

(1) 土木分野における知識情報処理とエキスパートシステム[文献 1]

(a) 知識情報処理技術

人工知能(AI: Artificial Intelligence)の分野において、最も進歩した分野の 1 つが知識情報処理である。人工知能とは、人間の知的な働きと同じ働きを、機械によって実現することを目的とした科学技術であり、多くの分野で応用がなされている。

知識情報処理をコンピュータにより実現したのが知識情報処理システムである。知識情報処理システムのうち、知識ベースエキスパートシステム(エキスパートシステム)は、顕著に発展を遂げたシステムとしてあげられる。

エキスパートシステムは、人間の知識をある定まった形式で集約した知識ベース(またはルールベース)を用いて、ある問題に対する解を探し出すシステムであり、この知識ベースには、限定された領域の専門知識が集約されている。これを用いて、専門家の行う業務をコンピュータに代行させたものを知識情報処理システムという。

(b) 土木分野でのエキスパートシステムの必要性

知識情報処理システムは、各分野の専門技術者の経験知識と専門的な領域知識を収集し、コンピュータに取り込むことで、誰もがその知識を共有して、対象領域での問題解決を行うことを目的としているもので、これをエキスパートシステムという。

エキスパートシステムの必要性は、様々な分野でいわれており、その必要性は、土木分野においても例外でなく、多くのエキスパートシステムに関する研究が行われている。

土木分野においては、多くの場合熟練技術者を中心とし、彼らの持つ専門知識や経験則を基に業務がなされている。

熟練技術者による判断は、学問的に明確にされていないような対象問題の解決についても的確な解を示している。また、実際の実務レベルにおいてもコストの高い熟練の上級技術者が直接現場で作業することは少なく、たとえ上級技術者が直接作業するとしても、多種多様な問題を少数の上級技術者で対処することは困難である。そのため、経験の浅い初級技術者や中級技術者の補助は必要不可欠となってくる。

そこで、土木分野におけるエキスパートシステムでは、専門家の支援や、非専門家の支援としての利用、あるいは、初級・中級技術者に対する教育、助言、技術者の作業緩和、知的財産の蓄積などを主眼において、システムの開発がなされている。

(2) 研究の背景[文献 1,2,3]

知識工学というものが、1977年 Feigenbaum によって提唱され、その後 80年代に入りエキスパートシステム構築に関する研究が盛んにされるようになった。これと同じくして、土木分野においてもエキスパートシステムの必要性がいわれはじめ、エキスパートシステムに関する推論技術の研究や、推論システムの開発が行われるようになった。

エキスパートシステムに関する技術、システム開発の事例は、以下に示すように、様々な問題を対象として報告されている。

1. 橋梁などの構造物の健全度，安全性，耐用性の評価[文献 4,5]
2. 構造解析における同定問題[文献 10]
3. 都市計画における計画，景観設計，景観を考慮した橋梁形式選定[文献 11,12]
4. 構造物の維持管理業務[文献 13,14]
5. コンクリート橋の損傷度や，損傷要因の選定[文献 15,16]
6. 橋梁の補修・補強方法の選定[文献 17-23]
7. トンネルの支保パターン選定[文献 24,25]

ここで示した事例は、報告されている事例のうちの一部にすぎないが、これらの多くは、コンクリート橋や鋼道路橋など橋梁に関するものであるといえる。

また、これらのシステム開発事例では、ルールベース推論、事例ベース推論、ファジィ推論、ニューラルネットワークなど様々な技術が用いられており、対象とする問題の特徴に応じて使い分けられている。また、それぞれの推論技術の長所を組み合わせた、ファジィルールベース推論や、ファジィエントロピー、ルールベース推論と事例ベース推論を用いた推論技術など、新たな推論技術の開発も行われている。

一方、最近では、コンピュータ技術の向上により、大規模なデータの蓄積・処理が容易かつ高速になったことから、データベースシステムを高機能化、あるいは高性能化するための知識処理技術の応用や、知識処理分野で有効利用可能な機能を備えたデータベースシステムの構築が各分野で行われている。[文献 26]

本論文において、皆川らによる既存の研究では、道路橋 RC 床版の損傷要因推定問題を、本研究では、鋼橋疲労損傷の補修方法選定問題を対象ドメインとして取り上げることとした。これらの対象問題解決のために最も基本的な推論技術として、ルールベース推論を用いた。また、上述のような対象問題は、専門家の経験的知識を取り入れる必要があるが、このような経験的知識は、ルール化することが困難な場合がある。この問題解決のため、皆川らは、ニューラルネットワークによる推論技術を取り入れることにした。ニューラルネットワークは、対象問題の入力と出力の値は分かっているが、それに関係づけるアルゴリズムが不明である場合に、高度な知的情報処理能力を発揮することから適用することとした。また、本研究において、ルールベース構築の際、知識のルール化を自動で行えるようにするため、データベースからのルール抽出アルゴリズムの確立を試みた。

(3) 研究目的 [文献 1]

土木構造物は、他分野の製品に比べて大規模で単品生産であり、しかも耐用期間が非常に長いことなどから、構造物の設計、施工や維持管理業務は、少ないデータ、知識、経験に基づいて行われているのが現状である。このような分野に対して、エキスパートシステムの活用は有効であることから、前述のように多くの対象問題に関する報告がなされている。

これらエキスパートシステムの知識ベース(ルールベース)構築にあたっては、専門家や熟練技術者の経験則をアンケートやインタビューによりルール化して、知識ベースを構築している方法が主流であるが、この方法には、次のような欠点がある。

- (1) ルールベース毎にルールの表現形式が異なる
- (2) 問題毎にルールベースを構築しなくてはならず、エキスパートシステム構築の初期段階でのルールベース作成に多大な労力が必要となる
- (3) 専門家の示す知識が経験的知識であるため、複雑な因果関係を持つことから、知識を単純な表現形式であるルールで表現することが困難な場合がある
- (4) 同様に専門家の示す知識が経験的知識であるという性質上、曖昧さを含むことから知識の信頼性を定量的に評価してルール化することが困難となる
- (5) エキスパートシステムの構築がある程度進んだ後、ルールベース内のルールの追加・修正を行う場合、追加・修正を行ったルールによって、ルールベース内に矛盾が生じたり、ルールベース内の整合性がとれなくなる場合がある

このため、現在でも知識のルール化、つまり知識獲得が困難であり、エキスパートシステム開発における最大のボトルネックとなっている。この問題解決のため、ルールの自動生成、知識獲得手法、機械学習など様々な方法が検討されているが、実用化に対して有効なアルゴリズムの確立はなされていない。

土木分野において、実用化に向けたエキスパートシステムの構築を図るならば、少なく点在している知識を共通して利用可能な知識として明確な形で記述、収集して、専門知識の共有化や再利用をすることが必要不可欠であるといえる。

本論文では、専門知識の共有、再利用を主とした知識獲得問題の解決を目的として、皆川らにより構築された知識洗練機能付エキスパートシステムの知識共有、再利用に関する有効性を検討した結果を示す。次に、ルールベース構築の初期段階での知識獲得問題を解決するため、データベースからのルール抽出技術を用いて、データベースを用いて、データベースからルールベースを構築するための、ルール抽出アルゴリズムの構築を試みた。

(4) 研究概要

第1章においては、知識情報処理と知識情報処理技術の1つであるエキスパートシステムについて述べる。

第1節では、エキスパートシステム開発が行われるようになった経過と、エキスパートシステムの特徴を述べる。

第2節では、エキスパートシステムの基本概念として、1項において、エキスパートシステムを構成する知識ベースと推論機構について述べ、2項では、実際に用いられている推論機構のうちの1つであるプロダクションシステムについて述べる。また3項では、エキスパートシステムが専門知識を扱うシステムであり、従来型の数値計算を主体としたシステムのように、全てを明確に宣言的に記述することが困難であることから、“不確実な情報の取り扱い”として不明瞭な情報や不確実な情報を取り扱うための方法を述べる。はじめに、エキスパートシステム開発において、確実性を扱う手法として多く採用されている確信度について述べ、次に、同じ確実性を扱う手法として、統計学的なアプローチによる Dempster-Shafer 理論について述べる。そして、多くのエキスパートシステムで推論機構として適用されているファジィ推論について述べ、同様に推論機構として多用されているニューラルネットワークについて述べる。最後に、前述した推論機構では問題解決のために必要な知識をある形式で知識ベースとして用意する必要のある推論機構であったのに対し、こういった知識ベースを構築する必要が必ずしもなく、過去の実際の事例を蓄積しておいて、過去の事例と現在の問題の間に存在する類似性を用いて推論を行うシステムである、事例ベース推論について述べる。

第3節では、エキスパートシステムの開発技術として、1項では、エキスパートシステムが従来型のプログラム開発と違い、知識をいかに収集・整理して記述するかという点に注目して開発が試みられていることを述べ、エキスパートシステムの開発手順と、その注意点について述べる。2項において、エキスパートシステム開発で最も重要で困難な部分とされる知識の収集・整理などの知識獲得手法について述べ、3項では、エキスパートシステムを適用して開発する時の対象ドメインのモデル化について述べる。また、4項では、実用的なエキスパートシステム開発の実現のため、従来ならば困難であった問題解決を行うことのできるエキスパートシステムと、大量かつ高速な計算処理を行える計算機システムの結合を行う必要性和その方法について述べ、5項において、エキスパートシステムの一般的な構築ツールの構成と、エキスパートシステムの現状について述べる。

第2章では、既存の研究である、道路橋 RC 床版の損傷要因推定を対象問題とした、ルールベース洗練機能付診断エキスパートシステムの開発について述べる。

第1節では、これまでの様々な研究を概観し、本研究の必要性和目的について述べる。

第2節では、ルールベース洗練機能付診断エキスパートシステムの推論機構について述べる。1項では、本推論システムが相互結合ネットワークにより構成され、教師付推論システムであることを述べ、推論方法として min-max 演算とニューラルネットワークで用いられる誤差逆伝播アルゴリズムを応用していることを述べる。また、ネットワー

ク内での仮説と仮説間の関係についての定義をし、すべての構成要素について確信度を付与して推論を行うことを示す。2項において、1項で述べた min-max 演算を用いた、仮説と仮説間の関係の属性値の決定アルゴリズム(推論アルゴリズム)について述べる。また、3項において、本研究で主眼においている知識獲得問題解決のために提案した1手法であるルールベース更新のためのアルゴリズム(洗練アルゴリズム)について述べる。

第3節では、本推論システムの有効性を検討するために、道路橋 RC 床版の損傷要因推定を対象ドメインとしたことを述べ、対象問題の推論を行うにあたり、三上、田中らにより開発された道路橋 RC 床版の損傷要因推定システムのルールベースを適用したことを述べる。

第4節、5節において、実際に本推論システムによる損傷要因推定を行った結果を示す。

第4節では、三上、田中らによるシステムのルールベースをそのまま適用して推論した結果を示し、第5節では、本推論システムの持つルールベース洗練機能によりルールベース洗練を行った後、損傷要因推定を行った結果を示す。本推論システムの機能の精度を検証するため、1項では、ルールベース洗練の際に用いる事例を唯1つとし、その後の仮説推論の際にもルールベース洗練に用いた時と同じ事例を提示して損傷要因を行った結果を示す。次に、ルールベース洗練の際に複数の事例を用いた場合に洗練されるルールベースの洗練精度の検討を行った。2項では、過去の事例として保持している8つの事例を全て用いてルールベース洗練を行った場合の推論精度を検討した。3項では、8事例のうち、損傷場所に類似性のある6つの事例を用いてルールベース洗練を行い、ルールベース洗練に用いた6事例についてそれぞれ仮説推論を行い、推論精度の検討を行った。また4項では、3項で用いた6事例のうち5事例をルールベース洗練のための事例とし、残りの1事例についての仮説推論を行った場合の推論精度の検討も行った。

第6節では、2章での研究の結果、本推論システムが知識の共有・再利用を中心とした知識獲得問題に対し、有効に機能していることを述べる。しかし、専門知識のルール化という点において、知識をルール化できなかった場合に対処し得ないという課題が残る。

そこで、皆川らは次に、専門知識のルール化が困難であった場合に、ニューラルネットワークによる推論をブラックボックスとして用い、はじめに仮説推論である損傷要因推定を行い、この推論結果を用いてルールベース洗練を行って知識の共有・再利用を達成することを試みた。

第7節では、6節までの研究の結果、ニューラルネットワークを用いた推論システムをルールベース洗練機能付診断エキスパートシステムの補完システムとして適用する必要性とその目的を述べる。

第8節では、適用したニューラルネットワークによる推論システムの学習アルゴリズムについて述べる。1項では、ニューラルネットワークの学習アルゴリズムとして多く用いられている誤差逆伝播アルゴリズムについて示し、2項では、誤差逆伝播アルゴリズムが複雑な問題に対しては不向きなアルゴリズムであるとされていることから、誤差逆伝播アルゴリズムに代わるアルゴリズムとして対向伝播アルゴリズムを適用したことを示す。

第9節において、8節で述べた2つのアルゴリズムを適用して、道路橋 RC 床版の損傷

要因推定を対象ドメインとして推論を行った結果を示す。

第 10 節では、ニューラルネットワークによる推論結果が、本システムでの推論結果と同等の結果を示し、ニューラルネットワークによる推論システムが、十分に損傷要因推定システムの補完システムとして機能することを示す。また、誤差逆伝播アルゴリズムの代替アルゴリズムとして提案した、対向伝播アルゴリズムが有効な推論を行うことも示す。

第 3 章では、2 章での研究結果から、対象問題を道路橋 RC 床版の損傷要因推定とした場合の推論は十分有効な結果を示したことが明らかになったことから、新たに対象ドメインを、鋼橋疲労損傷の補修方法選定として、推論システムが対象問題によることなく機能するかを検証した。

第 1 節では、対象問題を鋼橋疲労損傷の補修方法選定として推論システムの有効性を検討することとした研究の目的を示す。

第 2 節では、鋼橋疲労損傷の補修方法選定を行うにあたり、田中らによる鋼橋疲労損傷の補修・補強方法選定システムで用いられたルールベースを適用した。1 項では、適用した田中らによるルールベースのネットワークの構成を示す。2 項では、1 項で示したネットワークを本推論システムに適用するための仮説の設定を行い、3 項において適用したルールベース内の提示方法を示す。

第 3 節では、2 節で提示したルールベースを用い、推論の入力情報として亀裂の外的要因、内的要因、継手の作用力、亀裂様式を提示して、実際の事例に対する補修方法の選定を行った。1 項では、入力情報の 4 項目を全て観測された既知の事実(既知情報)として、仮説推論を行い、その推論精度の検討を行った。2 項では、1 項での結果を踏まえ、既知情報とした 4 項目のうち亀裂の外的要因を除く 3 項目について場合によって観測が不明であった(未知情報)として推論を行い、推論精度の検討を行った。

また 4 節では、田中らにより開発された、事例ベース推論による補修・補強方法選定システムによる推論結果である事例群を適用して、本推論システムにおいて補修方法の選定を行った結果を示す。

5 節では、これらの検討より、ルールベース洗練機能付診断エキスパートシステムが、知識の共有・再利用における知識獲得問題解決の 1 手法として、対象ドメインによることなく十分有効に機能することが実証できたことを述べる。しかし、ルールベース構築の初期における、知識のルール化に対する知識獲得問題解決の機能を本推論システムは保持していない。

そこで、第 4 章において、知識のルール化に対する知識獲得問題を解決するため、データベースからのルール抽出アルゴリズムの開発として、最近注目を集めているデータマイニング技術を取り入れた、ルール抽出アルゴリズムの開発を試みた。

第 1 節では、知識獲得の初期における、知識のルール化に対する課題を示し、その解決のための 1 手法として、データベースからのルール抽出(KDD)アルゴリズムを提案することを示す。

第 2 節において、はじめに KDD がデータベース分野と知識ベース分野の両分野からの

アプローチにより生まれた技術であることを述べ、データベース、知識ベースの明確な区別をし、両分野の技術の融合の必要性を示す。次に、KDD とデータマイニングの定義付けをし、一般的な KDD プロセス、及びデータマイニングの手法を示す。

第 3 節では、平均圧縮情報量を用いたデータベースからのルール抽出のためのアルゴリズムを示す。1 項では、データベース中の要素の情報量が、ルールを持つことによって情報量が圧縮されることを示し、この圧縮される情報量を基に平均圧縮情報量を求め、抽出したルールの評価値としたことを示す。2 項では、データベース内の事例、及び要素の定義付けをし、3 項において、平均圧縮情報量(ACE 値)のアルゴリズムを示す。

第 4 節において、対象ドメインを 3 章で適用した、鋼橋疲労損傷に対する補修方法選定として ACE 値を評価値としたルール抽出を試みた適用例を示す。1 項において、ACE 値によるルール抽出の妥当性を検討するため、既存ルールベースから仮想事例を作成し、その仮想事例によるデータベースを用いてルール抽出を試みたことを示す。2 項において、仮想事例を基に抽出したルールから鋼橋疲労損傷の補修方法選定のためのルールベースを構築し、3 項において、構築したルールベースを基に補修方法選定を行った結果を示す。4 項では、3 章での補修方法選定結果と 3 項の結果との比較を行い、ACE 値によるルール抽出の有効性の検討を述べる。

第 5 節では、本章で提案した、平均圧縮情報量によるデータベースからのルール抽出が十分有効に行われたことをまとめとして述べる。

1. エキスパートシステムの歴史と特徴 [文献 2]

1.1 人工知能と知識処理研究の歴史

人工知能という術語が最初に使われたのは、1956年のダートマス会議からである。この中で McCarthy, J. や Minsky, M. らは、

“学習やその他の知能のどんな特徴もあらゆる面で原則的には、常に正確に記述でき、機械がそれを模倣できる”

という主張をした。

1960年代になって人工知能研究の中から Simon らによって“あらゆる問題に適用可能な一般的な問題解決機構 (GPS ; General Problem Solver)”が提案され、

また、Minsky は、人工知能研究の課題は、“探索、パターン認識、学習、問題解決・計画”の4つに分類されこの4つにまとめられることが主張された。この主張は、現在我々が注目していることと同じで、非常に優れた主張であるといえる。

しかし、これ以後人工知能は何の役にも立たないといわれはじめ、冬の時代といわれた。しかし、この間に、導出原理 (resolution principle) という機械的な推論原理など、今日の人工知能技術を支える重要な基礎研究がなされていた。

人工知能研究がふたたび注目されるようになったのは、専門家のもつ経験的知識を利用するようになってからである。これが、今日のエキスパートシステムの基本的な考え方である。すなわち、エキスパートシステムとは、

“専門家の持つ知識を計算機に組み込んで人間に代わって、あるいは人間を助けて問題解決を行うことを目的とした計算機システム “

である。初期のエキスパートシステムとしては、1960年代終わりから Feigenbaum らによって開発された有機化合物の構造物を推定するシステム DENDRAL が有名である。このプログラムにはプロダクションシステムの考え方が実用的な問題に適用された最初のシステムであり、この概念は次第に洗練され、MYCIN をはじめ多くのシステムに利用され、現在のエキスパートシステム技術の中心的な技法の1つとなっている。

1970年代になって、エキスパートシステムの分野が拡大するにつれて、それに適した工学的な分野の確立が望まれるようになってきた。そこで“知識は力である”の言葉をもとに Feigenbaum によって知識工学が提唱された。そして、人工知能技術あるいは知識処理技術を新しい計算機利用技術の一つとして、従来の人工知能の研究と一線を画すこととなった。

1.2 人工知能研究と知識処理研究の定義

ここでは、人工知能研究と知識処理研究を区別するため、それぞれの術語の定義をする。

人工知能研究： 研究者が興味を持つ特定の問題を、人間の問題解決方法を忠実にまねて解く方法を研究することである。

知識処理研究： 人工知能研究から得られた記号処理中心の技術を適用して、専門家が扱うような実際的な問題を解くための研究である。

1.3 従来型システムと知識処理システム

計算機上における問題解決において知識が必要なことについては、従来型システムも知識処理システムも何ら相違はない。要は、知識の使われかたに違いがあるといえる。以下で、従来型システムと知識処理システムの違いを示す。

従来型の計算機システムでは、解析に必要な専門知識をアルゴリズムという形で定式化し、最終的にプログラムの形で記述して実行する。その結果、従来型システムでは、知識は全てプログラム中に埋め込まれる。

これに対し、知識処理システムは、プログラムに埋め込まれるはずの知識をできるだけプログラムの外に出して表現し、この外に出された知識を表現する部分を**知識ベース**とする。これによって残されたプログラムは、特定の問題に依存する必要のない制御機構とすることができる。言い換えれば、知識を操作するしくみだけとすることができる。これを**推論機構（推論エンジン）**とする。

知識処理システムで専門家の知識を取り扱うにあたり、これらの知識は、直感に基づく知識、基準に基づく知識、深い知識の3つに分類される。

直感に基づく知識：

専門家ならば特に意識しなくても通常の問題解決に伴う一連の行為を実行できるという種類の知識である。これは、極めて効率の良いのが特徴であるが、反面、この種の知識は、“いうにいわれぬノウハウ”であり、記号的に記述するのは非常に難しいものである。

基準に基づく知識：

手順書などにルール（規則、基準）の形で整理されたものを指す。このような知識は、If ~ then などの形式で記述することが比較的容易で、大量に集めることができる。したがって、現在の技術レベルで有用なエキスパートシステムを開発しようとする場合は、基準に基づく知識をうまく定式化できるかどうか重要な問題となる。しかし、基準に基づく知識だけに頼っていては、経験したことのない状況に対応することができない。この点に、現在のエキスパートシステムの持つ限界が現れている。

深い知識：

以上のような場合、人間は経験則に加え、対象システムの情報をもうまく使って、未経験の事象にも対処している。このような知識を深い知識といい、現状では、この知識を取り扱うことが、人工知能研究、知識処理研究の重要な課題のひとつとなっている。

1.4 エキスパートシステムの特徴

知識処理技術を計算機システムに導入しエキスパートシステムを構築する意義は次の4点にあるといえる。

1. 従来、それぞれ専門家固有のものであった専門知識を計算機システム全体で利用できるように共有化すること。
2. 漠然とした形で保持・記述されていた専門家の知識・思考を計算機で処理できるように明文化すること。

この2つは、エキスパートシステムの本質が、知識や思考の明文化にあるという考えに基づいている。問題解決に使う知識を紙の上に書くことさえできれば専門家の知識を共有できるようになり、知識を容易に扱いうるエキスパートシステムの意義が存在する。

3. 新しいプログラム開発手法として、動的な処理を行う方式とそれに必要な静的な専門知識をうまく分離し、組み合わせること。
4. 新しいプログラム開発手法として、使用方法の決定から実現までの開発期間を著しく短縮できること。

この2つは、エキスパートシステムの開発が従来型システムの開発よりも生産性の高いことに基づいている。エキスパートシステムの開発方式では、利用者の要求をうまく書きとめさえすれば、その段階でシステムを動かしてテストすることができる。すなわち、知識ベースの記述は専門家の問題解決知識の厳密な仕様に相当し、この仕様は推論機構（推論エンジン）の働きで自動的に解釈することができる。

これらの意義を踏まえると、現在の技術レベルで比較的容易に実現できるエキスパートシステムの特徴として次の5つの分野があげられる。

1. **狭い分野**：対象領域の性質が明確であり、狭く深い問題を取り扱う。同時に、専門家の持つ問題解決のための知識を知識処理技術で利用可能な形に整理することができる分野を選ぶ。
2. **高度な専門知識が必要な分野**：少数の専門家でなければ解決できない分野の問題を扱う。さらに、実際にその問題を比較的短時間で解ける専門家が存在しており、計算機システムによってその代替・支援を行うことに意義がある分野を選ぶ。
3. **記号処理が問題解決の中心となる分野**：従来型の数値計算プログラムに加えて、数値情報化しにくい専門家のヒューリスティクス*を、記号処理の原理によってプログラム化・効率化できる分野を選ぶ。
4. **緊急度が低い分野**：計算機システム化するための要請は十分に強いがその開発に時

間・コストをかけなくてはならない問題であり，実験システムとしての性格を持つ問題を選ぶ．また，システムの実行にあたっては，時間変化に伴って対象システムの状況が変化することを前提とする必要がない分野を選ぶ．

5. 対象システムについての知識の利用： 経験則に頼るだけではなく，何らかの意味で対象システムについての知識を利用して問題を解決できる分野を選ぶ．

*ヒューリスティクス(heuristics)： 効率の良いアルゴリズムが不明な場合や，存在しない場合に，コンピュータで問題を解くには，対象領域の性質をうまく利用して状態空間を調べるという方法を使う．このようなやり方をヒューリスティクスという．

2. エキスパートシステムの基本概念 [文献 2,27]

2.1 知識ベースと推論機構 [文献 28]

一般的なエキスパートシステムは、専門家の経験的知識などを整理・収集して作られる知識ベースと、この知識に基づいて推論処理を行う推論機構の2つの主要なコンポーネントから構成される。図 1.1 に従来型システムとエキスパートシステムとを比較してエキスパートシステムの特徴を示す。

エキスパートシステムは、図 1.1 にも示したように、知識の追加・修正が比較的容易で柔軟に対応できる。しかしその反面、知識(ルール)ベースシステムにおいて、知識をいかにルールとして表現するかは、最も重要な問題の1つであり、知識をモデル化するためにその表現方法に対し、考慮すべき点は、以下の2点である。

一様な表現: 推論制御機構が簡潔なものとなり、知識の管理を容易にすることができる。
わかりやすい表現: 知識を提供する技術者、またそれを利用する利用者の両者が扱いやすいものでなくてはならず、そうしなければ、知識の獲得や知識の評価に支障をきたすことになる。また、取り扱う問題が大規模で複雑な場合、知識の構造化とモジュール化が不可欠となってくる。

これらの点を考慮した代表的な知識表現モデルを以下に示す。

- (1) 論理モデル (Logic-Based Model)
- (2) ルールモデル (Rule-Based Model)
- (3) フレームモデル (Frame-Based Model)
- (4) 意味ネットワークモデル (Semantic Network Model)
- (5) オブジェクト指向

(1) 論理モデル (Logic-Based Model)

論理モデルは、1階述語論理の体系の中で知識を表現し、三段論法によって推論を行うモデルである。論理モデルの特徴は、知識表現モデルの中で、唯一つの理論的根拠を持ったモデルであり、形式的に厳密な定義と推論形体が実現できることである。しかし反面、理論的根拠を持つために、比較的規模の小さい問題しか取り扱うことができないモデルであるといえる。

(2) ルールモデル (Rule-Based Model)

ルールモデルは、俗にプロダクションルールといわれ、専門知識、もしくはその断片の知識を If-then ルールの集合で表現されるモデルである。この表現形式は、文章として書かれることが多く、知識の記述が比較的容易にできるという特徴がある。この知識表現モデルを用いて構築した知識ベースシステムをプロダクションシステムといい、多くのエキスパートシステムで用いられている推論アルゴリズムである。

なお、プロダクションシステムについては、次節で詳しく述べることにする。

(3) フレームモデル (Frame-Based Model)

フレームは、知識に構造を持たせて表現する方法である。すなわち、特定の対象や事象に関する情報を意味的に関連させ、しかも個々の情報の独立性（モジュール性）が高くなるようにまとめて蓄積するデータ構造となっている。

この概念は、M. Minsky によって考案された概念で、一般にエンジニアリングシステムの構成情報のような静的な情報を表現するのに向いている。フレームは、図 1.2[文献 2]に示すように、適当な項目名が記述されたカードとそれらの関係を紐で結んだ構造となっている。

(4) 意味ネットワークモデル (Semantic Network Model)

意味ネットワークは、人間に連想記憶のモデルとして提案されたものであるが、これを知識表現形式として利用したものが多く、宣言的な知識の記述に適しており、階層性、継承といった性質を表現するのに適している。

意味ネットワークは、対象を個体や概念を表すノードと、これらノード間の関係を表すリンクで構成されるネットワークとして表現したものである。

(5) オブジェクト指向

知識を内部状態、メッセージ送信機能、メソッドからなる抽象データ型で表現したモデルである。

2.2 プロダクションシステム

プロダクションルールについては、より詳述するとともに、これを利用したプロダクションシステムについても述べることにする。

1950 年代中頃から AI の研究が盛んになってきたが、1965 年にスタンフォード大学の Feigenbaum らが開発した DENDRAL という質量分析計のデータからの分子の化学構造式の決定を支援するエキスパートシステムが、後の AI 研究に大きな影響を与えることになった。推論方式に研究の重点が置かれていたそれ以前の研究と違って、むしろ問題としている対象に用いられる専門知識の利用に注目したことが、このシステムのそれまでと違う点である。

2.2.a プロダクションルール

[IF (条件)]ならば [THEN (結果) である。 / (行動) しろ。]

プロダクションルールとは、知識を上のように単純な 1 種類の構文で表現したものであり、我々人間の知識の多くは、この表現方法の集合体として表すことができる。この表現方法において、条件部の項目の手続きは、1 つのルールの中に 2 つ以上あっても構わない。以下にプロダクションルールの特徴を示す。

1. 専門家も含め我々が日ごろ慣れ親しんでいる表現であって、直感的で理解しやすい。
2. 表現方法が IF-THEN と単純で 1 種類に限られているため制御機構が単純化できる。
3. 1 つ 1 つのルールのモジュール性がよく、ルールの追加・修正が容易である。

エキスパートシステムを構築する際、はじめに必要な最低限度の専門知識を専門家から獲得し、その知識に基づいて小規模なシステムを構築する。次に、そのシステムを専門家に使用してもらうことにより知識の追加・修正を行い、より高度なシステムに拡張してエキスパートシステムを完成させる。

しかしこの時、専門家が必ずしもコンピュータの知識を多く持ち、プログラム言語に精通しているとは限らないため、システムに用いる専門知識が専門家にとって単純明快に表現されていなくては、知識の獲得や整理・管理だけでなく知識の追加・修正などにも大きな障害をもたらすシステムの開発効率を低下させることになる。

以上のことから、単純で明確に記述できるプロダクションルールは、知識の表現方法として有効であると考えられる。

2.2.b プロダクションシステム

プロダクションシステムとは、プロダクションルールに基づいた知識ベースを処理する推論システムであり、多くのエキスパートシステムで、この推論機構が用いられている。しかし、実用的なプロダクションシステムを構築するためには、IF 部と THEN 部とを結ぶための機構を複雑化しなくてはならず、また組み合わせ爆発にも対処しなくてはならないといった問題も残されている。

以下に、最も基本的なプロダクションシステムの構成について述べ、その関係を図 1.3[文献 2]示す。

1. ルールベース：IF(前提・条件)、THEN(結論・結果)の形をしたプロダクションルールの集合で構成される。
2. 作業領域(ワーキングメモリ)：観測事実・推論途中経過を保持する。
3. 推論機構：作業記憶の状態を見て適合するルールを取り出し実行する。
4. 競合解消：推論で2つ以上のルールが適用可能なとき、最適ルールを選択する。

また、プロダクションシステムは、得意とする対象問題により、次の対照的な2つの形体をした推論システムに分けることができる。

1. **AND/OR 木型**：分析型(診断型)の問題解決に適する。
2. **認識一行動サイクル型**：合成型(設計型)の問題解決に適する。

AND/OR 木型プロダクションシステム：

ルール全体を事実と結論を結び付ける AND と OR の木で構成し、この木を評価して推論を行う。推論の基本的な方法には、**前向き推論**、**後ろ向き推論**、**双方向推論**がある。

前向き推論：

与えられたデータから始まり、途中で否定されたノードで評価を中止し最後まで残ったときの木の最上位に相当する仮説を結論として採用する。この方法では、結論に直接関係のないデータの評価が多くなるという無駄が起こってくる。

後ろ向き推論：

結論に関係のある部分木のみが評価されるという点においては無駄を省くことができるが、推論中になかなか否定も肯定もされないとき無駄に木を生成してしまうという問題が生じる。

双方向推論：

まず少量の与えられたデータを評価して仮説を選び(前向き推論)、次にこの仮説が妥当であるかどうかを判定するために必要なデータを要求し評価する推論方法である。この推論方法は、前の2つより柔軟性の高いシステムの実現が可能となる。

認識－行動サイクル：

プロダクションルールの前提部において真または偽の値がとられるわけであるが、前提部で真の値をとったときのみ作業記憶に対して記号の追加や削除などの更新を繰り返すことを言う。この型のシステムは、以下の3要素を含むシステムである。

ルールベース：プロダクションシステムの集合

データベース：事実の集合

インタプリタ：推論を実行する推論機構

推論は、図 1.4[文献 27]に示すような“認識－行動サイクル”の繰り返しとして実行され各サイクルで選択されたルールの実行部がデータベースを更新していく。この結果データベースの内容が初期状態から目標状態へと変換されていくことになる。この時適用できるルールが複数存在するときの競合解消については以下のようないくつかの方式が用いられる。

1. 重要度優先方式
2. 詳細ルール優先方式
3. 最近実行ルール優先方式
4. 最近データ優先方式
5. 並列実行方式

2.2.c プロダクションシステムの長所・短所

プロダクションシステムでは、ルールの選択と実行・評価の単純な繰り返しが推論のほとんどになり、その度ごとにルールベースに対して照合操作を行う必要が出てくる。そのためルールの数が多くなると推論速度が極端に遅くなってしまふ。したがって、大規模な問題を取り扱うにあたっては、推論機構に対する工夫が必要となってくる。

ここで考えられる工夫は、ルールのグループ化とデータベースの構造化である。推論過程において関連あるルールを前もってグループ化しておきルール選択の範囲をある程度限定しておく。また、ルールが作用する対象も一般に階層性を持つ場合が多いので、データベースを構造化しておくことが推論速度の向上に有効である。

以下にプロダクションシステムでよく知られる長所と短所を示しておく。

長所：

1. 個々のルールは作りやすくわかりやすい
2. 知識の追加・修正が容易である
3. 推論機構の表現が簡単である

短所：

1. ルールの相互関係が不明瞭である
2. 知識の全体像がつかみにくい
3. 処理効率が悪い
4. 推論の柔軟性に欠ける

2.3 不確実な情報の取り扱い

従来型の数値計算を主体として問題解決を計る場合、対象とする問題・現象のモデル化を行いその解法を定式化するという手順でプログラムを作成して解を求めるため、プログラムにおいて不確実な情報を取り扱う必要性がほとんどないのが普通である。これに対し、エキスパートシステムの手法においては、「専門知識」を定式化する時点で対象システムの近似を明確に意識せずに打ち込むことになり、また宣言的に記述した知識をそのまま記号的に操作するため、非常に複雑な処理が必要となってくる。つまり、エキスパートシステムでは、不確実な情報を取り扱うにあたり、専門知識の近似の容易な実現が必要となってくる。

そこで、エキスパートシステムの推論機構では、推論処理を効率的に正しい解を導く方法として近似度を表す不確実性の尺度を用いることがある。特に、診断型エキスパートシステムでは、全てのデータが得られておらず疑わしいデータや誤った(負)のデータを解釈するための知識が不備な場合、不確実性の尺度が重要となってくる。

2.3.a 確信度

確実性を扱う手法としてよく知られており、エキスパートシステム構築ツールの多くに採用されているのが**確信度(Certainty Factor)**の概念である。確信度は漠然とした判断に基づく推論を表現するために考え出されたものである。

確信度は、IF-THEN ルールの形式で整理した知識の断片に、その確からしさを表す情報として-1 から 1 までの値を付与しておき、推論処理に生じる曖昧さを定量的に判断する方法である(図 1.5[文献 2])。

前提部の確信度がそれぞれ違った形で与えられたとき、結論部の確信度の決定方法を図 1.6[文献 2]に示す。また、2 つ以上の根拠によって 1 つの結論が支持、あるいは反証されるとき、それぞれ図 1.7[文献 2]の方法により結論部の確信度を決定する。

2.3.b Dempster-Shafer 理論 [文献 27,29]

1967年,A.P.Dempsterにより,統計学の論文として基本確率,上限確率,下限確率という概念を提案して不確実な可能性の表現をした.その後1976年G.Shaferがこの概念をエキスパートシステムの曖昧測度として理論を整理し,下限確率を確度,上限確率を尤度としてエキスパートシステムに導入した.

