2通りある。

しかし、**ファジィ論理**では、論理値は0以上1以下の任意の実数である。2値論理と同様に、1は真、0は偽を表すが、ファジィ論理ではそれが拡張され、0と1の中間の値は、1に近いほど真に近く、0に近いほど偽に近いという、あいまいな真理値を表す。

2値論理の基本的な演算 **and**, **or**, **not** も拡張されている。このスライドにあるように、

$P \text{ and } Q = \min(P, Q)$: 最小

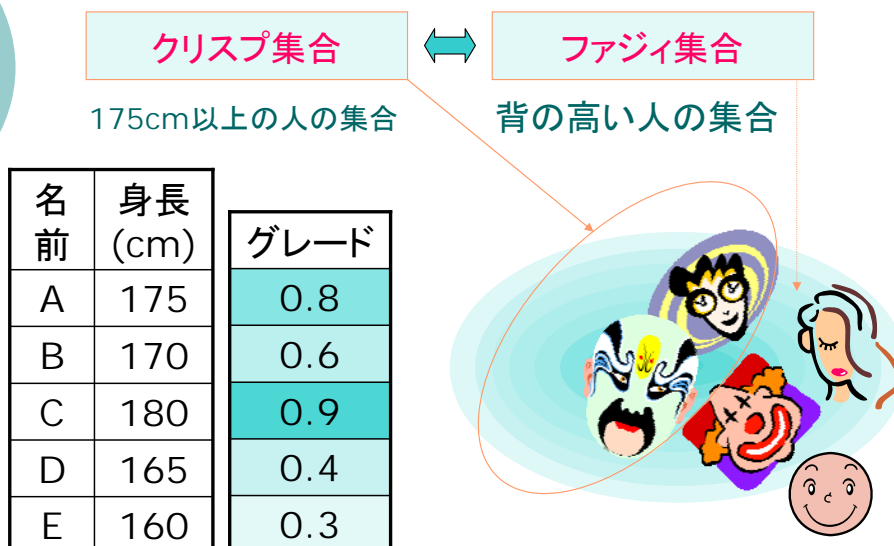
$P \text{ or } Q = \max(P, Q)$: 最大

$\text{not } P = 1 - P$: 補数

と定義すれば、これは2値論理でも成り立つ式になっている。ファジィ論理では、これを P, Q が0と1の間の実数値のときにも成り立つとして、論理演算を定義している。

ファジィ集合(1) ファジィ集合

Fuzzy set



ファジィ集合は、数学の「集合」の概念を拡張したものである。(ただし、伝統的な数学者には無視されている。)

例として、「175cm以上の人の集合」というふつうの集合を考えよう。これに対し、「背の高い人の集合」というのは、ファジィ集合である。たとえば、170cmの人がこの集合の要素に含まれるのかどうか、あいまいで、きっかりと判定できないからである。

ファジィ集合を数学的に厳密に取り扱うために、各要素毎に**グレード**という数値を導入する。これはファジィ論理の論理値のようなもので、0と1の間の実数値である。1に近いほどその要素はこの集合に属している度合いが強い。0に近ければ、あまり属していない。

眠たく、うとうとしているときに、「君はこの授業に参加している人か？」と聞かれたら、グレード=0.7と答えるとよい。

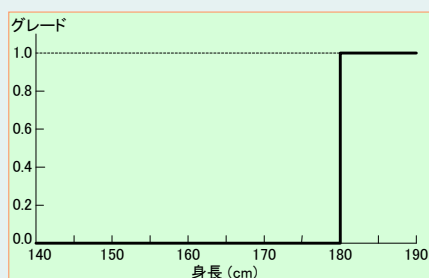
ファジィ集合と明確に区別する必要があるときには、ふつうの集合を**クリस्प集合**と呼ぶ。クリस्प(crisp)というのは、ポテトチップスのようにパリパリした様子を表す形容詞である。ファジィ集合は境界線があいまいで、どこまでを要素として含み、どこからは要素でないのか、ぼんやりしている。それに対し、クリस्प集合は境界線が、パリッと明確なのである。

