活水論文集　第36集

# 多次元ファジイ集合における和集合と共通集合の高速演算法

川　場　　　隆

# Quick Operating Method of Union and Intersection for Multi-dimensional Fuzzy Sets

Takashi KAWABA

## 1．はじめに

特に関数形を前提しない連続なファジイ集合を計算機で取り扱うためには、ファジイ集合の要素座標とそのグレードの組（以下、定義点と呼ぶ）をいくつか与える方法が一般的である[1)]。この方法では、実際の集合演算において、定義点として与えられなかった要素のグレードが必要になった場合、適当な補完法によって推計したグレードを用いて、連続なファジイ集合としての取扱いを可能にしている。この方法は、補完法を多次元へ拡張することで、多次元ファジイ集合へも容易に適用できるが、欠点として、計算量の爆発的な増加があげられる。

例えば、n次元の実数軸を全体集合とする二つのファジイ集合で共通集合を計算する場合、要素座標を1.0刻みに取るのと、0.1刻みに取るのとでは、時間計算量にして、$10^n$倍の違いが生じる。和集合と共通集合は、応用的なファジイ集合演算の中で用いられる基礎的な演算であり、特に、多次元ファジイ集合においては、これらの計算量を小さくすることが、演算全体の高速化に直結する。

筆者は、推論や意志決定などのファジイ応用システムを記述するプログラミングシステムとして、$C^{++}$を拡張した$F^{++}$を開発中であり[2)]、現在大幅な改訂を行っている所であるが、上の問題を克服するため、ファジイ集合を保持する効率的なデータ構造と高速演算が可能なアルゴリズムを考案した。

## 2．ユニット法

考案したユニット法は、一次元のファジイ集合演算で行う簡易な計算法を多次元に拡張する手法である。ただ、多次元に拡張するとファジイ集合を保持するためのデータ構造は複雑になる。以下では、まず、一次元のファジイ集合演算における実際を説明し、ついで、その多次元への拡張について述べる。

**2．1．一次元の場合による考察**

補完法として直線補完を仮定し、１次元のファジイ集合だけを扱うのであれば、メンバシップ関数は直線であるから、二つのファジイ集合の演算において、和集合や共通集合は、メンバシップ関数の交点座標を計算することにより容易に求められる。この場合、計算のためには、実際に与えた定義点と、わずかな補完計算のみで可能である。

これは、実際には以下のように行う。

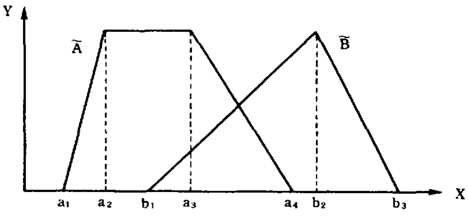


図１　ファジイ集合の例示

二つのファジイ集合$\tilde{A}$、$\tilde{B}$が、図１のように定義されているものとする。定義点は実数軸Ｘを全体集合とする部分集合$\tilde{A}$，$\tilde{B}$の要素であり、$\tilde{A}$は区間［$a_1$, $a_4$］、$\tilde{B}$は区間［$b_1$, $b_3$］上のファジイ集合として次のように与えられている。

$$\tilde{A}=\alpha_1/a_1+\alpha_2/a_2+\alpha_3/a_3+\alpha_4/a_4 \quad a_1,a_2,a_3,a_4,\in X,\ 0\leqq\alpha_i\leqq 1$$
$$\tilde{B}=\beta_1/b_1+\beta_2/b_2+\beta_3/b_3 \quad b_1,b_2,b_3,\ \in X,\ 0\leqq\beta_i\leqq 1$$

また、定義点と定義点の間では、ある要素$x$のグレード$F(x)$は以下のような直線補完により求める。

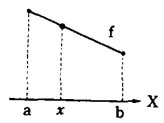


図２　直線補完

$$F(x)=La(x)\cdot f(a)+L_b(x)\cdot f(b)$$

ただし、$La(x)=\dfrac{x-a}{b-a}$, $L_b(x)=1-La(x)$

例えば、$\tilde{A}$，$\tilde{B}$の共通集合$\tilde{C}$を計算するケースでは、このように定義点間のグレードが直線補完によっていることから、二つの定義点で区切られる区間の両端でのみ比較を行えば十分である。