(1) 基本確率

一般に確率を定めるとは,図 1.8 のように,全体が 1 の重さを起こりうる**各事象**に割り振るものと考えることができる.これに対し,Dempster-Shafer理論では,図 1.9 のように,全体の重さの割り振りを各事象にではなく全事象集合の**部分集合**のそれぞれに対して行う.ここで割り振られる重さを**基本確率**と呼ぶ.つまり,各部分集合に割り振られた基本確率は,次の2つの場合が考えられる.

1. 1つの事象からなる部分集合

2. 複数の事象からなる部分集合

1 の場合は一般の確率と同じようにその事象が起こると判断される重さを表す.

2 の場合は,部分集合内の事象のどれかが起こると判断される重さを表すが,そのうちのどれが起こるかを判断することができない重さを表すことにもなる.

ここで,全事象集合を A_0 , A_0 の部分集合を A_1, A_2, \dots, A_n とすると,このときの基本確率 $m(A_i)$ は, $[0,1]$ の値をとり,式(1)の条件を満たす.

$$\begin{cases} m(\phi) = 0 \\ \sum_{A_i \subseteq A_0} m(A_i) = 1 \quad \dots(1) \end{cases}$$

(2) 下限確率, 上限確率

下限確率・上限確率共に,基本確率をもとに全事象集合の部分集合のそれぞれに割り振られる.ある部分集合に割り振られた下限確率は,その部分集合の要素となる事象のどれかがおきる確率の下限を表す.例えば $\{a_1, a_2\}$ に割り振られる下限確率が 0.3 のとき,これは,“ a_1 か a_2 の起きる確率は少なくとも 0.3 である”であることを表す.

また, 上限確率はその集合の要素となる事象のどれかが起きる確率の上限を表す. 例えば, $\{a_1, a_2\}$ に割り振られた上限確立が 0.6 のとき, これは “ a_1 か a_2 の起きる確率は多くとも 0.6 である” ことを表している. 下限確率を $PL(A_i)$, 上限確率を $PU(A_i)$ とすると, これらは基本確率を用いて式(2),(3)で定義される.

$$PL(A_i) = \sum_{A_j \subseteq A_i} m(A_j) \quad (2)$$

$$PU(A_i) = 1 - \sum_{A_j \subseteq A_i^c} m(A_j) = 1 - PL(A_i^c) \quad (3)$$

この定義より下限確率は, 部分集合に必ず割り振られる重さを表し, 上限確率は少しでも部分集合に割り振られる可能性のある重さを表すことが分かる.

このように Dempster-Shafer 理論は, 確率論的な裏付けの基に不確実性を表現している.

2.3.c ファジィ推論 [文献 2,30]

人間が日常行う推論には、言葉の意味や概念自体に曖昧さを含んでいることが多い。従来、あいまいさを確率的手法により定量的に取り扱ってきたが、この方法では存在するすべての曖昧さを取り扱うのに限界があるため、ファジィ理論が提案されたと考えられている。

L.A.Zadeh は、このような主観的意味を含む曖昧な知識に関して、ファジィ理論を提唱し、数学的表現と演算方法で推論する方法を提案した。これ以後、理論と応用の両面から研究が進められて、多くのエキスパートシステムで適用されるようになった。

ファジィ理論において通常は、ルールに用いる用語そのものに曖昧な性質を認め、エキスパートの主観的な曖昧さがもつ知識を表現してルールベースとし、その処理に先に述べた Dempster&Shafer 理論を導入して推論を行うことが多い。

(1) ファジィ理論

ファジィ論理は、真と偽(0,1)のどちらかの値をとる通常の論理学を拡張した考え方であり、“xは大きい数である”といった主張を [0,1] の実数区間の値で特徴づけたものである。例えば、“xは大きい”を次のような値で表現すると次のようになる。

$$\begin{array}{l}
 \text{xは大きい} \\
 \begin{array}{l}
 0 \leq x \leq 10 \quad \Lambda \quad 0.1 \\
 10 \leq x \leq 1000 \quad \Lambda \quad 0.2 \\
 1000 \leq x \quad \Lambda \quad 0.7
 \end{array}
 \end{array}$$

ここで 0.1,0.2,0.7 の値は、それぞれの主張の確からしさを表す値となり、これらの値はファジィ論理ではメンバーシップ関数で与えられ、[0,1] の実数区間の値を与える。メンバーシップ関数とは、ある集合の特性を表す特性関数である。

(2) ファジィ制御によるファジィ推論

ファジィ推論は、if~then~というファジィ条件命題に基づいて複数個のルールが用いられて推論が行なわれる。この形式で記述された言語的制御規則をファジィ制御規則といい式(1) で表される。

$$\text{if "x is } \tilde{A}_i \text{" and "x' is } \tilde{A}_{2i} \text{" then "y is } \tilde{B}_i \text{"} \quad (1)$$

(i = 1, 2, ..., N)

式(1)で $\tilde{A}_i, \tilde{A}_{2i}, \tilde{B}_i$ はそれぞれ「大きい」「小さい」「高い」等のファジィ変数を表していて、図 1.10[文献 30]のような実数区間を全体集合としたファジィ集合で与えられる。ここではメンバーシップ関数に式(2)を用いてファジィ集合を三角型で表現した。実際には他に釣鐘型や指数型による表現が用いられることもある。

$$\mu_{\tilde{A}} = \begin{cases} \frac{1}{r-p}(x-r)+1 & (x \leq r) \\ -\frac{1}{p-r}(x-r)+1 & (x > r) \end{cases} \quad (2)$$

$\mu_{\tilde{A}}(x)$: \tilde{A} に関するメンバーシップ関数 r, p : パラメータ

(3) ファジィ制御の推論方法

ファジィ制御の推論方法には、直説法や間接法といった推論方法があるが、ほとんどの場合、推論には直説法が用いられていることから、ここでは直説法による推論方法を示すことにする。ここでは、以下に示すファジィ推論を対象として直説法を考えることにする。

前提1 $\tilde{P}^+ : "x \text{ is } \tilde{A}^+"$

前提2 $\tilde{P} \rightarrow \tilde{Q} : "if \ x \text{ is } \tilde{A} \text{ then } y \text{ is } \tilde{B}"$

結論 $\tilde{Q}^+ : y \text{ is } \tilde{B}^+$

\tilde{A}^+, \tilde{A} は集合 $X=\{x\}$ 上のファジィ集合

\tilde{B}^+, \tilde{B} は集合 $Y=\{y\}$ 上のファジィ集合

直説法では、条件命題 $\tilde{P} \rightarrow \tilde{Q}$ を集合 $X \times Y$ のファジィ関係 $\tilde{R}_{\tilde{P} \rightarrow \tilde{Q}}$ によって定める。また、

$\tilde{P} \rightarrow \tilde{Q}$ はファジィ制御規則を用いると $\tilde{A} \rightarrow \tilde{B}$ となることから $\tilde{R}_{\tilde{P} \rightarrow \tilde{Q}}$ は $\tilde{R}_{\tilde{A} \rightarrow \tilde{B}}$ と表すことができる。

よって、結論 \tilde{Q}^+ のファジィ集合 \tilde{B}^+ は $\tilde{R}_{\tilde{A} \rightarrow \tilde{B}}$ と \tilde{A}^+ との間の $\text{sup} \cdot \text{min}$ 合成演算によって式(3)のように表すことができる。

$$\tilde{B}^+ = \tilde{A}^+ \circ \tilde{R}_{\tilde{A} \rightarrow \tilde{B}} \quad (3)$$

ここで $\text{sup} \cdot \text{min}$ 合成演算は、それぞれ以下のように定義されている。

$$\text{min 演算 } \wedge : 1 \wedge 1 = 1, 1 \wedge 0 = 0 \wedge 1 = 0, 0 \wedge 0 = 0$$

$$\text{sup 演算 } \vee : 1 \vee 1 = 1, 1 \vee 0 = 0 \vee 1 = 1, 0 \vee 0 = 0$$

また、 $\tilde{R}_{\tilde{A} \rightarrow \tilde{B}}$ を集合 $\tilde{A} \times \tilde{B}$ 上のファジィ関係と考えると式(3)は式(4)で書き換えることができ、これをそれぞれのメンバーシップ関数で書き換えると式(5)のようになる。

$$\tilde{B}^+ = \tilde{A}^+ \circ (\tilde{A} \times \tilde{B}) \quad (4)$$

$$\begin{aligned} \mu_{\tilde{B}^+} &= \vee_{x \in X} (\mu_{\tilde{A}^+}(x) \wedge \mu_{\tilde{A} \times \tilde{B}}(x, y)) \\ &= \vee_{x \in X} (\mu_{\tilde{A}^+}(x) \wedge \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(y)) \quad (5) \\ &= (\vee_{x \in X} \mu_{\tilde{A}^+}(x) \wedge \mu_{\tilde{A}}(x)) \wedge \mu_{\tilde{B}}(y) \end{aligned}$$

式(5)は、2つのファジィ集合 \tilde{A}^+, \tilde{A} の積 $\tilde{A}^+ \cap \tilde{A}$ からメンバーシップ値の最大値 α を求め、 α と \tilde{B} との積によって \tilde{B}^+ のメンバーシップ関数を決定している。この推論の流れを図 1.11[文献 30]に示す。

2.3.d ニューラルネットワーク [文献 2]

ニューラルネットワークシステムは、人間の脳機能を真似た仕組みを工学的に実現しようとしたもので、神経細胞モデルの細胞の繋がりが自立的に形成されるメカニズムに注目したものである。ニューラルネットワークを構成する単位はニューロンで、ニューロン同士は、シナプスを介して結合している。このようにニューラルネットワークは、ニューロンの相互結合から成り立っており、ニューロンの数学モデルとして考案されたアルゴリズムである。

ニューラルネットワークシステムは、事例でしか表せない知識をシステムに与え、システムの持つ学習機能を利用することで、専門家の判断論理をシステム内に生成させようとするものである。この学習機能は、事例データから複雑な判断論理を作り上げるため、ルールベースシステムやファジィ推論システムでは適用が困難であった対象ドメインに対して、有効に作用すると考えられている。

2.3.e 事例ベース推論(CBR) [文献 3]

人間は、新たな問題に遭遇した時、過去に経験した大きく類似している状況と、その時とった問題解決のための行動を思い起こし、その行動を修正して新たな問題に対処しようとする。

このように、失敗も含んだ過去の経験または、事例に基づいて現在の問題を解決することを事例ベース推論(Case-Based Reasoning)という。

(1) 事例ベース推論によるシステムの開発要求

ルール型エキスパートシステムに代表される知識情報処理システムが、実用化段階に入り、以下に挙げるような問題点が指摘されるようになった。

1. 知識情報処理システムの問題解決能力が融通性と柔軟性に乏しい
2. 人間が“実経験の蓄積と記憶”に基づいて問題解決を図る際の適応能力と比較して格段の隔たりがある
3. 従来の教科書的な知識は獲得・記述は比較的容易であったが実用的ではない
4. より人間の思考・推論の形態に類似した知識処理技術の開発要求
5. 専門家固有の直感的な問題解決を裏付けるような知識の効率的な獲得の要求
6. 熟練技術者が各種の複雑な制御タスクを苦もなくやってのける際に、暗黙のうちに用いているような技能知識の効率的な獲得

これらの問題点が指摘される理由として、実用化を目的とする以前は、対象問題が比較的単純で、知識の記述も明確に行えるものであったのに対し、実用化を目的とすると、対象問題の構造が複雑になり、また専門家や熟練技術者の有する知識が直感的で不透明であるため、知識を定式化して記述することが困難であることが挙げられる。この問題

を解決するのに要求されてくることは、新たな問題に対し、いかに過去に収集された事例や知識を活用しながら問題解決を行うかにあるといえる。

そこで、これらの問題点を解決しうる推論機構として、事例ベース推論が注目されるようになってきた。

(2) 事例ベース推論の特徴と推論機構

事例ベース推論による知識情報処理システムでは、次のような推論処理を行うことから、知識のルール化を行う必要がないため、(1)で述べた知識の定式化をエキスパートシステム構築の作業の中から省くことができる。このため、事例ベース推論による推論機構を適用することにより、現状のエキスパートシステムが持つ、知識の獲得の困難さを低減しうると考えられている。

事例ベース推論の構成と推論処理(図 1.12 [文献 31])

1. 図中の問題解析部で、入力されてくる新たな問題から事例検索性情報を取り出す。
2. 事例検索性部で、事例検索性情報を用いて類似事例の検索性を事例ベースから行う。
3. 検索性された事例中の解が、直接新たな問題に対して適用可能ならば、それを解とし、直接適用できないならば、解修正部において、解の修正を行う。
4. 3で解の修正に失敗した場合、その失敗の原因を解析し、その原因を取り除くことが可能であれば、解修復部において解の修復を行う。
5. 事例格納部では、扱った問題が、事例ベース内に格納されていない特徴や、失敗を持っていた場合、それに応じた事例検索性情報を書き換えて次の問題に備えて、事例ベースに格納する。

事例ベース推論では、以上のような推論機構により推論が行われている。しかし、解の修正、解の修復に関しては、人間による領域知識を用いてその作業を行っているのが現状である。

次に、事例ベース推論の長所と短所を以下に示す。

事例ベース推論の持つ長所

1. 知識獲得の容易性：事例ベースの構築は、実事例を蓄積するので、知識をルール化するのに比べて容易に知識を獲得することができる。
2. 部分照合：事例ベース推論では、新たな事例に対する類似事例の検索性をすることから、部分的に一致する事例を検索性でき、解の示唆を得ることができる。
3. 失敗事例の利用：失敗原因を一般化することで、失敗の予測を行うことができ、それに基づいて失敗を回避した推論をすることができる。
4. 効率的な探索が可能：0 から出なく、類似事例を検索性し、それを修正することから、より新たな問題に対する解に近いところから、探索による推論を行うことができる。

事例ベース推論の持つ短所

1. 解空間の包含性が不明：現在与えられている事例ベース内の事例で、すべての解空間が包含されているかが不明なため、得られた解が最適解である裏付けができない。
2. 事例間の整合性の欠如：矛盾した事例が、事例ベース内に存在した場合に、その矛盾を検出することができない。

3. エキスパートシステムの開発技術 [文献 2]

エキスパートシステムを開発する上で、従来型のプログラム開発と違った特有な部分は、知識をいかに収集・整理して、記号的に記述するかという点にある。しかし、エキスパートシステムの開発技術は、現在のところ完成されたものとはいえ、個々のシステムに応じて様々な方法が取られている。

3. 1 エキスパートシステムの開発手順

エキスパートシステムを効率良く開発するためには、1. 4でも述べたが、次のような条件を満たす適切な問題を選択することが重要と考えられる。

1. 狭い分野
2. 高度な専門知識が必要な分野
3. 記号処理が問題解決の中心となる分野
4. 緊急度が低い分野
5. 対象システムについての知識が利用できる分野

上の 5 つの条件を満たすような対象問題が決定したとき、実際のシステム開発を行うにあたっては、通常、ウォーターフォール型の開発手法、あるいは、スクラップ・アンド・ビルドを繰り返す段階的開発方式が取られる。

ウォーターフォール型の開発手法とは、ソフトウェアのライフサイクルを図 1. 13[文献 2]に示すようなフェーズに分け、各段階毎に作業が完結するように、トップダウン的に作業を行っていく方式である。

この方式の利点は、開発の比較的早い段階で利用者の要求が確定することから、ライフサイクルコストの削減に役立つことにある。しかし、各フェーズごとに作成される文書の量が膨大になるため、システム開発中において利用者の要求の変化に対応することが困難であることが欠点といえる。

段階的開発方式とは、図 1. 14[文献 2]に示すように、開発の途中でプロトタイプシステムを順次、利用者(専門家)に提示して、機能を充実させていくという、手探りの開発をしていく方式である。この方式は、エキスパートシステムが専門知識という把握しにくいものを計算機に取り込み、また開発当初から要求仕様を厳密に定めることが困難であるという性質上、ウォーターフォール型の開発手法よりも適しているとされている。

一般に、この開発手法を探索型プログラミングという。この手法は、問題解決手法や実現アルゴリズム自体が不明確であるという、特に、エキスパートシステム開発の初期段階において、システム化の方式を定める目的で試行錯誤を繰り返す方法である。

探索型プログラミングにおいて、専門知識の断片を If-then 型のルールで記述することを考えると、このルールは、構築されるシステムの動きの一部をになうことから、システムの仕様の一部を明確化したことになる。しかも、記述したルールは、そのまま推論機構により実行することができることから、システム構築に対する要求や、解決が不明

確な状況において、手探り的にプログラム開発が行えることになる。これが探査型プログラミングの利点である。

これにより、エキスパートシステムを開発する上で、次のような効果が得られる。

1. 推論システムの検証を、ルールの実行という形で達成できるため、複雑な問題の取り扱いにおいても、テストを比較的容易に行える。
2. 開発速度を速めることができるため、利用者の要求の変化に合わせた、柔軟なシステムの構築ができる。

3. 2 知識の収集・整理手法

エキスパートシステム開発において、最も困難であり、現在でも大きな課題として残っているのが、知識の収集・整理など知識獲得に関する部分である。

専門家から知識を獲得するには、必要な情報を専門家からのインタビューにより獲得したり、問題解決の過程で発話される情報を分析するプロトコル分析などの方法がある。しかし、エキスパートシステムの開発において、問題解決の知識を確定するまでに数多くの議論の必要な場合が多く、専門家→インタビュー→知識技術者といった単純な構造では知識獲得は困難であり、実際に知識獲得を行うには、専門家と知識技術者との間の協力関係が重要になる。この協力関係を結ぶ方法として、次の3通りのパターンが考えられる。

1. 専門家の側が、知識技術者の技術を身につけ、エキスパートシステムの開発を主導する方法
2. 知識技術者が、専門家の作業を学びながらエキスパートシステムの開発を行っていく方法
3. 問題の対象分野の専門家と知識技術者との間に、両分野の述語や概念をある程度理解できる橋渡しの技術者をおいてエキスパートシステムの開発を行う方法

1. は、現在、最もよく取られている方法といえる。現在のエキスパートシステムの研究者は、もともとが他の分野の専門家であったケースが多く、土木分野においても、そのほとんどが土木技術者の側からエキスパートシステムの開発を行っているといえる。

2 の場合、明文化しにくい専門知識を知識技術者自身が理解することから、これらの専門知識をシステム構築に適した表現に変換・蓄積しやすいという利点がある。このため、この方式は、公共構造物の診断問題のような息の長い問題を取り扱う場合において、エキスパートシステムの拡張やメンテナンスを継続していく上で重要な方法となる。

3 の方式は、特にシステム開発の初期において、専門家と知識技術者の議論を円滑に進める際に重要といえる。

次に、獲得した知識を知識ベースに整理して組み込む方法について述べる。

一般に、人工知能や知識処理の概念は、応用分野の特性に依存せず、トップダウン的に与えられるものが多いため、目標とするエキスパートシステムで利用する概念に合う

ように専門知識を変換することが知識を記述する段階において重要な作業となる。

そこで、知識を知識ベース内に効率良く獲得し、記述していくために、知識表現・記述に関して、以下のようなことが必要となってくる。

1. 知識記述のための単語とその意味を明確に定める
2. 広範な知識を知識ベース化するために、知識ベース間の相互関連性を明確にし、正しく定める

次に、ある程度獲得された知識を記述する方法を考える。現在のエキスパートシステムにおいてよく用いられているのが、KJ法や決定表などである。

KJ法：

ボトムアップ的な知識の記述・整理手法であり、知識が収集された時点では、知識間相互の関連性が明らかでない個々の情報(個々の具体問題や、様々な見解など)を整理・記述し、情報間の相互の関連性を見出していく方法である。

決定表：

知識を記述する用語、それに関する条件及び、条件が成立した場合に行うべき処理内容を1つの表にまとめて表した手法であり、これは、ルールを記述する際に生ずる網羅的な条件を漏れなく記述することが可能である。

これにより、知識、特にルールの抜けを防ぐことができ、また表の形を整理することで、必要なルールの数もある程度少なくすることが可能である。

このほか、知識の記述・整理の手法には、システム工学的な手法として、ISM、DEMATEL法、FTAやETA等もある。しかし、これらのシステム工学的なアプローチによる手法は、基本的に人間が知識を整理・記述して、人間が利用することを前提に提案されているため、記述された知識を直接計算機で操作できるほど意味を明確にしていないため、これらの知識をエキスパートシステムで用いる、知識表現手法によって推論処理を実行できるレベルに“不確実な情報の取り扱い”等の手法を用いて、改めて変更しなくてはならない。

3.3 対象システムのモデル化

土木構造物の診断などエンジニアリングシステムを対象とするエキスパートシステムの場合、システム自体が、問題解決に必要な経験則の他に、構造物の構成要素といった対象システムそのものの知識も備えている必要がある。そのため、2節で述べたような、獲得された知識を通常は、フレームやルールなどの概念により、対象をモデル化することを行っている。これらのモデルは、推論処理に利用されるだけでなく、専門家向きの利用者インターフェースとしても役立っている。

すなわち、対象システムのモデル化は、知識を提供する側の専門家にとっては、知識の獲得を支援するものであり、知識を利用する側の専門家にとっては、専門知識の伝達を容易にするものとなる。

この2つについて、Faught.W.Sは、前者の役割を、専門家用ワークステーションとし、後者の役割を、エキスパートシステムとして区別している。そして、この両者の共

通の課題として、専門知識をどのようにして陽に表示するか、すなわち、対象とする問題をどのレベルで抽象化、モデル化するかという点にあるとしている。

3. 4 従来型システムとの結合

エキスパートシステム技術は、知識を陽な形で表現することで、従来ならば困難であった問題解決を可能にするものである。しかし、現状の技術レベルでは、本質的に、知識を提供した専門家であれば正しい解を出せるレベルのシステムにとどまっているといえる。これに対し、従来型の計算機システムは、大量かつ高速な計算処理を行えるため、複雑な問題にも大きな役割を果たしている。

そこで、土木構造分野においても、実用的なエキスパートシステムを開発していくためには、経験的知識を利用するエキスパートシステムの手法に、従来型システムの持つ高速計算処理の機能などをうまく取り込んでいく必要がある。エキスパートシステムに結合する従来型システムには、大きく数値計算システム、データベースシステム、利用者インターフェースシステムの3種類が考えられ、以下にこれらのシステムとの結合による一般的な利用方法を述べる。

1. 数値計算システムとの結合

数値計算システムを結合することによる利点は、その計算能力を生かして、経験則だけで解決できない構造物の状態を詳しく知るのに役立つことにあると考えられる。

数値計算プログラムを結合する方法には、数値計算プログラムをメインシステムとし、パラメータの設定など、問題解決の一部にエキスパートシステムの技術を用いる方法と、逆に、エキスパートシステムをメインシステムとし、サブルーチンとして数値計算を行う方法が考えられる。前者は、エキスパートシステムの持つ変更容易性を利用することになり、後者は、エキスパートシステムの判断能力を利用することになる。

2. データベースシステムとの結合

データベースの利用は、データベースシステムにより、大量の情報を容易に表現、蓄積できることから、フレームやルールなどの知識ベースの構築の際に大変有効と考えられる。

データベースシステムの利用については、データベースからのルール抽出技術として、後の章で詳しく述べることにする。

3. 利用者インターフェースとの結合

利用者インターフェースについては、エキスパートシステムに求められている、“高度なユーザインターフェースによる利用”と強く関係するものである。すなわち、エキスパートシステムに求められている“柔軟な対話型処理機能”の実現には、推論処理の過程の説明や、図示など従来型システムに大きく依存するものが多い。そのため、従来型システムを用いた利用者インターフェースの結合を欠かすことはできないと考えられる。

3. 5 エキスパートシステム構築ツールの一般的構成と現状

エキスパートシステムは一般に、図 1.15[文献 2]に示すように知識ベースと推論機構を中心に、推論制御を行うアジェンダ機能、判明した事実や推論の目標を保持する作業記憶、大量の関連情報を保持するデータベースなどにより構成される。

以下に主なエキスパートシステム構築ツールについて述べる。

[1] ルールベースシステム

ルールベースシステムでは、図 1.16[文献 2]のように、知識ベース（ルールベース）、推論機構、競合解消のためのアジェンダと、作業記憶を主な機能として持つ。ルールベースシステムでは、データベース機能は省かれることが多い。

[2] フレーム、オブジェクトベースシステム

図 1.17[文献 2]のように、作業記憶とデータベースの機能をフレーム、またはオブジェクトの概念で表している。推論制御機能については、付加手続きやメソッドとして表している。

[3] ブラックボードベースシステム

図 1.18[文献 2]のように、作業記憶とアジェンダの機能を、構造を持ったブラックボードの機能を用いて表している。そして、知識ベースが幾つかの知識源として分割されているのが特徴である。

このように、エキスパートシステム構築のため、上に述べたようなツールが用いられしており、現在では、これらの特徴をそれぞれ生かして幾つかのツールを複合的に用いているものも少なくない。

エキスパートシステムは、ルールを利用して経験的知識を記述し、これを対象分野の知識と組み合わせて利用するといった、新しいプログラミング技術として確立してきたように考えられる。