ファジィ集合(2) メンバーシップ関数 membership function

特徴関数(characteristic function)

$$f_A(x): X \rightarrow \{0,1\} \quad \text{crisp}$$

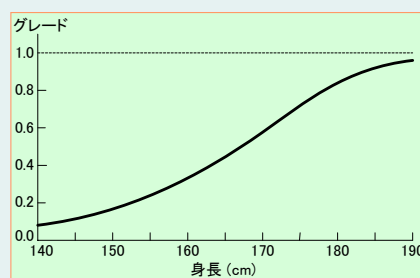
$$f_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$



fuzzy

メンバーシップ関数

$$\mu_A(x): X \rightarrow [0,1]$$

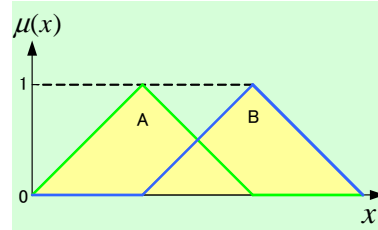


ふつうの集合は**特徴関数**というもので表現できる. 集合Aの特徴関数 f_A は, このスライドで式で定義されているように, 要素xを引数として与えられたときに, xがAの要素か否かにより, 1または0の値をとる. たとえば, その定義の下に描かれているグラフは, 「身長**180cm**以上の人の集合」, より正確には「**180**以上の数の集合」を表す特徴関数のグラフである.

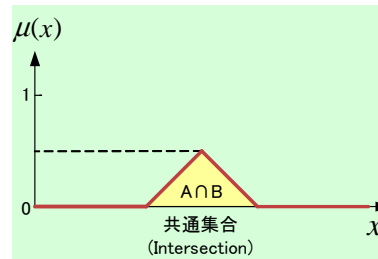
特徴関数の概念をファジィ集合に拡張したものを**メンバーシップ関数**という. メンバーシップ(membership)という英単語は, 日常的には「会員」という意味だが, 数学的には「所属性」すなわち要素が集合に所属するかどうかを示す \in という記号が表す概念を意味している. ファジィ集合Aのメンバーシップ関数は, 要素xを引数として与えられたときに, xがAに属する度合いを表すグレードを値とする. スライドの右下のグラフは, 「身長が高い人の集合」, より正確には「高い身長を表す数の集合」を表すメンバーシップ関数のグラフである. もちろん, このような関数は主観的に決められるものである.

メンバーシップ関数を表す記号として, membershipの頭文字の m に相当するギリシャ文字の μ (ミュー) が用いられることが多い.

ファジィ集合(3) 共通集合 Intersection



$$\begin{aligned}\mu_{A \cap B}(x) &= \mu_A(x) \wedge \mu_B(x) \\ &= \min[\mu_A(x), \mu_B(x)]\end{aligned}$$



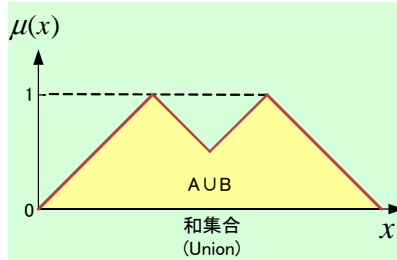
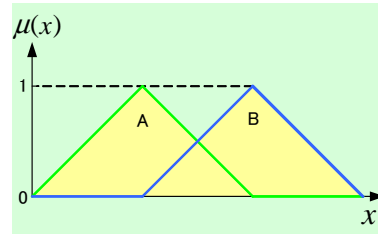
ふつうの集合と同様に、ファジィ集合間での演算として、共通集合(交わり)、和集合(むすび)および補集合が定義されている。

2つのファジィ集合の**共通集合**は、それぞれのメンバーシップ関数の最小値を与えるメンバーシップ関数により定義される。この定義は、特別なケースとして、クリस्प集合の共通集合の定義を含んでいることに注意しよう。