まず、図３のように共通集合$\tilde{C}$の定義域区間［$b_1$, $a_4$］に該当する部分集合を取り出す。次に、$\tilde{A}$，$\tilde{B}$の定義点を昇順に並べてマージし、$\tilde{A}$，$\tilde{B}$間でグレードを比較すべき要素座標を求める。例では、$b_1$, $a_3$, $a_4$の３点である。最後に、$\tilde{A}$，$\tilde{B}$のグレードを求めてmin演算を実行する。例えば、要素座標$b_1$では、$\tilde{B}$のグレードは、所与であるが、$\tilde{A}$のグレードは直線補完によって推計し、両者を比較する。

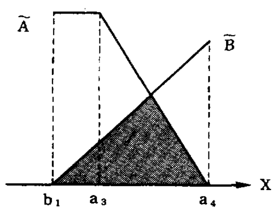


図３　$\tilde{A}\cap\tilde{B}$

| 定義点 | グレードの比較結果 | | |
|---|---|---|---|
| | $<\tilde{A}>$ | | $<\tilde{B}>$ |
| $b_1$ | $\mu_{\tilde{A}}(b_1)$ | $>$ | $\beta_1$ |
| $a_3$ | $\alpha_3$ | $>$ | $\mu_{\tilde{B}}(a_3)$ |
| $a_4$ | $\alpha_4$ | $<$ | $\mu_{\tilde{B}}(a_4)$ |

表１　グレードの比較

min演算の結果は、直前に行った結果と比較する必要がある。なぜなら、$\tilde{A}$，$\tilde{B}$のグレードの大小関係が逆転していれば、その区間ではファジイ集合の交わりが発生しているからである。例では［$a_3$, $a_4$］の区間がこのケースに当たる。

このような区間については、補完直線からメンバシップ関数の交点を計算し、求めた交

点座標とグレードを$\tilde{C}$の定義点に加えておく。この定義点を$\gamma$／pとすると、例の共通集合$\tilde{c}$は次のようになる。

$$\tilde{C}=\beta_1/b_1+\mu_{\tilde{B}}(a_3)/a_3+\gamma/p+\alpha_4/a_4$$

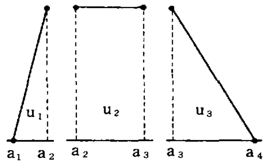


図4　$\tilde{A}$のユニット

なお、メンバシップ関数の交差を判定するために、定義域区間のどの部分でグレードの比較結果が逆転するか調べる必要があるため、ファジイ集合を図のように定義点で部分集合に分割して、保持しておくと都合が良い。この部分集合を以下ではユニットと呼ぶ。一次元ファジイ集合の場合、ユニットは二つの定義点によって定義されるファジイ部分集合である。

ユニットを使って、以上の手順を整理すると次のようになる。

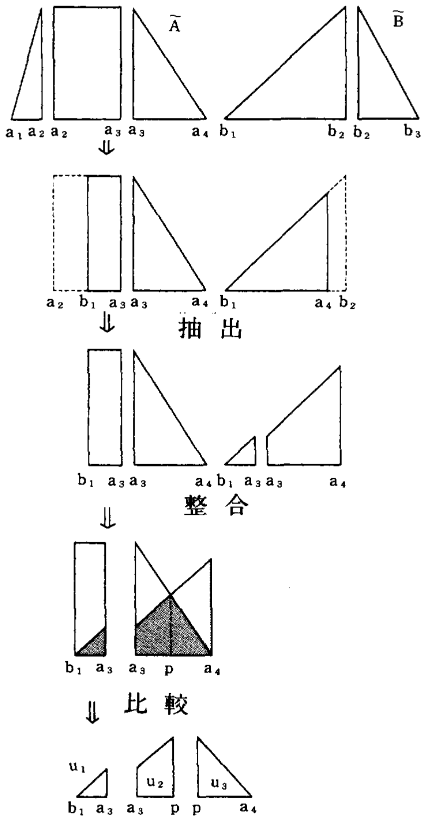


図5　ユニットによる$\tilde{A}\cap\tilde{B}$

1）抽出

計算結果であるファジイ集合$\tilde{C}$の定義域を求め、二つのファジイ集合$\tilde{A}'$, $\tilde{B}'$から対応する部分集合$\tilde{A}'$, $\tilde{B}'$を取り出す。この場合、$\tilde{A}$, $\tilde{B}$では、部分的にユニットの分割が発生する。ここで、分割とは、本来のユニットをさらに小さなユニットに分けることをいう。