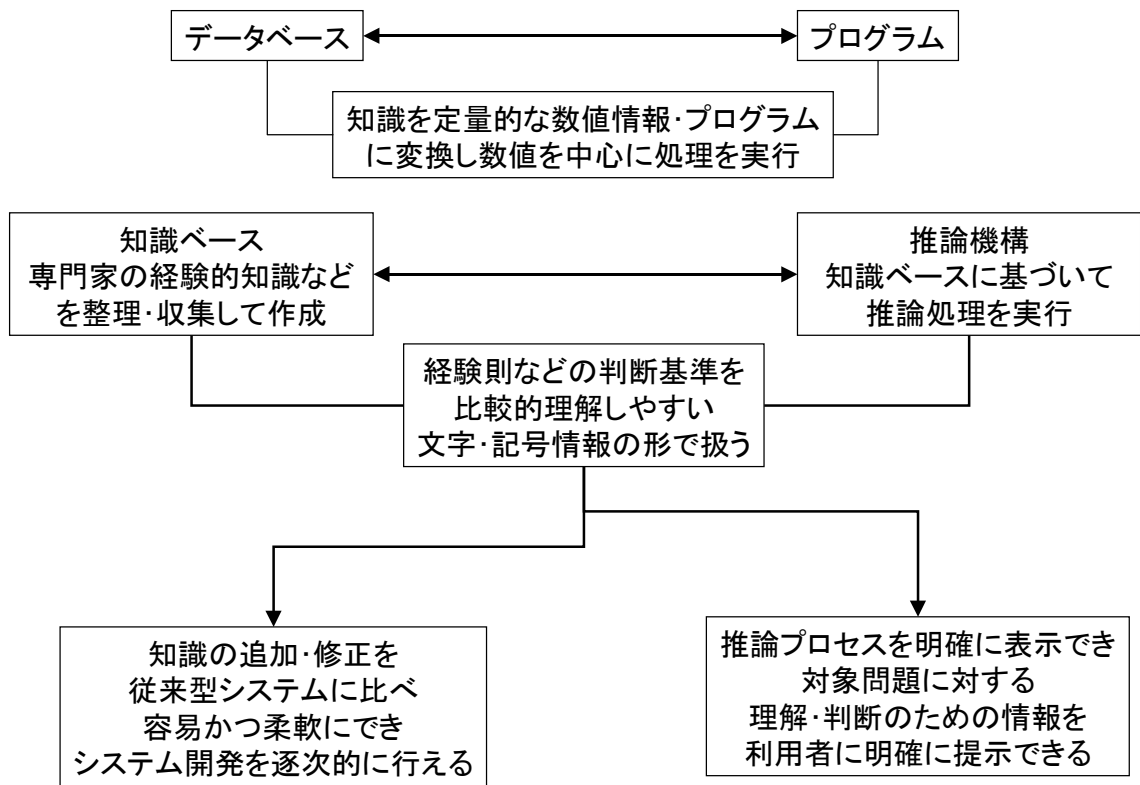


図 1.1 知識ベースと推論機構

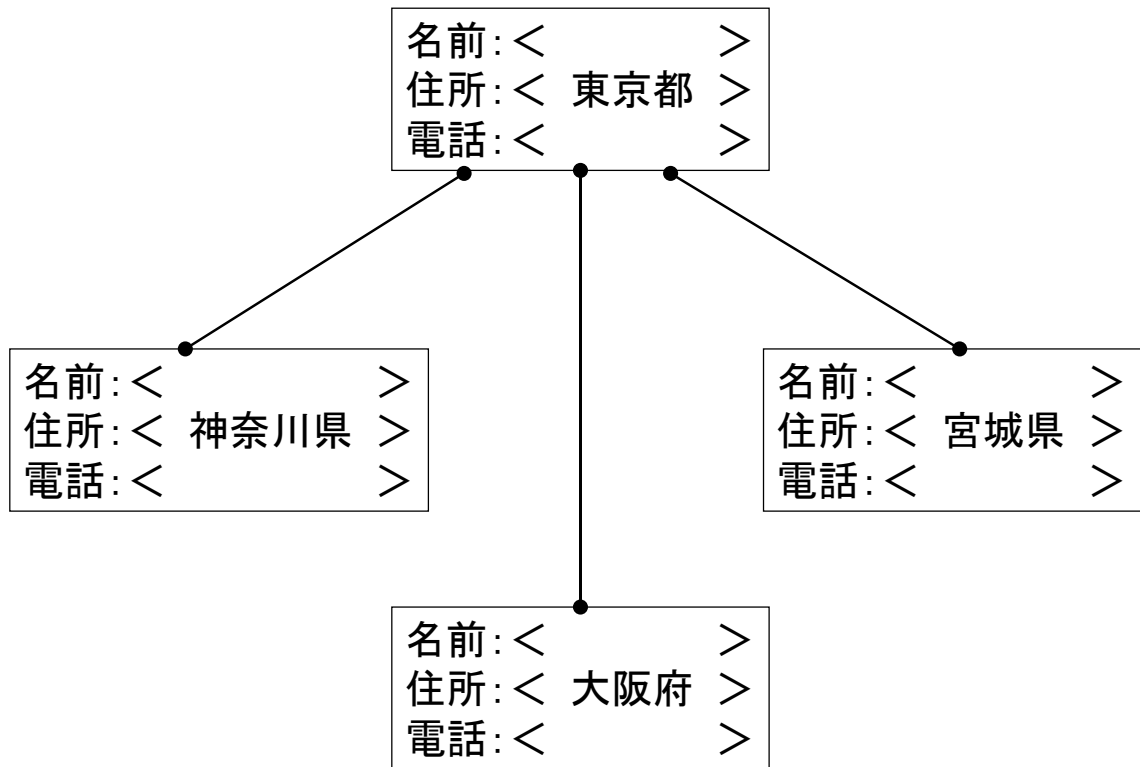


図 1.2 フレームによる概念 [文献 2]

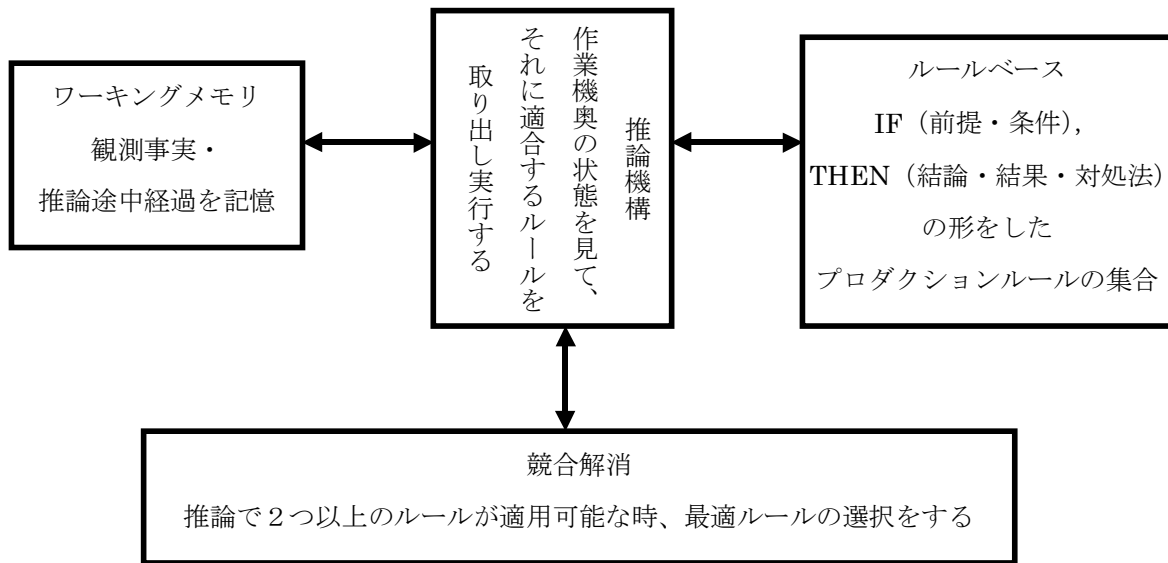


図 1.3 基本的なプロダクションシステムの構成 [文献 2]

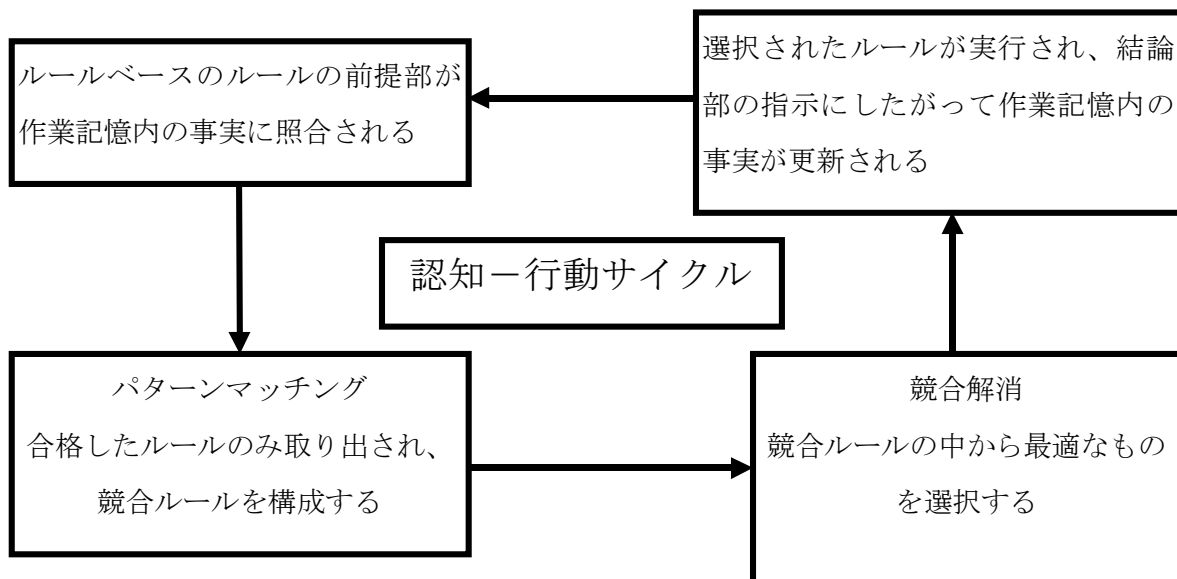


図 1.4 認知-行動サイクル [文献 27]

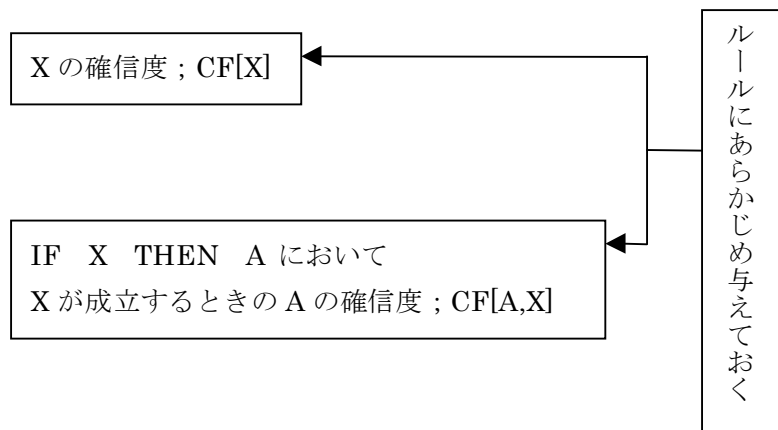


図 1.5 確信度の表現方法 [文献 2]

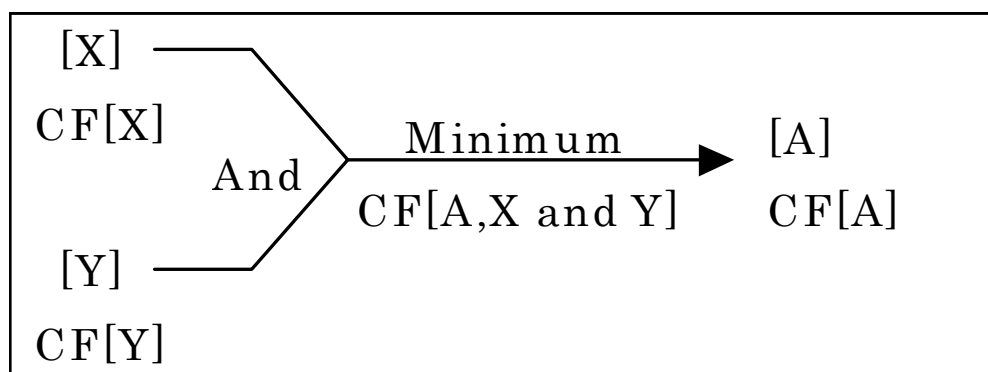
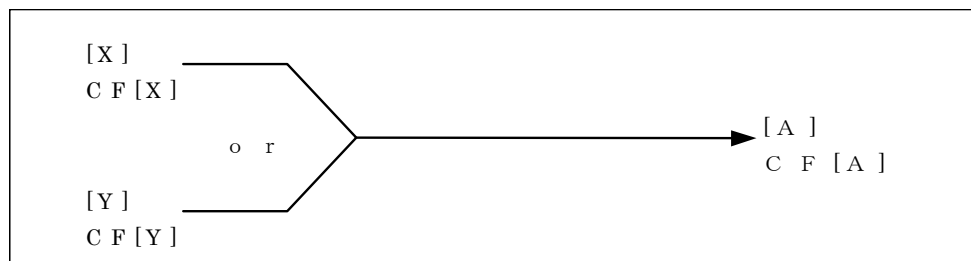


図 1.6 確信度の決定法 [文献 2]

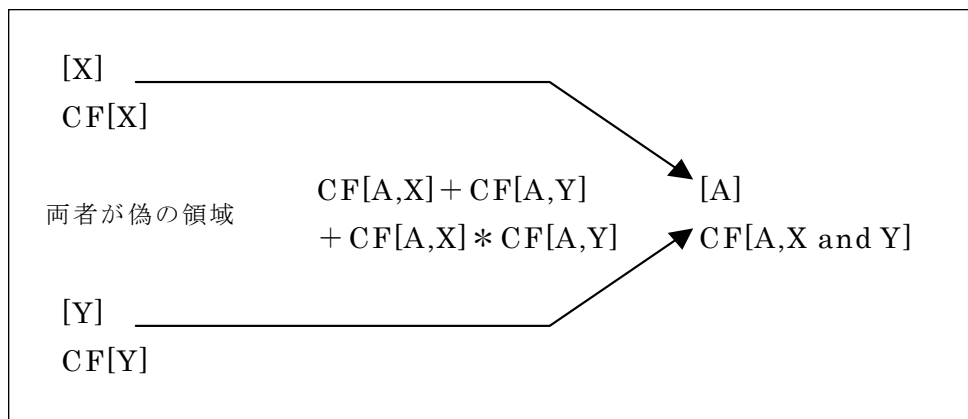
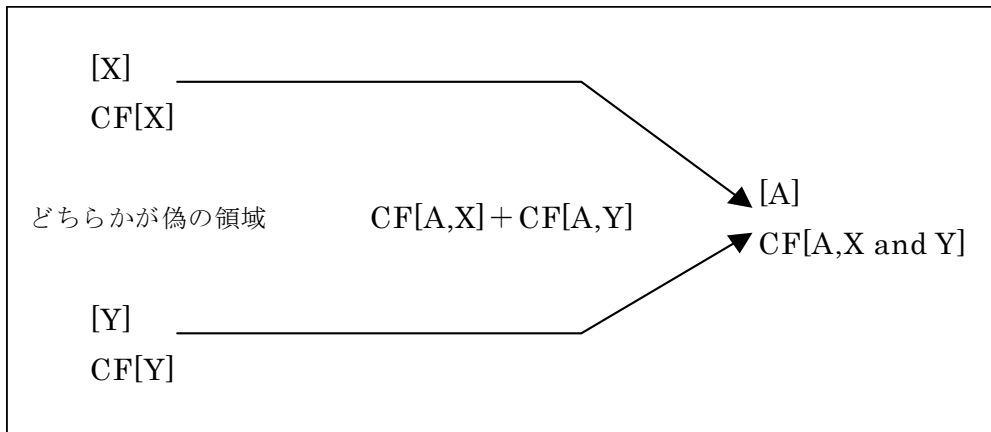
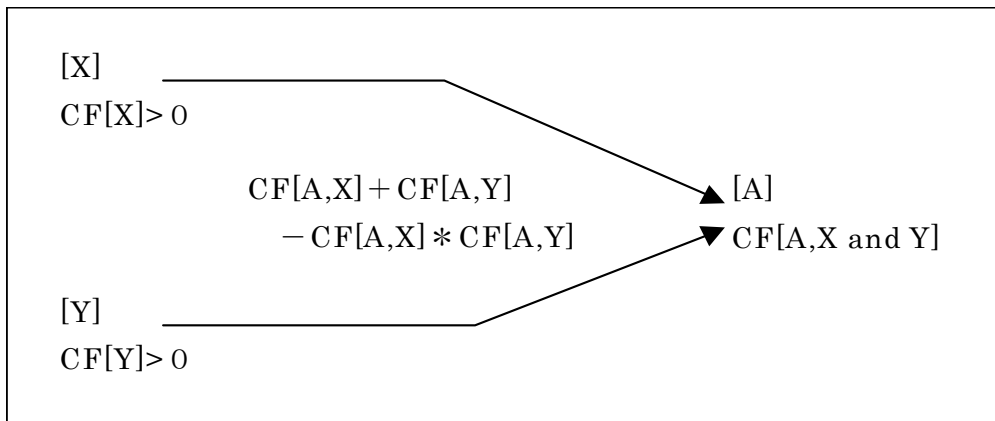


図 1.7 結論部が 2 つ以上のルールと関係を持つ場合の確信度の決定法 [文献 2]

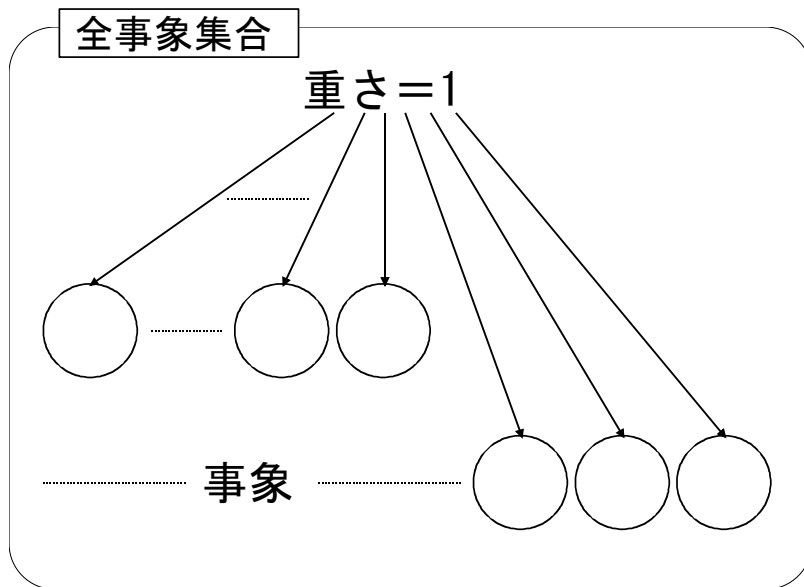


図 1.8 一般的な確立の決定法

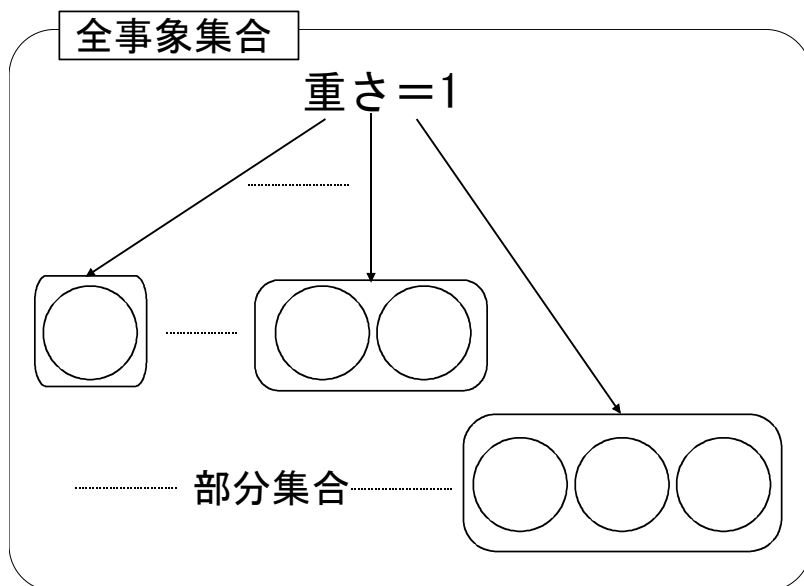


図 1.9 Dempster-Shafer 理論による基本確立

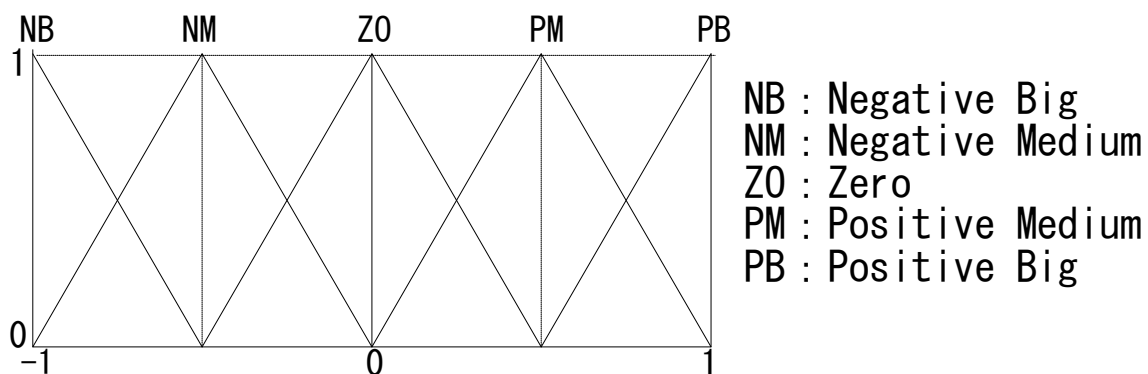


図 1.10 三角型で表現したファジィ集合 [文献 30]

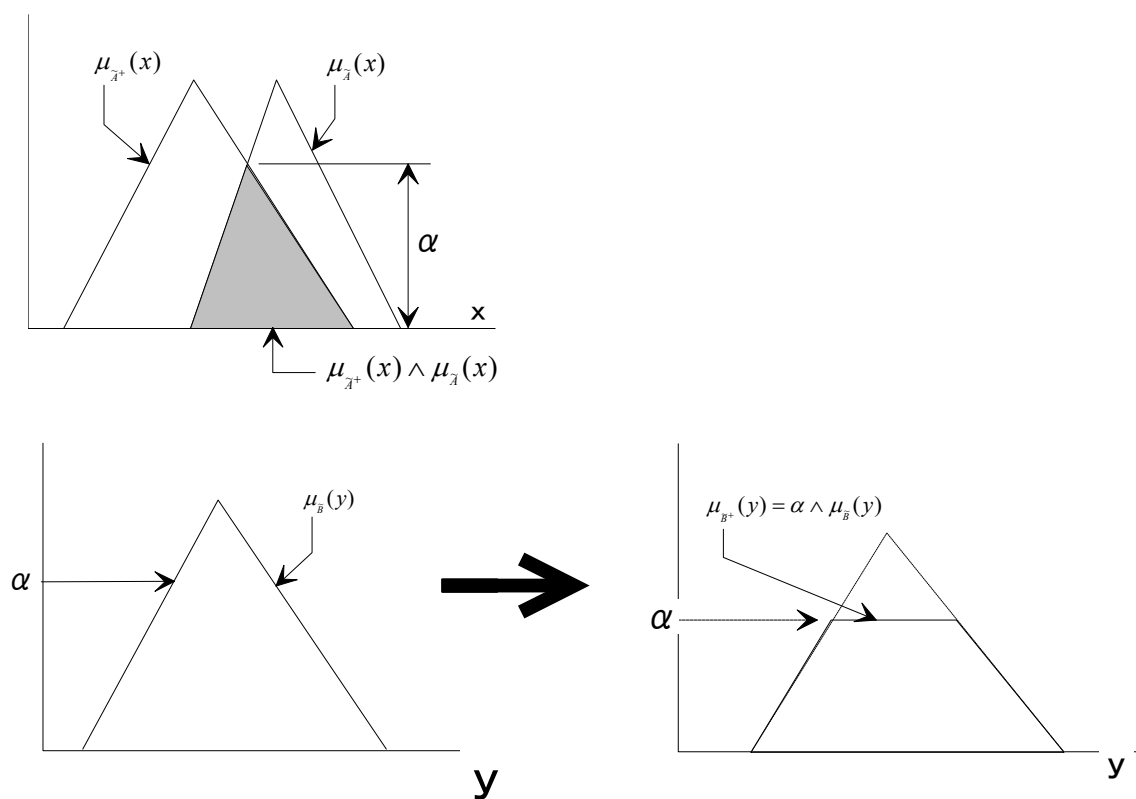


図 1.11 直説法によるファジィ推論の流れ [文献 30]

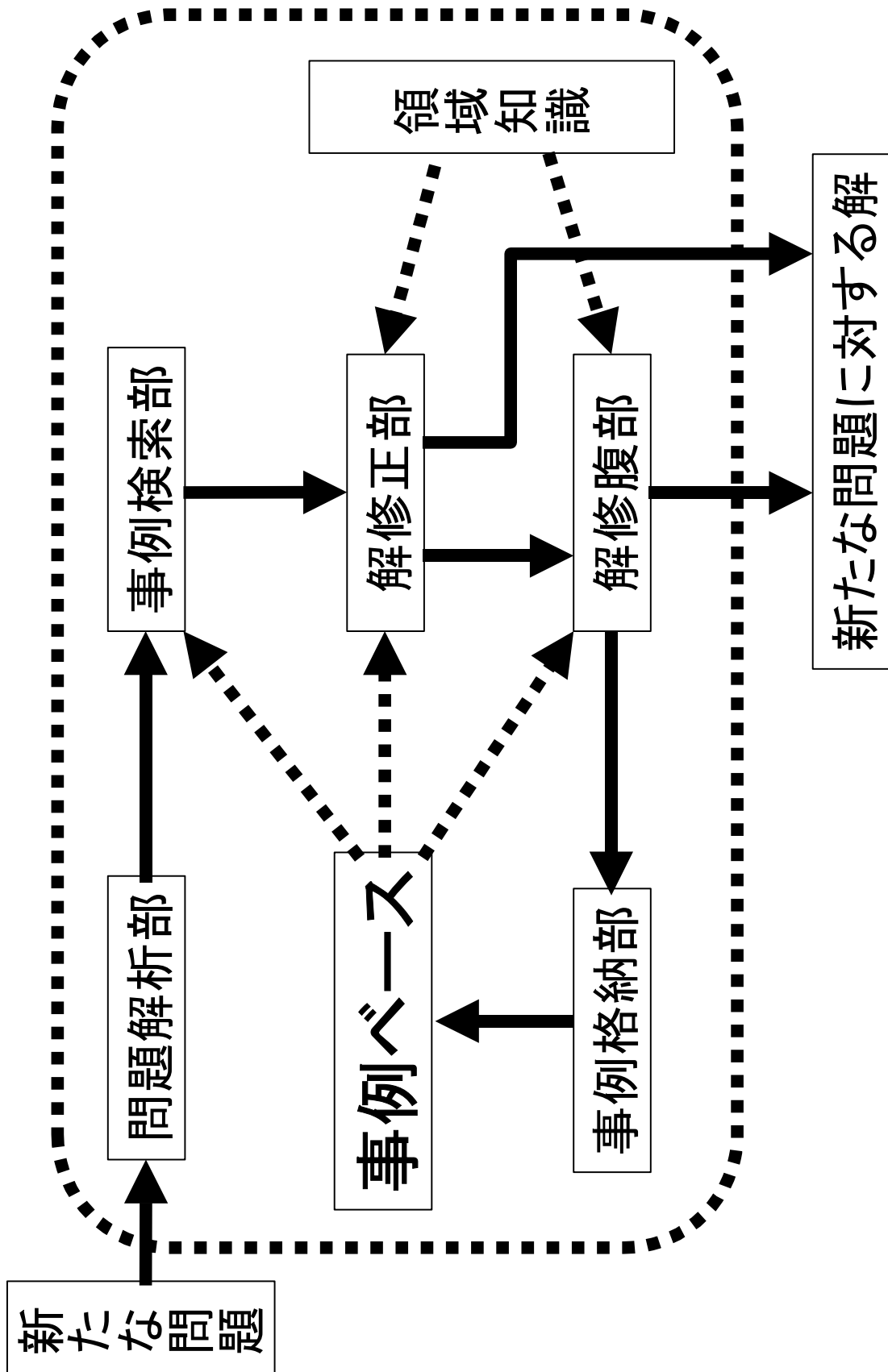


図 1.12 事例ベース推論の構成と推論処理 [文献 31]

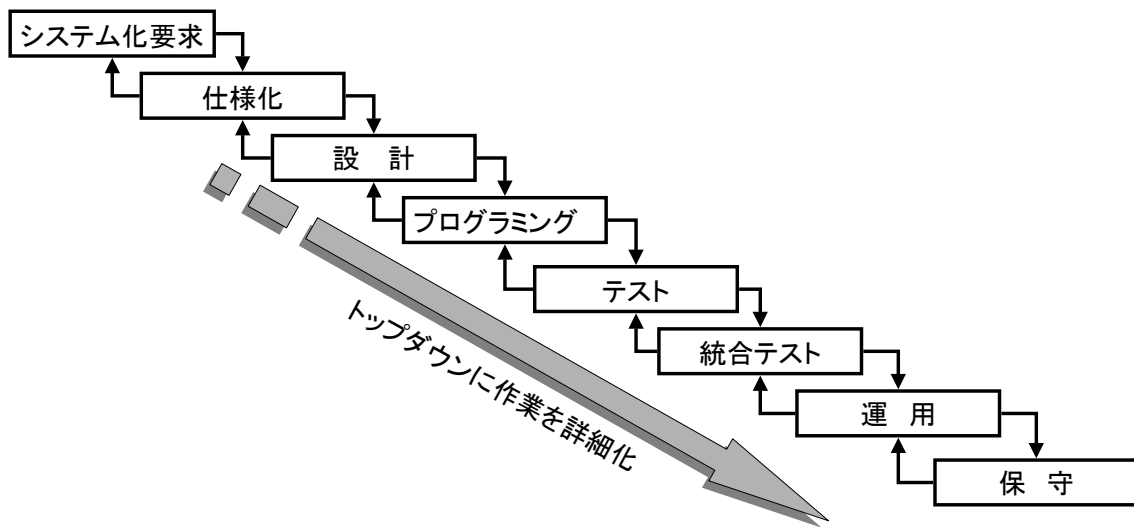


図 1.13 ウォーターフォール型開発手法 [文献 2]

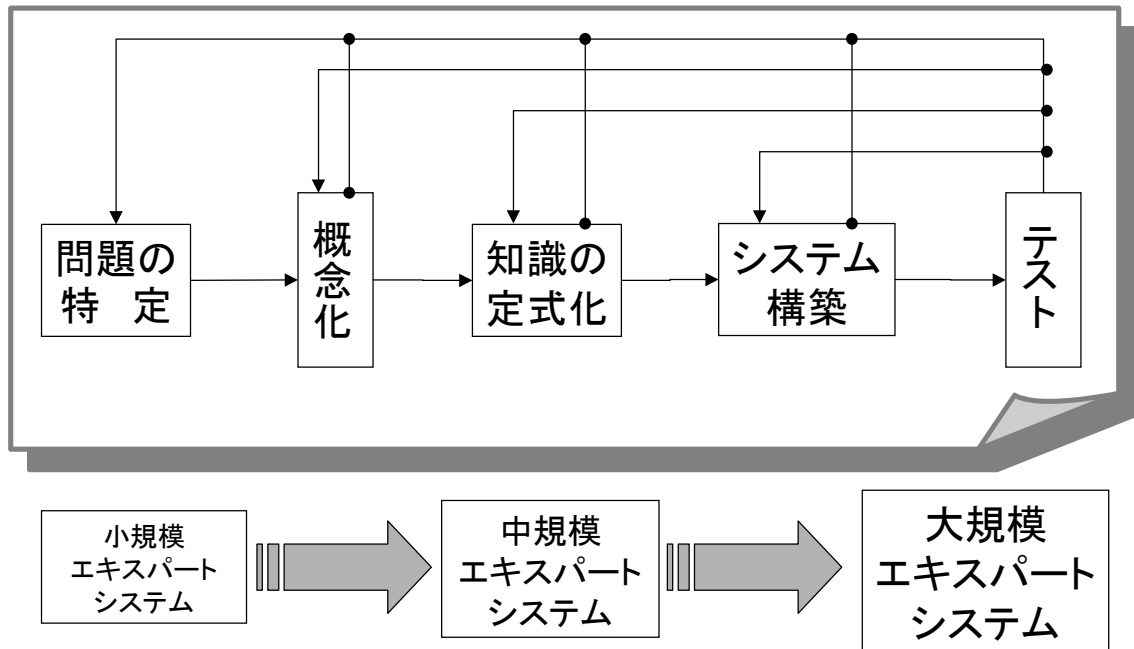


図 1.14 段階的開発方式(探索型プログラミング) [文献 2]

