3G-10

ＦＵＺＺＹ分類型シェル（ＦＵＺＺＢＯＸ）における推論方式

重村淳一[1]　木口重幸[2]　宥免達憲[3]　矢野真枝[3]
([1]沖縄日本電気ソフトウェア㈱　[2]日本電気㈱　[3]中国日本電気ソフトウェア㈱)

１．はじめに

診断型あるいは分類型と呼ばれるエキスパートシステムにおいて、そのルール記述の中に曖昧さの度合を表す尺度として、確信度やファジィ性等が多く用いられている。我々は、ルール記述において従来用いられていたＭＹＣＩＮ流の確信度を拡張した概念としてファジィ性をとらえ、確信度とファジィ性を同じ枠組みの中で取り扱うことにより、人間の主観や感性をも表現可能な知識推論システムの構築支援環境（ＦＵＺＺＢＯＸ）を試作している。これは、ある結論仮説を説明するための条件仮説が順に階層的な推論木を構成し、各条件仮説からなるノードには、ファジィ性を与えることができる。また、各ノードでの推論方式は、ＡＨＰ的手法[1]を取り入れて、複数入力にたいして重み付け演算を施し、１出力を与える方式で行うことにより、現実の意思決定問題に適用できることが期待される。

２．分類型問題における知識表現の問題点

分類型（診断型）エキスパートシステムの知識表現にファジィ性を用いて曖昧さを記述する場合、２値論理における２項演算がファジィ関係の演算の基本となり、ＡＮＤ／ＯＲに対応してＭＩＮ／ＭＡＸからなるファジィ演算が用いられることが多い[2]。

一方、信頼性の問題では、ｍ個の中のｎ個真であれば全体を真であるとみなしたり、選択決定の問題では、意思決定要因の数が多ければ多いものほど、また重要度が強ければ強いものほど選択されることになる。従って、これらの問題に知識処理を用いるときには、多数決原理や多項演算に基づく論理演算を用いた方がふさわしいように思われる。

３．Ｆｕｚｚｙ分類型問題におけるルール表現と推論方式

ファジィ性を取り入れた分類型問題を本稿では、Ｆｕｚｚｙ分類型問題、その知識ベース構築環境をＦｕｚｚｙ分類型シェルと呼ぶ。

本稿におけるルール（厳密には、結論仮説）は、いくつかの属性を持つ。まず、ファジィ関係の演算に関するものであり、従来のＡＮＤ／ＯＲと本稿で述べる多項演算を基にした論理演算とを区別するものである。二つ目が、実行制御に関するものであり、各仮説（条件仮説、結論仮説）の論理演算以外の関係を記述することで表現する。三つ目は最下位のファクト（条件仮説）に関するもので値を入力する時に呼び出されるメンバシップ関数を格納する。

３．１　重み付けルール表現

重みを付加したルールの表現の例を図１に示す。

rule1: if $x_1(w_1), x_2(w_2), \cdots x_n(w_n)$ then y　　（１）
rule2: then x_1
rule3: then x_2
…
rulen: then x_n

図１　重み付きルールの例

ここで、rule1中のx_i（$i=1, n$）が条件仮説であり、w_i（$i=1, n$）は、x_iの重みであり、ｙは条件仮説x_i（$i=1, n$）より導き出される結論仮説を表している。この重みw_iは、条件仮説x_iが結論仮説ｙを判定するのにどの程度重要であるかを示す値であり、エキスパートシステム構築者の主観や感性等を基にＡＨＰ的手法で決定する。本稿では重みの値を0.0～1.0とし、値が大きいほど重要度が増すものとする。これら条件仮説と結論仮説は、それ自身が他のルールの条件仮説あるいは結論仮説（中間仮説）となることでネットワーク（以降、推論木と呼ぶ）を構成する。

条件仮説ｘが、もしファクト（条件仮説が一個も

Inference method for fuzzy classification domain shell
J.Shigemura[1], S.Kiguchi[2], T.Yumen[3], M.Yano[3]
[1]NEC Software Okinawa,Ltd.　[2]NEC Corporation　[3]NEC Software Chugoku,Ltd.