2）整合

$\tilde{A}'$, $\tilde{B}'$の定義点座標をとりだし、マージする。マージ結果により、$\tilde{A}'$, $\tilde{B}'$をさらに分割し、ユニットの定義域を同じ大きさに揃える。

3）比較

ユニット毎に$\tilde{A}'$, $\tilde{B}'$のグレードを比較する。すなわち、同じ定義域を持つユニット同士で、両端のグレードを比較し大小関係を判別する。min演算では、グレードの小さなユニットが解となり、max演算ではグレードの大きなユニットが解となる。

ただし、左端と右端でグレードの大小関係が逆転していれば、ユニット内で$\tilde{A}'$, $\tilde{B}'$のメンバシップ関数が交差していることを意味する。その場合、補完直線の交点を計算し、その交点座標においてユニットを二つに分割する。各々のユニットでは、$\tilde{A}'$, $\tilde{B}'$のグレードはどちらが小さいか（大きいか）確定するので、適当な方を解のユニットとする。

### 2．2．n次元ファジイ集合におけるユニット法

#### 2．2．1．再帰的データ構造

直線補完を二次元に拡張すると次の様になる。

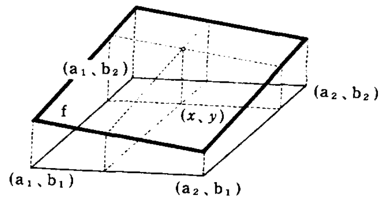


図6 二次元における直線補完

$$F(x,y)=\sum_{i=1}^{2}\sum_{j=1}^{2} La_ib_j(x,y)\cdot f(a_i,b_j)$$

ただし、$La_ib_j(x,y)=La_i(x)\cdot Lb_j(y)$

二次元の場合、任意の要素のグレードは、その定義点座標を囲む定義域平面上の4点から補完推計される。この4点で囲まれる矩形領域を定義域とするファジイ部分集合が、二次元ファジイ集合におけるユニットである。

一般に、n次元ファジイ集合の場合には、グレードの補完式は次のようになり、ユニットの定義域は$2^n$個の頂点を持つ超立方体となる。

$$F(x_1,x_2,\cdots,x_n)=\sum_{i_1=1}^{2}\sum_{i_2=1}^{2}\cdots\sum_{i_n=1}^{2} Lx_{1i_1},x_{2i_2}\cdots x_{ni_n}(x_1,x_2,\cdots x_n)\cdot f(x_{1i_1},x_{2i_2},\cdots,x_{ni_n})$$

ただし、$Lx_{1i_1},x_{2i_2}\cdots x_{ni_n}(x_1,x_2,\cdots,x_n)=\prod_{k=1}^{n}L(x_{ki_k})$

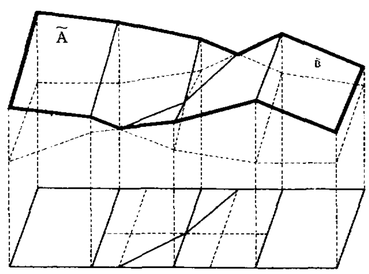


図7 二次元ファジイ集合の和集合演算(Ã∩B̃)

初期データとして与えられるファジイ集合では、ユニットは規則的に並んでいる。しかし、集合演算を行うことにより、図7のように局所的に分割が発生し、ある部分だけが細かく細分された状態になるため、当初の格子状の規則性は失われてしまう。