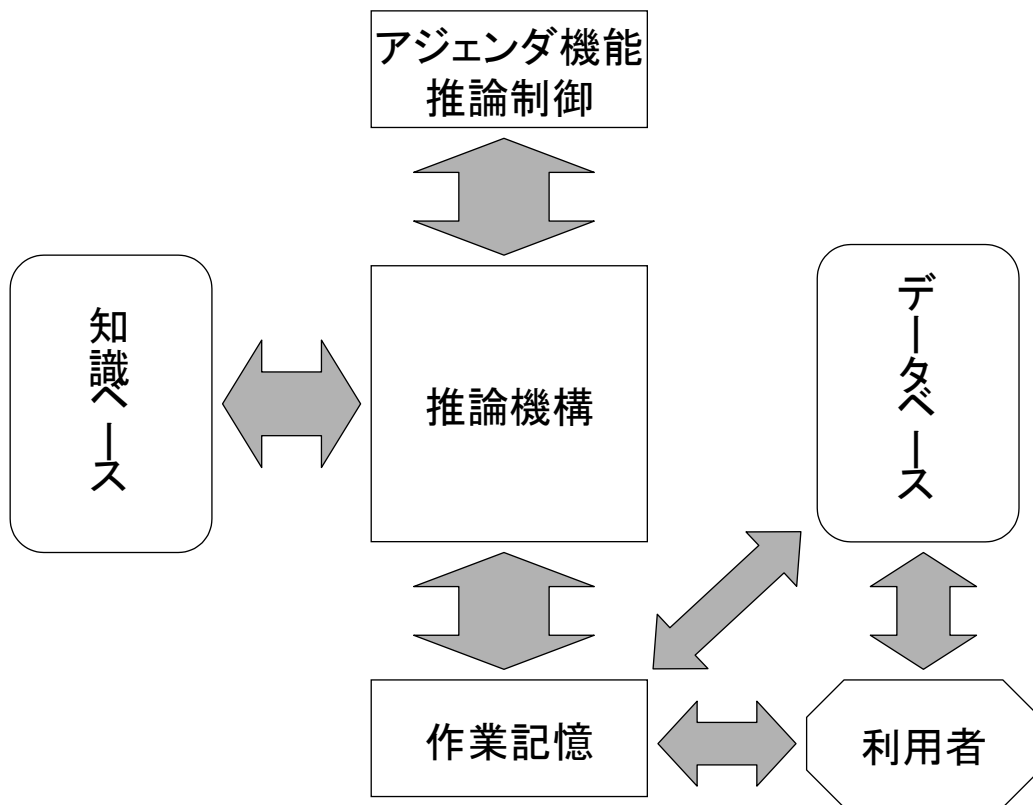


図 1.15 エキスパートシステムの構成 [文献 2]

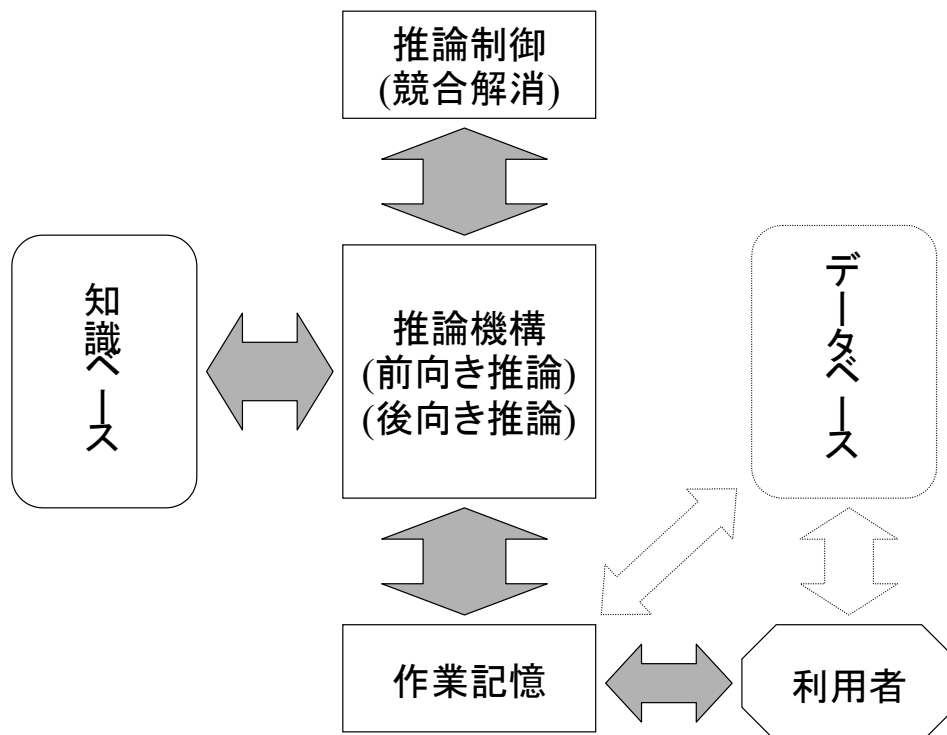


図 1.16 ルールベースシステムの構成 [文献 2]

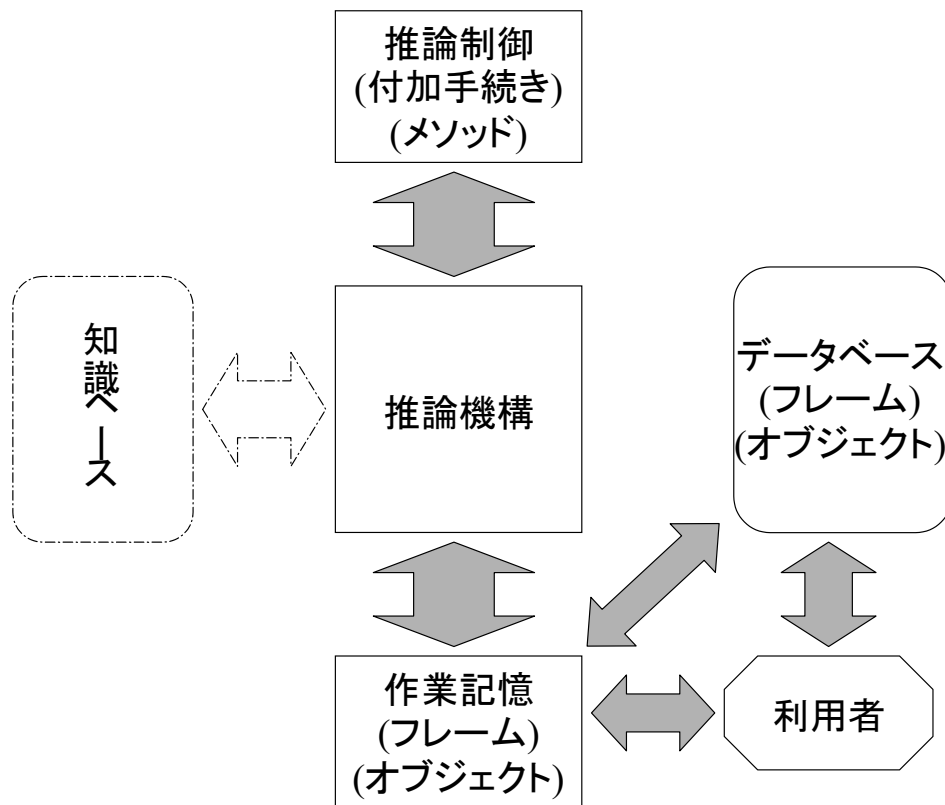


図 1.17 フレーム、オブジェクトベースシステムの構成 [文献 2]

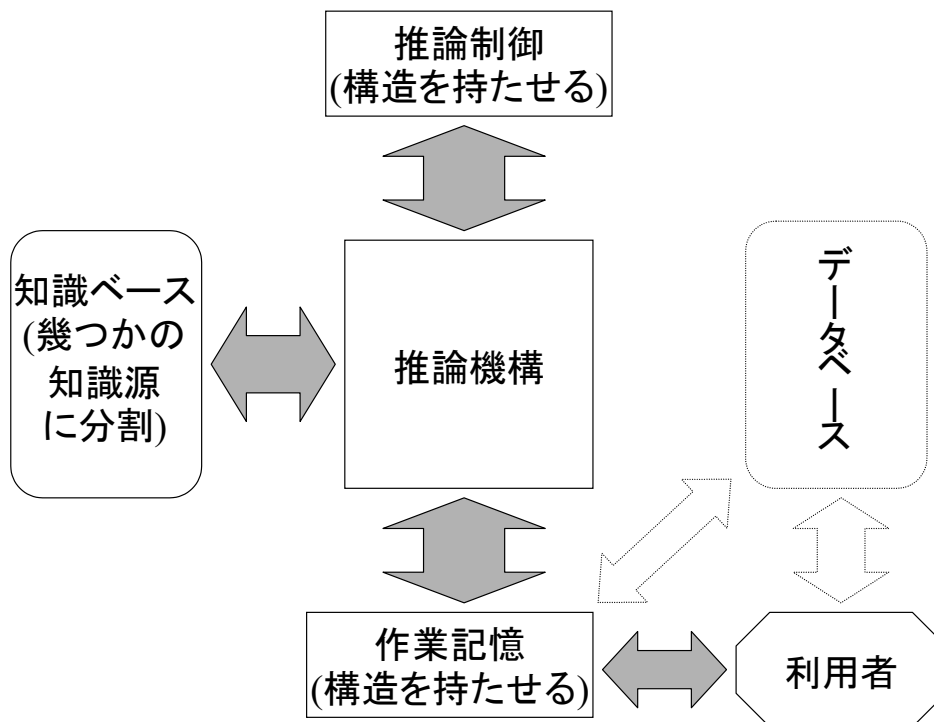


図 1.18 ブラックボードベースシステムの構成 [文献 2]

1. 研究の背景

エキスパートシステムを構築する上で、最も解決が困難とされる問題は、専門家や熟練技術者が持つ問題解決能力や経験的知識を明文化(ルール化)する、知識の獲得作業である。これを知識獲得問題と呼んでいる。

知識獲得は、システム構築の初期における専門家の知識の体系化と、知識がある程度獲得された後の、知識の追加・修正の過程に大きく分けられる。[文献 27,32]

システム構築の初期においては、一般に、専門家や熟練技術者へのアンケート調査等から、対象問題の主な特徴点を抽出してルール化し、ルールベースを構築している。この段階では、手作業によりルールベースを構築することから、多大な労力を要する困難な作業となる。また、知識の追加・修正においては、あるルールが表現すべき因果関係に、ルールベース内において矛盾が生じたり整合を取れなくなる場合があり、いかにルールベース内の整合性を保ち、信頼性の高い評価をしてルールの追加・修正を行うかが問題となる。

一方、専門家や熟練技術者の示す知識の多くが、経験則に基づく経験的知識であることから、示される知識は、複雑なものであったり、曖昧さを含み真偽が明確でないものであったりする。

複雑に示された知識を、簡単で定式化されたルールで表現することは困難である場合があり、曖昧さを含んだ知識の信頼性を定量的に評価してルールベース内に取り込むこともまた困難である。

このように、ルールの追加・修正も含めた知識獲得問題は、現在においてもエキスパートシステムを構築する際、最も重要な課題の一つとなっており、土木分野においても、知識獲得問題に関する研究がなされている。

宮本らは、対象問題をコンクリート橋診断としたエキスパートシステムの開発を行っている。その中で、ファジィ集合論を用いた知識の更新手法の提案[文献 8]や、ニューラルネットワークと双方向連想記憶を組み合わせた知識更新機能を持つ推論システムの構築[文献 9]、経験的知識に内在する曖昧さをファジィエントロピーの概念を適用して定量的評価を可能にしたアルゴリズム[文献 7]の提案をしている。

三上らは、対象問題を鋼橋疲労の補修方法選定としたエキスパートシステムの構築を行っている。ここでは、推論機能にニューラルネットワークの学習機能を持たせ、知識の追加・修正、削除を可能[文献 17]にし、さらに類推推論機能と負の学習を考慮する機能[文献 18]も実装した。また、最少化解釈、最適化解釈、最大化解釈の機能による知識の再利用の手法も提案している。[文献 19]

また、三上らは、事例ベース推論を用いた、鋼道路橋疲労損傷の補修方法選定[文献 20,22,23]や橋梁形式選定システムの開発[文献 11]、事例ベース推論とルールベース推論を併合した鋼道路橋の疲労亀裂に対する補修・補強方法選定システム[文献 21]の開発などの報告もしている。

そこで、皆川、佐藤らは、ルール洗練機能を持つ汎用的な相互結合ネットワークによ

る推論システム(以後, 本システムとする)を構築した. 本システムは, min-max 演算とニューラルネットワークで用いられる誤差逆伝播アルゴリズムを応用して通常の推論とルールベースの洗練を行うことができるシステムとした. 皆川らによる既存の研究(以後, 既存の研究)では, 三上, 田中らによる道路橋 RC 床版の損傷要因推定システムのルールベースを適用し, 本システムの持つルールベース洗練機能に対する有効性を検討した.

2. システムの推論機構

2.1 システムの概要

既存の研究では、仮説を表すノードと仮説間の関係を表すリンクを推論システムの構成要素とし、それらを相互結合ネットワークで表現してルールベース洗練機能付汎用型推論システムを構築した。

仮説の成立する可能性と仮説間の関係の強さをそれぞれノード及びノード間の関係とし、これらの属性値としてノード値と結合係数を区間[0, 1]の実数値でそれぞれ表現した。更に各属性値に対して確信度を区間[0, 1]で付与した。また、各ノードは過去の事例に関する推論結果を教師データとして持つことができるものとした。以上より、仮説を表すノードは[ノード値, ノード確信度]とこれに対応する[ノード教師値, ノード教師確信度]を属性値として持ち、仮説間の関係を表すノード間の結合は[結合係数, ルール確信度]の属性値を持つ。このように確信度を付与することで情報の不確かさを表現すると共に、min-max 演算による仮説推論機能及び、誤差逆伝播アルゴリズムの応用によるルールベース洗練の機能を制御することとした。

2.2 推論アルゴリズム

本システムの全体構成を図 2.1 に示す。図中のノード i_1, i_2, i_3 は第 i ルールの条件部を示し、 i_0 は結論部を示している。また、第 i ルールのノードを $[a_i, c_i]$ で定義し、ノード間の結合を $[w_i, c_{w,i}]$ で定義する。各ノードは一方ではルールの条件部となり、他方では結論部となり得る。

はじめに、1つのルールで導かれるノード値を式(1)の min 演算より求める。また、この時の確信度は式(2)より求める。

$$\bar{a}_i = \wedge \left(\wedge_{\bar{l}} a_{i_{\bar{l}}}, w_i \right) \quad (1)$$

$$\bar{c}_i = c_{i_{\bar{l}}} \cdot c_{w,i} \quad (2)$$

$i_{\bar{l}}$: 第 i ルールの条件部に相当するすべてのノードを表す

\bar{l} : 式(1)の \wedge に関する min 演算で採用された l を表す

この時、もし唯1つのルールによってのみノード間が結合しているならば、式(3)及び(4)に示すように結論部のノードは式(1)及び(2)より求まる値を採用し、複数のルールによってノード間が結合しているならば、はじめに各ルールについて $[\bar{a}_i, \bar{c}_i]$ を求め、次に式(5)及び(6)により、結論部のノードの属性値を決定する。

$$a_{i_0} = \bar{a}_i \quad (3)$$

$$c_{i_0} = \bar{c}_i \quad (4)$$

$$a_{i_0} = \dot{\vee}_m(\bar{a}_m) \quad (5)$$

$$c_{i_0} = \dot{\wedge}_m(\bar{c}_m) \quad (6)$$

m : i_0 を結論部とするルールの全てを表す

$\dot{+}$: $a \dot{+} b = a + b - a \cdot b$ を表す

$\dot{+}(\)$: $c_{m_1} \dot{+} c_{m_2}$ for $\forall_{m_1, m_2} \in m$ を表す

2.3 洗練アルゴリズム

通常のルールベースシステムは、ルールベースを確定情報として仮説推論を行っている。本システムでも、ルール確信度をノード教師確信度より高くすることにより仮説推論を行うことができる。逆に、ノード教師確信度をルール確信度より高くするとルールベースの洗練が行なわれる。以下にノード確信度とルール確信度の大小関係を用いた仮説推論(ノード値の更新)とルールベース洗練(結合係数の更新)の制御アルゴリズムを示す。推論により得られた第*i*ルールの結論部の[ノード値, ノード確信度]と同結論部の[ノード教師値, ノード教師確信度]との差を式(7)及び(8)より求める。

$$\Delta a_{i_0} = a_{i_0} - t_{i_0} \quad (7)$$

$$\Delta c_{i_0} = c_{i_0} - c_{t, i_0} \quad (8)$$

式(8)より $\Delta c_{i_0} \geq 0$ の場合, $c_{t,i_0} \leq c_{i_0}$ であることから式(9)及び(10)により[ノード教師値, ノード教師確信度]の更新を行う.

$$t_{i_0} \leftarrow t_{i_0} + \eta \cdot \Delta a_{i_0} \quad (9)$$

$$c_{t,i_0} \leftarrow c_{t,i_0} + \eta \cdot \left| \Delta c_{i_0} \right| \quad (10)$$

ここで, η は学習率を表す.

また, $\Delta c_{i_0} < 0$ の場合 $c_{i_0} < c_{t,i_0}$ であることから推論により得られた「ノード値, ノード確信度」の更新を行う. この際, 式(1)及び(5)の min-max 演算において条件部のノード値が採用された場合, 採用された条件部の[ノード値, ノード確信度]の更新を式(11)及び(12)により行い, ルールの結合係数が採用された場合, 採用された[結合係数, ルール確信度]の更新を式(13)及び(14)により行う.

$$a_{\bar{m}_l} \leftarrow a_{\bar{m}_l} + \eta \cdot \Delta a_{i_0} \quad (11)$$

$$c_{\bar{m}_l} \leftarrow c_{\bar{m}_l} + \eta \cdot \left| \Delta c_{i_0} \right| \quad (12)$$

$$w_{\bar{m}} \leftarrow w_{\bar{m}} + \eta \cdot \Delta a_{i_0} \quad (13)$$

$$c_{w,\bar{m}} \leftarrow c_{w,\bar{m}} + \eta \cdot \left| \Delta c_{i_0} \right| \quad (14)$$

\bar{m} : 式(5)の max 演算で採用された m を表す

このように本システムは, 誤差逆伝播の考えを用いて教師データとの比較・演算により, すべてのノード値と結合係数を一定値に収束させて解を導く.

3. 適用した知識ベース

本システムの適用例として、三上・田中らにより開発された道路橋 RC 床版の損傷要因推定システム[文献 16] (以後, 既存システム) のルールベースを用いた。既存システムは確信度付プロダクションシステムであり, 図 2.2 に示すように, はじめに目視できる損傷から損傷種類を求め, 次に求められた損傷種類と輪荷重通行位置, 適用示法書, 損傷場所から損傷要因を推定するものである。推定の対象とする損傷要因を表 2.1[文献 16]に示し, 既存システムに提示された入力情報を表 2.2[文献 16]に示す。また, 表 2.3[文献 16]には, 既存システムを適用した場合の入力情報に対する推論結果を示す。なお, 表中の番号 28 から 47 は, 表 2.1 で示した損傷要因の番号に対応している。

ルールベースは, [目視できる損傷-損傷種類],[損傷種類-損傷要因],[適用示法書-損傷要因],[損傷場所-損傷要因]の 4 つの因果関係から構成されており, 表 2.4[文献 16]に例として, 損傷場所がハンチ部付近での[損傷種類-損傷要因]についての因果関係を示す。これらの関係に対してその強さに応じて確信度 0.5, 0.3, 0.1 を付与している。

本システムでは, これらの確信度に代えて結合係数として $3/4, 2/4, 1/4$ の値をそれぞれ対応させて付与し, ルールベースを構築した。また, ノードの総数を 63 個, 結合の総数を 872 として, 本システムのネットワークを構成した。

4. 本システムによる損傷要因推定(ルールベース洗練前)

はじめに、ルールベースを確定情報とし、損傷要因の可能性を推定した。用いた事例は、表 2.2 に示したものである。例として、図 2.3 に事例 1 の推論結果と教師データを、損傷要因の可能性を縦軸に、損傷要因のノード番号を横軸にとって示した。ここでいう、教師データとは、表 2.4 に示した、各事例に対して既存システムが推論した損傷要因の可能性である。更に、教師データと本システムによる推論結果との誤差を次の式(15)により求め、縦軸に誤差、横軸に事例番号をとって図 2.4 に示した。

$$\text{誤差} = \frac{\sum_i |t_i - a_i|}{n} \quad (15)$$

ここで、 t_i は既存システムにより求めた第 i 損傷要因の可能性、 a_i は本推論システムにより求めた第 i 損傷要因の可能性を示しており、 n は対象としている損傷要因の個数を表している。

これらの結果で、既存システムと本システムの推論結果に相違が現れた原因は、既存のルールベースを移植する際に用いた 3/4, 2/4, 1/4 という結合係数に明確な根拠が必ずしもないこと、推論過程において min-max 演算を用いていることがあげられる。しかし、傾向としては、事例 1 以外も含めて損傷要因の可能性の大小の分類において比較的良く一致している。

5. 本システムによる損傷要因推定(ルールベース洗練後)

本システムの持つ知識洗練機能の精度を検討するため、次の4通りの場合のルール洗練を行い、それぞれの場合について仮説推論を行った。この結果の妥当性の検討のため、既存システムによる推論結果を提示し、本システムによる結果との比較を行った。

1. 各事例を個別に提示した場合
2. 8事例すべてを同時に提示した場合
3. 特定の損傷場所で損傷を生じた6事例を提示した場合
4. 3の6事例のうち5事例を提示し、残りの1事例についての仮説推論を行った場合

ここで、ルールベースの洗練の際には、途中で教師データが変化しないようにすべてのノード教師確信度を1.0にし、すべてのルール確信度を0.1に設定した。また、仮説推論の際にはノード値及びノード確信度をそれぞれ0.5と0.1に設定した。

5.1 各事例を個別に提示した場合

はじめに、事例1から事例8の教師データを各々単独に提示し、ルールベースの洗練を行った。例として、事例2の結果を、損傷要因の可能性を縦軸に、損傷要因のノード番号を横軸にとり、図2.5に示した。また、図中には、洗練における反復回数(以後、“洗練回数”と呼ぶ。)の影響を見るため、洗練回数が異なる場合の結果も示した。また、比較のために教師データも示した。更に、式(15)により各事例の誤差を算定し、誤差と洗練回数の関係を図2.6に示した。

これらの結果から、単独事例を提示してルールベースを洗練すると、各事例の推論結果が教師データとほぼ一致することが分かった。しかし、単独事例によるルール洗練ということは、各事例に対して異なるルールを求めることになるため、実用上の価値は低い。そこで、2項から4項では、複数の事例を同時に提示した場合、提示された事例によって、ルールベースの洗練後の推論結果及びその他の事例に対する推論結果がどのような影響を受けるかを検討した。

5.2 8事例すべてを同時に提示した場合

8 事例をすべて同時に提示し、ルールベースの洗練を行った。その後、各事例に対する損傷要因の可能性を推定した。その推論結果を図 2.7 に示した。図中には、比較のため既存システムの推論結果及び単独事例を提示した場合の結果も併せて示した。更に、既存システムと本システムによる推論結果の誤差を式(15)で算定し図 2.8 に示した。

これらの結果から、すべての事例を同時に提示してルールベースを洗練した場合、単独事例による洗練より、教師データと一致する割合は低くなった。これは、すべての教師データに合致することを目標にルールベースが洗練されているために、個々の事例に対する推定精度が低下していることによる。しかし、本来、ある時点で得られる情報から決定されるルールベースは 1 つに特定されるべきであり、図 2.8 に示した結果は、全事例を教師データとしてただ 1 組のルールベースを求めたにも関わらず誤差は小さく、ルール洗練の効果が顕著であるといえる。このことは、既存システムの推論性能を取り込んだ推論システムを、きわめて容易に構築できることを示している。

5.3 特定の損傷場所で損傷を生じた 6 事例を提示した場合

損傷要因と損傷種類の因果関係が損傷場所の影響を強く受けることから、ここでは、損傷場所が同一の事例のみを同時に提示してルールベースを洗練した。

損傷場所の一致する事例は、表 2.2 より事例 1, 4, 5, 6, 7 及び 8 の計 6 つである。これらはすべてハンチ部付近に損傷を生じた事例であり、これらをすべて同時に提示してルールベースの洗練を行った。例として、事例 4 の推論結果を図 2.9 に示した。図中には、比較のために、8 つの事例を同時に提示した結果と教師データも併せて示した。更に、各事例の推論結果と教師データとの誤差を式(15)で算定し、図 2.10 に示した。

これらの結果から、損傷場所の情報が異なる事例 2 及び 3 を提示しない場合と、それらも同時に提示した場合(図 2.8)を比べると、当該場所に対する教師データとの一致度は同程度であることが分かる。それに対して、損傷場所の異なる事例 2 及び 3 に対する推論結果の誤差は未洗練の場合(図 2.4)よりも更に大きい。

5.4 6事例のうち5事例を提示した場合

これら6つの事例から5つの事例を選択し、それらを用いてルールベースを洗練した。事例の組み合わせは以下に示す6パターンとなる。

Ex.6-8:事例 1, 事例 4, 事例 5, 事例 6, 事例 7

Ex.6-7:事例 1, 事例 4, 事例 5, 事例 6, 事例 8

Ex.6-6:事例 1, 事例 4, 事例 5, 事例 7, 事例 8

Ex.6-5:事例 1, 事例 4, 事例 6, 事例 7, 事例 8

Ex.6-4:事例 1, 事例 5, 事例 6, 事例 7, 事例 8

Ex.6-1:事例 4, 事例 5, 事例 6, 事例 7, 事例 8

ここで、例えば、Ex.6-8は6事例の中から事例8を除いたものである。

各パターンの事例を用いてルールベースを洗練し、それぞれ提示事例に含まれない事例について推論結果を求め、教師データとの誤差を算定し図2.11に示した。

事例として提示していない事例に対する推論結果を見ると、すべての事例を推定した場合(図2.8)や損傷場所が同一のすべての事例を提示した場合(図2.10)に比べると、誤差が同程度のもの(事例4,6,7)と大きく開いているもの(事例1,5,8)がある。しかし、ルールベース洗練をしない場合(図2.4)に比べれば、明らかに推定精度は向上している。このことは、提示する事例の質が良い場合には、未提示事例に対しても良い推論結果を与える機能を本システムがもっていることを示すものである。

6. 結論

本研究では、仮説と仮説間の関係を構成要素とする相互結合ネットワークで表現されたルールベース洗練機能付汎用型推論システムを構築した。本推論システムは、エキスパートシステムの知識をルール化することにより、通常の推論を行えるばかりではなく、具体事例に対する既存システムの推論結果を教師データとして用いることによって、ルールベースを洗練することができる。適用例としては、三上・田中により開発された道路橋 RC 床版の損傷要因推定エキスパートシステムのルールベースを用いて本システムの性能を検討した。得られた主な結論を以下に示す。

- (1) ルールベースを確定情報として、損傷要因の可能性を推定した場合、傾向としては損傷要因の可能性の大小の分類が可能である。
- (2) 既存システムの推論結果を確定的な教師データとして提示してルールベースを洗練し、その後、洗練されたルールベースを用いて損傷要因を推定した場合、
 - ・ 単独事例を提示してルールベースを洗練すると各事例の推論結果が教師データとほぼ一致する。
 - ・ 複数事例を同時に提示した場合、提示された事例によって、ルールベースの推論結果及びその他の事例に対する推論結果は影響される。
 - ・ 複数事例をすべて同時に提示しても、単独事例提示の場合と比べて推定精度の悪化は著しくない。
 - ・ 未提示事例に対する推論では、洗練のための教師データの良否が推論精度に強く影響する。

このように本論文で提案した手法を用いることによって既存システムと同等の推論性能をもつルールベースシステムを容易に構築することができる。また、提示する事例が適切に選択されれば、未提示事例の推定精度を確保することが可能である。今後の課題として、複数のルールベースシステムの機能を併せ持つシステムの構築とそれによる知識の共有・再利用を計ることが挙げられる。

7. 研究の背景

第6節までの研究において、仮説と仮説間の関係を構成要素とし、min-max 演算とニューラルネットワークの誤差逆伝播アルゴリズムを応用した学習アルゴリズムを用いてルールベース洗練機能付推論システムの開発を行った。また、適用例を道路橋 RC 床版の損傷要因推定のための既存エキスパートシステムルールベースを用いて、4節、5節の5つの場合について、それぞれルールベース洗練、あるいは仮説推論を行い、その結果を比較することで提案したシステムの有効性を検討した。この結果、提案した推論システムが、知識のルール化がなされていると、ルールベースの洗練を行え、知識の追加・修正に対して、有効な機能を持つシステムであると検証できた。しかし、知識によっては、その因果関係を明確にできず、提案した推論システムによる仮説推論の結果に支障をきたすことが考えられる。

そこで、7節以降では、知識の因果関係を明確にルール化できない場合に、因果関係ルールを必要とする本推論システムを補完する目的で、ニューラルネットワークによる推論を適用した。ニューラルネットワークは、1章の2.3で述べたような性質を持つことから、ここでは、ニューラルネットワークをブラックボックスとして用いて推論を行い、本推論システムでいう仮説推論による推論結果を獲得し、その結果を用いてルールベースの洗練を行うことで、知識の追加・修正を達成することを検討した。

8. ニューラルネットワークを用いた推論

ニューラルネットワークによる推論での学習アルゴリズムは、一般的に誤差逆伝播アルゴリズムが多用されている。しかし、誤差逆伝播アルゴリズムは、その学習方法から学習速度が遅く、複雑な問題に対して不向きであるといわれている。そこで、誤差逆伝播アルゴリズムの他に、学習速度が速いとされる、対向伝播アルゴリズムの適用を試みることにした。

8.1 誤差逆伝播アルゴリズム [文献 33]

誤差逆伝播では図 2.12[文献 33]に示すように入力層、中間層、出力層と信号が伝わり最後に出力信号を出力する。式(16)により入力層の出力値 a_i とユニット i, j 間の重み w_{ij} の間で和 S_j が計算された後、中間層の出力値 a_j は、式(17)で示したシグモイド関数を用いて決定する。同様の方法で出力層の出力値 a_k も決定される。

$$S_j = \sum_i a_i w_{ji} \quad (16)$$

$$a_j = f(S_j) = \frac{1}{1 + e^{-(S_j + t_j)}} \quad (17)$$

順伝播終了後、出力値 a_k が教師値 t_k と比較されて式(18)よりエラー値 δ_k が計算される。

$$\delta_k = (t_k - a_k) f'(s) \quad (18)$$

このエラー値が中間層に逆伝播されて式(19)により中間層の誤差 δ_j が求められた後、式(20)を用いて重み調整量 Δw_{ji} が算出される。

$$\delta_j = \left[\sum_k \delta_k w_{kj} \right] \quad (19)$$

$$\Delta w_{ji} = \eta \delta_j a_i \quad (20)$$

η : 学習率

この重み調整量により重み w_{ji} が更新されて新たな重み値を用いて再度順伝播を行う。このように誤差逆伝播では、順伝播と逆伝播を繰り返し実行して出力値と教師値との差を減少させていき最終的な出力値を決定する。

8.2 対向伝播アルゴリズム [文献 33]

対向伝播は、図 2.13[文献 33]で示すように入力層, 競合層, 書き換え層で構成されている。対向伝播法では、入力層に入力信号(入力値) a_i が示されると式(21)より入力値に応答した競合層のユニット値 a_j が決定される。ここで式(22)により、活性化されたユニットとして競合層のユニット値 a_c が唯 1 つに絞られ、それ以外のユニットは不活性化されたユニットとなる。

$$a_j = \sum_i a_i w_{ji} \quad (21)$$

$$a_c = \max a_j \quad (22)$$

式(23)より活性化されたユニットは勝者ユニットとしてユニット値 a_c が 1.0 に設定され、逆に不活性化されたユニットは敗者ユニットとなりユニット値 a_j が 0.0 に設定される。

$$a_c = 1.0, \quad a_j = 0.0 \quad (j \neq c) \quad (23)$$

勝者ユニットはユニット値が 1.0 であることから式(24), (25)により出力層に強い影響を及ぼすのに対し、敗者ユニットはユニット値が 0.0 であることから出力層のユニットに何ら影響を及ぼさない。

$$a_k = \sum_j a_j w_{kj} \quad (24)$$

$$a_k = w_{kc} \quad (25)$$

a_k : 出力層のユニット値

この後、勝者ユニットと結合する第 1 層の重みが式(26)により調整され、次にネットワークの出力層の重みが教師値と比較されて、第 2 層の重みが式(27)の Widrou-Hoff の規則により修正される。

$$w_{ji}^{new} = \begin{cases} w_{ji}^{old} + \alpha(a_i - w_{ji}^{old}) & j = c \\ w_{ji}^{old} & j \neq c \end{cases} \quad (26)$$

$$w_{kj}^{new} = \begin{cases} w_{kj}^{old} + \beta(t_k - a_k) & j = c \\ w_{kj}^{old} & j \neq c \end{cases} \quad (27)$$

