

図 6-6 パターンベクトル X とウエイトベクトル W 及び応答関数との関係

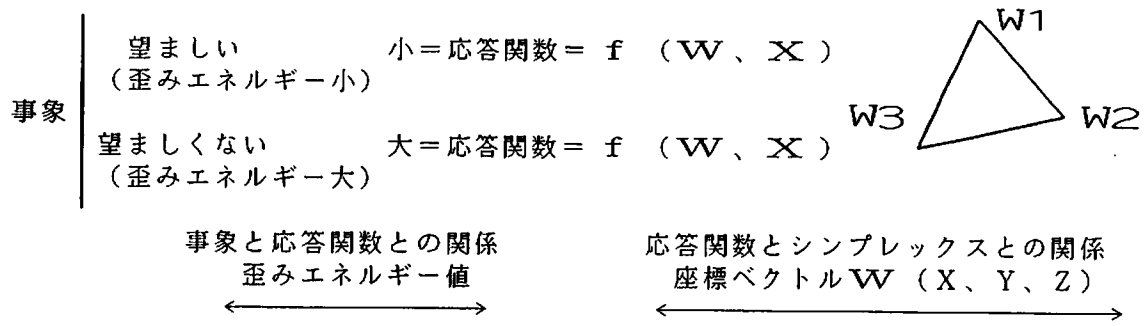
分類問題に利用される時、望ましい結果は分類率が高い事であり、望ましくない結果は分類率の低下である。この関係を応答関数とリンクする事が必要である。この場合、応答関数は望ましい結果の値が小さく、望ましくない結果の値が大きい事を要求する。さらに、分類率と応答関数の値は反比例関係にあるようにしなければならない。

分類問題においてはこのような関係を満たす目的で、誤分類率（分類率が高い時に値が小さく、分類率が低い時に値が大きくなる）を応答関数の出力値とする事が一般的に行われている。応答関数はウエイトベクトルを用いる事で判別関数とリンクされている。

事例 2 : 分子力学の歪みエネルギーの最適化

分子力学では歪みエネルギーを小さくする事が目的であり、この歪みエネルギーを応答関数に用いれば事象とのリンクは問題ない。従って、応答関数は化合物の歪みエネルギーの算出式をそのまま利用できる。

また、応答関数に利用されるベクトルとして化合物の原子の座標を利用すればシンプレックスとのリンクがなされ、シンプレックスの反転に従って原子の座標も変化する。この原子座標を用いて歪みエネルギーを計算（応答関数の出力）する事で、シンプレックスの展開が可能となる。



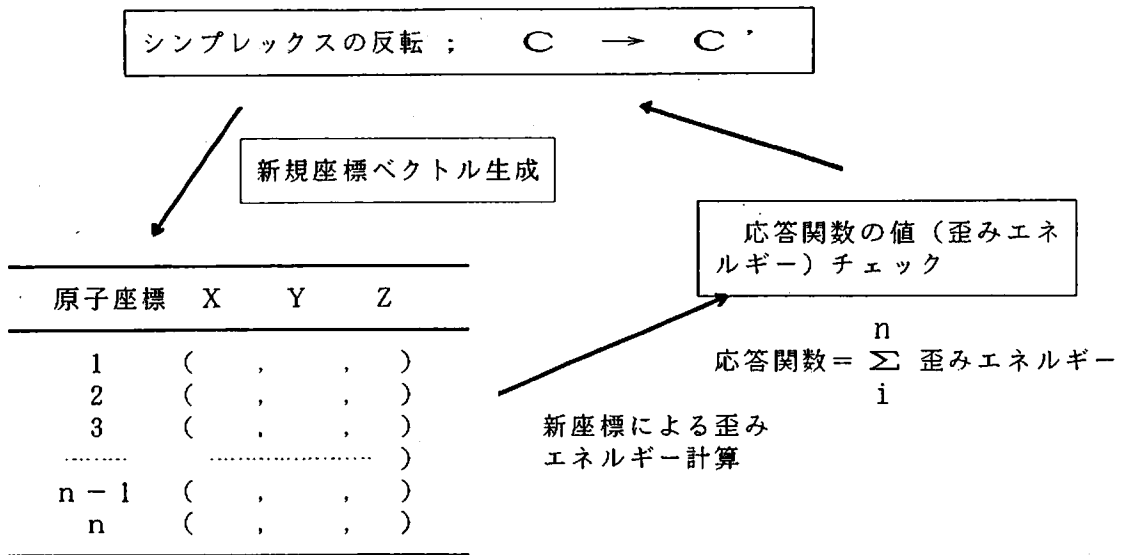


図 6-7 パターンベクトル X とウエイトベクトル W 及び応答関数との関係

以上の事例からもわかるように、シンプレックス法を自分の問題に利用する時に必要となるポイントは

- 変化させたいと思う数値データをベクトル W と関連づける。
- 変化の結果が望ましい時に値が小さくなる関数を見出し、応答関数とする。

の 2 点である。

□ ポリトープ法 (Polytope Method) (シンプレックス法の変形タイプ)

シンプレックス法と基本概念は同じであるが、最適値を探す時に用いるシンプレックスをポリトープ (N次元でN+1個の頂点を持つ多面体) に置き換え、このポリトープを移動させながら最適値を探す手法である。

* Nelder, J.A. and Mead, R.: Computer Journal, 7, 308-313, 1965.

6. 3. ダイナミックプログラミングによる最適化

ダイナミックプログラミングは最適問題を効率良く解決するものとして開発された手法である。他の手法との根本的差異は、他の手法が一回の計算で総ての変数を少しずつ変化させ、この過程を繰り返す事で最適値を求めるのに対し、ダイナミックプログラミングでは変数を1個ずつ最適化し、最終的に全変数を最適化する逐次的なアプローチを取っている点である。

* Bellman, R.: Dynamic Programming, Princeton University Press (1957).

* 尾形克彦、「ダイナミックプログラミング」、培風館、1973.

□ ダイナミックプログラミングの基本

ダイナミックプログラミング(動的計画法とも呼ばれている)による最適化は「最適性原理」というものを基本としており、この原理のもとで再帰方程式を用いてステップバイステップで最適解を求める手法である¹⁾。

最適性原理:

全体としての最適な状態は、最初に部分的な決定がなされたならば残りの部分の状態は、最初に部分的に決定された状態の下での最適な状態となっている。

つまり、全体を見通しての最適状態は、ある一部分の状態が決定されているならば、残りの部分は少なくとも最適な状態になっていなければならないという原理である。この原理に従い、最初に全体の極一部分(最小単位)について最適化を行い、この最初に最適化された部分をあしがかりとして段階的に最適化状態の部分拡大し、最終的に全体の最適化を行うのがダイナミックプログラミングの基本思想である。

このようなダイナミックプログラミングの逐次的な最適化というものは、丁度結晶の生成過程に類似している。つまり、最初は小さな結晶が部分的に成長し、次にその結晶を中心として結晶が広がり、ついには全体が一つの大きな結晶となる過程である。

□ ダイナミックプログラミングによる最適化手順

実際の問題に対し最適化を行う為の主な手順を以下にまとめる。

- ① 全体を構成する状態に関し、最小/大を規定する関数を決定する。
- ② 全体についての最適状態を規定する関数 $f(X)$ と、一つ前の部分的な状態に関する最適状態を規定する関数 $f(X-1)$ との関数再帰方程式を考慮する。
- ③ 関数再帰方程式を解いて最適決定列を決める。この最適状態における最小/大値を求める。

以下ではこの順番に従って説明をする。

- ① 最小/大を規定する関数(ここでは最小値を求めるものとする)を決定する。

$$f_N(c) = \min_{\substack{x_1+x_2+\dots+x_N=C \\ x_i \geq 0}} [g_1(x_1) + g_2(x_2) + \dots + g_i(x_i) + \dots + g_N(x_N)]$$

$f_N(c)$ の N はこの段階が最適過程において、現在 N 番目の過程にある事を示す。 $g_i(x_i)$ は最適化を行う対象となる変数を示す。また \min の下は変数 $g_i(x_i)$ を決定する時の制限条件を示している。つまり、 x_i は正及び $x_1 + x_2 + \dots + x_N = C$ の制限条件下、 $\sum g_i(x_i)$ の値が最小となる x_i の値の組を探す事である。

- ② 関数再帰方程式を求める。

最適性原理に従って①に関する関数再帰方程式を求めるならば、

$$f_N(c) = \min_{0 \leq x_N \leq c} [g_N(x_N) + f_{N-1}(c - x_N)] \quad (N \geq 1)$$

ここで $g_N(x_N)$ は N 番目の変数を示している。 $f_{N-1}(c - x_N)$ は $g_N(x_N)$ を除いた残りの変数の最適状態を示す関数である。

この関数再帰方程式を求める為には、関数 $f_N(c)$ の N の値を1から初めて目的とする値まで順にもとめる事が必要である。この時、最初に使う最小単位の関数 $f_1(c_1)$ の値を決定しておく事が必要となる。

$$f_1(c_1) = \min_{x_1=c_1} [g_1(x_1)] = g_1(c_1)$$

制限条件より $c_1 = c - x_N - x_{N-1} - \dots - x_2$ である。

③ 総ての変数について、最適値を決める。

以上の2式より、 $f_1(c_1)$ から出発して $f_N(c)$ まで順次拡張し、最終的に最適状態を示す変数の組 (X_1, X_2, \dots, X_N) と、その全体値 $f_N(c)$ を求める事が可能である。以上がダイナミックプログラミングの概要である。

□ 初期条件と最適解との関係

ダイナミックプログラミングは最初に与えられる（若しくは決定する）変数を基本とした解を求めるアプローチである。従って、初期変数が異なれば新たな最適解を求める事が可能であり、このような初期条件の異なる一連の問題を解決する時、最初に求めた計算に多少の付加計算を行う事で簡単に求める事が可能である。

この事実がダイナミックプログラミングの一つの特徴でもある。

□ ダイナミックプログラミングの化学分野での利用

ダイナミックプログラミングを用いた利用例は多数存在するが化学分野では少ない。このダイナミックプログラミングが利用される例として典型的なものは最短経路問題、最大収益を目指す配分問題等がある。

これを化学分野で考えるならば、試薬の値段と使用量、最終製品の値段と収率とを考慮しつつ最大の収益をあげる為の試薬使用量の決定などが典型的な問題として考えられる。その他ダイナミックプログラミングが利用出来る可能性があるものとして、シンプレックスの事例2で示した化合物の歪みエネルギー計算による最適形状の算出等も考えられる。この時は化合物中の一つの原子、或いは結合を最小単位として出発し、その回りの原子/結合を順に最適化する事で全体の歪みエネルギーの小さな形状を求めるという手順となる。

このような手続きは、丁度人間が分子模型を用いて化合物を組み立てる時、一つの原子/結合から出発し（この時、既に作成された部分の歪みはあるていど最適化された状態になっている）、最終的に全化合物の最適構造を求める事に似ている。興味ある方は試みると良いであろう。ちなみに他の最適化手法は全原子/結合を同時に少しずつ位置を変化させる事で最適構造を求めている。

□ バイオテクノロジー分野での利用例

これは複数のDNAや蛋白質の配列を比較し、最も似た形（同一アミノ酸配列部分が最も多くなった時）となる状態を探し出すもので“マルチプルアラインメント (Multiple Alignment)”と呼ばれている。この最も似た形を捜し出す手法としてダイナミックプログラミングが利用されている。

$$d(i, j) = \min \begin{cases} d(i, j-1) + n-1 \\ d(i-1, j-1) + \sum_{k=1}^{n-1} W[a_k^{(i)}, s_k^{(j)}] \\ d(i-1, j) + n-1 \end{cases}$$

$d(i, j)$ はマルチプルアラインメントにおける距離を示している。重み関数 $W(X, Y)$ は、 $X=Y$ の時0、 $X \neq Y$ の時1である。また、 $a_k^{(i)}$ は $n-1$ 番目の (k, i) 番目のエレメントを $s_k^{(j)}$ は n 番目のシーケンスの j 番目のエレメントを示す

この式に従ってダイナミックプログラミングを実行する事でマルチプルアラインメントを効率よく実行する事が可能となった。

□ ダイナミックプログラミングの特徴

演算時間について

ダイナミックプログラミングの特徴として従来手法に比べ、計算時間が早い（仕事の内容にもよる）という事がよく指摘されている。これはダイナミックプログラミングの特徴を考える時、良く理解出来る。つまりダイナミックプログラミングの最適化が部分から始まり全体に到るという手続きを踏んでおり、この手続きによる思考過程を考慮すると、以下に示されるようなツリー構造が考えられる。