ファジィ集合(4) 和集合

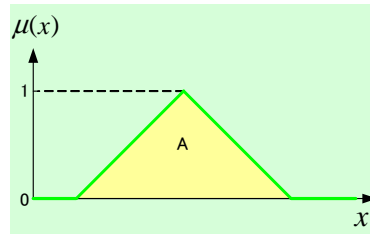
Union

$$\begin{aligned}\mu_{A \cup B}(x) &= \mu_A(x) \vee \mu_B(x) \\ &= \max[\mu_A(x), \mu_B(x)]\end{aligned}$$

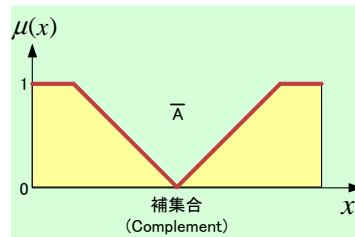


2つのファジィ集合の**和集合**は、それぞれのメンバーシップ関数の最大値を与えるメンバーシップ関数により定義される。この定義は、特別なケースとして、クリस्प集合の和集合の定義を含んでいることに注意しよう。

ファジィ集合(5) 補集合 complement

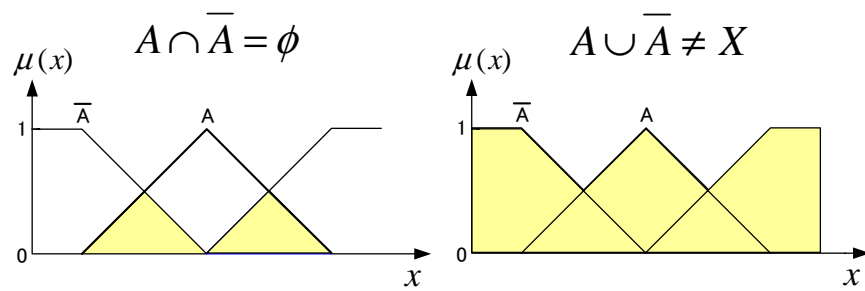


$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$



ファジィ集合の**補集合**は、1 (恒等的に1の値をとる関数)とメンバーシップ関数の差により定義される. この定義は、特別なケースとして、クリस्प集合の補集合の定義を含んでいることに注意しよう.

ファジィ集合(6) 成り立たない性質



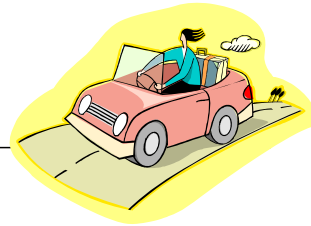
多くの場合、クリस्प集合について成り立つ数学的な性質は、ファジィ集合についても成り立つ。

しかし、成り立たない性質として、たとえば、このスライドのようなものがある。



ファジィ推論とファジィ制御

ファジィ推論の例題(1) ファジィルール



ルール IF x is A
AND y is B
THEN z is C

ルール1 カーブがきつく、速度が小さいならば、速度を保持

ルール2 カーブがきつく、速度が大きいならば、速度を下げる

ルール3 カーブがゆるく、速度が小さいならば、速度を上げる

ルール4 カーブがゆるく、速度が大きいならば、速度を保持

Question カーブ半径=60m, 速度=50km/hならば, 加速度=?

一般に、論理的な形式で与えられている**知識**と何らかの**事実**から**結論**を導き出す計算手続きを**推論**という。特に、「もし～ならば、～である」の形式の知識は、**IF-THEN** **ルール**あるいは単に**ルール**とも呼ばれ、応用範囲が広い。