ない結論仮説）であるならばメンバシップ関数を設定することができ外部より入力された観測値をメンバシップ値に変換する。図１中のrule2からrulenまでは、ファクトを定義しているルールである。

３．２　推論方式

重み付きルールによる推論方式について説明する図２は、推論木の一部を表しており、前述の図１に対応する。

$x_1\ (w_1)$
$x_2\ (w_2)$　→ y
…
$x_n\ (w_n)$

図２　推論木を用いて表されるルールの例

結論仮説ｙの真理値は、ルールを構成する条件仮説ｘの真理値と、ｘの結論仮説ｙに対する重みを乗算した値の総和で求められる。（式（２））

$$\mu_y = \sum x_i \cdot w_i \qquad (2)$$

ここで、μ_yは結論仮説ｙの真理値を表す。

ルールの選択実行は並列的に行われる。ただし、結論仮説ｙの真理値を計算するためには、各条件仮説x_iの値が定まっている必要があるため、推論は基本的に下位のルールより並列的に発火され、推論木中を下から上に向かって実行される。この時、実行する結論仮説が、ファクトであるならば、設定されたメンバシップ関数を用いて、外部からの入力値をメンバシップ値へと変換する。

しかしながら、エキスパートシステムの実行には、より複雑な制御構造が記述できるのが望ましい。そのため、推論木中の各ルールには当該ルールを発火させるために必要な条件（ここでは、必要な結論仮説）を記述しておき条件が満足されるまでルールの発火は抑制される。図３に一例を示す。

$x_1\ (w_1)$
$x_2\ (w_2)$　→ y_1
$x_3\ (w_3)$
　　　　　　↑
$x_4\ (w_4)$
$x_5\ (w_5)$　→ y_2
$x_6\ (w_6)$

rule1:　if　$x_1(w_1), x_2(w_2), x_3(w_3)$　then　y_1
　　　　(trigger　y_2)
rule2:　if　$x_4(w_4), x_5(w_5), x_6(w_6)$　then　y_2

図３　ルールの発火抑制

図中、波線の矢印は発火の抑制方向を示している。上図において結論仮説y_1を（y_2の）ターゲットノードと呼び、結論仮説y_2を（y_1の）トリガーノードと呼ぶ。ここで、上図の推論木で表されるエキスパートシステムが起動された場合、条件項目x_1，x_2，x_3の真理値が求められているならば、rule1により、結論仮説y_1の真理値を算出しようとするが、結論仮説y_1は、結論仮説y_2によって実行が抑制されているのでrule1は,rule2の終了を待って発火されることになる。

４．適用例

以上の表現方法を用いた簡単な例を図４に示す。

症状$_1$(0.8)
症状$_2$(0.2)　→ 病名$_1$

症状$_3$(0.6)
症状$_4$(0.4)　→ 病名$_2$

rule1: if 症状$_1$(0.8), 症状$_2$(0.2)　then　病名$_1$
rule2: if 症状$_3$(0.6), 症状$_4$(0.4)　then　病名$_2$
rule3: then 症状$_3$ (trigger 病名$_1$)
rule4: then 症状$_4$ (trigger 病名$_1$)
…

図４　適用例

図４において、システムの実行メッセージが送られた場合、rule3, rule4は、病名$_1$によって実行が抑制されているので、それを条件仮説とするrule2は病名$_1$の疑いが生じるまで考慮されないことになる。

５．おわりに

本稿において多数決原理や多項式に基づく論理演算を用いたルールの一表現方法と、推論方法について述べた。本表現方法では、人間の持つ主観や感性を重みとして表し、ルールに付加する。今後、現実のエキスパートシステムへの適用と、知識を獲得するための支援ツールの実現を行う予定である。

［参考文献］

［１］　Saaty,Thomas L., The Analytic Hierarchy Process, McGraw-Hill, 1980
［２］　水本雅晴，ファジィ理論とその応用，サイエンス社，1988