α, β : 学習率

9. 推論結果

3 節で適用した事例群を入力値とし、既存システムの推論結果を教師値として、誤差逆伝播アルゴリズムと対向伝播アルゴリズムをそれぞれ学習アルゴリズムに適用して推論を行った。

2つの学習アルゴリズムの学習速度の違いを比較するため、図 2.14 には、誤差逆伝播アルゴリズムを適用した場合、図 2.15 には対向伝播アルゴリズムを適用した場合の、教師データの数の違いによる収束状況の推移を、横軸に学習回数を取り、縦軸に学習回数毎の、教師値と出力値との間の rms 値をとって示す。

また、図 2.16 には、ニューラルネットワークによる推論結果が、本推論システムと同等の推論結果を示すことが可能かを検討するため、教師データの数を 50 個としたときの 2つのアルゴリズムによる推論の出力値と、既存システムによる推論結果を併せて示す。図 2.16 は、横軸に損傷要因を取り、縦軸に各損傷要因の推論結果をとった図である。

この結果、誤差逆伝播による学習の収束状況は穏やかな曲線を描いているのに対し、対向伝播による学習では早い時期に収束を完了している。更に対向伝播による学習では、教師データの数が収束状況に大きく影響していることが読み取られる。また、図 2.16 の推論結果をみると個々の出力値では誤差逆伝播法に比べて多少精度は低いですが、損傷要因の可能性の傾向でみると既存システムと変わらない傾向を示しており、学習速度の利点を考慮すると対向伝播による推論は有効であると考えられる。

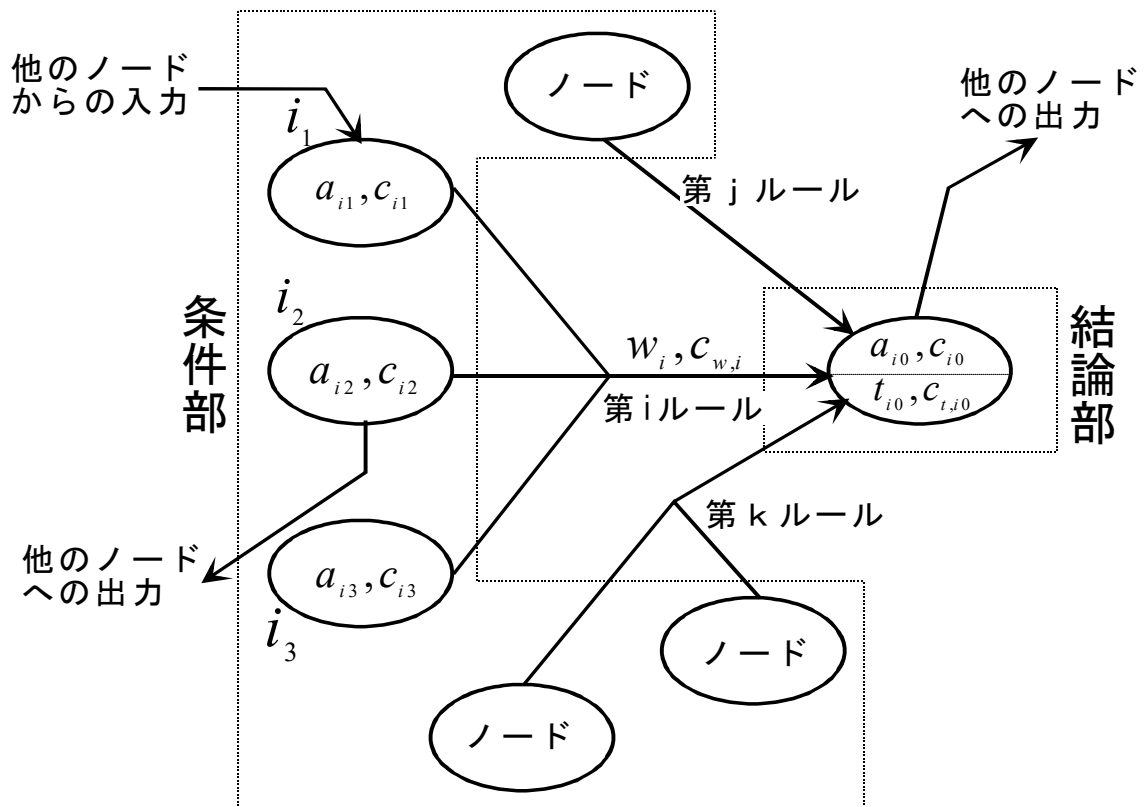


図 2.1 システムの全体構成

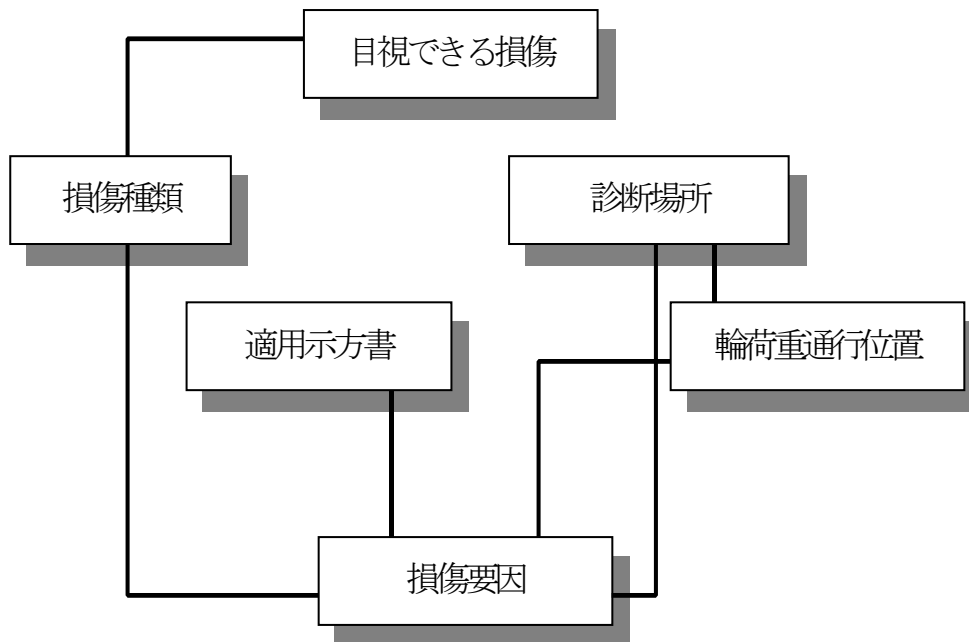


図 2.2 既存システムのネットワークの概要

表 2.1 推定の対象となる損傷要因 [文献 16]

荷重等	28 過大な輪荷重の作用 29 衝撃作用 30 輪荷重通行位置と支持桁配置の関係
設計 ・ 構造上	31 床版厚による剛性不足 32 鉄筋量不足による剛性不足 33 配力鉄筋量の不足 34 主鉄筋の曲げ上げ位置の不適性による鉄筋量の不足 35 主桁が拘束するための乾燥収縮による引張応力 36 主桁の不等沈下による付加曲げモーメント 37 連続げた橋、アーチ系橋梁等で、載荷状態における床版の負の曲げモーメント、引張力の作用 38 荷重分配横桁の有無
施工上	39 コンクリートの低品質 40 打設時における冬場の凍結影響 41 養生の施工不良 42 施工打継目の処理不十分 43 配筋の誤差 44 かぶり不足
その他	45 気象作用による凍結、融解等の影響 46 塩分 47 表面排水の良否

表 2.2 提示された入力情報 [文献 16]

事例		事例1	事例2	事例3	事例4
事象					
目視 できる	ひびわれ	縦横方向	縦横方向	縦横方向	縦横方向
	剥離	—	—	—	—
損傷	不純物の流出	遊離石灰	遊離石灰	遊離石灰	遊離石灰
損傷場所		ハンチ部付近	支間中央部付近	桁端部	ハンチ部付近
適用示方書		昭和39年3月示方書	昭和39年3月示方書	昭和39年3月示方書	昭和42年9月示方書
輪荷重通行位置		支間1/4付近	—	—	支間1/4付近

事例		事例5	事例6	事例7	事例8
事象					
目視 できる	ひびわれ	縦横方向	縦横方向	縦横方向	縦横方向
	剥離	—	—	浮き有り	—
損傷	不純物の流出	遊離石灰	漏水	—	—
損傷場所		ハンチ部付近	ハンチ部付近	ハンチ部付近	ハンチ部付近
適用示方書		昭和55年2月示方書	昭和39年3月示方書	昭和39年3月示方書	昭和39年3月示方書
輪荷重通行位置		支間1/4付近	支間1/4付近	支間1/4付近	支間1/4付近

表 2.3 既存システムによる推論結果(教師データ) [文献 16]

損傷要因(表1) 提示事例(表2)	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
事例	0.73	0.73	0.92	0.9	0.88	0.92	0.85	0.4	0.71	0.77	0.74	0.32	0.32	0.32	0.55	0.38	0.38	0.24	0.18	0.58
事例	0.9	0.77	0.5	0.73	0.73	0.6	0.3	0.4	0.59	0.65	0.6	0.4	0.52	0.56	0.58	0.1	0.2	0.1	0	0.05
事例	0.86	0.86	0.47	0.76	0.86	0.85	0.3	0.33	0.31	0.31	0.31	0.37	0	0.5	0	0.48	0.54	0	0	0.75
事例	0.73	0.73	0.92	0.9	0.88	0.89	0.85	0.4	0.71	0.77	0.74	0.32	0.32	0.32	0.55	0.38	0.38	0.24	0.18	0.58
事例	0.73	0.73	0.92	0.85	0.83	0.89	0.78	0.16	0.59	0.77	0.63	0.32	0.32	0.32	0.55	0.38	0.38	0.24	0.18	0.58
事例	0.73	0.73	0.9	0.88	0.86	0.9	0.85	0.4	0.72	0.73	0.73	0.32	0.32	0.32	0.48	0.38	0.38	0.24	0.18	0.45
事例	0.72	0.72	0.88	0.85	0.85	0.86	0.85	0.3	0.68	0.68	0.68	0.25	0.25	0.25	0.35	0.27	0.35	0.2	0.35	0.18
事例	0.53	0.53	0.65	0.65	0.65	0.65	0.65	0.3	0.53	0.53	0.53	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0	0.05

表 2.4 ハンチ部付近の損傷要因と損傷種類の関係 [文献 16]

損傷種類	損傷要因	ひびわれ						剥離				鉄筋露出					
		13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
ひびわれ	28	○	△	○	○	○	○	○	○	◎		○	◎				
	29	△	○	○	○	○	○	○	○	○		△		◎			
	30																
剥離	31	○	○	◎	◎	◎	◎	○	○	◎	○	◎		◎			
	32	◎	○	◎	◎	◎	◎	○	○	◎	○	◎		◎			
	33	◎	◎	◎	◎	◎	◎	○	○	○	◎	◎		○			
	34	◎	△	◎	◎	◎	◎	○	○	◎		○	◎				
	35											○					
	36		○	○	○	○	○										
	37		○	○	○	○	○				○	○	△				
	38		○	○	○	○	○				△	△	△				
鉄筋露出	39	△	△	△	△	△	△	△	△	△	△	△		△	△	△	△
	40	△	△	△	△	△	△	△	△	△	△	△		△	△	△	△
	41	△	△	△	△	△	△	△	△	△	○		△	△	△	△	△
	42		△	△	△	△	△	△	○	○	○	○	○	○	○	○	○
	43	△	△	△	△	△	△	△	△	△	△	△		○	○	○	○
	44		△	△	△	△	△	△	○	○	○			◎	◎	○	○
その他	45		△	△	△	△	△										
	46														△	○	○
	47						△	△			△	◎	◎		△	○	○

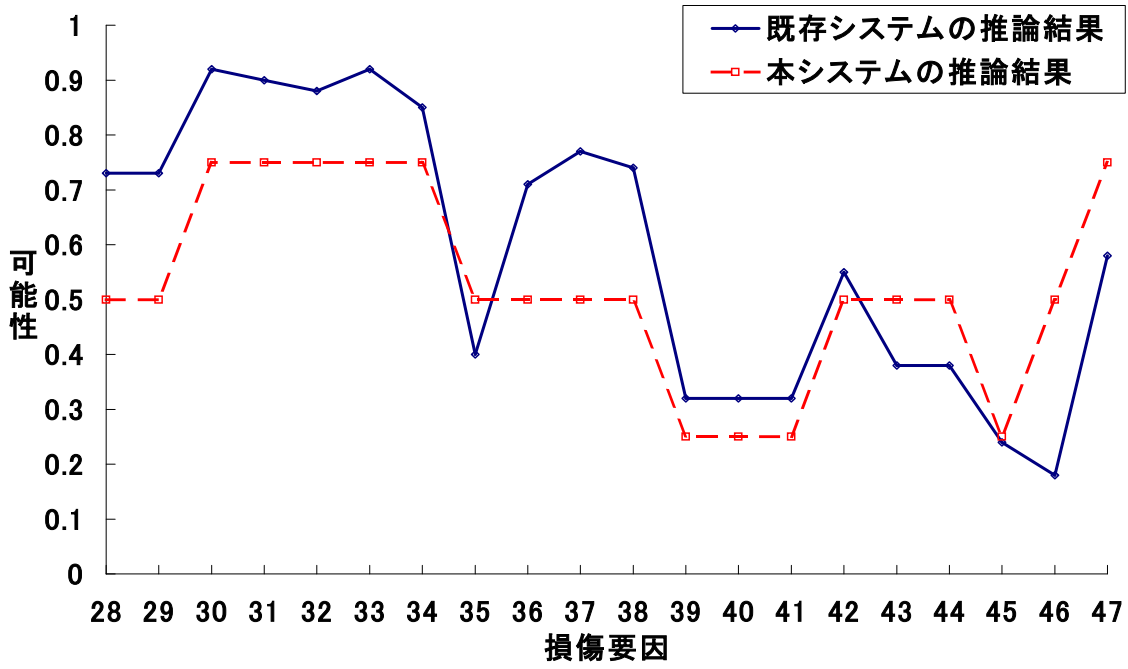


図 2.3 事例 1 の損傷要因推定結果(ルール洗練前)

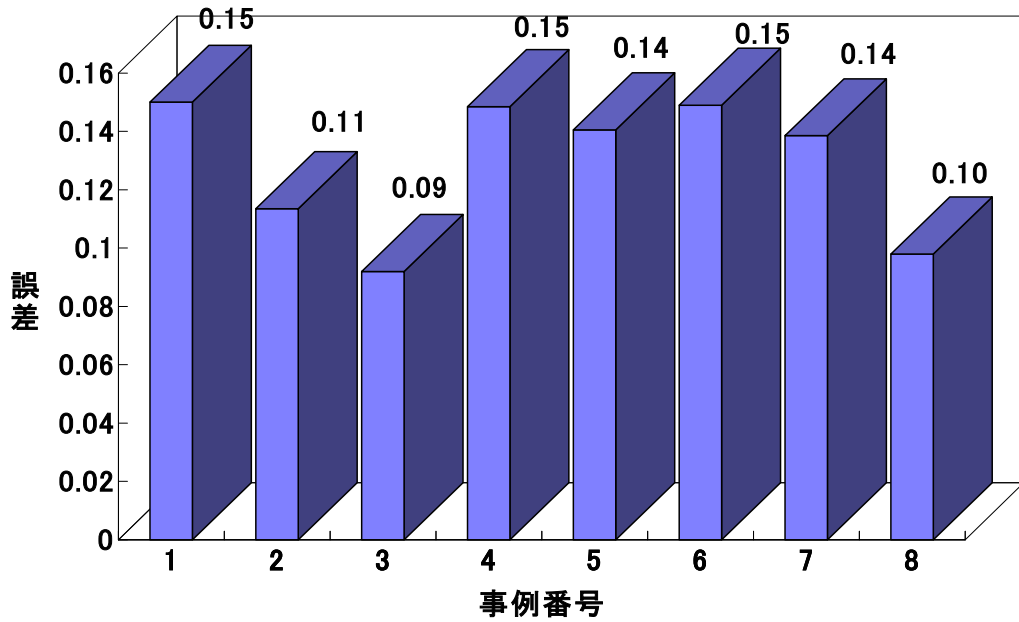


図 2.4 既存システムと本システムの推論結果との間の誤差(ルール洗練前)

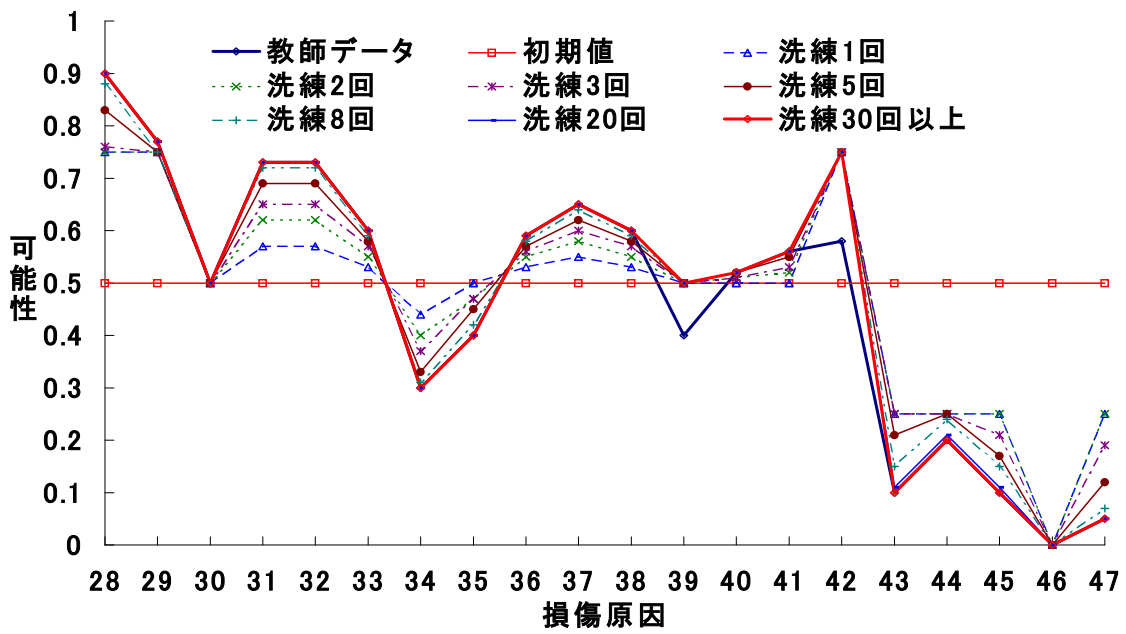


図 2.5 事例 2 の損傷要因推定結果(各事例を個別に提示)

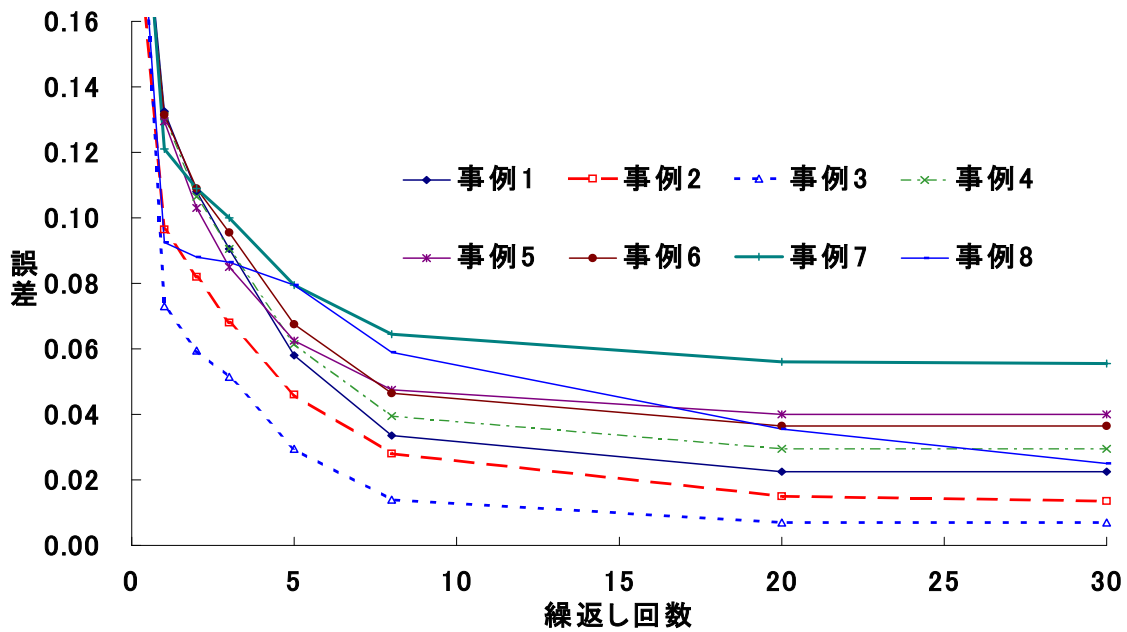


図 2.6 ルール洗練の回数による誤差の変化量(各事例を個別に提示)

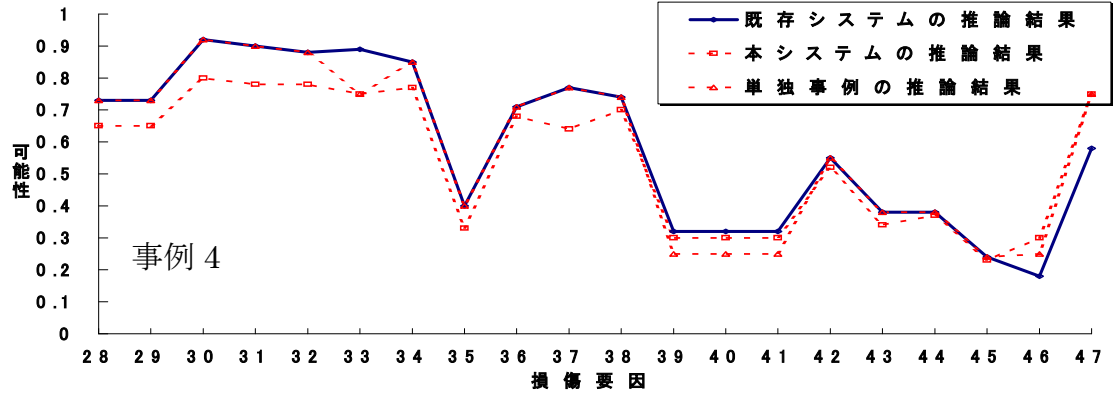
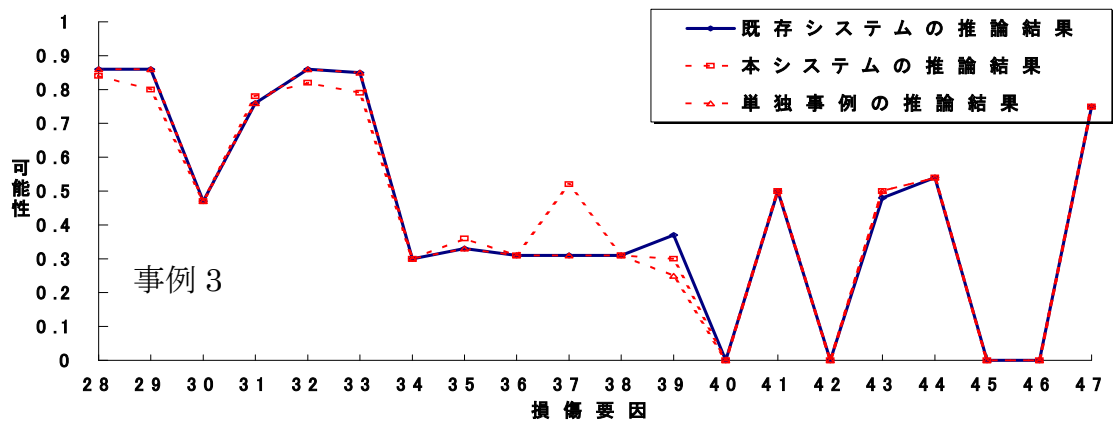
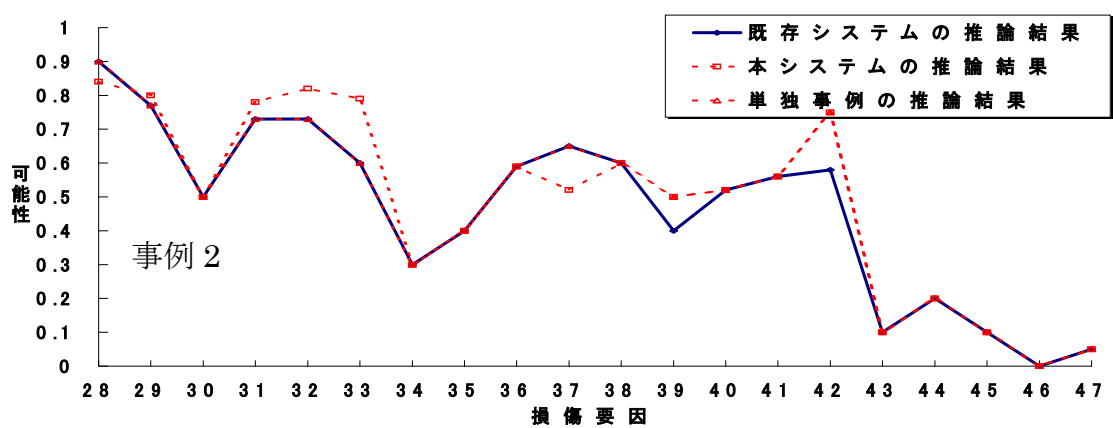
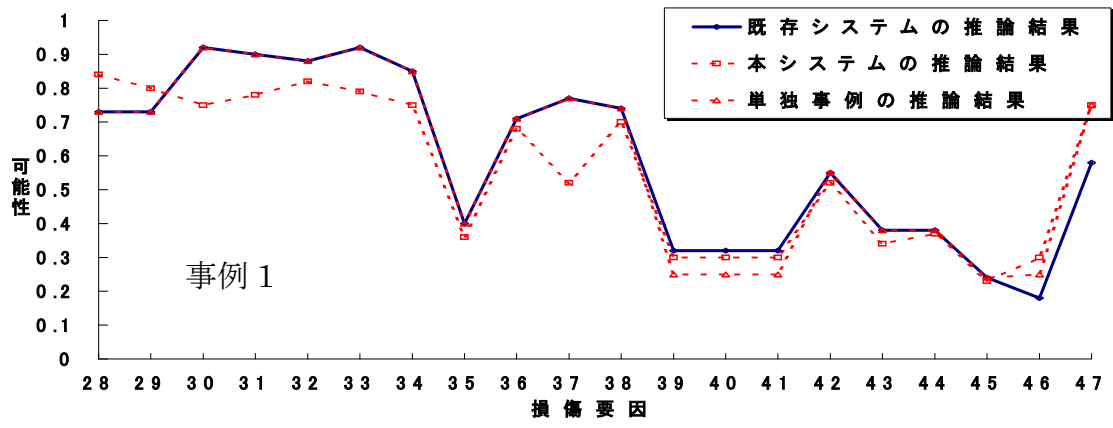


図 2.7 各事例の損傷要因推定結果(8 事例を同時に提示)

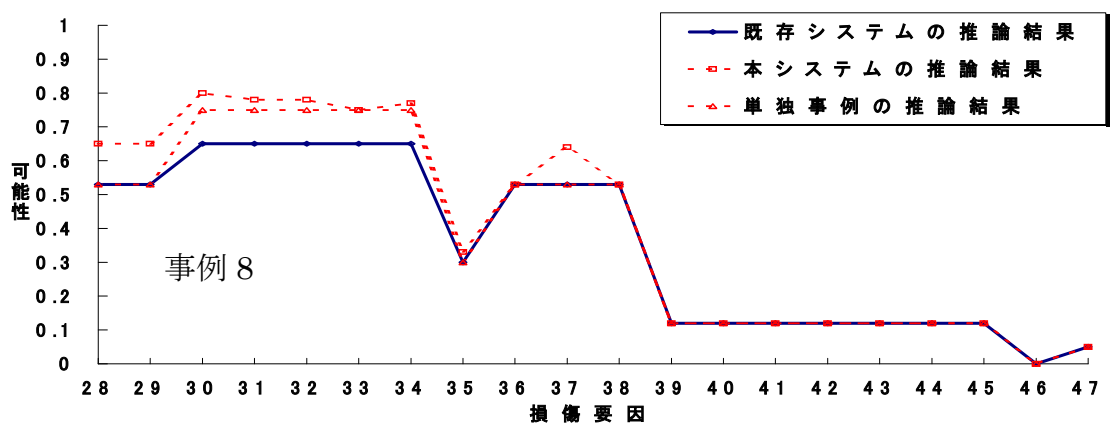
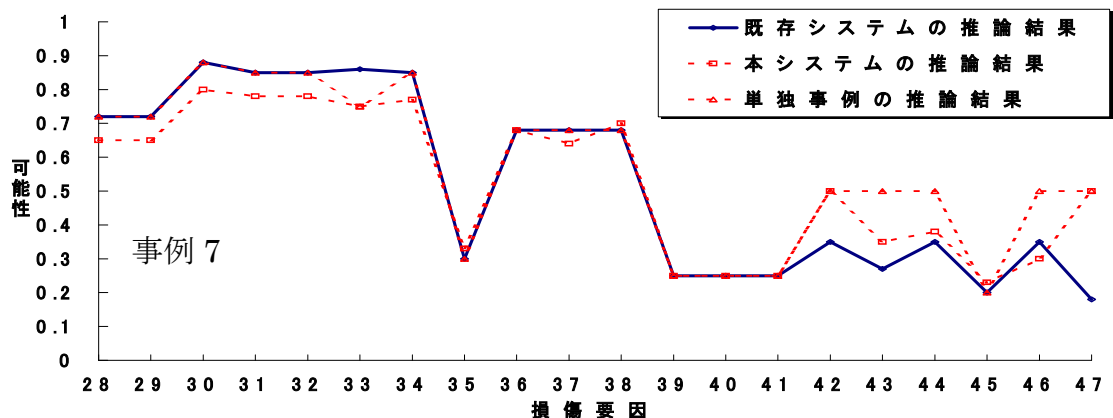
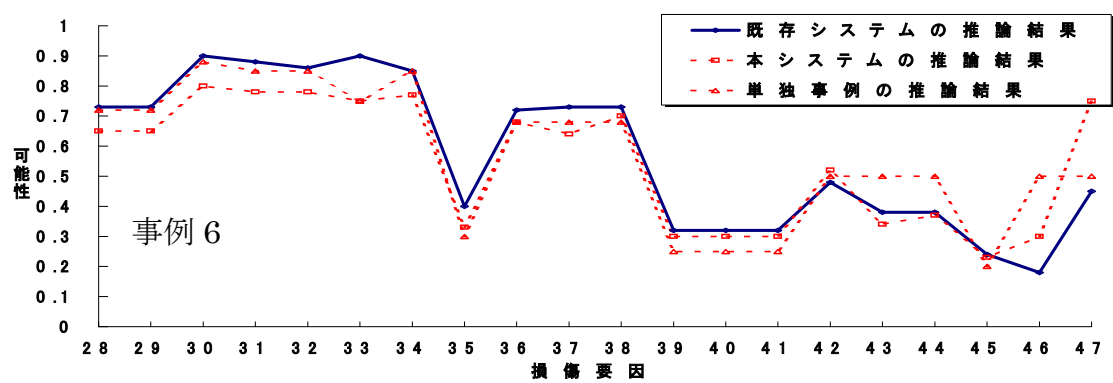
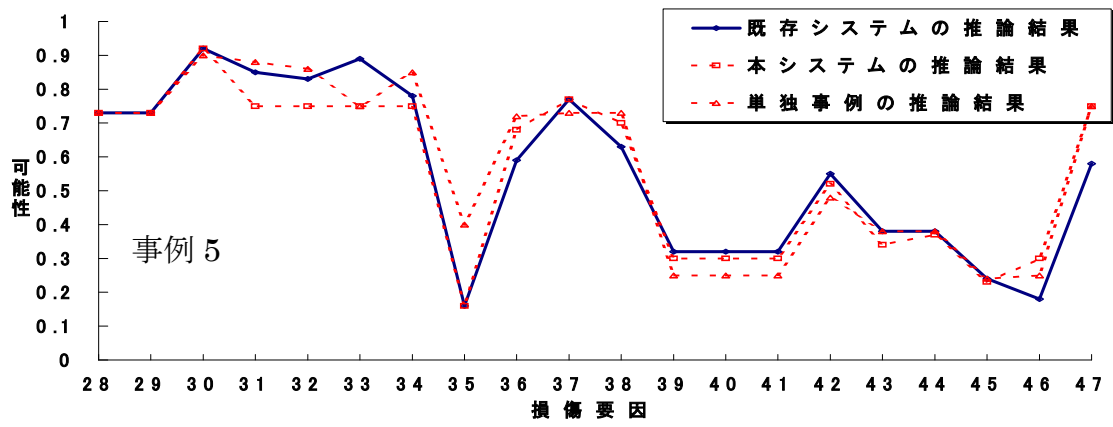