ファジィ推論は、あいまい性を含むルールをファジィ論理に基づいて取り扱うものである。ルールの形式は、典型的には

IF x is A **AND** y is B **THEN** z is C

の形となる。

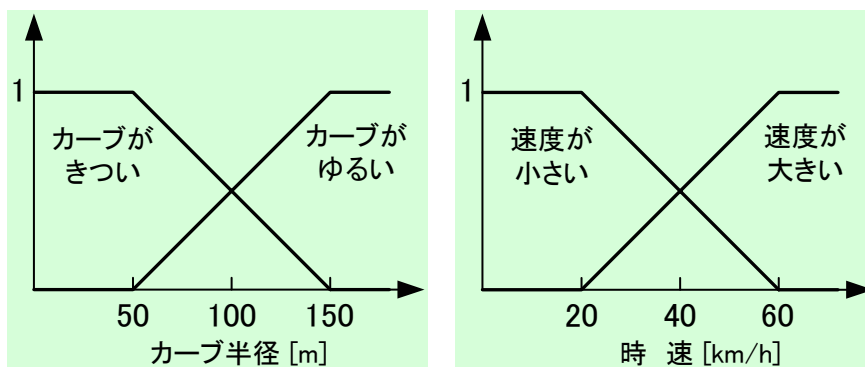
ここで、**IF**, **is**, **AND**, **THEN** は、ルールを記述するための構文要素(キーワード)である。論理学の用語にならって、**IF** から **THEN** の前までの部分を**前件部**、**THEN** 以降の部分を**後件部**という。

x, y, z は何か(ふつうは数)を表す変数、**A, B, C** はファジィ集合である。「 x is A」は「 $x \in A$ 」すなわち「 x はファジィ集合 **A** に属する」という意味を表している。あるいは、「 x は **A** である」と読み、あいまい性を含むファジィ論理の命題を表していると考えてもよい。

人工知能技術の発展により、最近ではコンピュータによって自動車を高速道路などで自動運転できるようになってきた。このスライドに示した例は、車をコンピュータで自動運転するときに、カーブのきつさ(カーブの半径)と現在の速度の大きさに依存して、速度の上げ下げ(加速度)を制御するための簡単な4つのルールを表している。ここで、たとえば、カーブ半径=60m, 速度=50km/h が与えられたときに、加速度を具体的にどんな値とすべきかを決定するのが、ファジィ推論の役割である。

なお、この例題は、萩原将文著:ニューロ・ファジィ・遺伝的アルゴリズム、産業図書(1994)の4.4節からの引用である。

ファジィ推論の例題(2) メンバーシップ関数(前件部)



この例題を解くために、「カーブがきつい」というあいまいな命題を、主観的であってもよいから定量的に表現する必要がある。そのために、「きついカーブの半径の集合」というファジィ集合のメンバーシップ関数を決める。この例では、スライドに示した折れ線、すなわち

$$\begin{aligned}\mu A(r) &= 1, && \text{if } 0 \leq r < 50 \\ \mu A(r) &= 1 - 0.01 * (r - 50), && \text{if } 50 \leq r < 150 \\ \mu A(r) &= 0, && \text{if } 150 < r\end{aligned}$$

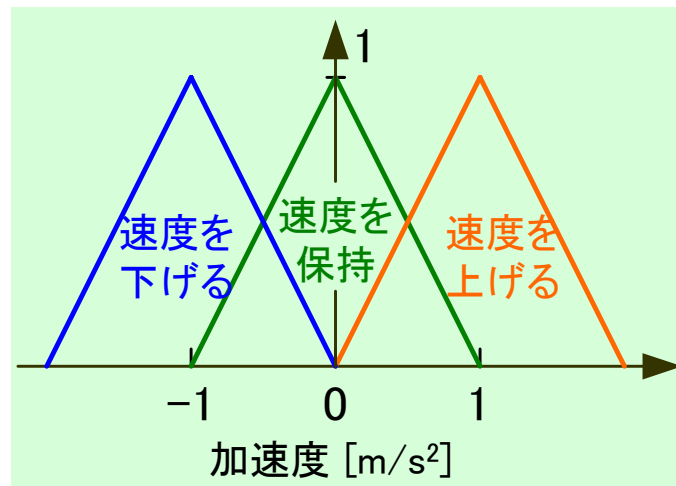
で「カーブがきつい」のメンバーシップ関数とした。ただし、 r はカーブ半径を表す変数、 A は「きついカーブの半径の集合」を表すファジィ集合である。