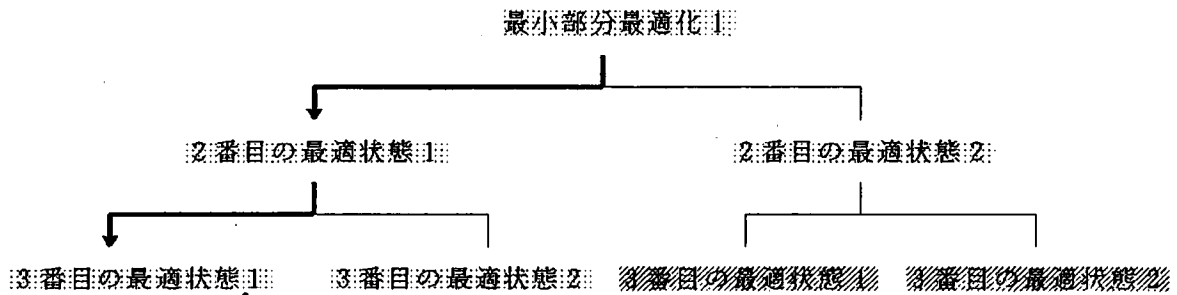


図6-8 最適問題解決の為の過程を表すツリー構造

いま仮に問題を簡単にする為、対象となる問題は最適過程で1段階毎に2通りの選択可能性が存在するとする。この時、前記の問題は図5で示したようなツリー図で書くことが可能である。この問題に対し、最適状態の発見手続きが上図の一の順に進んだとするならば、最適過程で考慮の対象となるものは網かけ部分のみとなり、斜線部分の問題は考慮の対象外となり、このぶん計算時間が短縮される。従って、このようなツリー構造が複雑で階層が深くなればなる程ダイナミックプログラミングによる解析の効果が出てくる。

この種の問題は人工知能分野で常に問題となるもので「場合の数の爆発」として有名である。つまり、ツリーの階層が深く且つ複雑に分岐すればする程考慮すべき場合の数が指数関数的に急増し、計算機の能力を簡単に超過してしまい物理的に解決不可能となるやっかいな問題である。

このような点でダイナミックプログラミングは有効な手段となり得るものである。

例) 20階層のツリーの時

$$\text{考慮対象問題数} = 2^{20} = 1048576 \text{通り}$$

$$\text{ダイナミックプログラミングによる考慮対象問題} = 2 \times 20 = 40 \text{通り}$$

故に総当たり方と比べると、ダイナミックプログラミング方は26214分の1について考慮する事で一つの最適解が得られる。

初期データ依存の問題

前記ツリー問題でも容易に想像出来るが、問題によっては初期データの取りかたにより、最終的に得られる最適値が大きく変化しうる事が考えられる。これは、他の最適化問題が常に抱えている極小/最大問題に該当するものであり、ダイナミックプログラミングといえども避ける事は出来ない問題である。

6. 4 遺伝的アルゴリズム (GENETIC ALGORITHM)による最適化

□ 遺伝的アルゴリズム (GENETIC ALGORITHMS) について

生物は気の遠くなるような長い歴史を通じ、進化や淘汰を繰り返してきた。この進化や淘汰は生物が基本として持っている巧妙な遺伝のメカニズムによりコントロールされてきた。このような厳しい自然環境に耐えてきた遺伝のメカニズムを問題解決の一つのアプローチとして導入しようとしたものが“遺伝的アルゴリズム”である。

先に述べたニューラルネットワークは生体における神経細胞の働きをそのままシミュレートするという点で展開されてきた。遺伝的アルゴリズム、そしてニューラルネットワークと、最近話題となっている技術は生体の複雑／巧妙な仕組みを真似ようとするものが多い。計算機のCPU速度がどんなに向上しようとも、まともなアプローチで計算機を用いれば計算だけで何年、あるいは何十年、何百年とかかる仕事も存在していることは事実である。現実的なものとして、人間はそのような問題を解決する手段を身につけてきた。このような手段はアルゴリズムを明確にすることが出来ないため、第6感／ノウハウ／インスピレーションとして説明されてきた。

3次元以上の問題を人間は解くことが出来ない。その欠点を補う為に展開されてきた統計／多変量解析／パターン認識といった近代的手法にも限界が見えつつある。いま再び生命の持つ驚異的な能力に着目し、新たな解析手法に結び付けようとするのはまさに時代の流れともいえるべきものだろう。

□ 遺伝的アルゴリズム概要

遺伝的アルゴリズムはHOLLANDにより1975年に提唱され、その歴史は比較的古い。このアルゴリズムは最近になり新たな最適化アルゴリズムとして注目されるようになってきた。

J. H. HOLLAND: Adaptation in Natural and Artificial Systems,
University of Michigan Press (1975)

生物は生死を数多く繰り返す過程で自然淘汰を受ける。この生死を繰り返す過程で、大切な情報を子孫に伝えるが、この情報伝達のためのメカニズムが遺伝である。この遺伝は生物の種を問わず全く同じメカニズム（一部変形もあるが例外である）を持つ。遺伝は4種類の塩基（A：アデニン、G：グアニン、C：シトシン、T：チミン）より構成されるDNAの相補コピーを基本として行われている。従って遺伝的アルゴリズムとはこのDNAのコピーメカニズムを模範とし、最適化問題に利用出来るように修正したものである。

DNAのコピーは単なるコピーと異なり、生物が故の複雑なコピーが行われる。このような複雑なコピーを繰り返すことで、進化／突然変異／自然淘汰が行われる。このコピーの主要なパターンとして1～3番のものが存在することが知られている。

1. 増殖 (MULTIPLICATION)
2. 交叉 (CROSSING-OVER)
3. 突然変異 (MUTATION)
4. 淘汰／選択 (SELECTION)

4番の淘汰／選択はコピーとは直接的な関係はないが遺伝的アルゴリズムでは重要な基本操作となるものである。これら4種類のパターンのなかで問題解決を行う際に重要となるものは、単純コピーとなる“増殖 (MULTIPLICATION) を除く3パターンである。

これらのパターンを情報を扱うという問題からみると、増殖は情報の単純伝達、交叉は情報の入れ換え、突然変異は情報の変化、そして淘汰／選択は情報の選択を意味する。これら4種類の基本的なパターンを繰り返す過程で、当初与えられた情報が様々な条件を満足するような理想的な情報へと変換されてゆく。これが遺伝的アルゴリズムによる最適化の基本である。

□ 遺伝的アルゴリズムの全体的特徴

遺伝的アルゴリズムは最適手法の一種であるが、この手法は従来の最適手法とは大きく異なる特徴を持っている。この特徴は以下に示す3点である。

- ① 固体（一点）を対象とするのではなく、集合体（多点）を対象とする。
- ② 固体全体を対象とするのではなく、固体の部分情報をも対象としている。
- ③ 目的関数の数式表現を特に必要としない。

この3つの特徴が従来のアプローチと大きく異なる点である。以下に順を追って説明する。

① この特徴は最適過程の探索手法に関係するものである。 遺伝的アルゴリズム以外の最適手法では1固体のみを観測対象とするために、ある出発点を基準としてその近傍の情報をみながら最適の部位を見出してくることになる。 従って、狭い場所からアンテナを伸ばして最適状態を探索することと同値であり、ローカルミニマの問題やパターンの最初の位置により最適解が異なるといった問題が常につきまとうことになる。

遺伝的アルゴリズムによる最適領域の探索は複数の固体を対象とする。 このため、最初に比較的広い範囲のなかから出発し、変換を繰り返すことで対象となる空間を狭めてゆく、いわば投網的な手法をとる。 従って、遺伝的アルゴリズムによる最適領域の探索では通常のアプローチで抱えるローカルミニマや初期座標といった問題を回避することが出来る。 この探索過程は従来のアプローチとは全く異なるもので、これが遺伝的アルゴリズムの最大の特徴である。

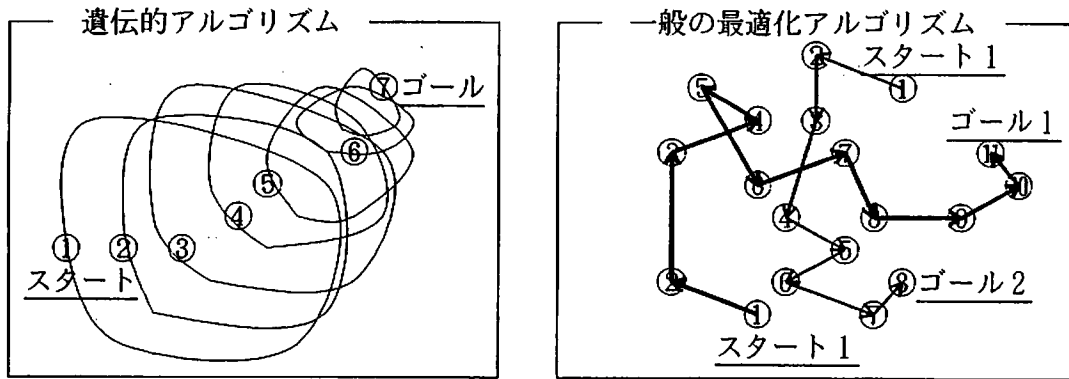


図3. 遺伝的アルゴリズムと一般の最適化アルゴリズムによる最適領域の探索過程

②この特徴は最適過程において、部分的特徴の保存が行われることを意味している。 従って、最適過程においては局所的ではあるが良質な情報というものが保持されて、相対的に良質な情報が順次増えてゆくことになる。 これにたいし、従来の手法は固体全体を視野にいれて最適化を行っていたために、部分的な情報というものは無視され、部分的には例えどんなに良質な情報をもっていても全体的な視点から必ず修正されてしまうといった問題があった。 この観点からも遺伝的アルゴリズムは従来の最適化手法には無い優れた能力をもっていることがわかる。

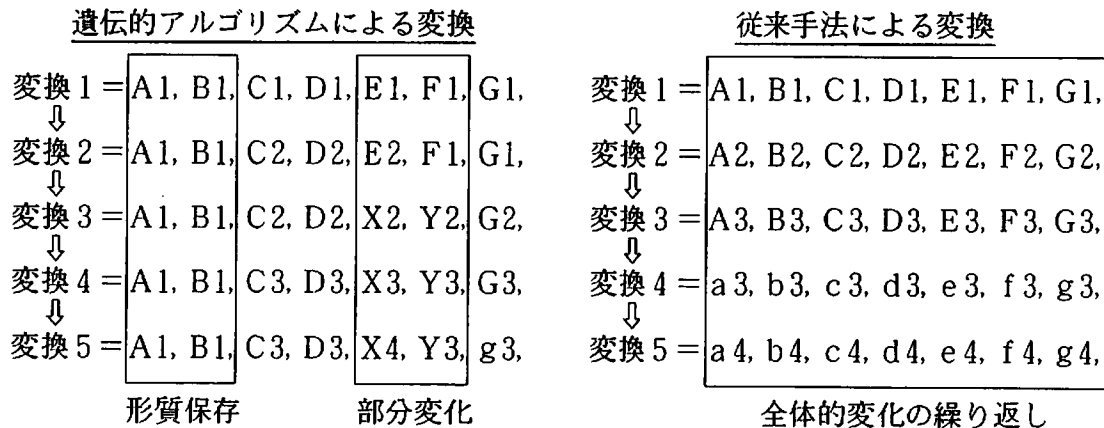


図4. 遺伝的アルゴリズムと従来手法による変換過程のイメージ図
 遺伝的アルゴリズムでは情報の部分的保存や部分変化等が行われ、従来手法と比べ柔軟な対応を可能としている。

③この特徴は、従来の最適化手法が最適化の為に何らかの形で目的関数を数式で表現することが必要であったのにたいし、遺伝的アルゴリズムではこのような関数表現をしなくとも最適化可能であることに関する。

最適化問題における一つの大きな山としてこの関数表現があった。 この関数表現が最適化を行おうとする解析目的とズレることがあれば当然のことながら良い最適解を導き出すことは不可能である。

最急降下法
 シンプルレックス法
 ダイナミックプログラミング
 ニューラルネットワーク
 遺伝的アルゴリズム

偏微分式
 応答関数
 関数再帰方程式
 エネルギー関数
 特に必要無し (評価/選択が可能な基準があればよい)

□ 遺伝的アルゴリズムの概要

ここでは話が分かりやすくなるように、遺伝子は0と1のバイナリコードで構成されているものとする。このバイナリコードが前記3種類の操作をうけることで徐々にへんかされ、問題となる条件をみたした理想的なコードへと変換されてゆく。以下では、前記4種類のパターンに従ってそれぞれの操作の具体的な手順を順番に説明してゆく。

1. 増殖 (MULTIPLICATION)

2. 交叉 (CROSSING-OVER)