図 2.7 各事例の損傷要因推定結果(8 事例を同時に提示)

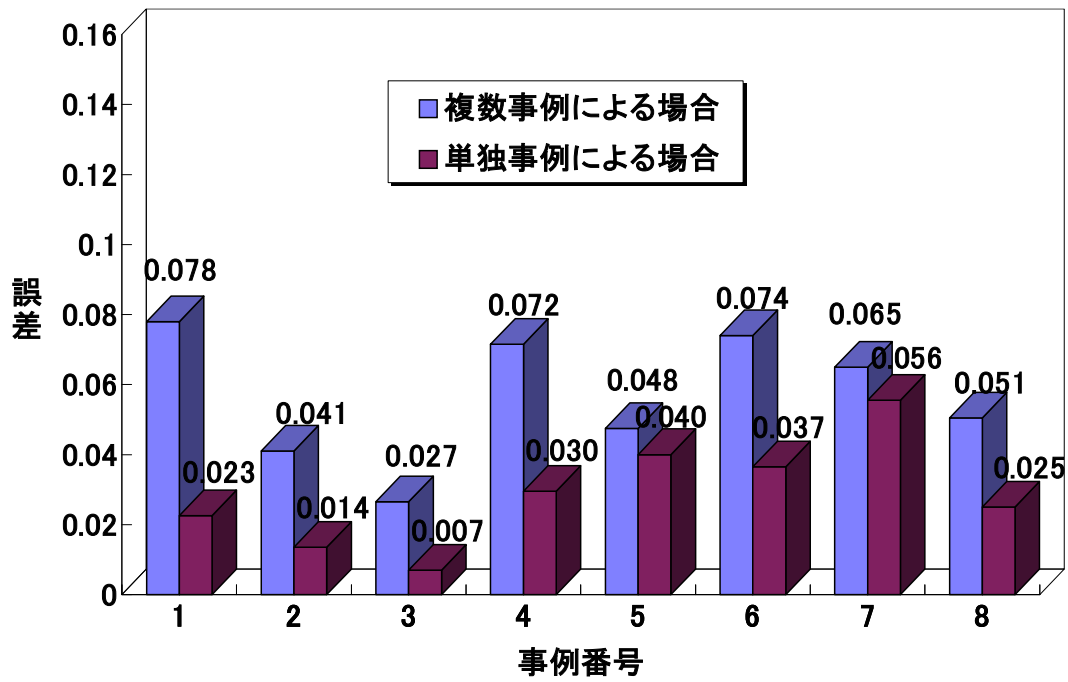


図 2.8 既存システムと本システムの推論結果との間の誤差(8 事例を同時提示)

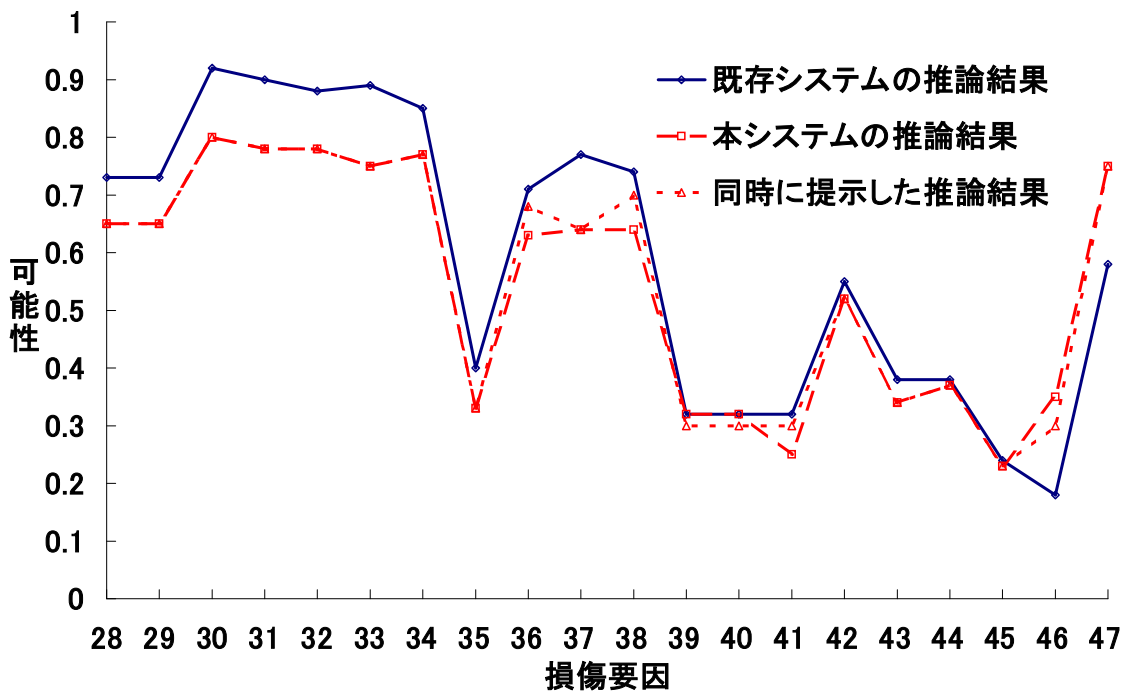


図 2.9 事例 4 についての損傷要因推定結果(同一な損傷場所の 6 事例を提示)

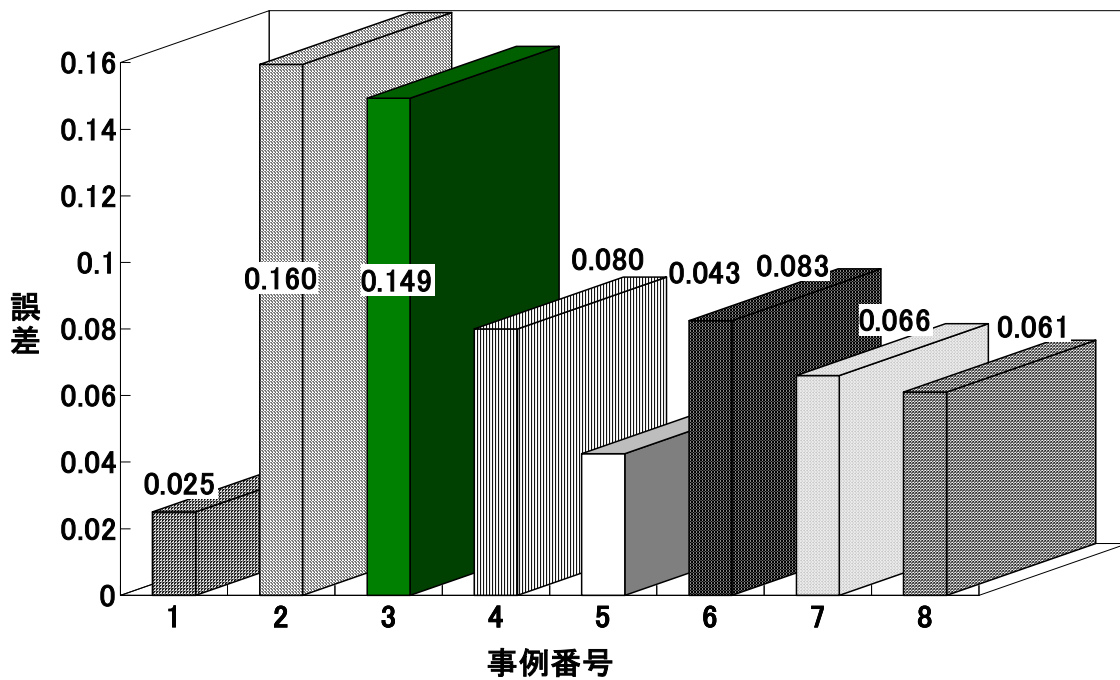


図 2.10 既存システムと本システムの推論結果との間の誤差
(同一な損傷場所の 6 事例を提示)

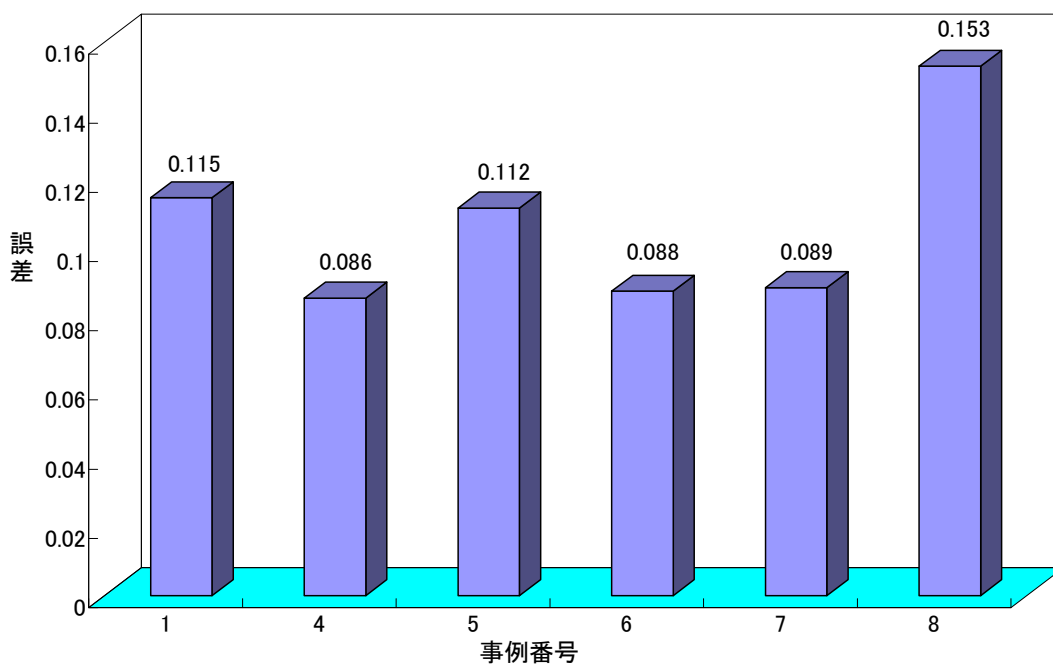
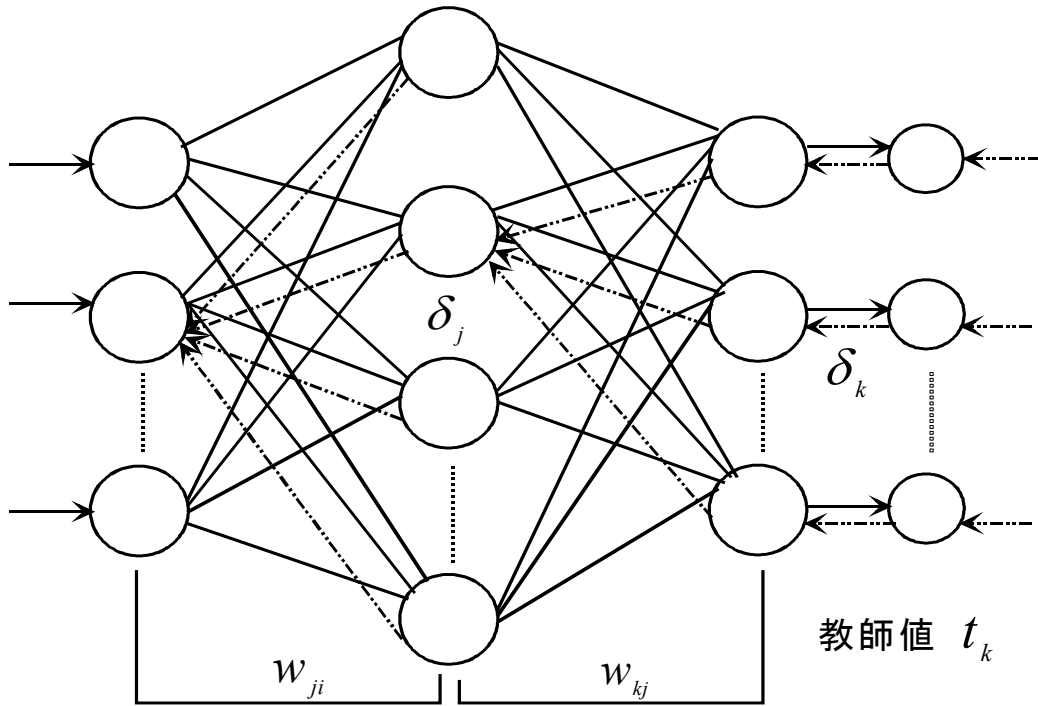
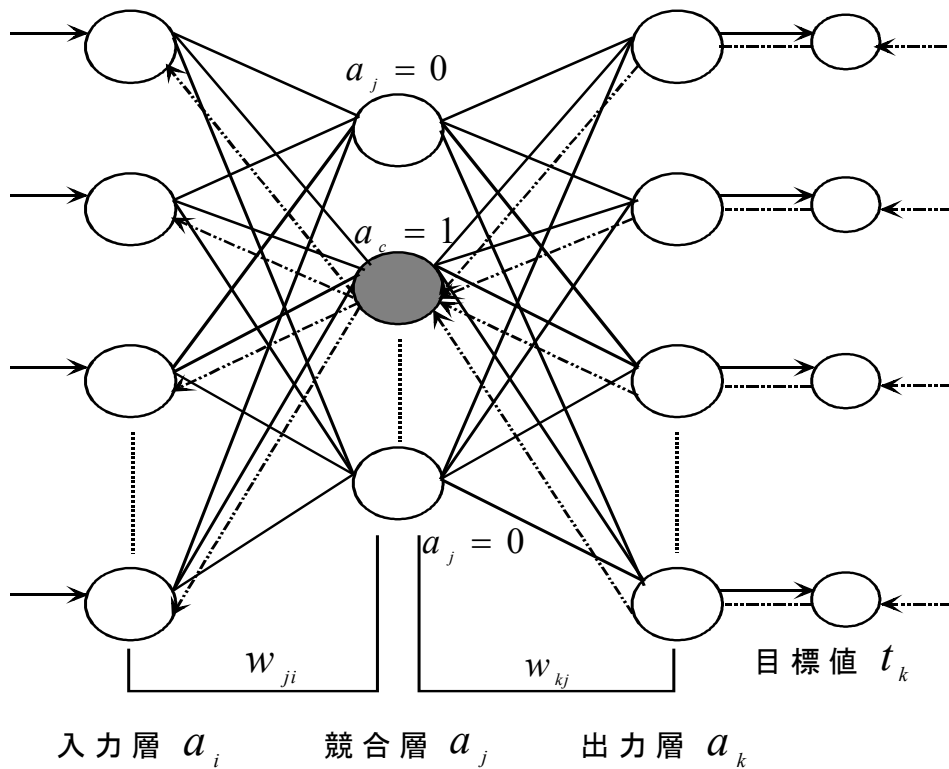


図 2.11 既存システムと本システムの推論結果との間の誤差
(未提示事例についての推定)



入力層 a_i 中間層 a_j 出力層 a_k

図 2.12 誤差逆伝播アルゴリズムの基本構造 [文献 33]



入力層 a_i 競合層 a_j 出力層 a_k

図 2.13 対向伝播アルゴリズムの基本構造 [文献 33]

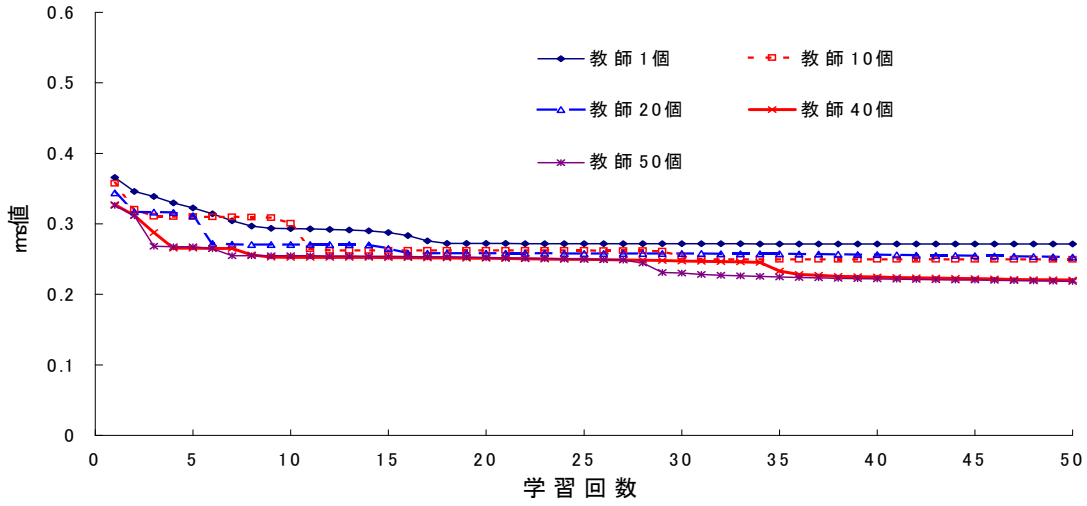


図 2.14 学習回数による rms 値の推移(誤差逆伝播法)

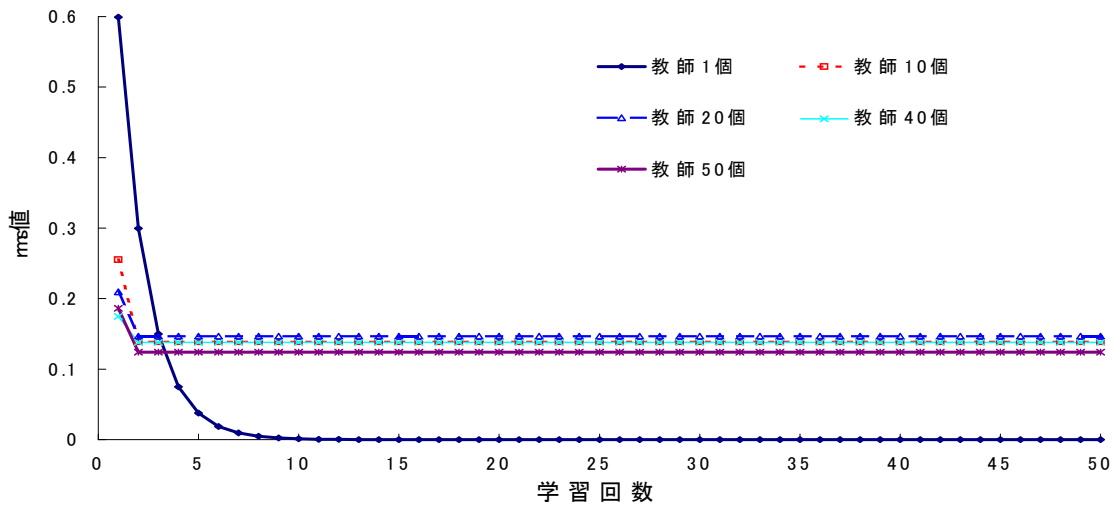


図 2.15 学習回数による rms 値の推移(対向伝播法)

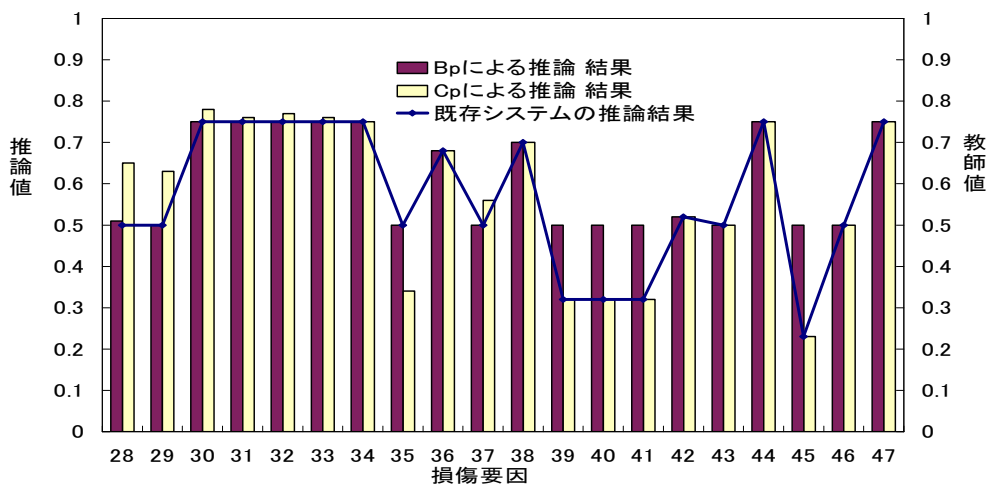


図 2.16 誤差逆伝播法と対向伝播法による損傷要因推定結果

1. はじめに

これまでに、既存のエキスパートシステムのルールベースの再利用を目的として、ルール洗練機能を持つ汎用的な相互結合ネットワークによる推論システムの構築と、ニューラルネットワークを用いた補完システムの有効性を、道路橋 RC 床版の損傷要因推定を対象問題として検討してきた。

その結果、対象問題が、道路橋 RC 床版の損傷要因推定の場合、システムとしての機能が十分果たされていることが示された。今後、本システムの実用化を目指すためには、現在の研究段階において、適用する対象問題を変え、推論システム内で構成するネットワークを変えて複雑化した場合においても、推論システムが十分機能し、精度の高い推論結果を示さなくてはならないと考える。

そこで、本研究では、本システムが、対象問題によらず機能するかを調べるため、対象問題を鋼橋疲労損傷の補修方法選定として、ルールベース洗練機能及び、仮説推論機能の有効性を検討した。

2. 適用したルールベース

本研究では、田中らにより開発された鋼道路橋に発生した疲労損傷の補修・補強方法選定を対象問題としたエキスパートシステムで用いられたルールベースを本推論システムに適用した。[文献 17,34]

2.1 ネットワークの構成

田中らによるシステムでは、図 3.1[文献 34]に示すように、点検リスト提示システムで、構造形式、桁形式、平面形から損傷事例に見られる損傷構造を探索し、点検すべき構造部分を点検リストとして提示し、このリストを基に点検、調査を行い、損傷が発見されれば、損傷構造、損傷部分、溶接種類、継手種類、損傷要素から要因・作用力推定システムにより、外的要因、内的要因、継手の作用力の推定を行っている。そして、補修の必要性があると判断した場合に、補修方法選定システムにより、要因・作用力推定システムで推定した外的要因、内的要因、継手の作用力の情報に加え観測された事実である亀裂様式の情報を用いて、補修方法の選定を行っている。しかし、点検リスト提示システム及び、要因・作用力推定システムでは、フレームを用いた推論を行っているため、推論過程がブラックボックスとなっている。

そのため、本研究では、亀裂の外的要因、亀裂の内的要因、継手の作用力を亀裂様式と同様、観測された事実である既知情報とみなし、仮説間の関係を相互結合させて、システム内のネットワークを図 3.2 のように構成した。

2.2 仮説(ノード)の設定

本推論では、亀裂の内的要因、外的要因、継手の作用力、亀裂様式及び、補修方法の項目を示すノードの総数を 54 個に設定した。このうち、観測された事実の入力情報として表 3.1[文献 34]に示す 30 個の入力項目(図 3.2 中のノード番号 1~30)を用意し、推論の対象である補修方法の項目は、表 3.2[文献 34]に示す 24 個の出力項目(同ノード番号 31~54)を用意した。

2.3 ルールの提示

田中らによるシステムのルールベースでは、各ルールの因果関係の強さを *Necessity*, *High Possibility*, *Possibility*, *Low Possibility* の 4 段階に分類していることから、本システムに適用するにあたり、結合係数 0.8, 0.6, 0.4, 0.2 をそれぞれ対応させ、同じく 4 段階に分類してルールベースの初期状態を構築した。なお、ルールベースのルールの総数は、267 個とした。

入力情報及び、推論対象の項目と、ルールの結合の関係を、図 3.2 中のノード番号とルールの結合番号を用いて以下に示す。

ノード番号 1-11 : 亀裂の外的要因
ノード番号 12-16 : 亀裂の内的要因
ノード番号 17-23 : 継手の作用力
ノード番号 24-30 : 亀裂様式
ノード番号 31-54 : 補修方法

結合番号 1-55 : [亀裂の外的要因－補修・方法]
結合番号 65-103 : [亀裂の内的要因－補修方法]
結合番号 104-151 : [継手の作用力－補修方法]
結合番号 152-203 : [亀裂様式－補修方法]
結合番号 204-253 : [亀裂の外的要因－亀裂の内的要因]
結合番号 256-267 : [亀裂の内的要因－継手の作用力]

3. 推論システムの適用例

3.1 入力情報を既知とした場合

2 節のルールベースに対して本システムを適用するにあたっては、既知情報とみなした、亀裂の外的要因・内的要因、継手の作用力、及び亀裂様式に対しては、ノード値として 1.0(真の場合)または 0.0(偽の場合)を、確信度として 1.0 を付与した。また、未知情報である補修方法に対しては、真偽が不明であるという意味から、ノード値 0.5 及び確信度 0.1 を付与した。

はじめに、表 3.3[文献 34]に示した事例について、田中らによるシステムの推論結果をルールベース洗練の際に提示する事例として用い、ルール洗練を事例 1,2 について行った。次に、事例中の補修方法が不明であるとして、仮説推論となる補修方法の選定を行った。図 3.3 a に事例 1 についての推論結果を示し、図 3.3 b に事例 2 についての推論結果を示す。図 3.3 は、横軸に表 3.2 の補修方法項目のノード番号をとり、縦軸に補修方法の可能性であるノード値をとり、学習率を 0.3 としたときの推論結果である。また、推論精度の検討のため、図中には教師データを併せて示した。図 3.4 a,3.4b には、推論過程における収束状況を比較するため、学習率を変化させた場合の学習回数に伴う教師データと推論によって得られたノード値との間の誤差を図中に定義した式により求め、横軸に学習回数を対数でとり、縦軸に誤差をとった結果を示す。また、図 3.5 a,3.5 b には、図中に定義した式より、洗練回数毎の結合係数の変化量を求め、横軸にルールベースの洗練回数を対数でとり、縦軸に変化量をとった結果を示す。

この結果、補修方法の選定では、実際に取られた補修方法であるストップホール、グラインダを最も高い可能性として推論しており、教師データとの誤差の値も 0.058 と微少な値に収束していることから、推論システムとしての機能は、十分果たしていると考ええる。また、ルールベースの洗練状況も学習率の違いにより変化の傾向は違うが、洗練回数が増すごとに収束していき、これに加え推論結果が妥当な解を示しているということから、本システム内のルール洗練機能も十分機能していると考ええる。

しかし、本来、本システムは教師付学習型のシステムであることから、本適用例のように教師データとして 1 事例を提示し、提示した事例についての推論を行う場合には、推論結果の値は、教師データに一致するべきである。

このように推論結果に多少の違いが見られた理由には、田中らのシステムでは、図 3.2 に示すように複雑なネットワークを構成して補修方法の選定を行っているのに対して、本適用においては、亀裂の外的要因、内的要因、継手の作用力を既知情報としてみなして推論を行ったことが挙げられる。

例えば継手の作用力に注目すると、継手の作用力は、ネットワーク構造より、補修方法選定のための入力情報となっており、かつ亀裂の外的要因、及び亀裂の内的要因との間に因果関係が存在することから、これらの情報から推定される出力先ともなっており、これらの全ての情報を基に継手の作用力と補修方法との間の推論結果を示すべきである。しかし、既知情報とみなしたことにより、ネットワーク構造による亀裂の外的要因や内的要因との間の因果関係知識が推論結果に反映されなかったと考えられる。このことは、

亀裂の内的要因と外的要因の関係などについても同様にいえる。その結果、補修方法選定に関しては、図 3.6 に示す 8 通りの推論経路が存在している。

そこで、次の 3.2 では、本節で既知情報とみなした入力情報が、推論過程及び、推論結果に及ぼす影響を調べることにした。

3.2 未知情報が含まれる場合

ここでは、次に挙げる 3 つの場合について推論を行い、前述で既知情報とみなした各情報が、推論結果に及ぼす影響を調べることにした。

(I) 亀裂の内的要因を未知情報とした場合

(II) 継手の作用力を未知情報とした場合

(III) 内的要因、作用力を未知情報とした場合

ここで、“未知情報”としたノードに対しては、[ノード値, ノード確信度]として[0.5, 0.1]を付与した。

事例 1, 2 について、上述の(I)から(III)における推論結果を図 3.7a,b から図 3.9a,b にそれぞれ示す。同図は、図 3.3 と同様横軸に補修方法のノード番号をとり、縦軸に補修方法の可能性をとった。また、比較のため、図 3.3 で示した入力情報が既知である場合の推論結果と、教師データを併せて示した。

図 3.6 に示したように、ここで未知情報とした項目は、どれも他の情報からの推論の対象になっている。そこで、図 3.10a,b から図 3.12a,b には、推論の対象となった場合に推論される未知情報のノード値を(I)から(III)についてそれぞれ示す。同図は、横軸に入力情報の項目番号をとり、縦軸に推論されたノード値をとった。

(I) 亀裂の内的要因を未知情報とした場合

事例 1 の推論結果(図 3.7a)、事例 2 の推論結果(図 3.7b)共に既知情報とみなした場合の結果とほとんど違いがなく、亀裂の内的要因が補修方法選定に及ぼす影響は少ないと考えられる。この事は、図 3.10a, 図 3.10b から示すことができる。図 3.10a では、既知情報の場合と同じ内的要因の項目が、ノード値 0.2 と低い値を示し、図 3.10b でも、既知情報とした場合と同じ項目のノード値が 0.6 で、それ以外、ノード番号 14 のノード値が 0.2 であることから、内的要因が補修方法選定に与えた影響は小さいと考えられる。

(II) 継手の作用力を未知情報とした場合

図 3.8a,b からわかるように、既知情報とした場合における推論結果との違いが非常に大きく、教師データとの一致度も小さいことがわかる。これは、継手の作用力が、構成するネットワークにおいて、補修方法選定に大きく影響していることを示している。このことは、図 3.6 で継手の作用力が経路内において、他の入力情報からの推論対象となっている割合が非常に高いことからいえる。