同様に、「カーブがゆるい」、「速度が小さい」、「速度が大きい」というあいまいな言語表現をメンバーシップ関数で表現する。

このようなメンバーシップ関数は、運転に熟練した専門家の知識や、実験による試行錯誤などによって決める。もともと厳密性を要求されるものではないので、この例のように折れ線のような簡単な関数を使うことが多い。

一般にメンバーシップ関数を設計する作業は時間のかかる大変なものではあるが、基本的には単純労働の繰り返しで決められるものである。難解な数学に基づく自動制御理論による設計よりは、低い開発コストで、満足の良い品質のシステムが開発できる可能性がある。

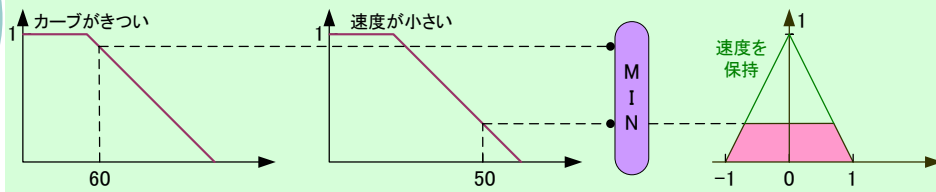
ファジィ推論の例題(3) メンバーシップ関数(後件部)



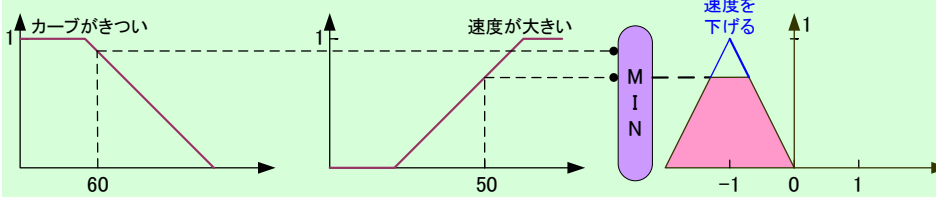
同様に、後件部に用いられているあいまいな命題「速度を下げる・保持・上げる」のそれぞれのメンバーシップ関数を作る。

ファジィ推論の例題(4) ルールの評価

(ルール1) カーブがきつく、速度が小さいならば、速度を保持



(ルール2) カーブがきつく、速度が大きいならば、速度を下げる



一般に、通常の論理学では

「AかつBならばC」

というルールと、AおよびBの2つの事実からCという結論を導き出す推論方法が知られている。人工知能の分野では、これは**前向き推論**と呼ばれている。この推論は、**事実A、B**とルールの**前件部**「AかつB」が一致するときに、**後件部C**を結論として生成する。

ファジィ推論はこの考え方を拡張したものであるが、ふつうの論理的推論と異なる点は、事実と前件部の比較に、一致する・しないの2値論理ではなく、あいまい性のあるファジィ論理を使うことである。

たとえば、すでに述べた自動運転の例のルール1を考えてみよう。

入力が、カーブ半径=60であるという事実は、ルールの前件部の「カーブがきつい」という条件とどの程度よく一致しているだろうか。これは、「カーブがきつい」ことのメンバーシップ関数を用いて、図のように**0.9**と決定できる。

同様に、速度=50である事実は、「速度が小さい」という条件と**0.25**だけ一致する。

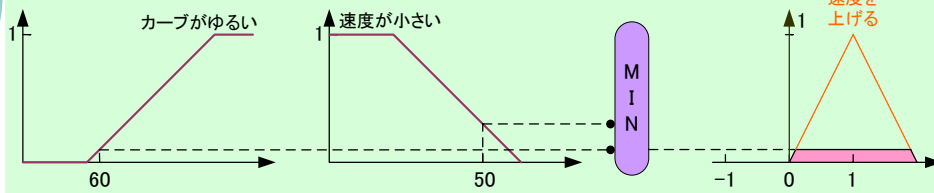
つぎに、ルールの前件部はこの2つのあいまいな命題が共に成り立つこと、すなわち**AND**の条件を要求しているので、いま求めた2つのファジィ論理値のファジィ論理積(ファジィAND)すなわち**最小値(MIN)**を求めると、**0.25**が得られる。つまり、ルール1の前件部は**0.25**程度成り立つということになる。この値をルール1の前件部の**適合度**という。