情報の取替により、情報の変換を果たすものである。

```

0110 00101 1011001001011  遺伝子A
10011100110000 11001001011  新遺伝子BA
0110 00101 101101110100011  新遺伝子AB
10011100110000101110100011  遺伝子B
  
```

異なる遺伝子パターンをもつ2本の遺伝子がある時、これら2本の遺伝子を交叉させる。この結果、新たに2本の遺伝子が誕生するがこれらの遺伝子は親となるAおよびBの遺伝子の形質を交叉させた点を中心としてそれぞれあわせ持つことがわかる。

3. 突然変異 (MUTATION)

この突然変異では遺伝子の持つ情報がある部位で変化して対立遺伝子となる。

```

10110100101 10110010010111  遺伝子A
      ↓ 突然変異
10110100101 10110010010111  遺伝子A'
  
```

4. 淘汰/選択 (SELECTION)

様々な変換パターンにより構築された遺伝子が淘汰により悪い形質 (問題解決に取って望ましくない) を持つものが取り除かれてゆくことである。生物学的にはまわりの環境に適合した形質を持つ固体だけが生き残り (選択され)、次世代へと情報 (形質) を伝えてゆくことを意味する。

101	111	遺伝子A	—————>	×
010	110	遺伝子B	—————>	010 110 遺伝子B
001	001	遺伝子C	—————>	×
.....	—————>
100	011	遺伝子Z	—————>	×

遺伝的アルゴリズムには、ここで紹介されたオペレータの他にも様々なオペレータが存在する。例えば、DNAのコピーに関するオペレータとして挿入、欠失、転移といったものがあり、そのケースバイケースにより使い分けされている。

□ 遺伝的アルゴリズムの実際

文字列の集合をスキーマ (SCHEMATA) と称する。スキーマはアルファベットの集合であるが、これに*記号 (0 or 1) を加えて表現される。
 S = *11*01*1の時、この集合は

{01100101, 01100111, 01110101, 01110111, 11100101, 11100111, 11110101, 11110111} から構成される。

・次数 (ORDER)

$$O = (\text{全長 } L) - (\text{ドント・ケア記号の数})$$

例) $O(*11*01*1) = 8 - 3 = 5$

・構成長 (DEFINING LENGTH)

$$D =$$

□ 遺伝的アルゴリズムの実際の問題への適用

遺伝的アルゴリズムを自分の仕事に結び付けようとする時、以下のポイントをクリアすることが必要である。

- ① 問題をスキーマ形式にブレークダウンする。
- ② 複数の初期スキーマを作成する。
- ③ 淘汰/選択を行う時利用される適応度を求める手法を決める。

例えば、配座解析により安定化合物構造式を求めようとするならば、化合物構造式を一旦なんらかの手続きによりスキーマ形式に置き換える必要がある。

		歪エネルギー
配座異性体 1	90, 120, 30, 280, 300	40.5
配座異性体 2	30, 90, 170, 120, 70	29.1
.....
配座異性体 N	230, 320, 250, 20, 130	32.8

回転可能な結合軸の回転角度をそれぞれの遺伝子座 (この値は 0 / 1 に限らずどのような値でも構わない) とする。結合軸が 5 個ある時、構成長は 5 となる。一個の配座異性体ごとに一個のスキーマを対応させる。ここまでで、先のポイントの②までをクリアすることができた。

つづいて②のポイント、評価のための手段を確立することが必要である。ここでは配座解析であるので個々の配座異性体の歪エネルギーを計算しその値の大きさを比較することで選択を行うことが出来る。この歪エネルギーを指標とした選択により③のポイントもクリアすることが出来る。

□ 遺伝的アルゴリズム適用時のプロセス

先に遺伝的アルゴリズムを実際の問題に適用する為の条件について述べた。ここでは先の条件を満たした後、実際に遺伝的アルゴリズムを運用してゆく為の具体的な手続きについて述べる。

- ① 遺伝的アルゴリズムを行う前の初期条件として、初期染色体集合 $G(0)$ を作る。この初期染色体集合は遺伝的アルゴリズムの出発時の領域 (図 3 の①) を特定することになるので可能な限り広範囲なサンプルを集める事が必要である。この集合体の世代は 0 である。
- ② 淘汰/選択により、新たな染色体集合 $G(t)$ を作成する。この淘汰/選択による染色体の選択は適応度を求め、その比較を行うことで実施される。この淘汰/選択には様々なアプローチが取られているが、最も単純なものは適応度の順位に応じて上位のものから順番に選択したり、一世代前の染色体集合の平均よりも高いもののみを残したり、適応度の高いものはそのまま次世代に残すといった様々なアプローチをとることが可能である。自分の問題に最も適するものを適用すればよい。
- ③ 交叉や突然変異等の操作により染色体集合 $G(t+1)$ を新たに作成する。
- ④ 世代数が予め予定した世代に達したならばその時点で終了する。これまでの過程で得られた適応度最大の染色体を最終解として取り出す。世代が指定代に達していない時には②にもどり③及び④の過程を繰り返す。

① 初期染色体集合の作成

初期染色体集合		歪エネルギー (適応度)
配座異性体 1	90, 120, 30, 280, 300	40.5
配座異性体 2	30, 90, 170, 120, 70	29.1
.....
配座異性体 N	230, 320, 250, 20, 130	32.8

② 淘汰／選択の実行

初期染色体集合		歪エネルギー (適応度)
配座異性体 37	90, 120, 30, 280, 300	25.3
配座異性体 15	30, 90, 170, 120, 70	26.1
.....
配座異性体 53	230, 320, 250, 20, 130	165.5

③ 交叉 (同一世代の2染色体をランダム／ルールに従って選択)
(交叉点等も1点や複数点をランダム／ルールに従って選択)

初期染色体集合		歪エネルギー (適応度)
配座異性体 37	90, 120	25.3
配座異性体 15	30, 90	26.1
	↓ 交叉	
配座異性体 37	90, 120	27.2
配座異性体 15	30, 90	20.6

③ 突然変異 (各染色体中の1遺伝子座をランダム／ルールに従って選択)
(選択された遺伝子座の遺伝子を変化(置換やルール等)させる)

初期染色体集合		歪エネルギー (適応度)
配座異性体 37	90, 120, 30 , 280, 300	25.3
	↓ 突然変異	
配座異性体 37	90, 120, 30 , 280, 300	27.0

④ 世代数のチェック: $t \geq T$ (停止世代) の時、世代交代の終了。この時点までに得られた最高の染色体を選択して解答とする。

初期染色体集合		歪エネルギー (適応度)
配座異性体 172	60, 241, 167, 220, 287	13.5

④ 世代数のチェック: $t < T$ (停止世代) の時、再び②にもどり世代交代を繰り返す。

以上、配座解析に事例を取って遺伝的アルゴリズムの適応という観点から眺めてみた。

残念ながら実際にこの実験を試みたわけではないので訴える力に乏しいかもしれないが、遺伝的アルゴリズムが化学者の身近な問題にどのように適用されるかというイメージはつかんでいただけたと思う。また、遺伝的アルゴリズムの特徴（ローカルミニアや初期座標の影響を受けにくい）を考慮したならば、従来から行われてきた配座解析問題に一石を投じるものと考えている。私は遺伝的アルゴリズムは化学分野の研究に大いに利用出来るだろうと感じている。

7. パターン認識に関するその他の手法

パターン認識にはその他さまざまなアプローチが存在する。ここではアルゴリズムが簡単で、分類手法の一つとして頻繁に利用されている最近隣法（K-NN法）について述べる。

□ 最近隣法（K-NEAREST NEIGHBOR METHOD）の基本概念

最近隣法（以下K-NN法とする）の基本原理は「パターン空間中で近い位置にあるパターンは似た傾向（帰属クラスが同じ）を示す」というものである。これはK-NN法が、パターン空間上で同じ傾向を示すパターンは互いに近い位置にある（クラスターを形成している）という経験的な概念を利用した手法である事を示すものである。

□ 最近隣法の実行手順

パターン空間中におけるパターン間の距離を求め、クラス未知パターンXのクラス帰属は、そのパターンXに近いパターンが帰属しているクラスの情報に従って決定されるという手法である。このクラス決定は、ある定められた数（Kの数を意味し、1、3、5というような奇数を取る^{*)}のパターンをパターンXに近い順に取り出し、この取り出されたパターンの属するクラスを参照し、その中で最も多数を占めるクラスにパターンXのクラスを帰属させるものである。

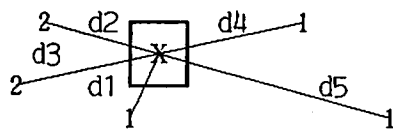
* 1. このクラス決定の時に、クラスパターン数が同数になるとクラス決定が不可能となるのでKの値は通常奇数をとる。

□ 最近隣法におけるパターン間の距離

K-NN法で必要となるパターン間の距離を求める為の距離尺度はユークリッド距離を用いて計算されるのが普通である。

$$D_{nm} = \left[\sum_{i=1}^d (X_{ni} - X_{mi})^2 \right]^{1/2} \quad (1)$$

D_{nm} はパターン X_n とパターン X_m との距離を示し、 d は次元数を示す。



パターン間の距離関係

$$d_1 < d_2 < d_3 < d_4 < d_5$$

	決定クラス	隣接パターン所属クラス				
		1	2	3	4	5
K = 1 の時	クラス 1	1				
K = 3 の時	クラス 2	1	2	2		
K = 5 の時	クラス 1	1	2	2	1	1

図 7-1 最近隣法によるクラス分類

□ K-NN法の特徴

K-NN法はクラス分類手法としては基本アルゴリズムが簡単で理解しやすく、多クラス分類（2クラス分類に用いる事が多い）が可能で、かつパターンの分布状態が非線型のものにも利用出来る（必ずしも線型で分離されるようなパターン空間の構造をしている必要がない）極めて汎用性の高い手法である。

通常K-NN法ではK=1の時が最も高い分類/予測率を示す事が経験的に知られている。特に、解析に用いる母集団のパターン数が多くない時はK=1が望ましい。Kの値を大きくすると、一般的に分類/予測率が低下してゆく傾向がある。この原因として幾つか考えられるが、以下に示す2つが主たる原因であろう。

① Kの値が大きくなると、クラス決定は常に最多数を占めるクラスに支配される事に

なる。従って、母集団パターンにおけるクラス毎のパターンの数に大きな差異がある時にはK-NN法のとり扱いは注意が必要である。

- ② Kが大きくなると、クラス決定が単なる多数決で決められている為、取り出されたK個のパターンのうち最も近いパターンと最も遠いパターンとのクラス決定に与える重みは同じであるのでK-NN法の基本定義から解離する事になる。この為クラス分類/予測率が減少するものと考えられる。

□ K-NN法の問題点

K-NN法の特徴に起因する欠点というものが存在する。これらを以下に簡単にまとめる。実際の適用では、これらの長所と欠点をよく理解した上で解析を行う必要がある。

- ① 計算にすべてのパターン間の距離を計算しなければならないので、パターンが増えると計算すべき距離の数が急激に増大し、計算時間がかかる。
- ② クラス決定が常に他のパターンの存在に支配される。
- ③ クラス決定が単純な隣接関係だけで行われ、隣接パターンとの距離関係の考慮がされない。すなわち同じ隣接関係にある限り、近い位置にあるパターンと遠い位置にあるパターンとの区別がされずにクラス決定が行われる。

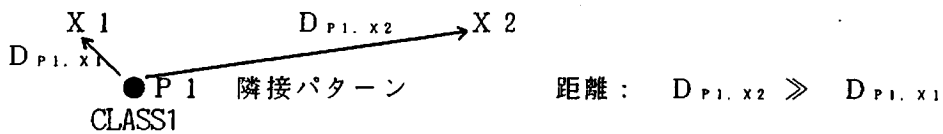


図7-2 K-NN (K=1) 法によるX1及びX2パターンの分類

6. 3 ニューラルネットワークによる最適化

□ ニューラルネットワークを用いた最適化

ニューラルネットワークを用いて最適化問題を解くことが可能である。第2世代のニューラルネットワークが注目され新時代を形成したのは、この最適化問題として古くから良く知られたトラベリングセールスマン問題にニューラルネットワークを適用し、従来のアプローチと比べ良好な結果を得たからである。

本手法によるアプローチはニューラルネットワークの普及に伴い、今後急速に普及してゆくものと期待される。

□ ネットワーク構造

最適化問題を解くのに用いられるニューラルネットワークは、相互結合型（自己学習型ともいう）と呼ばれるネットワーク構造が閉じた構造をもつものである。このような閉じたネットワーク構造をもつものとしては、ホップフィールド型のニューラルネットワークとボルツマンマシンがある（図1）。