そこで、このような不規則性を吸収するために、図8のような再帰的なクラスオブジェクトとして、ファジイ集合を保持する。

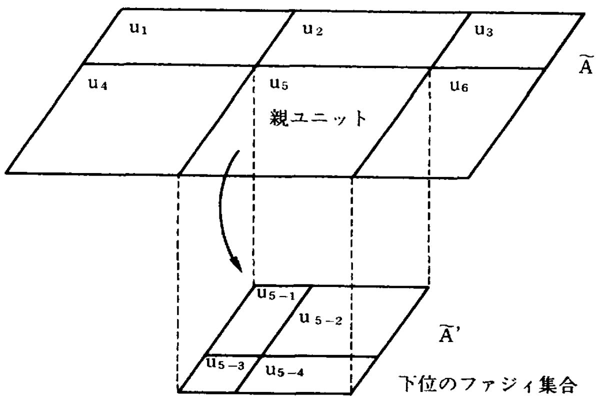


図8 再帰的なファジイ集合オブジェクト

すなわち、集合演算によりユニットが細かく分割された場合は､元のユニットを親ユニットとして残し、それらはひとつにまとめて親ユニットから指される独立したファジイ集合オブジェクトとして扱うのである。このデータ構造により、ユニットは常に格子状に並んだ規則的な構造を保ち、取扱いは格段に容易になる。

このために実際に定義したクラスを示せば次の通りである。

1）FuzzySet／ファジイ集合クラス

総称的なファジイ集合本体のクラスである。集合定義情報 (FzDef) オブジェクトから派生し、集合属性 (Panel) オブジェクトを追加的なメンバオブジェクトとして保持する。集合属性とは使用するmax, min演算メソッドの種別などである。ユーザーインターフェースとなるさまざまな集合演算は、このクラスで定義してある。

2）FzDef／ファジイ集合定義体クラス

実質的なファジイ集合の定義情報に当たるクラス。ユニットをまとめるユニット配列 (UnitArray) オブジェクトと定義域の範囲、最大値、最小値などの付加情報をメンバオブジェクトとして保持している。

3）UnitArray／ユニット配列クラス

ファジィ集合の定義域を格子状に区切ったとき、各区画に位置する規則的なユニットの並びからなる配列のクラスである。すなわち、n次元ファジイ集合の定義域が各軸座標において、各々 $\alpha_i$個 (i＝1,2,…,m) の区間に区切られている時、この配列クラスは、$\prod_{i=1}^{m}\alpha_i$ の数だけの要素ユニットを持つ。ただし、$\Pi$は各$\alpha_i$を掛け合わすことを意味する。

なお、この配列クラスは、格納要素数の増大に対して配列サイズの自動拡張機能を持ち、また、定義座標をキーとするソーティッド配列である。

4）Unit／ユニットクラス

ユニット本体であり、ユニット配列クラス (UnitArray) の要素になる。メンバオブジェクトとして、定義点とファジイ集合 (FuzzySet) へのポインタを保持している。

n次のファジイ集合の場合、ユニットクラスオブジェクトは、$2^n$個の定義点を保持している。定義点は要素座標とそのグレードを保持するグレード要素オブジェクトとして定義され、実際には$2^n$個の要素を持つ配列となっている。

ファジイ集合へのポインタには、通常NULLが割り当てられている。しかし、集合演算によりこのユニットが分割された場合には、それら個々のユニットをまとめてファジィ集合とし、それへのポインタを格納する。すなわち、このユニットを親ユニットとする再帰的なデータ構造が形成される。分割が起こった時は、親ユニットの定義点情報は定義座標のみが残され、グレードは意味を持たなくなる。

### 2．2．2．演算アルゴリズム

二次元の場合を例にとって説明する。メンバシップ関数はユニット上で、既に見た図6のような平面となる。

そこで、ファジイ集合同士のグレードの大小比較は、先に掲げた抽出、整合の操作の後、ユニットを規定する四つの定義点でのみ行えば良い。4点は、軸方向に平行な四つの区間を形成するが、ファジイ集合の交わりはこれらの区間のうち、その両端におけるグレード