この適合度とルール1の後件部のファジィ論理積、つまり最小値を求めると、ルール1から得られる結論が得られる。ここで、後件部との最小値といっても、後件部は1つの値ではなく、「速度を維持する」ことを表す「速度」に関するメンバーシップ関数であることに注意する。そこで、適合度のほうも、単なる数値(=0.25)ではなく、常に**0.25**の値を取る恒等関数(グラフは水平線)でメンバーシップ関数を表しているものと考えよう。この2つの関数の最小値というのは、グラフの横軸を表す変数である「速度」の各々の値ごとに最小値をとって得られる関数である。その結果をグラフに描くと、もともとの後件部のグラフの**0.25**より値が大きい部分をカットして水平線に置き換えたグラフとなる。これがルール1から得られる推論結果である。

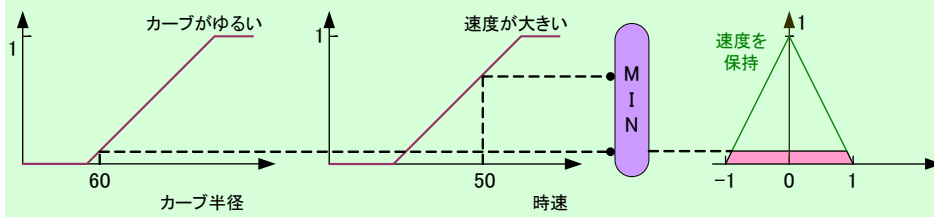
同様な計算をルール2, 3, 4についても行い、このスライドと次のスライドのように推論結果を求める。

ファジィ推論の例題(4) ルールの評価 (続き)

(ルール3) カーブがゆるく、速度が小さいならば、速度を上げる



(ルール4) カーブがゆるく、速度が大きいならば、速度を保持



休憩



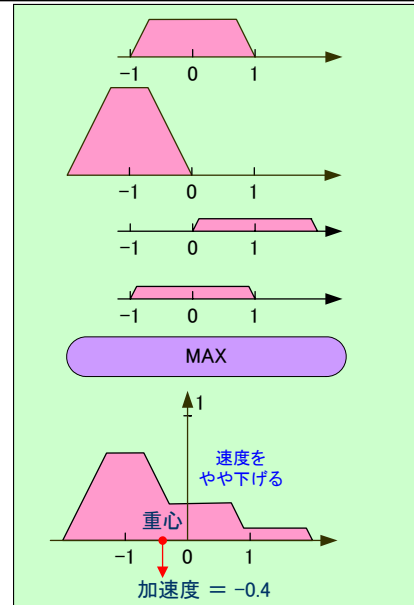
ファジィ推論の例題(5) 出力の合成

重心 (center of gravity)

$$COG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx}$$

↓
離散化
(重み付き和)

$$COG = \frac{\sum_{x=a}^b \mu_A(x) x}{\sum_{x=a}^b \mu_A(x)}$$



最後に、4つのルールから得られたそれぞれの推論結果を合成して総合的な推論結果を得る。

通常の論理では、結論は、各ルールのいずれか少なくとも1つから得られるならば正しいものとされる。その意味では、複数のルールから得られる結論は **OR** で結ばれている。

この考え方をファジィ推論でも採用し、各結論のファジィ論理和(ファジィOR)、すなわち最大値(MAX)を最終結論とする。その結果は、図のように、推論結果を表す各メンバーシップを重ね合わせたときの、最大値部分をなぞったグラフとなる。