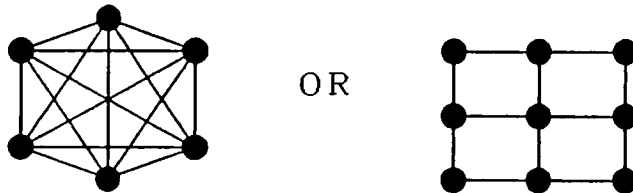
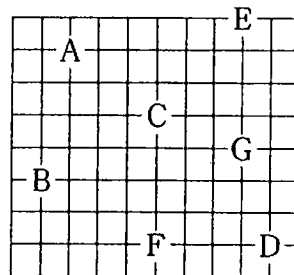


図1. ボルツマンマシン及びホップフィールドネットのネットワーク構造

このような閉じたネットワーク構造を持つニューラルネットワークの特徴は、学習において教師データを必要としない自己学習を行うということである。これに対し、第章で述べた階層型のネットワーク構造を持つバックプロパゲーションでは学習に教師データが必要不可欠である。

相互結合型ネットワークにおける自己学習はなにも指標がなく適当に行うというのではなく、内部で定められた条件（例えばホップフィールドネットで用いられるエネルギー関数等）を満たすという一つの目標があり、この条件を満たすように学習が行われる。この学習の条件を最適化のための条件とうまくリンクするように工夫すると、学習の結果得られたネットワークは与えられた最適化問題の解を与えるものとなる。

□ ホップフィールドネットのトラベリングセールスマン問題への適用例



A～Gはセールスマンが立ち寄る都市であり、枠目上の地点が個々の都市の場所と相互距離を示している。

図2. トラベリングセールスマン問題

この問題は昔から極小/最小値を求める問題として有名なものであった。つまり、セールスマンがある都市を出発点とし、指定された各都市を必ず一回たどり、この経路が最短距離となるようにして出発点にもどるという問題である。

この歴史的な問題には様々な手法をもってアプローチがなされてきたが、なかなか決定的な良い方法が見つからなかった。この問題に対しホップフィールドらは自らが考案したネットワークを持つニューラルネットワークに適用し、従来手法よりも格段に優れた（計算機時間が少なく済む）結果を導き出した。

この問題を解決するにあたり、トラベリングセールスマン問題をネットワーク構造に対応させる事が必要となる。ホップフィールドらは各都市とその都市が何番目にたどるかという事をあらわしたマトリックス（訪問した時は1、しない時は0）（表1）を作成し、このマトリックスを構成する要素値にそれぞれのニューロンを対応させている。

表1. トラベリングセールスマン問題におけるホップフィールドネットの応用

		訪 問 順 番						
都 市		1	2	3	4	5	6	7
A		0	0	0	///	0	0	0
B		0	0	0	0	///	0	0
C		0	0	///	0	0	0	0
D		0	0	0	0	0	0	///
E		0	///	0	0	0	0	0
F		0	0	0	0	0	///	0
G		///	0	0	0	0	0	0

表1の行は訪問すべき都市を、カラムは都市の訪問順を示している。各行中、1の値を持つカラムの番号がその行の都市を訪問すべき順番を示している。

この問題を解くにあたり、幾つかの拘束条件が設定されている。

第1拘束条件： 一度訪れた都市は2度と訪れない（即ち、一つの行のなかでは訪れた事を意味する1は一個しかない）。

第2拘束条件： 2つ以上の都市を同時に訪問する事は出来ない（即ち、一つの列内に二つ以上の1は立たない）。

第3拘束条件： 都市間の距離の情報はネットワークの強度として表現する。

第1の拘束条件は、1の立っている（発火状態）ニューロンは同じ行にあるその他のニューロンと抑制関係にあるように結合させ、抑制されているニューロンからも発火状態にあるニューロンを抑制するように結合させ、相互抑制回路を組むようにさせる事で実現される。この制限条件は、一つの都市が1となったならば、その都市に該当する残りの行のユニットにつながるウエイトベクトルは総て0となる事で実現されている。

第2の拘束条件は、同時に複数の都市を訪問する事は不可能である事を示している。この制限条件により、同列内に複数の1は立ちえない事になる。この条件も第1の拘束条件と同様に、ある順番での訪問都市が決定されたならば同じ列内の他のユニットに繋がるウエイトベクトルは総て0となる事になる。

第3の拘束条件は都市間の距離情報がネットワーク強度という形でネットワークの中に埋めこまれる事を意味する。

この制限条件は先のエネルギー関数（第2式）のウエイトベクトル W_{ij} の代わりに都市（ i 及び j ）間の距離 d_{ij} で置き換える事で実現される。

$$E = -\frac{1}{2} \sum_i \sum_j d_{ij} U_i U_j + \sum_i \theta_i U_i \quad (5)$$

これらの拘束条件下、ホップフィールドネットを実行するとエネルギー関数値 E が減少して平衡状態になる。この平衡状態にある時の発火ニューロンの並びを検討する事で訪問すべき都市の順番が判明する。

表1では解析の結果、都市の訪問順はG E C A B F Dである事がわかる。

文献

Hopfield, J. J., and Tank, D. W. : Neural computation of decisions in optimization problems. Biol. Cybern., Vol. 52 : 141~152, 1985.

4. 2. ファジイ理論の利用

□ ファジイ (Fuzzy) とは？

ファジイ (Fuzzy) とは事象の曖昧さを扱う理論である。

例 1)

最近、東京近郊では気温の高~~い~~日が続~~いて~~いるが、湿度の低~~い~~日が多~~い~~ので比較的過~~ぎ~~し~~な~~い。

例 2)

抗腫瘍活性の高~~い~~化合物中、分子量が大~~き~~く且つLOGP値の低~~い~~ものは毒性が少~~な~~く好~~ま~~し~~い~~が、合成~~し~~に~~難~~し~~い~~、分解~~し~~易~~い~~化合物が多~~い~~。

□ ファジイによるエキスパートシステム検討中及び実施中のシステム

・ファジイによる地下鉄自動運転システム

(知識例)

次の駅が近~~く~~なったら、始めは強~~く~~ブレーキをかけ、駅の手前で一旦緩~~め~~て、停止位置の近~~く~~で再び少~~し~~強~~く~~ブレーキをかける。

・ファジイによる犯人しぼりこみシステム

(知識例)

目付きが鋭~~い~~、鼻の高~~い~~、丸顔で 疲~~れ~~た感~~の~~、串肉串背の西洋太風の若~~い~~男性

□ 曖昧の種類について

① 確立的曖昧さ (Randomness) 確立

一度事象が発生すれば、曖昧さはなくなる。

例) さいころ、宝くじ、その他

② 概念の曖昧さ (Fuzziness) ファジイ

本質的に曖昧なもの

例) 美味しい、美人、中年、大量、甘い、その他

③ 知識/情報の不十分さに起因する曖昧さ (Incomplete)

本質的には曖昧さは存在しないが、単に情報不足から正確な結論を出す事が出来ない、または誤差が多い事。

例) 天気予報、地震予知、その他

④ 解釈が何通りもあって曖昧である (Ambiguity)

一元多項対応の為に曖昧となっているものである。

例) 分子式等 $C_8H_{12}O_2$ 、 $C_{10}H_{18}N_2O$

⑤ 正確でない事に起因する曖昧さ (Imprecision)

ノイズや誤りが含まれている為、本質が不明となり曖昧なもの。

例) 不純物を多く含んだ化合物のスペクトル

曖昧にはこの他にも様々な種類がある。ここにあげた例はその中の一例にしかすぎ

ない。これら数多くの曖昧さのうち、ファジイ理論で扱う事の出来る曖昧さとは②の概念の曖昧さ (Fuzziness) である。

P U Z Z L E :

① 以下の文章中から曖昧な表現あるいは曖昧な情報を含む事象をを取り出し、その曖昧の種類に従って分類せよ。

- 明日の天気は晴れそうだから、何処かにピクニックにでもいこうか？
- 「はし」というのは何の事ですか？
- 君の父さんは中年というよりは実年かな？
- 明日の競馬で大当たりしたらヨーロッパ旅行でもしよう。

② だまし絵に含まれている曖昧性はどのような曖昧性か？

R A N D O M N E S S F U Z Z I N E S S I N C O M P L E T E A M B I G U I T Y I M P R E C I S I O N

□ 曖昧さの数学的取扱

ファジイには大きな基本として「ファジイ集合」と「ファジイ関係」とがある。

- ① 「ファジイ集合」は言葉の意味／概念に含まれている曖昧さを定量的に扱う為の集合概念であり、ファジイ理論の基本である。
- ② 「ファジイ関係」とは物と物との曖昧な関係をファジイ集合を基本として展開されたものである。従って、2つの要因に関する相互関係 (AとBはかなり似ている、AとBはほぼ等しい、AはBより大きい、AはBと少し離れている、その他の関係) をファジイ理論を基本として論じるもので、マトリックスの展開が主体となる。

この2つの基本的概念を用いて様々な展開が試みられている。以下、順番に説明する。

□ ファジイ集合

全体集合をXとする。また、この全体集合Xの要素xが部分集合Aに含まれる度合いを μ で示すと、この μ は以下の式で示される。

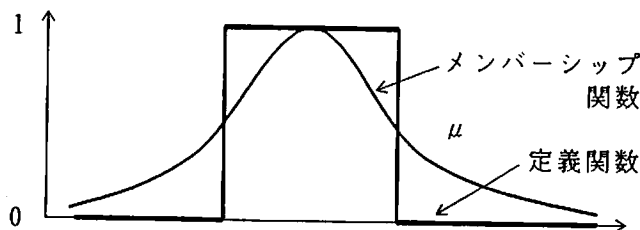
$$\mu_A : X \rightarrow [0, 1]$$

この時、 μ は部分集合Aのメンバーシップ関数という。

*但し、 $[0, 1]$ は0～1までの実数区間の集合を意味する。この0と1の意味は、～らしさが100%の時に1 (全体集合Xの要素xが部分集合Aに含まれる時を意味する) であり、～らしさが0%の時に0を (即ち要素xが部分集合Aに含まれない時を意味する) とするもので、この0と1の間の値は～らしさの程度を表す連続量となっている。この0～1までの値はグレード又は適合度と呼ばれている。

図1にメンバーシップ関数と定義関数とが示されている。

* $\{0, 1\}$ は従来の集合を意味し、これは0と1のみで構成される集合であり、定義関数或いはクリस्प関数と呼ばれる。



定義関数は0と1の値で構成されている。このように連続性の無い関数ステップ関数、或いはクリस्प関数という。

図1. メンバーシップ関数と定義関数

□ ファジイ集合AとBの集合の関係について簡単にまとめる

ファジイ集合の演算は先に述べたメンバーシップ関数を基本として定義されている。以下にその定義中、和集合、共通集合、及び補集合の定義を示す。

和集合 $A \cup B$: $\mu_{A \cup B}(X) = \max\{\mu_A(X), \mu_B(X)\}$
 $= \mu_A(X) \vee \mu_B(X)$

共通集合 $A \cap B$: $\mu_{A \cap B}(X) = \min\{\mu_A(X), \mu_B(X)\}$
 $= \mu_A(X) \wedge \mu_B(X)$

補集合 \bar{A} : $\mu_{\bar{A}}(X) = 1 - \mu_A(X)$

* $\max(a, b)$ や $\min(a, b)$ はそれぞれ a と b の値の内、大きい値と小さい値をとる事を示す。これは略記方により $a \vee b$ 及び $a \wedge b$ で表記される。

□ メンバーシップ関数による和集合、共通集合の意味

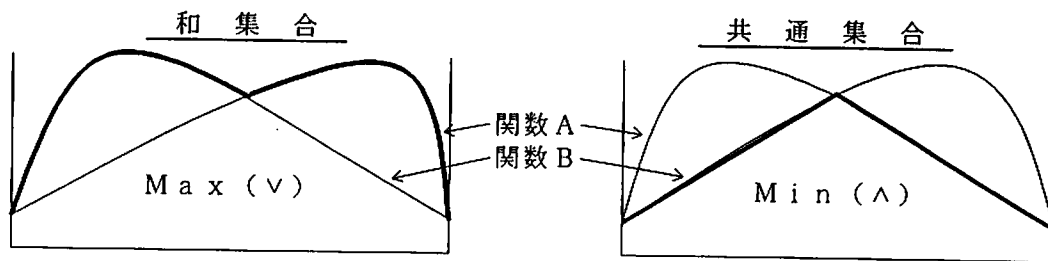


図2. メンバーシップ関数による和集合及び共通集合

図2にメンバーシップ関数を用いた和集合及び共通集合の概念図がしめされている。この図からもわかるように和集合では関数AとBの値の中大きい値を取り、共通集合では関数AとBの値で小さいものを採用する事がわかる。

□ ファジイ集合の代数的演算式

二つの集合の演算式を以下に示す。ファジイ演算では代数積、代数和の他に限界積、限界和及び限界差が存在する。いずれの演算式においても値が1を越えないようになっている事が特徴である。