特に、事例 2 についての図 3.8b では、教師データ、既知情報とした場合の推論結果が

共に低い可能性を示している項目について、0.8 と非常に高い可能性を示していることがわかる。この原因としては、図 3.11b に示すように、既知情報とした場合は、ノード番号 23 のノード値が唯 1 つ 1.0 となっているのに対し、未知情報とした場合は、ノード番号 17, 18, 20 の 3 つのノードがノード値 0.8 と非常に高い値を示しているにもかかわらず、ノード番号 23 については、ノード値 0.2 と非常に低くなっているため、継手の作用力と補修方法との間の因果関係に大きく影響し、その結果、補修方法選定にも大きな影響を与えたと考えられる。

(Ⅲ) 亀裂の内的要因, 継手の作用力を未知情報とした場合

図 3.12a,b より、内的要因を未知情報としているため、継手の作用力のノード(ノード番号 17, 18, 20)が、(Ⅱ)の場合に比べて、低い値で活性化されていることから、継手の作用力と補修方法の間の因果関係に与えた影響が小さくなり、推論結果への影響も小さくなったため、(Ⅱ)の場合より教師データに近い、精度の良い結果を示したと考えられる。

以上、(Ⅰ)から(Ⅲ)の推論を試みた結果、鋼橋疲労損傷の補修方法選定で構成したネットワークにおいて、特に、継手の作用力に関する情報が、推論結果である補修方法選定に大きく影響していることが示された。また、複雑なネットワークを構成して推論を行うと、図 3.12a,b や図 3.13b で特に顕著に表されているように、幾つかのノードが、高いノード値で存在した場合に、これらの出力先となる結論部のノードが、どの条件部のノードを優先させて選択すべきかが不明になり、推論結果に影響を及ぼすことが示され、今後の課題となる。

この課題の解決策としては、本システムで適用している確信度を用いたルール及び、ノードの優先度の決定などによるネットワーク内の各構成要素の順位付けを体系化することが有効であると思われる。

4. おわりに

3 章では、田中らによって開発された鋼道路橋疲労損傷の補修方法選定システムのルールベースを適用して、皆川らによるルール洗練機能付エキスパートシステムの有効性を検討した。

この結果、本システムが対象問題によらず、本来エキスパートシステムに要求される対象問題の可能性の分類を行い、技術者支援システムとして機能していることが示された。また、2 章、3 章で特に注目していた本システムのルール洗練機能が、目標とする知識の共有・再利用といった知識獲得作業に対し、有効に機能していることも示された。今後、実用的な大規模ルールベースを構築し、精度の良い推論を可能にするためには、先に述べたように、確信度を用いて、ノード及び結合係数の順位付けを体系的に確立する必要があると考える。

表 3.1 入力情報項目 [文献 34]

	外的要因
1	風による振動
2	活荷重の作用
3	地震による振動
4	低温
5	横分配作用
6	ディテールの不適性
7	2次的変形
8	材質不良
9	溶接不良
10	製作誤差
11	輸送架設荷重
	内的要因
12	応力集中
13	2次応力の応力集中
14	2次応力
15	応力集中による座屈2次応力
16	残留応力
	継手の作用力
17	T継手の作用力②
18	T継手の作用力③
19	T継手の作用力⑥
20	T継手の作用力⑦
21	突合せ継手の作用力⑨
22	突合せ継手の作用力⑩
23	重ね継手の作用力⑫
	継手の亀裂様式
24	T継手の亀裂様式(a)
25	T継手の亀裂様式(b)
26	T継手の亀裂様式(c)
27	T継手の亀裂様式(d)
28	T継手の亀裂様式(f)
29	重ね継手の亀裂様式(i)
30	重ね継手の亀裂様式(j)

表 3.2 補修方法項目 [文献 34]

31	ストップホール
32	ガウジング
33	グラインダ
34	ビーニング
35	腹板ギャップの増大
36	腹板厚の増大
37	切り抜き
38	再溶接
39	フランジと補剛材の溶接
40	再溶融
41	補剛材付き添接板
42	高力ボルト
43	添接板
44	挿入板
45	ケーブルによる吊材相互の緊結
46	横桁と主桁のフランジ連結
47	対傾構と主桁のフランジ連結
48	ダイヤフラムと主桁のフランジ連結
49	横桁とアーチリブのフランジ連結
50	縦桁支承の交換
51	主桁の交換
52	連結板の交換
53	新補剛材の設置
54	制振装置の設置

表 3.3 提示する事例 [文献 34]

橋梁名	事例 1	事例 2
国名	米国	米国
架設年度	1958年	1973年
損傷発見年度	12年	0年
構造形式	単純橋	連続橋
桁形式	格子合成プレートガーター	合成プレートガーター
平面形	直橋	直橋
損傷構造	主桁のカバープレートと取り付け部	主桁の垂直補剛材取り付け部
損傷部分	主桁下フランジ カバープレート	主桁腹板 垂直補剛材
溶接種類	隅肉溶接	隅肉溶接
継手種類	重ね継手	T継手
損傷要素	隅肉溶接	主桁腹板
外的要因	活荷重の作用	輸送架設荷重
内的要因	応力集中	2次応力
継手の作用力	重ね継手の作用力 12	T継手の作用力 3
亀裂様式	重ね継手の亀裂様式 (i)	T継手の亀裂様式 (d)
亀裂方向	溶接に平行な亀裂	溶接に平行な亀裂
補修方法	ビーニング 再溶融 高力ボルト 添接板	ストップホール グラインダ
損傷状況		

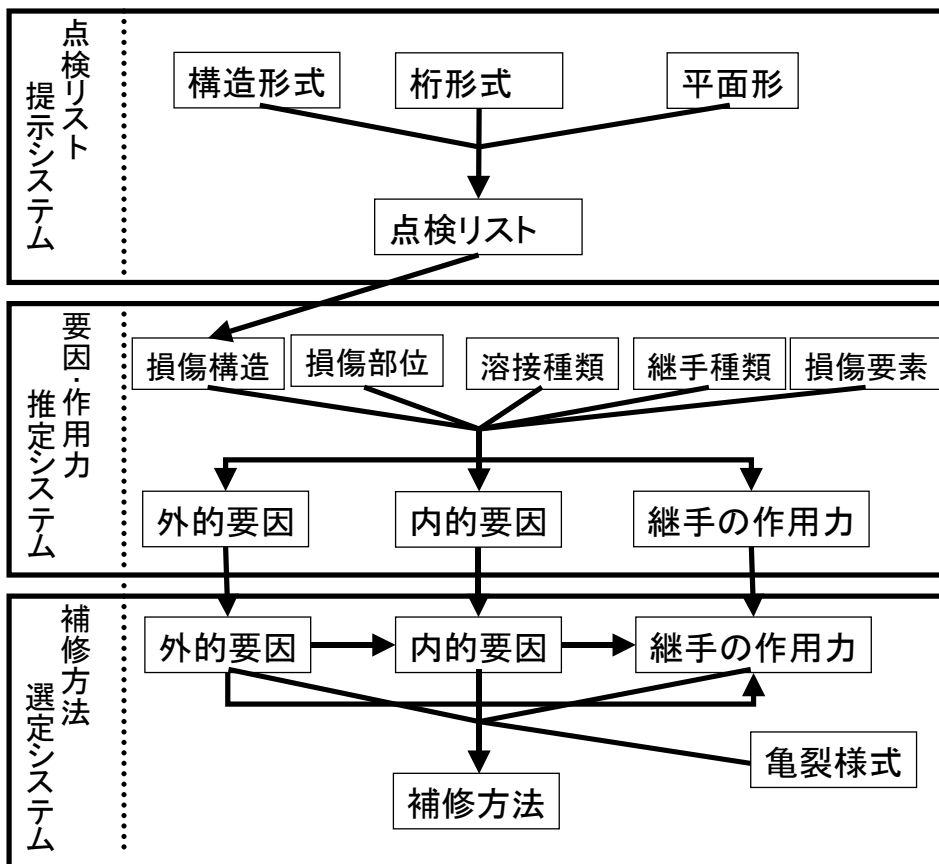


図 3.1 田中らによる推論システム [文献 34]

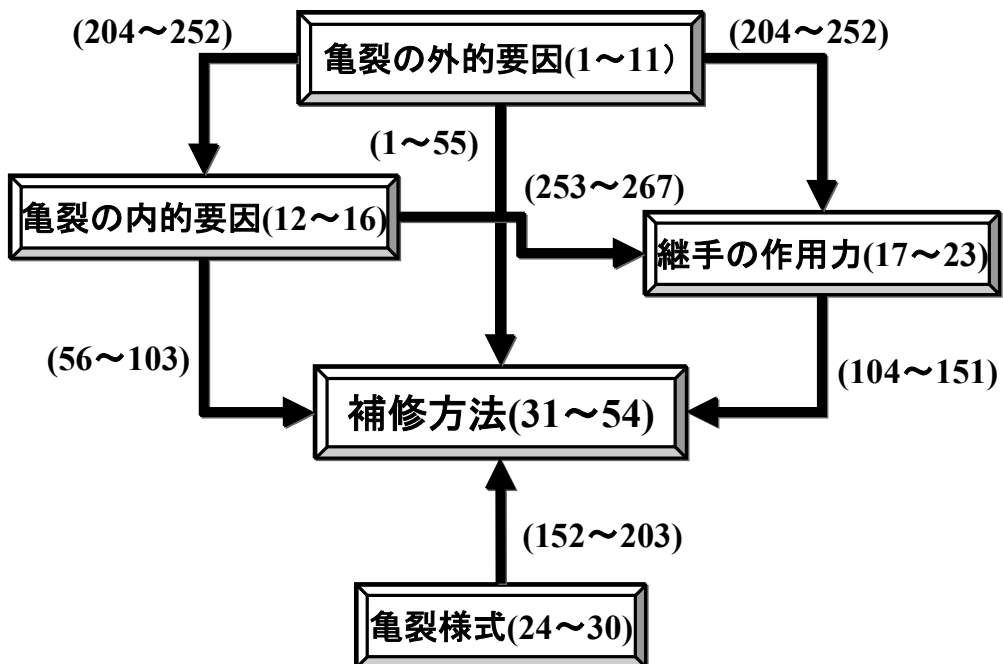


図 3.2 ネットワークの構成

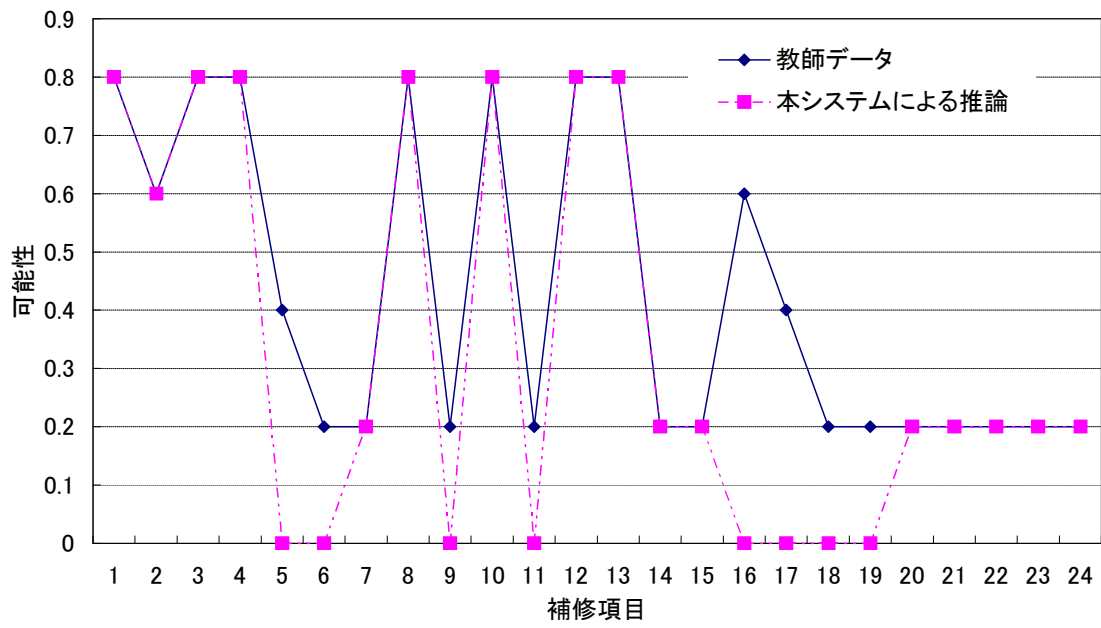


図 3.3a 補修方法選定結果(事例 1)

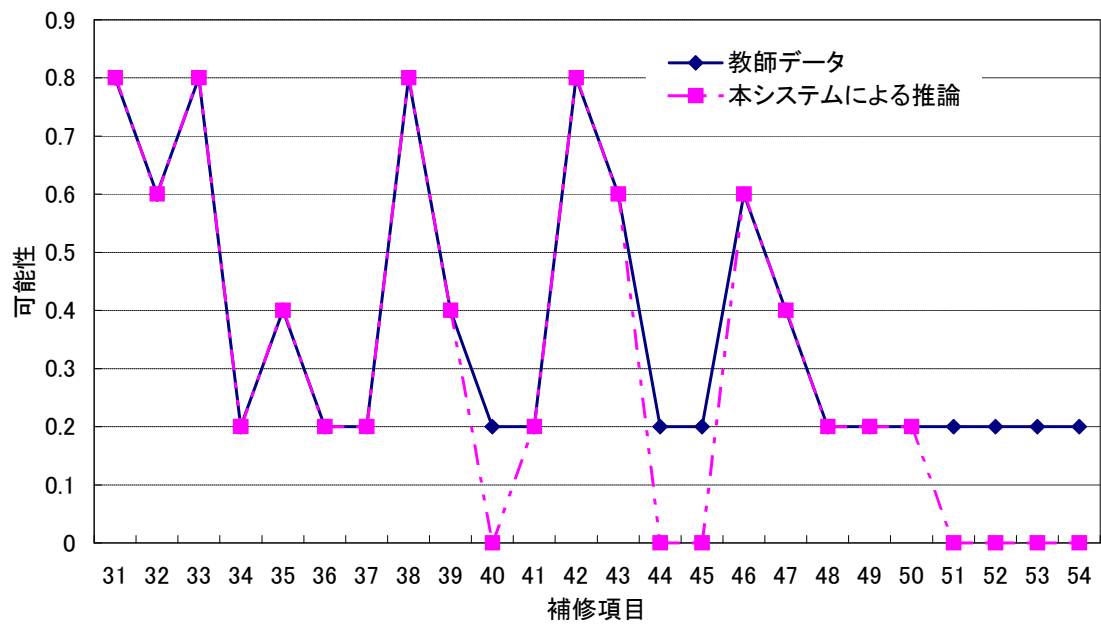


図 3.3b 補修方法選定結果(事例 2)

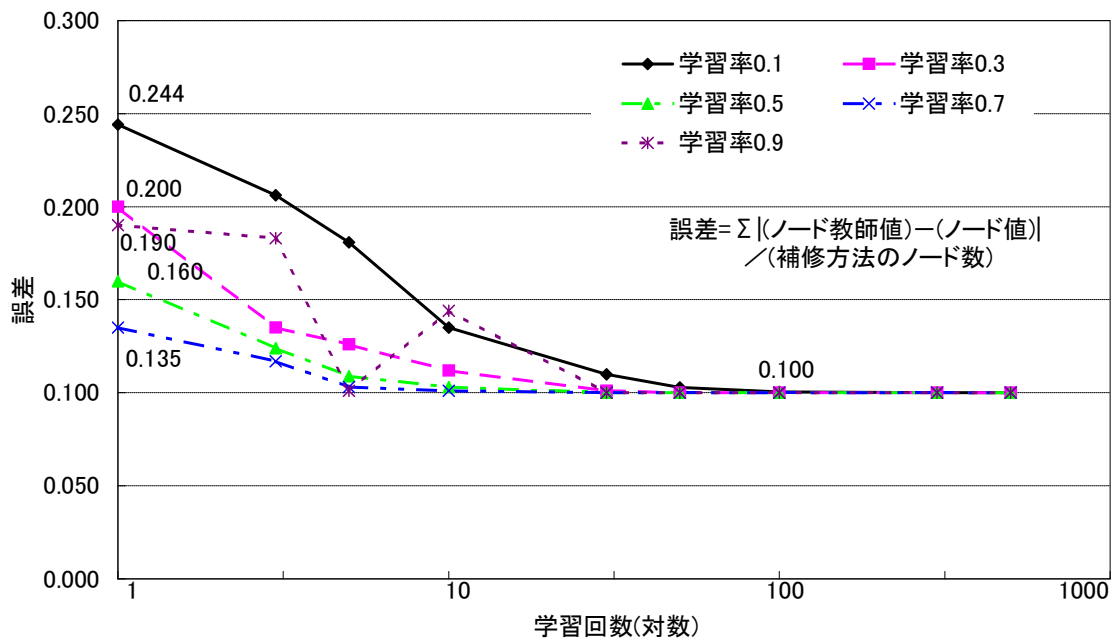


図 3.4a 推論過程における収束状況(事例 1)

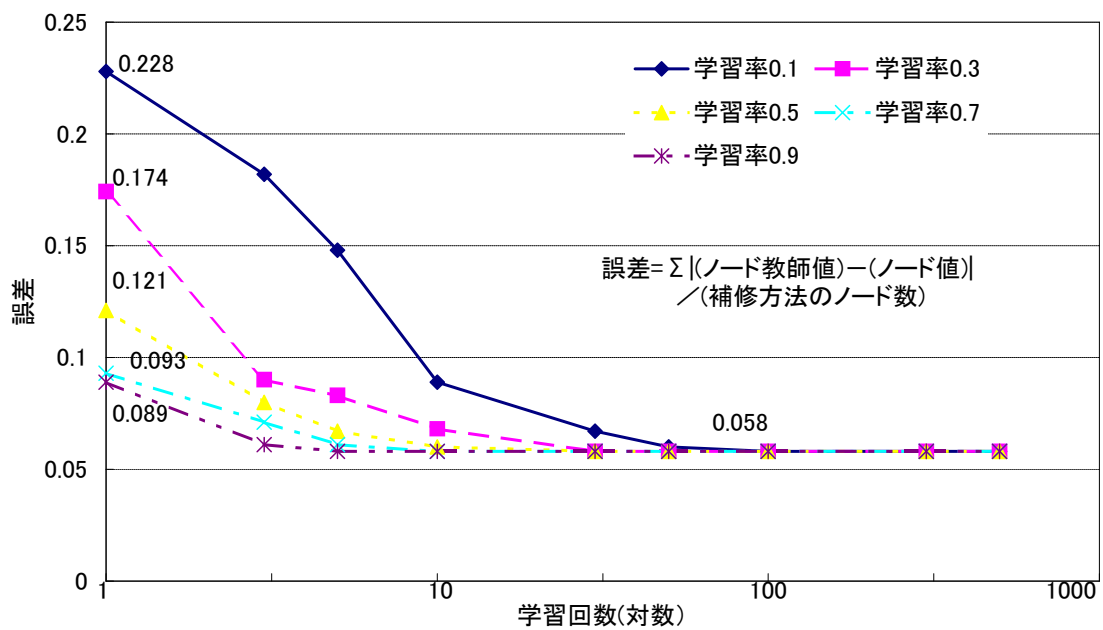


図 3.4b 推論過程における収束状況(事例 2)

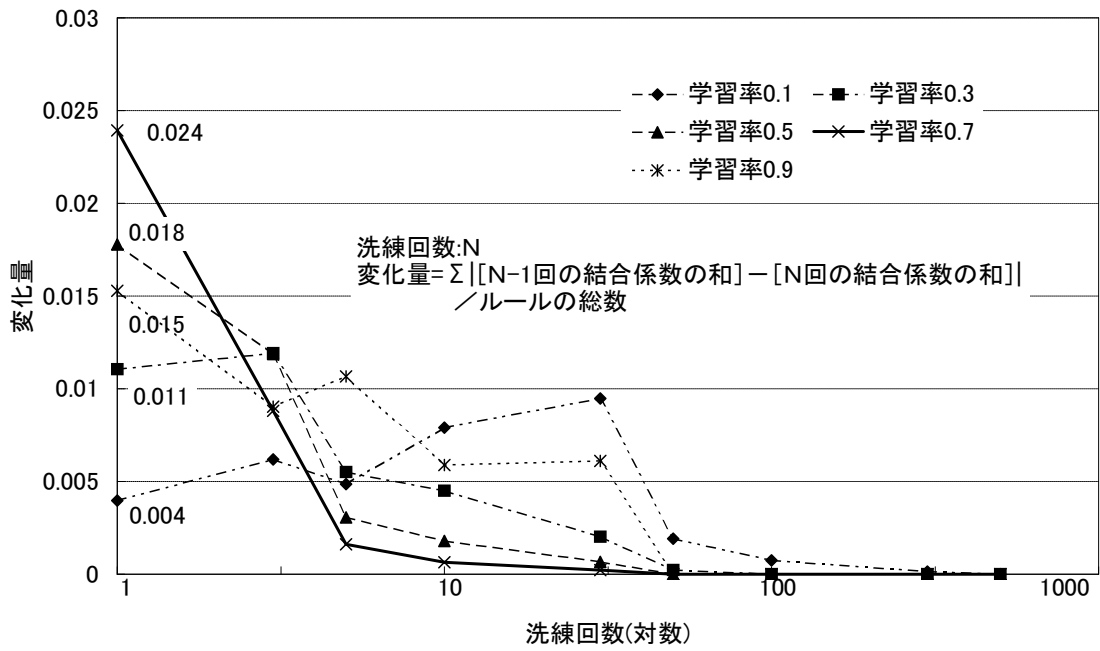


図 3.5a 結合係数の変化量(事例 1)

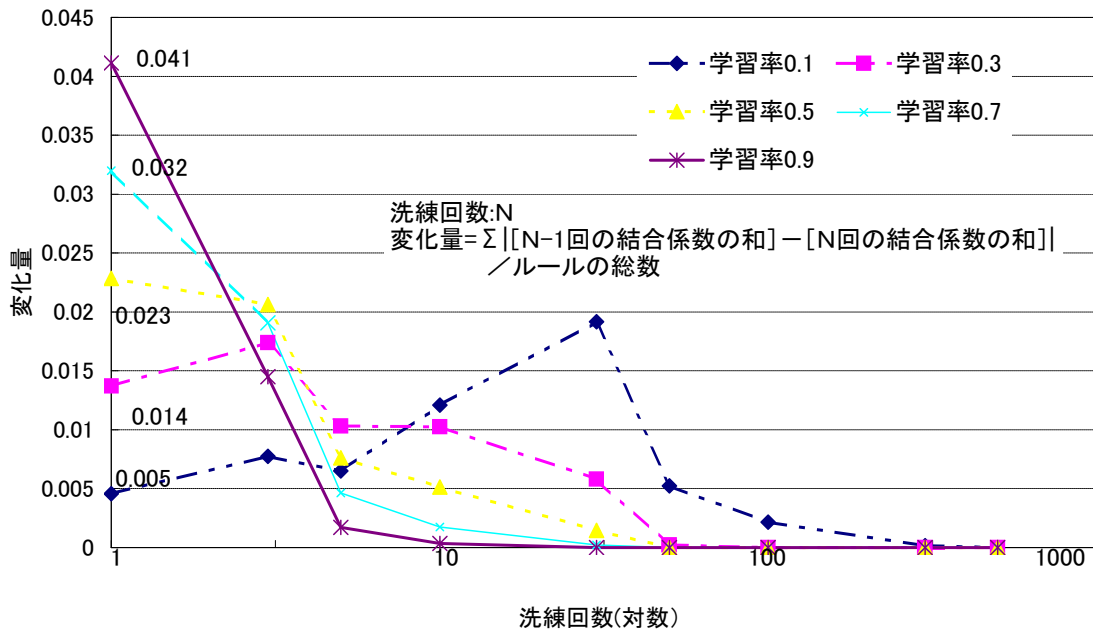


図 3.5b 結合係数の変化量(事例 2)

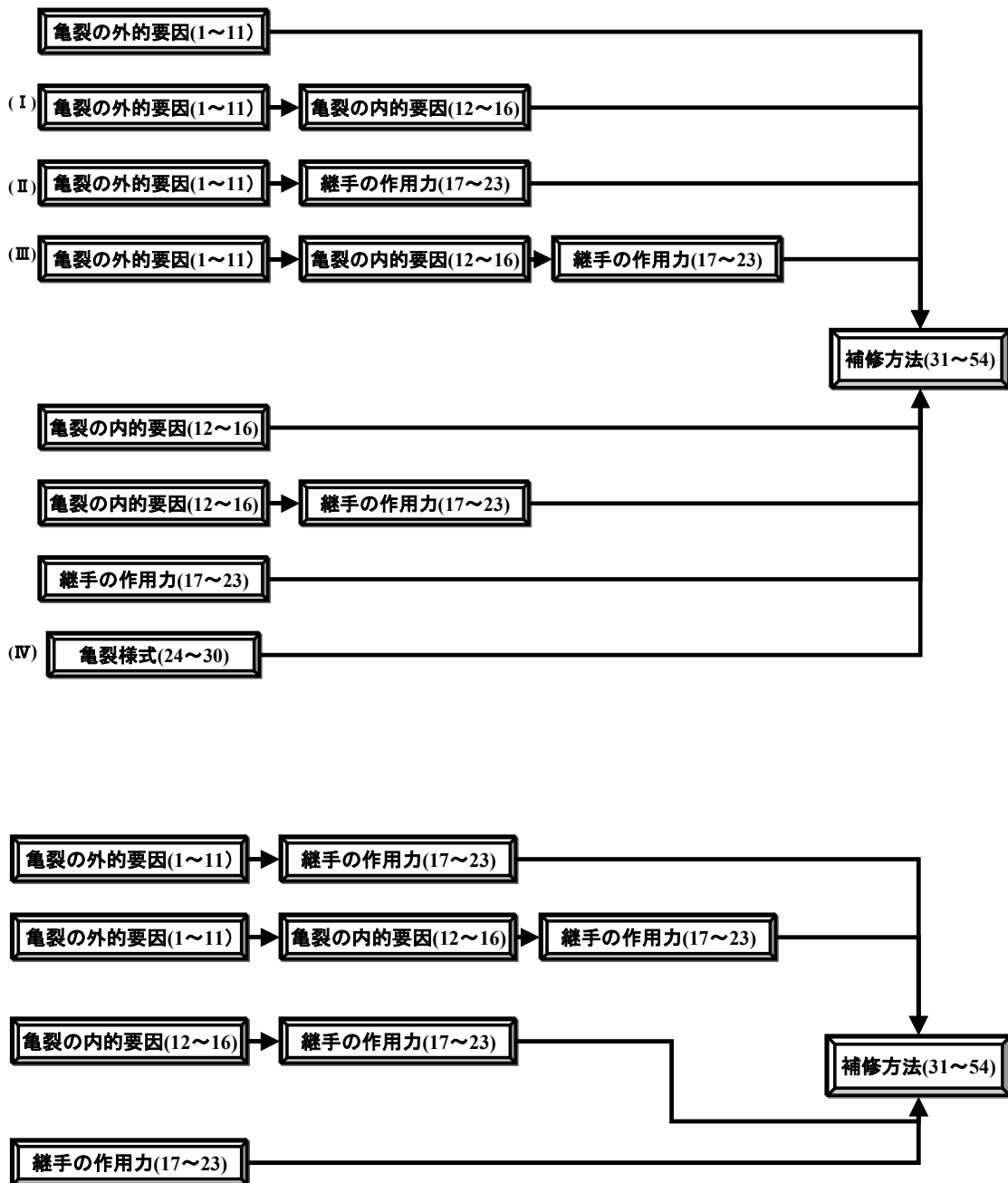


図 3.6 ネットワークの推論経路

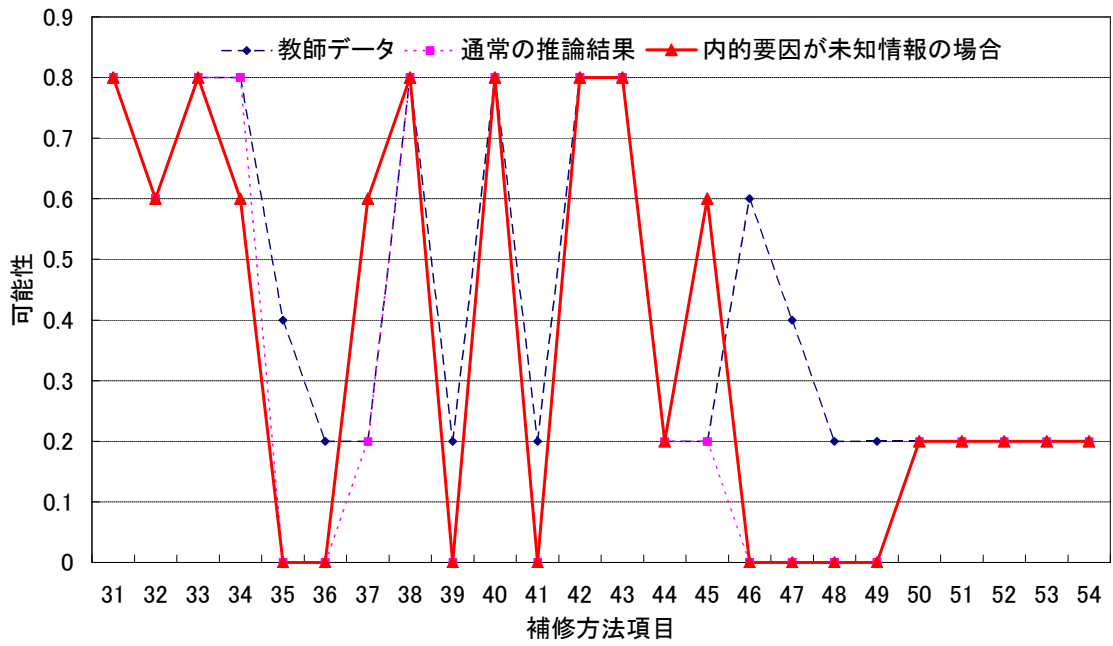


図 3.7a 内的要因が未知情報の場合の補修方法選定結果(事例 1)

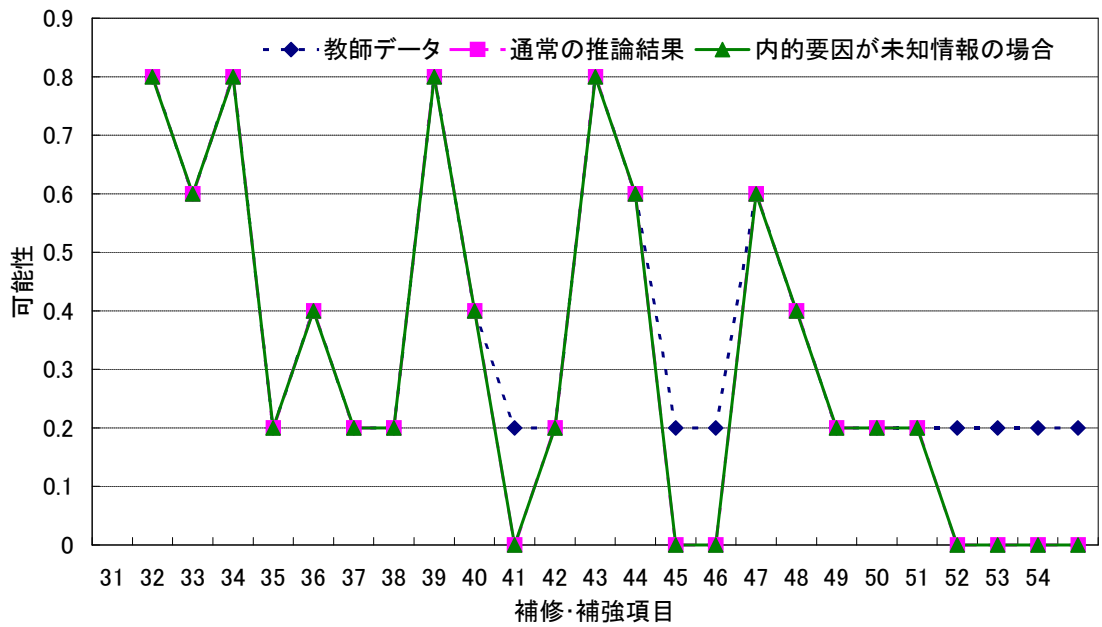


図 3.7b 内的要因が未知情報の場合の補修方法選定結果(事例 2)