このメンバーシップ関数をあえて言語的に表現すると、「速度をやや下げる」という感じになるか。そうだとすると、カーブ半径=60m、速度=50km/h のときには、「速度をやや下げる」という結論が得られたことになる。

この結論は、ファジィ推論の結果なので当然あいまい性をもっている。しかし、実際の運転制御システムに指令を与えるには、これではまずい。機械はあいまい性を理解できないからである。そこで、あいまい性のない数値として結果を出力する必要がある。そのための処理を**非ファジィ化**あるいは**脱ファジィ化**という。

非ファジィ化でよく用いられるのが**重心法**である。これは推論結果を表すメンバーシップ関数の重心の x 座標を出力するものである。この例題の場合には、およそ、加速度=-0.4という値が出力されることになる。

重心は物理学で習う力学的な重心と同じ意味の言葉である。重心に支点を置いて物体を支えると、その周りの力のモーメント(支点からの距離×力)の和が釣り合い、物体をバランス良く支えることができる。いまの例では、横軸(加速度の座標軸)を細長い棒と考え、その上の無数の各点にメンバーシップ関数のグレード値が表す大きさの力が上から鉛直に加えられていると考え、このスライドで示した重心においてその棒をバランス良く支えることができるのである。

重心の計算方法はスライドの数式に示されている。定積分が含まれているので、それを数値計算で求めるために、 x を離散化して近似値を求めるとよい。

すげの
菅野の方法(1) ルールの形式

ルール IF x is **A**
AND y is **B**
THEN z is $f(x,y)$

あいまいでない
数値にする

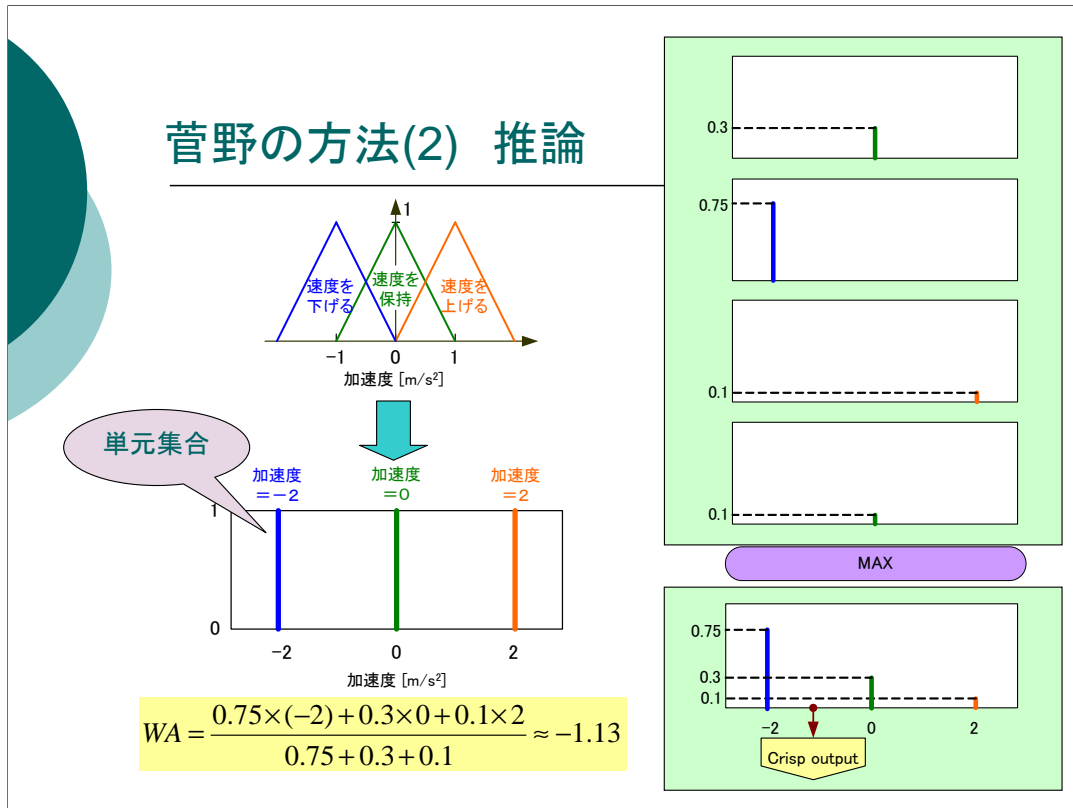
定数でもよい

- ルール1 カーブがきつく、速度が小さいならば、**加速度=0**
- ルール2 カーブがきつく、速度が大きいならば、**加速度=-2**
- ルール3 カーブがゆるく、速度が小さいならば、**加速度=2**
- ルール4 カーブがゆるく、速度が大きいならば、**加速度=0**

Question カーブ半径=60m, 速度=50km/hならば、**加速度=?**

これまで見てきた計算をもっと簡略化するために、菅野の方法が知られている。この方法では、後件部には「 z is **C**」の形の一般的なあいまいな命題ではなく、「 c is $f(x,y)$ 」の形で、入力 x,y から一意に定まるクリस्पな値を指定する。特別な場合として、このスライドの4つのルールのように、 x, y に依存しない定数としてもよい。