代数積	$A \cdot B$	$\mu_{A \cdot B}(X) = \mu_A(X) \mu_B(X)$
代数和	$A + B$	$\mu_{A + B}(X) = \mu_A(X) + \mu_B(X) - \mu_A(X) \mu_B(X)$
限界積	$A \cdot B$	$\mu_{A \cdot B}(X) = \max(0, \mu_A(X) + \mu_B(X) - 1)$ $= 0 \vee (\mu_A(X) + \mu_B(X) - 1)$
限界和	$A \cdot B$	$\mu_{A \cdot B}(X) = \min(1, \mu_A(X) + \mu_B(X))$ $= 1 \wedge (\mu_A(X) + \mu_B(X))$
限界差	$A \cdot B$	$\mu_{A \cdot B}(X) = \max(0, \mu_A(X) - \mu_B(X))$ $= 0 \vee (\mu_A(X) - \mu_B(X))$
激烈積	$A \cdot B$	$\mu_{A \cdot B}(X) = \mu_A(X)$ if $\mu_B(X) = 1$ $= \mu_B(X)$ if $\mu_A(X) = 1$ $= 0$ if $\mu_A(X) \text{ AND } \mu_B(X) \neq 1$
激烈和	$A \cdot B$	$\mu_{A \cdot B}(X) = \mu_A(X)$ if $\mu_B(X) = 0$ $= \mu_B(X)$ if $\mu_A(X) = 0$ $= 1$ if $\mu_A(X) \text{ AND } \mu_B(X) \neq 0$

□ ファジイ集合の計算例 1

前節の結果からもわかるように、ファジイ集合間の演算は極めて簡単である。基本的には二つの値の比較の問題に帰結されてしまう。つまり、 $\max(\vee)$ は大きい値を取り、 $\min(\wedge)$ は小さい値をとるだけである。

例) $A = 0.6 + 0.2 + 0.7 + 0.9 + 0.3 + 0.5$
 $B = 0.3 + 0.6 + 0.2 + 0.8 + 0.5 + 0.7$ の時、

$$A \cup B = (0.6 \vee 0.3) + (0.2 \vee 0.6) + (0.7 \vee 0.2) + (0.9 \vee 0.8) + (0.3 \vee 0.5) + (0.5 \vee 0.7)$$

$$\begin{aligned}
&= 0.6 + 0.6 + 0.7 + 0.9 + 0.5 + 0.7 \\
A \cap B &= (0.6 \wedge 0.3) + (0.2 \wedge 0.6) + (0.7 \wedge 0.2) + (0.9 \wedge 0.8) + (0.3 \wedge 0.5) + (0.5 \wedge 0.7) \\
&= 0.3 + 0.2 + 0.2 + 0.8 + 0.3 + 0.5 \\
A \cup B &= 0.18 + 0.12 + 0.14 + 0.72 + 0.15 + 0.35 \\
A + B &= 0.72 + 0.68 + 0.76 + 0.78 + 0.65 + 0.85 \\
A \cdot B &= 0 + 0 + 0 + 0.5 + 0 + 0.2 \\
A \cdot B &= 0.9 + 0.8 + 0.9 + 1 + 0.8 + 1 \\
A \cdot B &= 0.3 + 0 + 0.5 + 0.3 + 0 + 0 \\
A \cdot B &= 0 + 0 + 0 + 0 + 0 + 0 \\
A \cdot B &= 1 + 1 + 1 + 1 + 1 + 1
\end{aligned}$$

□ メンバシップ関数 (曖昧さを的確に表現出来るように自由に定義する)

メンバシップ関数はファジイを実用的な問題に適用する時に最も重要となる関数である。この関数は全体集合中の要素 x がある領域に含まれている度をあらわすものであり、統計や確立で厳密に支配されている関数ではない。この関数は、作業目的や経験に応じ、人が自由に最適なものを設定する事ができる。確立や統計と異なり、解析に必要な関数を自由に設定出来る点がファジイの特徴の一つである。この結果、メンバシップ関数は作業目的のみならず、関数を設定する人の個人差によっても異なってくる。

例えば、車の速度を速いと規定する時、ある人は速い速度は 40 K 以上と感じ、別の人は 60 K 以上と感じるというように、個人により異なる事はファジイにとり自然な事である。

□ ファジイ関係 (Fuzzy relation)

ファジイ関係とはファジイ集合 A 及び B の各要素が互いに何らかの相関関係が存在する時、その相関関係を示す。この時、 A と B とが m 及び n 個の要素からなる時、この関係は m 行 n 列のマトリックス (ファジイ関係行列 Fuzzy relational matrix) R として示される。このファジイ関係における演算も基本的にはファジイ集合の演算と同じパターンで行われる。

ファジイ RUS の和 : $\mu(X, Y) = \max\{\mu(X, Y), \mu(X, Y)\}$
 $= \mu(X, Y) \vee \mu(X, Y)$

ファジイ R ∩ S の交わり : $\mu(X, Y) = \min\{\mu(X, Y), \mu(X, Y)\}$
 $= \mu(X, Y) \wedge \mu(X, Y)$

補ファジイ関係 \bar{A} : $\mu(X, Y) = 1 - \mu(X, Y)$

□ ファジイ合成 (Fuzzy composition)

R S : $\mu(X) = \max \min\{\mu(X), \mu(X)\}$
 $= \vee\{\mu(X) \wedge \mu(X)\}$

□ ファジイ関係の計算例 2

基本的に前のファジイ集合の演算と全く同じである。ただ、扱うのがファジイ関係マトリックスであるので、マトリックス間の演算が必要というだけである。

例)

$$R = \begin{bmatrix} 0.6 & 0.3 & 0.8 \\ 0.3 & 0.7 & 0.5 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.2 & 0.5 \\ 0.8 & 0.3 \\ 0.1 & 0.7 \end{bmatrix}$$

RUS =

$$\begin{bmatrix} (0.6 \vee 0.2) & (0.3 \vee 0.8) & (0.8 \vee 0.1), & (0.6 \vee 0.5) & (0.3 \vee 0.3) & (0.8 \vee 0.7) \\ (0.3 \vee 0.2) & (0.7 \vee 0.8) & (0.5 \vee 0.1), & (0.3 \vee 0.5) & (0.7 \vee 0.3) & (0.5 \vee 0.7) \end{bmatrix}$$

$$\begin{bmatrix} (0.6) & (0.8) & (0.8), & (0.6) & (0.3) & (0.8) \\ (0.3) & (0.8) & (0.5), & (0.5) & (0.7) & (0.7) \end{bmatrix}$$

R S =

$$\begin{bmatrix} (0.6 \wedge 0.2) \vee (0.3 \wedge 0.8) \vee (0.8 \wedge 0.1), & (0.6 \wedge 0.5) \vee (0.3 \wedge 0.3) \vee (0.8 \wedge 0.7) \\ (0.3 \wedge 0.2) \vee (0.7 \wedge 0.8) \vee (0.5 \wedge 0.1), & (0.3 \wedge 0.5) \vee (0.7 \wedge 0.3) \vee (0.5 \wedge 0.7) \end{bmatrix}$$

$$\begin{bmatrix} (0.2) \vee (0.3) \vee (0.1), & (0.5) \vee (0.3) \vee (0.7) \\ (0.2) \vee (0.7) \vee (0.1), & (0.3) \vee (0.3) \vee (0.5) \end{bmatrix}$$

$$\begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.5 \end{bmatrix}$$

QUIZ

設問： 以下の計算をせよ。

$$R = \begin{bmatrix} 0.3 & 0.5 & 0.4 \\ 0.5 & 0.7 & 0.2 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.2 & 0.5 \\ 0.5 & 0.6 \\ 0.1 & 0.3 \end{bmatrix}$$

の時、

- ① $R \cap S$
- ② $R \cup S$

従来手法及びファジイ理論に基づいた分類の差異について

曖昧概念に基づいた分類問題では、従来手法による分類とファジイ理論に基づいた分類とは大きく考え方が異なってくる。この基本的な考えの差を理解する為、以下に体重に基づいた「瘦身」、「中肉中背」、「肥満」という概念を従来手法のクリスプ的な考えによる分類とFuzzy理論的考え方に基づいた分類とで試みる。

従来のアプローチによる分類

従来のアプローチでは、ファジイの概念が無い為、分類はある点(重さ)を境として急激に変化するクリスプな事象を基本として実行される(図3)

分類基準) 瘦身 ~ 40Kg未満

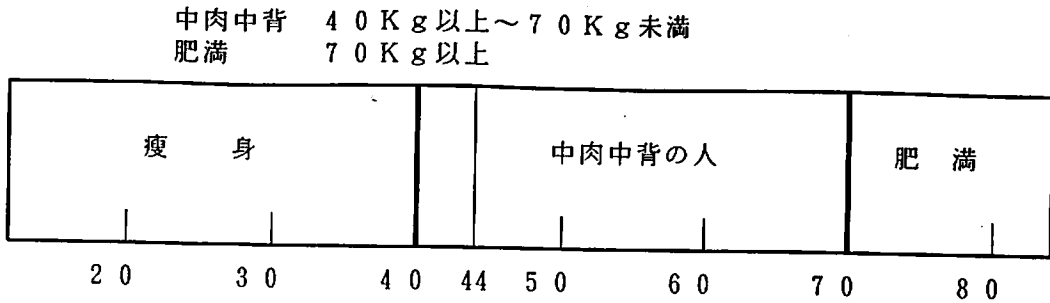


図3. 従来のクリスプな事象を基本とした分類

従来の（クリスプ的）考え方による分類の特徴

前記3タイプの分類は厳格に分離されており共存する事がない。ここでは44Kgの人は中肉中背であり、それ以外の要素は存在しない。（クリスプ集合）

この分類にはいくつかの際立った矛盾が存在する。

- ① 境界ゾーンは本来曖昧なものであるが、むりに境界を設けて分類している。
- ② 境界が存在する為、同じクラスに分類された人にも大きな差異が存在する。
例) 40Kgと69Kgの人とで29Kg離れていても、全く同じ中肉に分類される。一方で、1Kgしか離れていない39Kgの人はクラスが異なっている。
- ③ 境界が存在する為、本来は変化が緩やかであるべき事象が急激な変化を伴う事象に変えられている。
例) 39Kgの人は痩せ、40Kgの人は中肉のように少しの変化で、分類は極端に変化する。

このように従来手法による分類では幾つかの矛盾を内包するものであったが、その矛盾は境界線の弾力的運用により補われてきた。つまり、人、立場、環境、ムード、等その時々条件により境界が変更され、また境界線の両脇に緩衝領域を設定して分類のグレーゾーンとして運用していた。従って、同じ人でも環境が変わると痩せから中肉へと変化した又その反対も存在しうる。また、言葉でその曖昧さをおぎなってもいた。「中年らしい」、「中年っぽい」、その他。

□ ファジイ概念導入による分類の基本的考え

同じ事象を対象とした分類であっても、Fuzzyに基づいた分類では様子が異なってくる。分類は従来手法のように境界線を基準として行われるのではなく、それぞれのクラス毎に設定されたメンバーシップ関数の値の大きさを比較する事で実行される。同じ分類を3種類のメンバーシップ関数で表現した時の様子が図4に示されている。

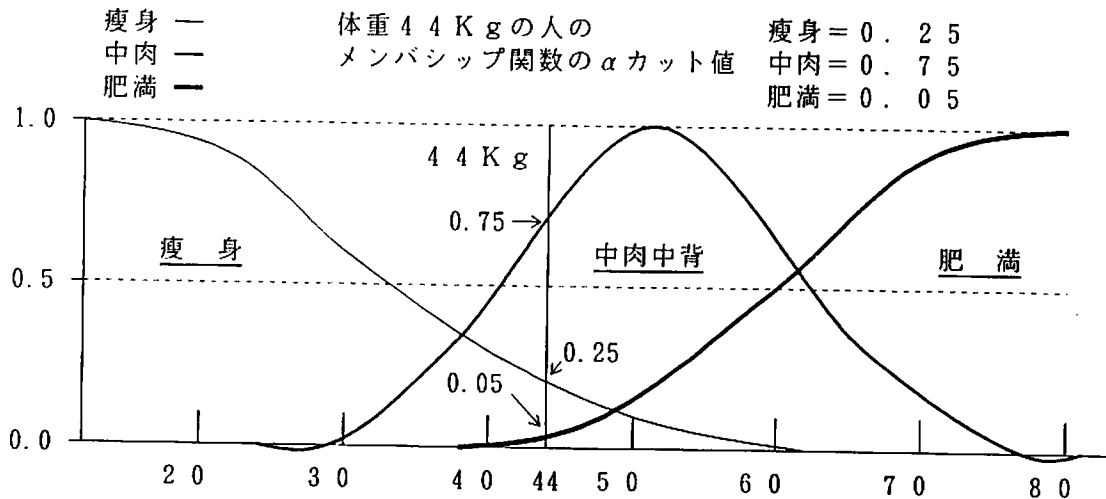


図4. 体形に関する3種のメンバーシップ関数

ファジイ的考え方による分類の特徴

ファジイ的事象の基本は連続的な変化であり、クリस्प関数的な急激な変化を伴わない事である。ここでは図4の3種のメンバーシップ関数を基本として、前節と同様に体重が44Kgの人の分類を例に取ってのべる。