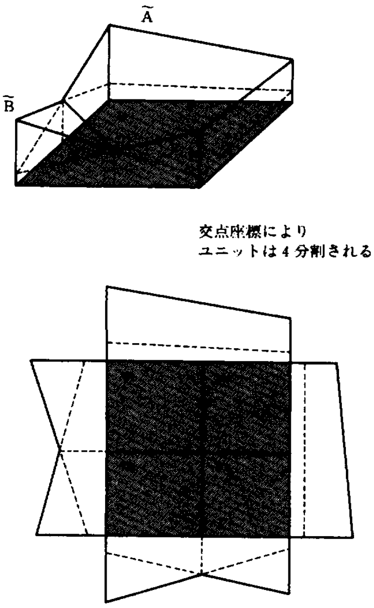


図９ ユニットの分割

の大小関係が逆転している区間で発生する。また、いづれの区間でもグレードの大小関係の逆転が見られなければ、一方のファジイ集合が完全に他方を含んでいることを意味する。

ユニット内においてファジイ集合が交わっている場合、すなわち、グレードの大小関係が両端で逆転している区間については、１次元ファジイ集合の場合と同様にして、補完直線から交点を計算する。ついで、その交点を境にしてユニットを分割し複数の新しいユニットを生成する。これらの分割されてできた新しいユニットは、部分集合として管理するためにファジイ集合としてまとめ、分割された親ユニットにはそのファジイ集合へのポインタを格納してリンクを張る。この結果、親のユニットには、定義域の情報だけが残され、グレードを含めた実態は部分集合として、ファジイ集合に移されることになる。

以上から、一般のｎ次元のファジイ集合$\tilde{A}$，$\tilde{B}$の和集合・共通集合の算出も、実際には抽出、整合、比較の三手順で実行される。ただ、その過程では、再帰的な処理が行われることに注意しなければならない。以下に、その手順を示す。

⑴抽出

$\tilde{A}$，$\tilde{B}$の定義域の和、または共通部分を算出し、$\tilde{A}$，$\tilde{B}$から該当部分を切り出して新しいファジイ集合$\tilde{A}$，$\tilde{B}$を生成する。この際、分割が起こるユニットが、ファジィ集合へのポインタを持つ親ユニットになっていれば、そのファジイ集合に対して、再帰的に抽出処理を適用する。

⑵整合

整合も、再帰的な処理となり、次のような手順で行う。

１）$\tilde{A}$，$\tilde{B}$の定義点座標を取り出し、昇順に並べてマージし、定義点座標ベクトルを作成する。

２）定義点座標ベクトルによって、$\tilde{A}$，$\tilde{B}$のユニットを分割する。この際、ユニットが親ユニットとなっている時は、下位のファジイ集合を再帰的に分割する。

⑶比較

１）max、min演算

ユニットの辺を形成する区間において、両端のグレードを比較する方法によりmax、またはmin演算を行う。なお、比較するユニットの少なくとも一方が親ユニットになっている場合は、そのユニットの定義域について、それらをファジイ集合とみなして、

(2)の整合処理から再帰的に処理を適用する。

2）ユニットの分割

ユニット内でメンバシップ関数が交差していれば、交点を境にして、軸方向に平行にユニットを分割する。これらのユニットは親ユニットの下で、独立したファジイ集合としてまとめ、親ユニットのファジイ集合へのポインタ欄にそのポインタ値を格納しておく。最終的な解として、親ユニットをユニット配列オブジェクトに格納する。

## 3．まとめ

n次元ファジイ集合の和集合、共通集合を効率よく高速に計算するための方法としてユニット法を提案した。ユニット法ではファジイ集合を、定義点を境とする複数のユニットに分割し、ユニット毎に集合演算を適用する。ユニット内でメンバシップ関数が交差する場合には、その交点で座標軸に平行にユニットを分割し、それらをまとめて新たなファジイ部分集合とする再帰的な処理を行う。直線補完を採用したため、メンバシップ関数の交点の計算は、一次元のファジイ集合に対して行うのと同じ方法でよく、極めて高速である。

## 参考文献

1）Umano.M.,Fuzzy-set manipulation system in lisp. Preprints of Second IFSA Congress,7,1987

2）川場隆，オブジェクト指向言語C++によるファジイ集合処理，活水論文集，第35集，ppl-14，1992

3）Zadh. L.A., Fuzzy sets. Information and Control,pp.338-353,1965

4）水本雅晴，ファジイ理論とその応用，サイエンス社，1988

5）Stroustrup.B.,The C++Programming Language second edition,Addison Wesley,New York,1989