菅野の方法(2) 推論

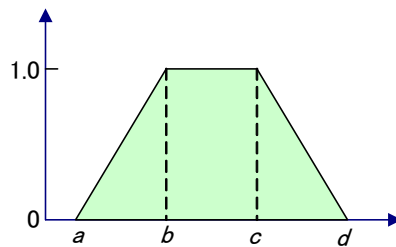


あとはこれまでの考え方と全く同じである。後件部がクリップであるため、各ルールから得られる結論のメンバーシップ関数は、図のように、ある1点を除くすべての点で **0** となる垂直な線分で表現される。これはファジィ集合の言葉でいうと、**単元集合** (要素が1個のみの集合) ということである。その結果、推論のための計算を大幅に簡略化できるのである。応用によっては、この程度の方法でもじゅうぶんに実用的なシステムを作ることができる。

メンバーシップ関数の設計

学習

- パラメータ化し、後にチューニングする



- じゅうぶんな数のルールを用意する

ここで、ファジィ理論の応用上で最も重要なメンバーシップ関数の設計について補足しておく。

メンバーシップ関数は、事前にはっきりと定義するのは困難なので、パラメータ化して定義しておき、後に実験などによって経験的にパラメータを決定するのがよい。たとえば、この図のように、4点 a, b, c, d で与えられる台形をメンバーシップ関数とするなどのモデルを事前に決めておき、これらのパラメータの具体的な値は後に決定する。人工知能の技術の1つである学習に関する手法を使ってもよいだろう。

また、さらに重要なのは、じゅうぶんな数のルールを用意することである。そうすると、そのうち1つくらいに疑問視されるルールがあったとしても、全体としては安定的な無難な推論結果を得ることができる。

ファジィ制御の応用

洗濯機

短い時間で、布傷みを少なくして洗う

- もし、布量が多く、かつ、布質がごわごわならば、水流を強くし、洗い時間を長くする。

掃除機

吸収力は強く、ノズルが床面に吸着しないように

- もし、ゴミの量が多く、かつ、床面がカーペット状ならば、吸収力を大きくする。

最後に、ファジィ制御の応用分野の1つとして、家庭電化製品の例を示す。

洗濯機や掃除機は、効率や品質あるいは使いやすさの面で、相反する要求を満たす必要がある。そこで、それらのバランスを微妙に制御するために、ファジィ理論に基づいた製品が開発された。これらは1990年頃に実際に「ファジィ機能付き」などの宣伝文句とともに販売されたものである。洗濯機の場合は、布量と布質に応じて、水流と洗い時間を制御するためのあいまいなルールが使われる。掃除機の場合は、ゴミの量と床面の状態に応じて、吸収力を制御する。

もちろん、ファジィ理論の応用はファジィ制御に限らず、幅広い分野で応用が模索されている。