この人の α カット値(グレード; 体重とメンバーシップ関数との交点)は瘦身0.25、中肉中背0.75、肥満0.05となっている。これらの値はそれぞれのクラスに対する所属度とよばれている。つまり、44Kgの人は中肉中背クラスに対する所属度が0.75で最も高いが、然し瘦身クラスの所属度も0.25有り、肥満の程度も少しだが0.05であると解釈する。

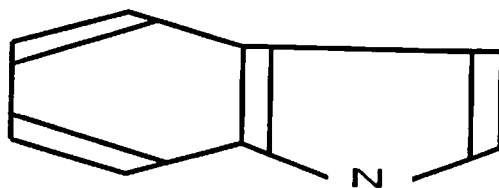
即ち、ファジイ的事象とは強制的に一つの事象に帰属されるものではなく、常に複数の事象の特性を兼ね備えており(多面性を持つ)、その複数の特性の帰属度を総合的に判断する事で様々な展開に導く事が出来る事を特徴とするものである。ファジイを理解する上では、この考え方を素直に受け入れる事が必要である。

従来の事象の捕らえ方	事象は常に一つのカテゴリーに帰属し、複数のカテゴリーに同時に帰属する事はない
ファジイ的事象の捕らえ方	事象は複数のカテゴリーに帰属する事が自然であり、多面性が基本である

丁度、この定義(クリस्प)関数とメンバーシップ関数の問題は、古典物理で光が波動か粒子かという問題で世界中の物理学者を悩ませていた時、「光とは波動と粒子の両方の特性を兼ね備える」という、古典物理から見れば矛盾の固まり的な発想転換を行う事で近代物理の幕が開けられた事を思い出させる。

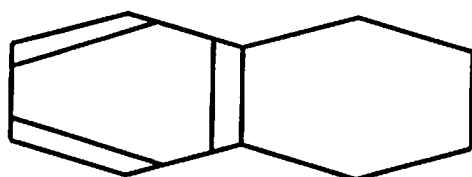
ファジイ的な考えを素直に受け入れるには、この種の発想転換が必要と思われる。

標準化合物

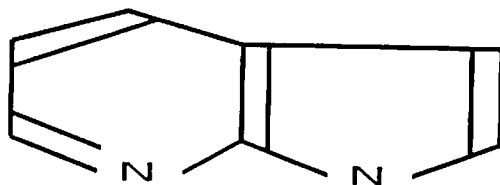


比較化合物

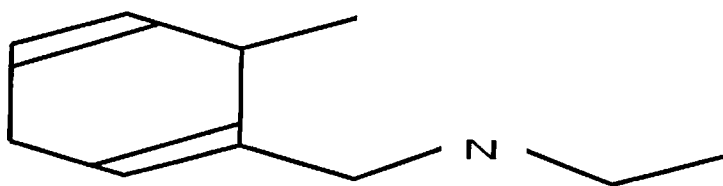
1



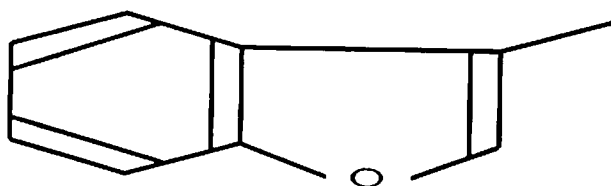
2



3

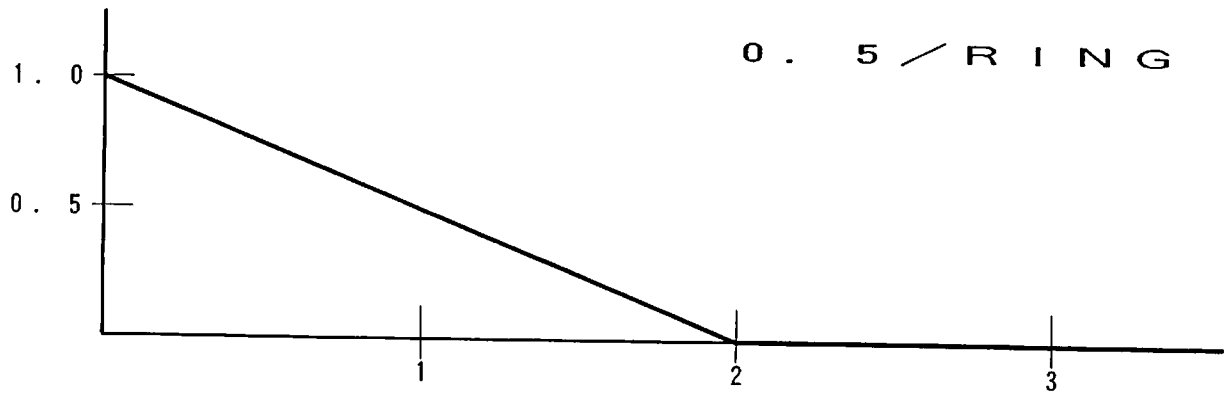


4

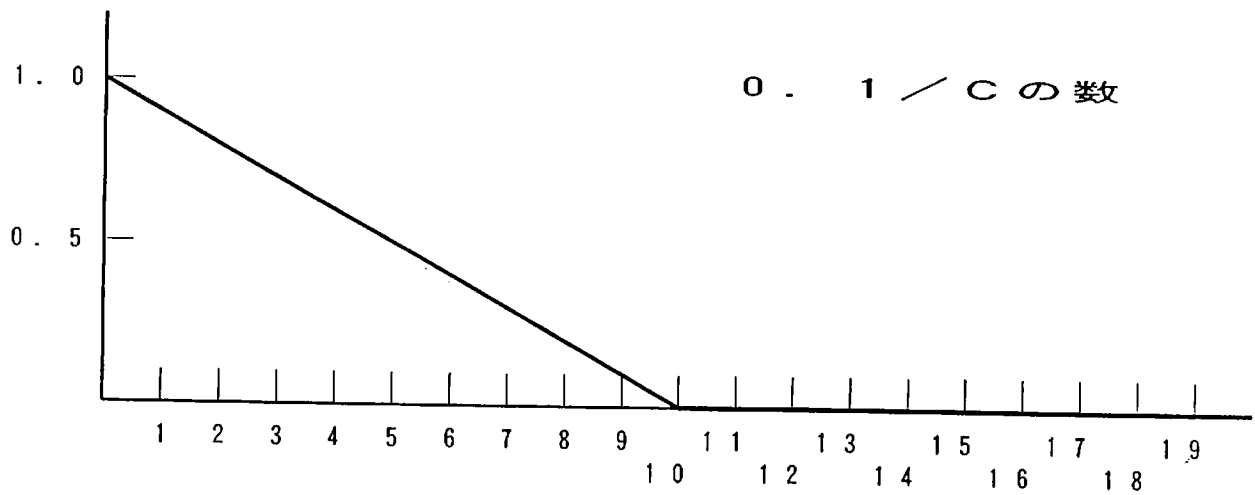


使用メンバシップ関数について

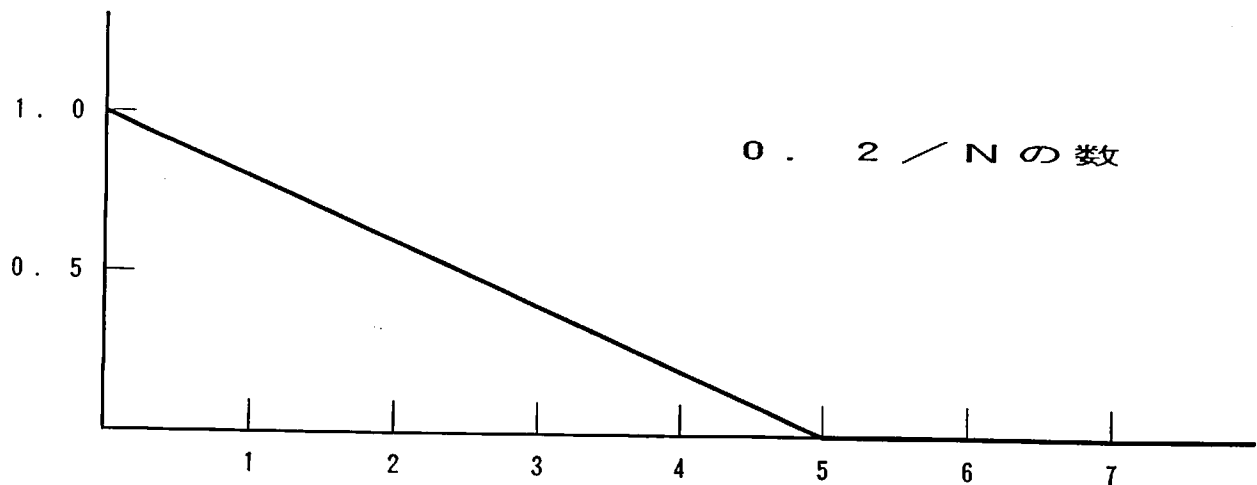
R I N Gの数に関するメンバシップ関数



Cの数に関するメンバシップ関数



Nの数に関するメンバシップ関数



類似化合物選定における従来法との比較

(1) 類似関数の利用による類似化合物選択

- 関数の適用範囲が狭い場合が多い
- 類似度を調べるのに用いられる情報量は少ない (パラメータ数が少ない)

(2) パターン認識法による類似化合物選択

- 関数の適用範囲は (1) と比べると広い、幅広い化合物に適用可能である
- 類似度を調べるのに用いられる情報量は多い (数多くのパラメータを使用出来る)
- 解は用いたパラメータにより一義的に決まる
- 計算量が多くなる

(3) ダイナミックプログラミングによる類似化合物選択

- 関数の適用範囲は (1) と比べると広い、幅広い化合物に適用可能である
- 類似度を調べるのに用いられる情報量は多い (数多くのパラメータを使用出来る)
- 解は用いたパラメータにより一義的に決まる
- 計算量が多くなる

(4) ファジイ理論による類似化合物選択

- 関数の適用範囲は (1) と比べると広い、幅広い化合物に適用可能である
- 類似度を調べるのに用いられる情報量は多い (数多くのパラメータを使用出来る)
- メンバシップ関数の設定が容易なので解は状況に応じて自由に変えられる (使用する人間の感覚にフィットした結果を得るように調節出来る)

□ 他の解析技術とファジイとの関係

ファジイ理論は様々な分野における基礎理論となり得る。従って、他の分野の技術にファジイ理論を導入する事も多く、寧ろ従来技術にファジイ理論を積極的に導入し、従来問題がかかえていた様々な問題点を解決する試みが数多く行われている。ファジイが従来技術に導入され、最も大きな成果を上げている分野としてはファジイ推論を基本とした制御工学分野があるが、この分野の解説は他書にゆずる。ここでは、パターン認識や多変量解析とファジイとの関係にポイントをおいて述べてゆく。

□ ファジイとパターン認識／多変量解析との関係

ファジイがパターン認識や多変量解析に導入される時、従来の手法と大きく異なってくるのは入力／出力データである事が多い。従来の手法ではこの入／出力データには唯一個だけの値が対応しており、複数の値が対応するという例はなかった。このような考えに対し、ファジイを導入すると1対多対応という考え方が必要となってくる。これは従来の解析の考えに慣れていれば奇異な感じがするが、本質的な問題から考えるならばこの方が自然である。クラス分類、クラスタリング、線形重回帰等様々な解析手法が存在するが、解析結果で総てのパターンが理想的（完全に一つのクラスに帰属出来る、あるいは回帰線上に完全に乗る等）なパターンである事は少ない。寧ろ、かなりのパターンは一つのクラスへの帰属が困難である事が多く、特に境界領域付近のパターンの取扱は1対多対応を考慮した解析がより自然である。

このような観点から考えた時、多変量解析にファジイを導入する事で、より自然な考えに従った解析が可能となる。特に、従来手法では誤分類等の原因となるパターンに関する配慮が可能となり、より高い信頼性を獲得する事が可能となる。

□ 解析結果にファジイネスを導入する

顔チャート／レーダーチャートの中に潜むファジイネス

顔チャートやレーダーチャートの最終的な解析は人間が行う。これらチャートの作成段階にはファジイの入り込む余地はないが、最終的に得られたチャートを判定する段階でファジイネスが入ってくる。これはチャートの作成では何ら定まった解をあたえる物でなく、解は結果を見て人間が判断することから、この判断過程ではいつてくる為である。

ここでは顔チャートを例に取って考える。

例) 顔チャートによる薬の効果判定：

表情		ファジイネスによる表情の程度
良く効く薬	笑っている	本気で笑う／大きく笑う／少し笑う
効かない薬	怒っている	本気で怒る／強く怒る／怒る／少し怒る

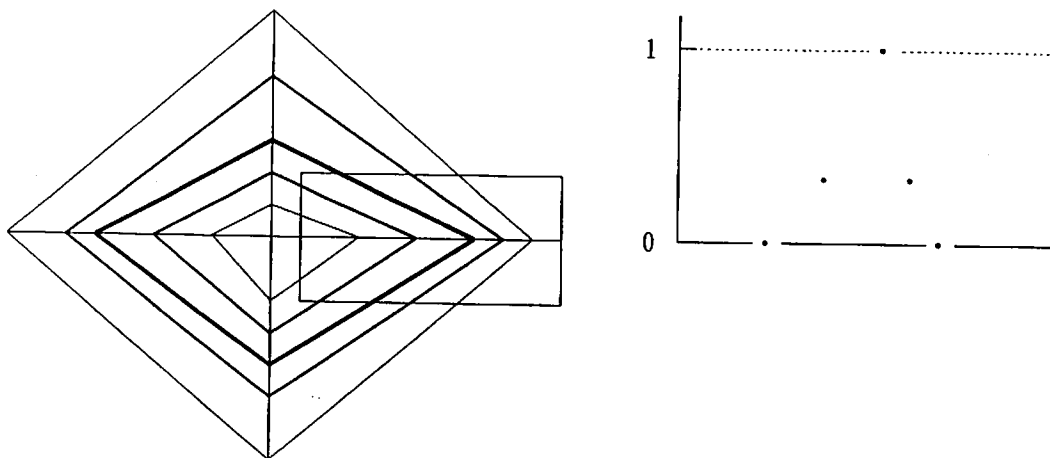


図1. レーダーチャートに対するファジイネスの導入

□ 従来の解析手法における考え方とファジィによる考え方の基本的な違い

従来の解析手法に対する考えは「1つの問題に対し、得られる答えは1つしか存在しない」というものである。これに対し、ファジィを導入した解析手法では「1つの問題に対し、解は複数個存在する」。これがファジィ導入の基本的考えである。

表1) 2クラス分類問題に対する従来手法とファジィネスを導入した時の解答例

	解答数	解答例
従来の分類問題 (2クラス分類)	1	クラス1
ファジィ分類 (2クラス分類)	2	クラス1 (0.8) クラス2 (0.3)

表1には2クラス分類問題における分類結果について、従来手法による結果とファジィ導入による結果とを示す。ファジィが導入された時は解答がクラス1と2それぞれに対し、ある帰属度を持ってアサインされている事がわかる。つまり、解析対象となるパターンは常に多面性を持っており、この為了解も一つに定まらないということである。しかし、現実の問題としてファジィネスの導入はこれだけでなく、解析過程の数式処理等の段階にも様々な形で取り入れられている。

しかし、これ以外にも様々な形で解析手法にファジィを導入する事は可能である。先に述べたように解析結果へのファジィネスの導入や、その外入力データ、解析過程等で様々な展開が可能である。最近ではファジィとニューラルネットワークとを組み合わせる事も行われている。

□ ファジィ導入による解析手法事例

クラスタリングへのファジィネスの適用 (ファジィクラスタリング)

クラスタリングへのファジィの導入は個々のパターンの帰属は複数のクラスターにわたるとい事が基本である。従って、従来のクラスタリングの結果で示される総てのパターンが1対1で一個のクラスターに必ず帰属されるというような事はない。

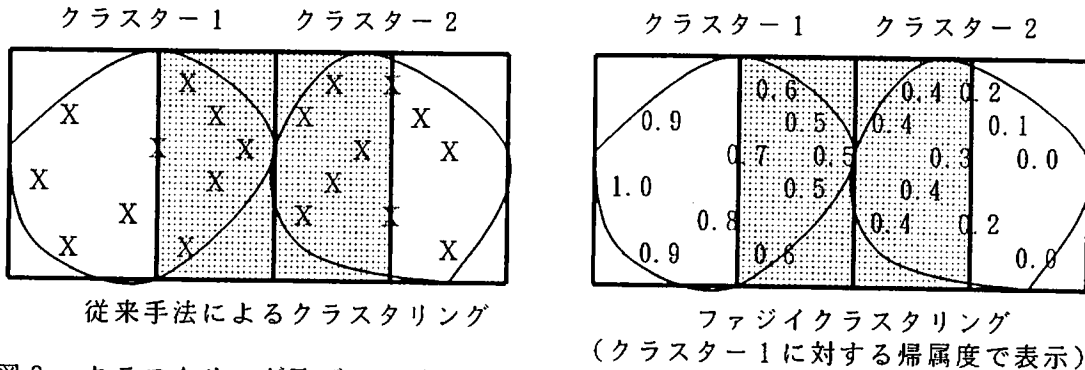


図2. クラスタリング及びファジィクラスタリングによる結果 (クラスター1に対する帰属度で表示)

図2の従来方式によるクラスタリングではパターンがクラスター1と2とに強制的に分類されている。個々のパターンで隣接する場所に他のクラスターが無いパターンは問題はないが、境界領域に近い部分に存在するパターンの帰属は常に不確実性が付きまとう事になる。図2のファジィクラスタリングにおいて個々のパターンに示されている値はクラスタリング1に対する帰属度が示されている。これから分かるように、他のクラスターと離れた位置にあるパターンは1や0に近い値を取り、境界領域にあるパターンの帰属度は0.5前後の値を取っている。このようにファジィクラスタリングではパターンの個々のクラスターに対する帰属度が分かるので、従来のクラスタリング手法と比べより詳細な情報を得る事が可能である。

BAYES判別分析へのファジィネスの適用 (ファジィBAYES判別分析)

超球法へのファジイネスの導入 (ファジイ超球法)

『超ボリューム概念』に従った分類手法「超球法」に対し、ファジイ理論を導入する事でより柔軟な分類が行われるように改良した手法である。このファジイの導入により、分類問題上での欠点の除去、解析精度の向上、解析時に得られる情報の増大等様々な利点を持つ事が可能となった。

このポイントは超球自体にファジイネスを持たせる事である。これは、超球そのものに密度を持たせる事で実現される。従来の超球法で扱う超球はその内部の密度が均一なものを利用して来た(図3.左図)。これに対し、ファジイを利用する超球法では超球中心部が1で表面が0となる密度関数(実際はメンバーシップ関数)で表される密度勾配を持っている(図3.右図)。

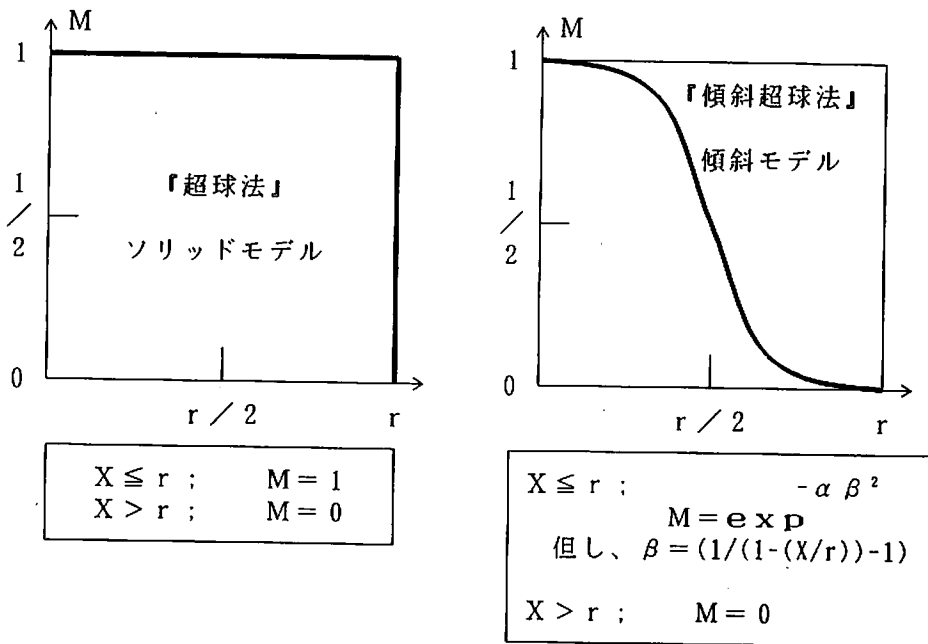


図3. ソリッドモデルと傾斜(ファジイ)モデルとの相違
傾斜(ファジイ)モデルに使用したメンバーシップ関数はGAUSS関数を用いている

超球法の定義ではその超球の大きさは、その超球が属しているクラスの特徴を保証する領域を示しているとする。従来のアプローチ（ソリッドモデル）では超球内どの部分であってもそのクラスの保証は全く同じである。つまり、超球の中心部分と周辺部分とでその「クラスらしさ」は全く同じ事を意味している。これは数学的な取扱を考えると便利であるが、現実的に考えるならば事象を素直に表現しているとは考えにくい。例えばクラスの異なる超球が隣接している時を考えるならば、特に隣接部分に近い周辺部分（図4中境界領域）では近い部分の環境に特性が似ず、逆に遠い部分の特性に似るという逆転現象が起きる事になる。

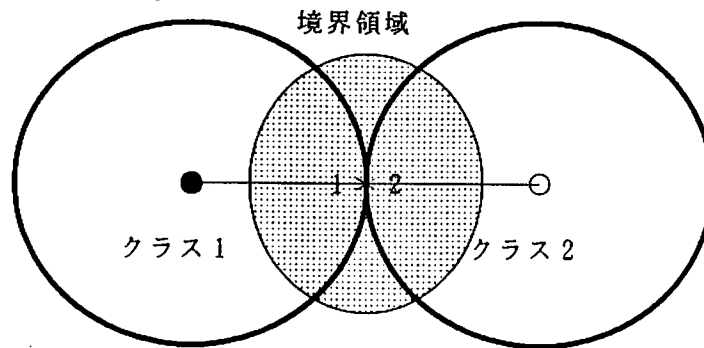


図4. 超球が隣接する時の境界領域に関する問題点

つまり、境界領域中にあるパターン1と2は互いに近い距離にある他のパターンに似る事なく、遠い距離にある超球中心部分の特性に支配される為、互いに隣接しながらクラスが異なるという矛盾した結果になる。

さらに複数の超球が重なった時、その重なった部分のクラス特性はどの超球に帰属させるのが適切であるかという問題（クラスの競合）が生じてくる。この時、クリस्पな関数を用いた従来手法による解析では多数決を取る等の手続きが取られている。このように単純な多数決では、超球同士の重なり程度を考慮した解析は不可能である。

ファジイ超球法ではこのクラスの特性が、超球中心部分で1、超球周辺部分に行くにつれて減少し、周辺部分では0、即ち超球中心部分のクラスと同じという保証は全くなくなるようになっていく。従って、クラス未知のパターンがある超球と中心上で重なった時、そのクラス未知パターンのクラスは重なった超球が帰属するクラスと全く同じである事になる。一方、超球の周辺部分で重なった時はそのクラス未知パターンのクラスが重なった超球と同じである保証は低いものとなる。

また超球が重なっている時はその重なり部分での重なり程度が数値データで示されているのでクラス未知パターンは重なり程度が最も高い超球と同じクラスに帰属される。従って、ファジイを導入した「ファジイ超球法」によるクラスタ分類は他の手法に比べて自然であり、またきめこまかな分類が可能となる。

□ ファジイ理論と他の解析手法との関係まとめ

ファジイ理論が様々な解析手法に導入する事が可能である事を示した。さらに従来の解析手法にファジイを導入する事で、従来の解析手法が持つ欠点を克服する事が可能であり、より自然な解析が可能になる事も示した。

ここで展開された解析手法以外にもファジイを導入した事例が多くなっている。また、多変量解析以外でも数理計画法やニューラルネットワークとの組み合わせが試みられ始めた。今後ますます多くの分野でファジイとの適合が試みられ、素晴らしい成果が得られると期待される。

□ ファジイの利用例

化学の分野においてファジイを導入して解析を行った事例は少ない。一つの例としてスペクトル解析にファジイを用いた例がある。これは石田らが行った仕事で、IRスペクトルの解析に利用する対照表にファジイを導入したものである。

ファジイを用いたスペクトル解析

通常IRスペクトルの解析はIRスペクトル吸収位置と吸収強度を基準として、様々な官能基の存在可能性をチェックする。この過程で、スペクトル吸収位置と官能基とを対応させた対照表が必要となるが、この対照表は総てクリスプな関数を用いて作成されていた。

例) 3100~3800 W -OH、-NH₂、.....
1600~1850 S ケトン、エステル、アルデヒド、.....

つまり、この表は定められた領域中にピークが存在する時は該当する官能基の存在確立が1で、領域外の時は0のクリスプな関数である。例えば、3800のピークは-OH、-NH₂とアサイン出来ても、3810のピークはなにも該当する官能基が存在しないことになる。この様な領域に存在する時は例外事項として扱う事で問題を解決してきたが、この事実は明らかに現実の解析業務とは掛け離れたものとなっている。

対照表の一般的特性として、個々の官能基の存在範囲を狭くするほどノイズが減少し解析精度が向上し、幅を広げるほどノイズデータが増えて解析精度が下がる。この対照表の設定で守らなければならない絶対条件は官能基の存在可能性を潰さない事である。つまり少しでも官能基の存在可能性がある時は、その事実をスペクトルチャートから確実にピックアップする事が必要となる。この危険率を小さくするという目的の為、計算機を用いたスペクトル解析で用いる対照表はスペクトルの範囲を必要以上に幅広く取る事になる。この結果、正解は絶対に逃さないがノイズデータが増大する事になる。ノイズが増える結果、計算時間が増大し、解析精度も落ちるといって望ましくない結果をもたらす。この為、現実的には計算時間と解析精度という2つの相反する項目のバランスを取りながら運用しているのが現状である。

この対照表の作成問題にファジイ理論を導入し、解析精度の向上を計ろうとする試みが行われている。

□ スペクトル対照表へのファジイの導入

前節でも述べたが、ノイズを減少させるには対照表中の範囲を狭くする事が望ましいが、正解を逃す可能性も増大する。この為、スペクトル対照表の取り方はかなり微妙な問題である。このような問題を解決するものとして、ファジイの導入が考えられる。

図1にはスペクトル対象表にファジイを導入したもので、そのメンバーシップ関数が示されている。この新しい対照表では、官能基の存在可能性は一つのピークをもつ山型の関数で示され、個々の官能基に対してこのような対照表が設定される。なおこのメンバーシップ関数はこの形だけでなく他には台形等も考えられる。

参考迄に従来手法による対照表が付記されている(右図)。

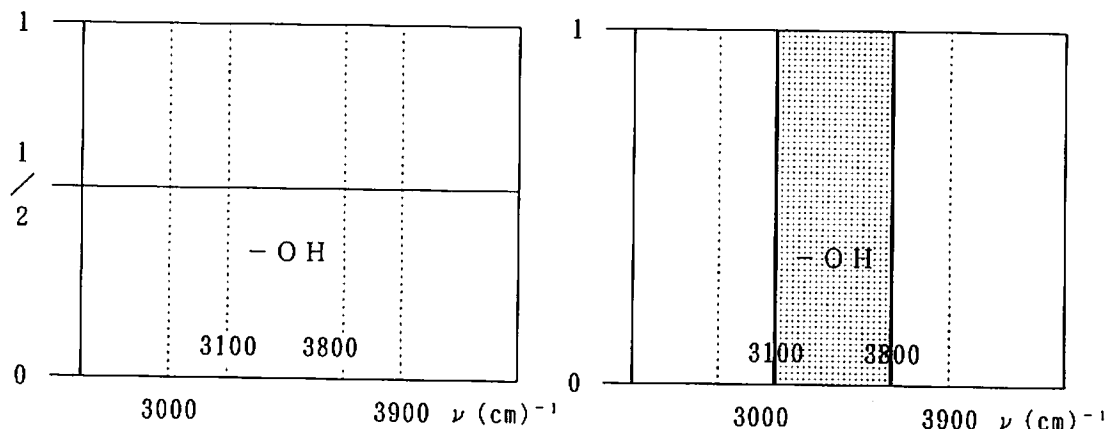


図1. 解析に用いられたスペクトル対照表(メンバーシップ関数と従来の対照表)

ファジイを用いた官能基の存在可能性出力結果

図2には図1に示されたようなメンバーシップ関数を用いた対照表を用いたスペクトル解析結果がしめされている。この構造式中、部分構造の上に記されている数値は個々の部分構造の所属度を示している。この値が大きい程メンバーシップ関数から求められるグレード値が大きく、その部分構造の存在可能性が高い事を意味している。

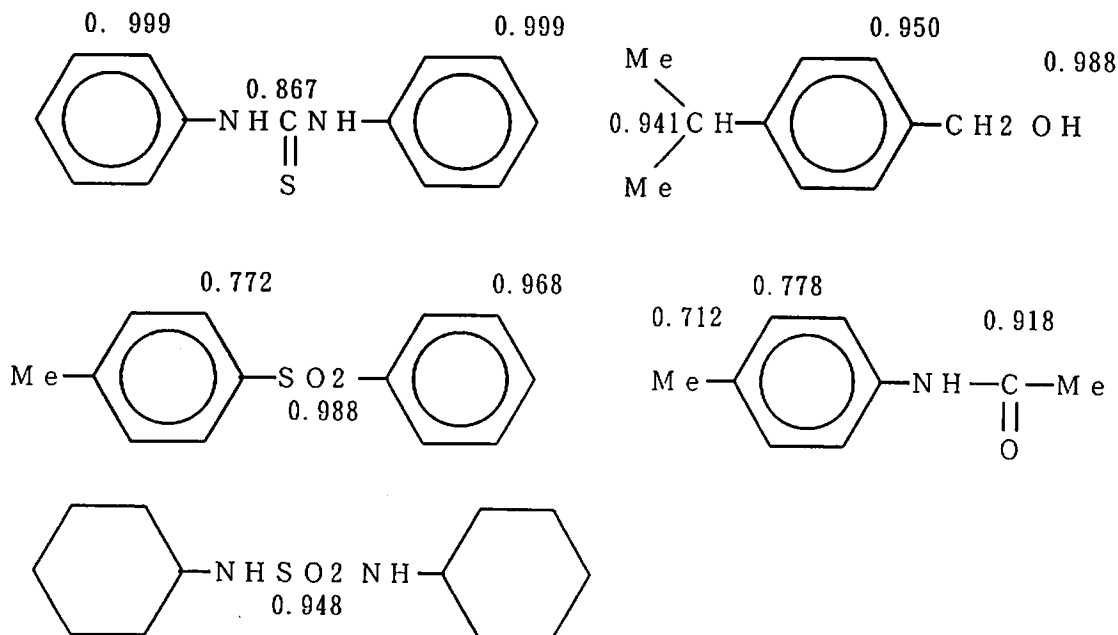


図2. スペクトルデータを基本として創出された化合物構造式と部分構造の所属度

従来の対照表を用いたスペクトル解析では部分構造の存在は絶対であり、その存在可能性の無い（対照表で部分構造の存在範囲内に無い）部分構造を持つ化合物の存在はあり得なかった。この点で化合物の部分構造が所属度付で表示されている事に抵抗を感じるかもしれないが、このような表示がファジイを導入した解析の特徴である。

3. パターン認識/多変量解析にもちいられる数値データ

3. 1. 数値データ概論

パターン認識/多変量解析に利用される数値データは、解析に利用される形態と数値データそのものの特性とで分類可能である。

数値データの利用形態としては目的変数と説明変数の2種類存在する。この2種類の分類では数値データの特性は全く問題にならない。単に解析に利用される形態が異なるだけである。

- ・ 数値データの利用形態： 目的変数、説明変数

$$Y = f(X) \quad \text{-----} \quad (1)$$

ここでYは目的変数を、Xは説明変数を意味する。一般的にXで示される説明変数のことをパターン認識分野では記述子と呼んでいる。

一方、数値データの特性としては連続変数及び不連続変数の2種類存在する。これらの数値データを目的変数として利用する時には、連続及び不連続変数の種類に応じて解析手法を変えることが必要である。また、説明変数に利用する時には連続及び不連続変数を混在させる時には注意が必要となる事がある。

さらに計算機を用いて解析を行う時には、整数型のデータか、実数型のデータかを意識して扱う必要がある。

- ・ 数値データの種類： (1) 連続変数、不連続変数
(2) 整数型、実数型

3. 2 N次元空間上のパターンはどのようにして表現されるか

□ パターンと数値データ（記述子）との関係

解析対象となるパターンPがn個の数値データで示されている時、そのパターンはn次元超空間上に分布している。従って、パターンベクトルPは以下のように表現される。

$$P = (X_1, X_2, \dots, X_n)$$

パターン認識や多変量解析ではこのパターンベクトルを用いて全ての解析が行われる。

3. 3 パターン間の距離

□ N次元パターン空間上のパターン間の距離

パターン認識や多変量解析の基本はパターン空間中におけるパターン相互の距離関係を様々な観点から検討することである。この為、多次元空間上でのパターン間の距離を表現する事が解析の第一歩である。

ここではこのようなパターン間の距離を表現する距離基準について簡単にのべる。以下に示す式では、パターン空間中のパターンM及びN間の距離をDMNとして表現する。また、パターンMとNはそれぞれ以下のようなd次元のパターンベクトルで示されているものとする。

$$PM = (X_{M1}, X_{M2}, X_{M3}, \dots, X_{M,d-1}, X_{M,d})$$

$$PN = (X_{N1}, X_{N2}, X_{N3}, \dots, X_{N,d-1}, X_{N,d})$$

□ 様々な距離基準について

パターン空間上のパターン間の距離を表現する距離基準（メトリクス）として様々なものが提唱されている。これらの距離基準は解析作業内容や数値データの特性等を考慮しながら使いわけることが望ましい。

- (1) ミンコフスキー (MINKOWSKI) 距離

$$D = \left[\sum_{i=1}^d (X_{Mi} - X_{Ni})^k \right]^{1/k}$$

- (2) ユークリッド (EUCLIDEAN) 距離

ユークリッド距離はミンコフスキー距離の式において、kが2の時にあたる。

$$D = \left[\sum_{i=1}^d (X_{Mi} - X_{Ni})^2 \right]^{1/2}$$

(3) シテイブロック (CITY BLOCK) 距離

シテイブロック距離は2パターン間の最短距離をとるのではなく、直交する2線の距離の総和をとるものである。

$$D = \sum_{i=1}^d [X_{M_i} - X_{N_i}]$$

(4) キャンベラ (CANBERA) 距離

$$D = \frac{\sum_{i=1}^d [X_{M_i} - X_{N_i}]}{\sum_{i=1}^d [X_{M_i} + X_{N_i}]}$$

(5) ハミング (HAMMING) 距離

ハミング距離は1/0のバイナリデータで利用される事が多いが、ORとANDの識別を効率良く行う事が出来る。

$$D = \sum_{i=1}^d [X_{M_i} + X_{N_i} - 2X_{M_i}X_{N_i}]$$

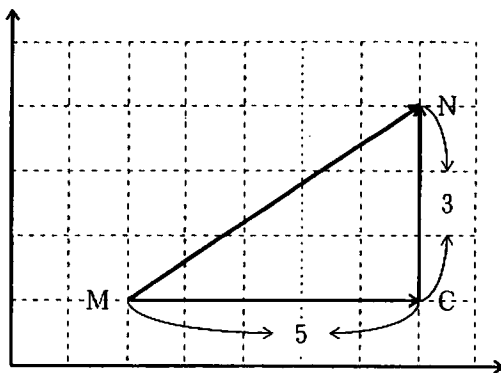
尚、1/0のバイナリデータを用いた時、ハミング距離とシテイブロック距離は同じものとなる。

(6) 谷本距離

谷本距離はハミング距離がその性格上、1が少ないデータは不利に評価されるといふ欠点を改良したものである。

$$D = \frac{\sum_{i=1}^d [X_{M_i} + X_{N_i} - 2X_{M_i}X_{N_i}]}{\sum_{i=1}^d [X_{M_i} + X_{N_i} - X_{M_i}X_{N_i}]}$$

図1にユークリッド距離とシテイブロック距離による表現例を示す。



2次元空間中でA点とB点との距離を計る時、ユークリッド距離では距離 D_{MN} を算出し、シテイブロック距離では距離 D_{MC} と距離 D_{CN} との距離を合わせたものをA点とB点との距離とする。

$M = (2, 1)$
 $N = (7, 4)$

ユークリッド距離	=	$D_{MN} = (5^2 + 3^2)^{1/2} = 5.83$
シテイブロック距離	=	$D_{MC} + D_{CN} = 5 + 3 = 8$

図1. ユークリッド距離とシテイブロック距離

3. 4. 数値データの意義及び分類

□ 連続変数/不連続変数 (カテゴリカル) 変数