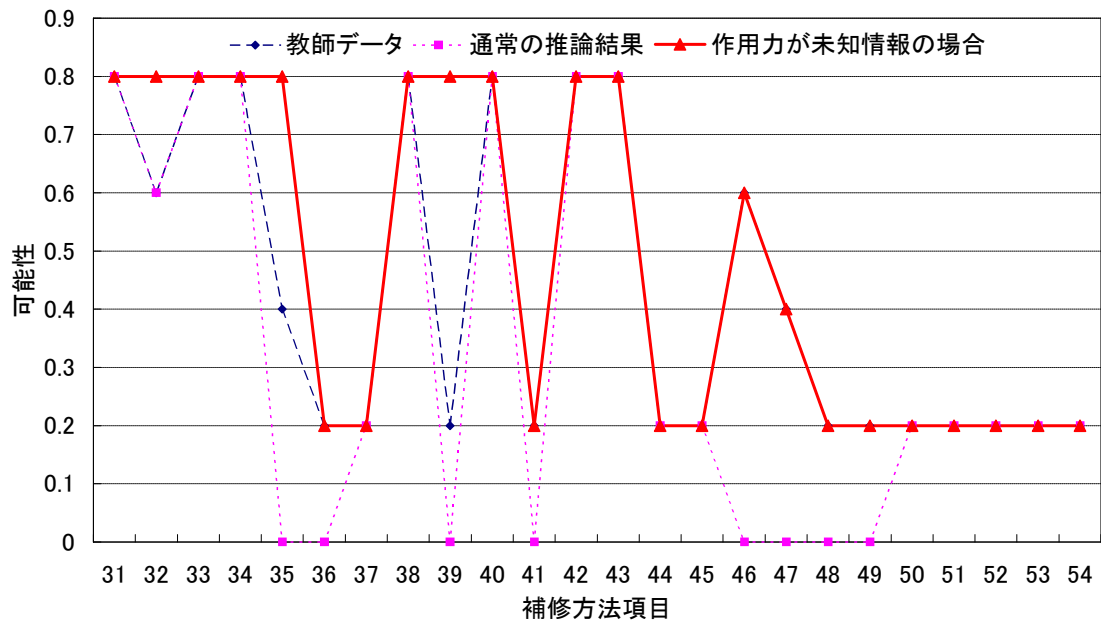


図 3.8a 作用力が未知情報の場合の補修方法選定結果(事例 1)

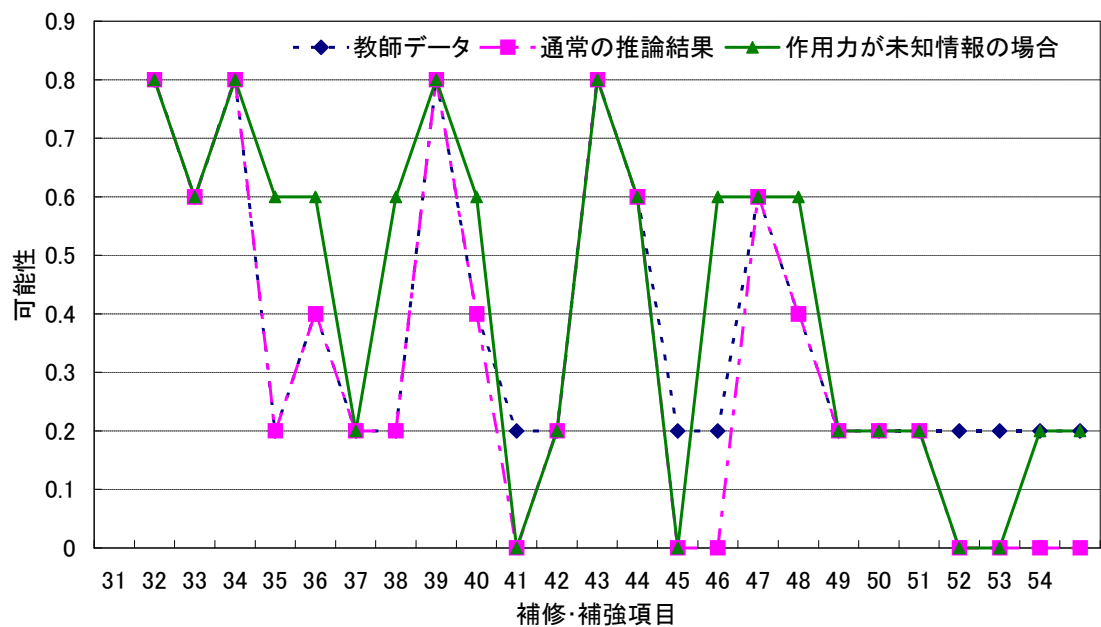


図 3.8b 作用力が未知情報の場合の補修方法選定結果(事例 2)

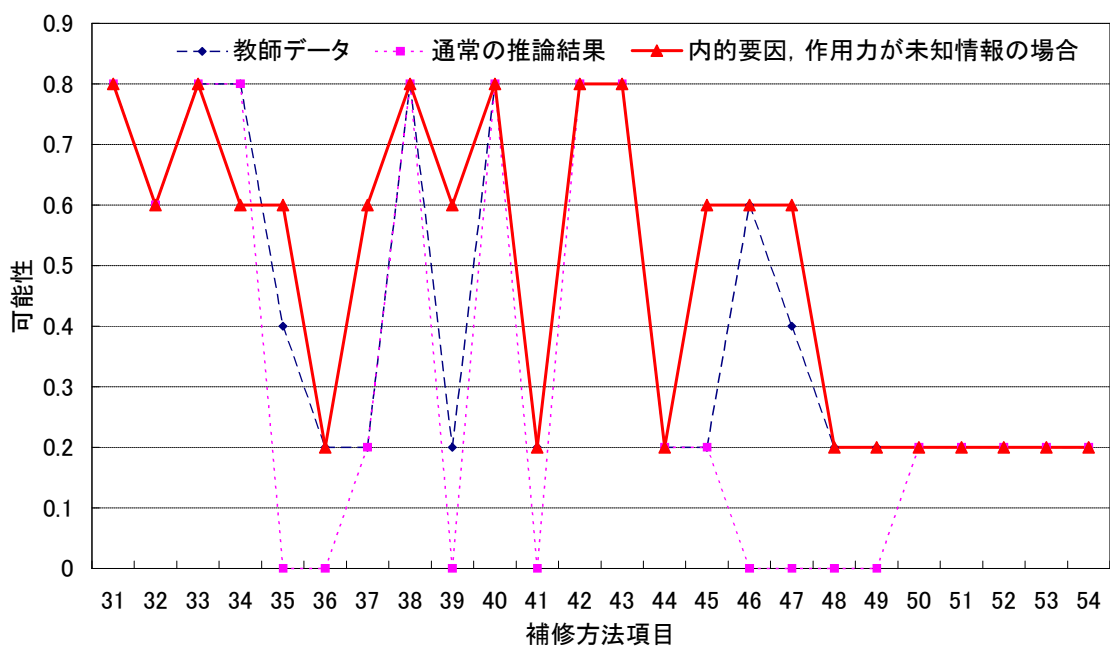


図 3.9a 内的要因，作用力が未知情報の場合の補修方法選定結果(事例 1)

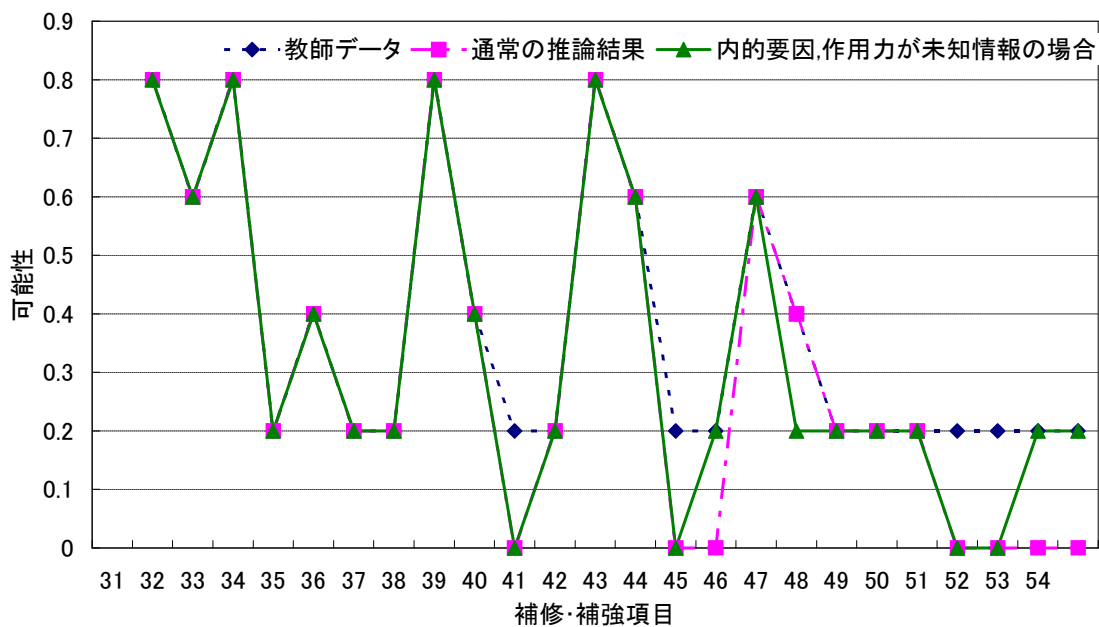


図 3.9b 内的要因，作用力が未知情報の場合の補修方法選定結果(事例 2)

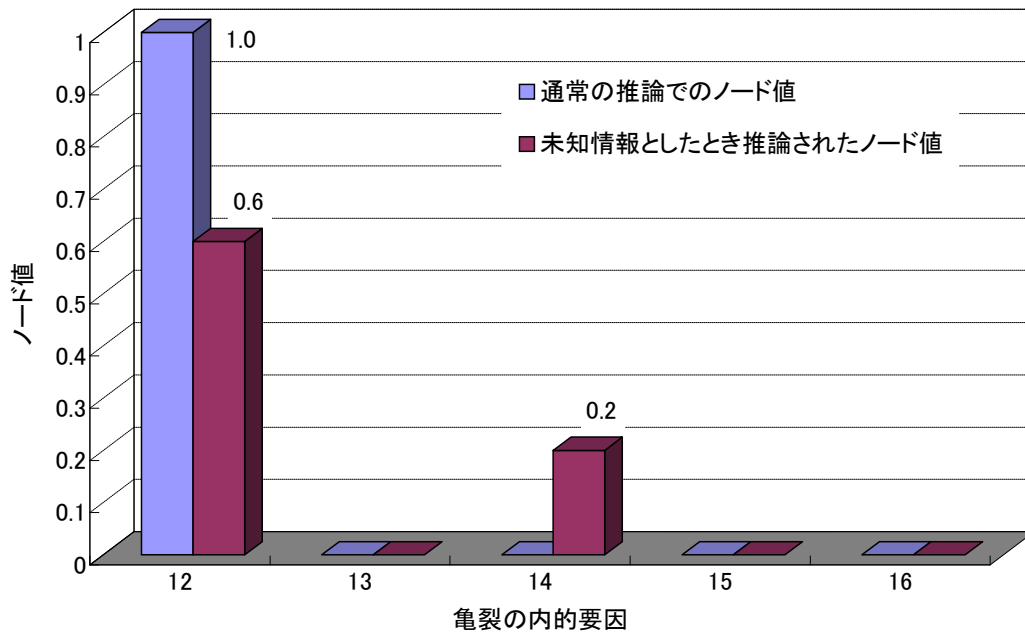


図 3.10a 推論される内的要因のノード値(事例 1)

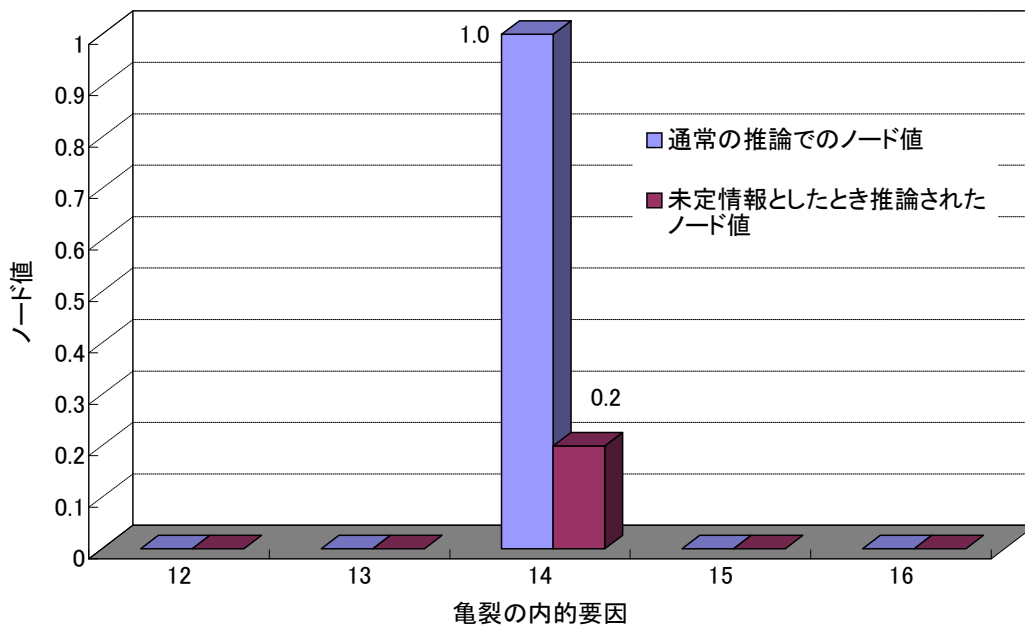


図 3.10b 推論される内的要因のノード値(事例 2)

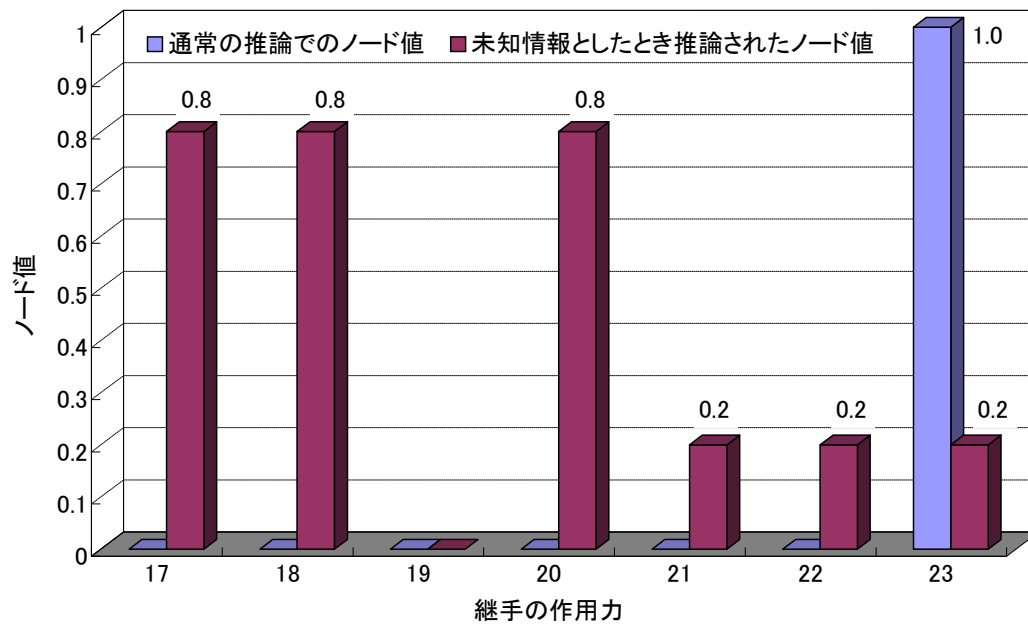


図 3.11a 推論される作用力のノード値(事例 1)

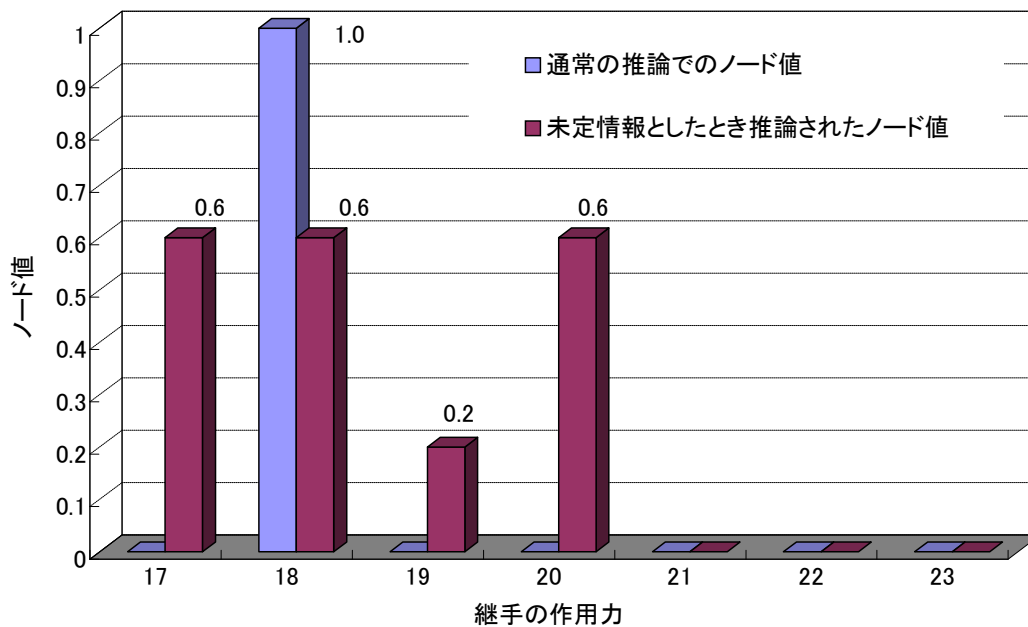


図 3.11b 推論される作用力のノード値(事例 2)

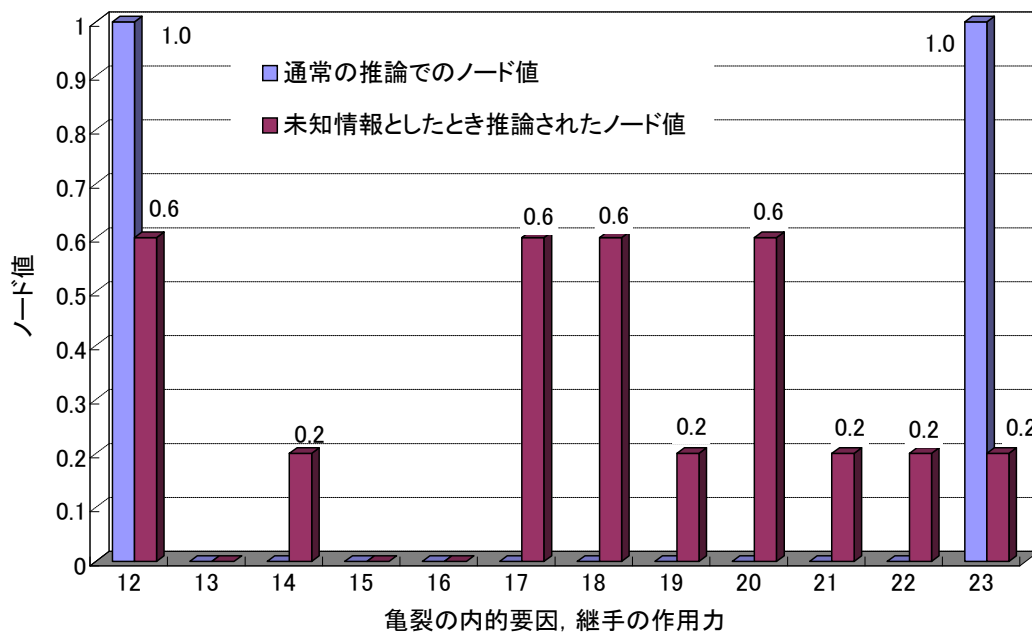


図 3.12a 推論される内的要因, 作用力のノード値(事例 1)

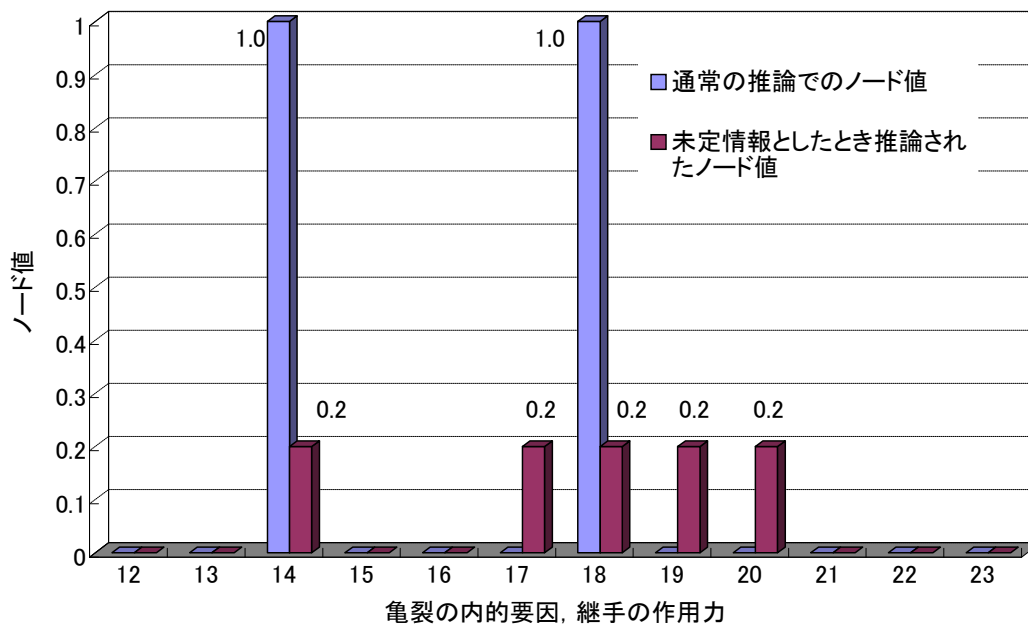


図 3.12b 推論される内的要因, 作用力のノード値(事例 2)

1. はじめに

エキスパートシステムの開発において、知識獲得問題は、解決すべき重要な課題であり、様々な研究が各方面でなされている。2章、3章においても問題解決のための一手法を述べてきたが、2、3章では、知識がある程度獲得された後の知識の追加・修正の過程に起こる、ルールベース内の整合性の取り方、信頼性の高い評価をしたルールの追加・修正、曖昧さを含む知識の信頼性の定量的評価などの問題に対する解決のためのアプローチを、既存の知識ベースの共有、再利用として述べてきた。しかし、皆川らのルール洗練機能付推論システムにおいては、知識獲得の初期における、専門知識の体系化の過程で指摘されている問題について、解決のためのアプローチは行っていない。

知識獲得の初期においては、専門家へのアンケート調査等からルールを抽出し、ルールベースを構築していることから、ルールベース構築に多大な労力を要する困難な作業であるという問題、及び経験的知識であるという性質上、示される知識が複雑であり、知識のルール化が困難であるという問題がある。これらの問題に対して、決定木及び遺伝的アルゴリズムによる知識獲得手法[文献44]、帰納学習を用いた帰納推論によるルールの獲得手法[文献46]や、事例ベース推論を用いた知識獲得手法[文献41,42,43]などによる、問題解決のためのアプローチがなされている。

こういった中、最近では、ラフ集合論による知識獲得・発見 [文献40] や、データベースからの知識発見 (KDD: Knowledge Discovery in Databases)、あるいはデータ発掘 (Data Mining) の考え方が注目を集めている。この背景には、次のような事柄がシステムを開発する上で要求されていることが考えられる。[文献26,35,36,37,45]

1. コンピュータ技術の向上により、大量かつ良質な生データを容易に蓄積することが可能になったことから、これらの膨大なデータを有効に活用できないかという実際的な要求。
2. データベース分野、知識ベース分野、統計的分野など、それぞれの分野における独立した理論を統合しようという理論的な要求。
3. 近年の成熟しつつある知識ベース分野やデータベース分野の技術を統合し、ソフトウェア化することで、大規模問題に適用させようという技術的な要求。

そこで、こういった背景をもとに4章では、ルールベース構築の支援、及びルールベースへのルールの追加を主な目的とした、知識獲得問題の解決を検討するため、データベースからの知識発見 (KDD) の考え方をを用いて、データベースからルールを自動的に生成するアルゴリズムを提案し、その有効性を検討する。

2. データベースからの知識発見 (KDD: Knowledge Discovery in Databases) [文献 27,35]

近年,コンピュータ技術の向上により,大規模なデータの蓄積・処理が容易かつ高速になったことから,データベースシステムを高機能化,あるいは高性能化するための知識処理技術の応用や,知識処理分野で有効利用可能な機能を備えたデータベースシステムの構築が各分野で行われるようになり,データベース,知識ベースの双方の分野において,お互いの技術の必要性が増してきている。

特に,大量のデータの中に内在する知識を獲得する研究は,データベース分野,知識ベース分野の両方の分野において注目されている.この種の研究は,知識ベースの分野からは,より洗練された知識ベースシステムの構築を目的とし,データベース分野では,大量のデータを 1 つの知識源として用いながら,対話的に情報検索をするシステムなどといった,データベースのアクセスに有効利用することを目的としている。

このような,大量データから知識を獲得することを“データベースからの知識発見 (KDD)”という。

そこで,以下にデータベース(システム),知識ベース(システム)の在り方と問題点をそれぞれ明確にし,データベースからの知識発見に関して検討する。

2.1 データベース

データベースとは,厳格なデータ定義と処理制御が行われ,管理がしっかりと行われている“大量・定型”の情報である.つまり,大量のデータ管理を目的とし,対象問題への応用とは独立した従来のコンピュータ技術の枠組みの中で作られたシステムといえる。

よって,データベースを利用する際には,まず対象データがあり,次にそれをどのように利用するかというように,ボトムアップ的に利用されるシステムである。

2.2 知識ベース

知識ベースとは,“問題解決にどのように有効に使えるか”に主眼を置いたシステムといえる.つまり,データベースとは対照的に,対象問題への応用に強く依存した,従来のコンピュータ技術とは異なる新たな枠組みで作られたシステムといえる.そのため,データベースのように知識ベースを独立なものとして扱うことは考えにくく,ある目的での汎用のものとして,幾つかの応用システムのもとで共有利用するといった,知識共有,知識再利用というものが非常に困難である。

また,データベースがボトムアップ的に利用されているのに対し,知識ベースや人工知能は,はじめに目的があって,次にそれを達成するために有効に使えると思われるものをトップダウン的に構築して利用されている。

2.3 データベースと知識ベースの融合[文献 35]

データベースからの知識発見(KDD)を確立するには,ボトムアップ的に利用されているデータベースとトップダウン的に利用される知識ベースの2つの対照的に利用されている技術を,例えば,あるところまでは,データベース中のデータをもとにボトムアップ的に知識を獲得し,次に,それまでに得られた知識をもとに,今度はトップダウン的操作により,有用な知識を獲得するというように,1つの連続的なつながりを持った技術に融合する必要がある.

一般に,データベースシステム利用において,ユーザは単純に検索結果を求めるだけでなく,仮説に従って検索を繰り返し,検索結果に応じて仮説を修正しながら何らかの知識を発見するまで検索を繰り返す.そのため,知識ベース分野から,より柔軟で高度な問い合わせ処理技術を提供することは,多様な情報の中に埋もれてしまいがちな知識を,大量データベースから発見する場合において非常に重要となってくる.よって,単純な検索を行うデータベースシステムだけでなく,その検索結果を利用して有効な知識を自動的に生成する知識ベースシステムの融合は非常に重要となってくる.

2.4 データベースからの知識発見技術[文献 35,36]

ここでははじめに,データベースからの知識発見(KDD)とデータ発掘(データマイニング)についての定義付けをし,次にKDDのプロセスとデータマイニングの手法について述べる.

2.4.1 KDDとデータマイニングの定義[文献 36]

KDDとは,妥当性,新規性,潜在的有用性,理解可能性のあるパターンをデータから同定するための一連のプロセスのことをいい,また,データマイニングとは,KDDプロセスの中の1ステップであり,データ分析とデータ発見のアルゴリズムを適用して,データ中のパターンを数え上げ,抽出することをいう.

2.4.2 KDDのプロセス

図4.1に一般的な,データベースを用いたKDDプロセスを示す.以下で,KDDプロセスの内容について述べる.

データ獲得: 応用分野の性質を理解し,事前知識と必要なデータ,KDDの同定を行う.

データ選択: 与えられたデータ集合からKDDに必要な目標データセットを選択する.

前処理: 目標データから,ノイズや異常値を除去し,データをクリーンなものにする

ために必要な前処理を行う.(連続データの離散化など)

データ変換: 前処理の終わったデータをデータマイニングのアルゴリズムが適用でき

るようにデータを変換する。

データマイニング：与えられた表現の中から、パターンを抽出し、分類ルール、関連ルール、決定木、クラスターなどを得る。

データ解釈と評価：抽出したパターンを解釈・評価して知識を得る。このステップで十分な知識が得られなかったら、繰り返しこれらのステップが行われる。

このように、KDD ツールは、従来からの機械学習などのシステムと比較しても、データマイニングの部分で用いられる手法が特徴的であること以外は、従来から行われてきた手法と同等のステップが踏まれている。そのため、本来ならば、すべてのステップを対話的なサポートによるツールにすることが望まれるが、以下にあげるような理由から、その実現に至っていない。

1. データマイニングの手法・アルゴリズムが多岐にわたっていること。
2. 大規模なデータベースの情報を効率的に処理することが困難であること。
3. KDD プロセス自体の研究が十分確立していない。

1 に関しては、次のデータマイニングの手法で詳しく述べることにする。

2 に関しては、本来、知識発見の対象となるデータベースは、知識発見を目的として構築されたものではないため、データマイニングに必要な全属性が必ずしも格納されているとは限らず、データマイニングを可能にするため、データの前処理やデータ変換などにより、データベース中に含まれるノイズを除去する必要があることから、これらを自動化するのは困難とされている。

3 に関しては、KDD 自体が近年の新しい研究分野であり、現状では、各方面で独立して研究・開発がなされている状況といえる。

2.4.3 データマイニングの手法

KDD プロセスは、その目的に応じて、大きく仮説検証型の KDD と仮説生成型の KDD とに分類することができ、データマイニングの手法や、データの解釈方法が異なっている。

データマイニングに必要な技術は、データベース技術に始まり、高性能計算技術、データ視覚化技術、統計的手法、人工知能の機械学習技術、推論技術など広範囲に及んでおり、これらを用いたデータマイニングの手法もまた数多く存在する。

仮説検証型の KDD は、利用者の持つ仮説を、与えられたデータを用いて検証することを目的としていることから、データマイニングの手法としては、データベース質問・操作手法、多変量解析手法、統計的検証手法などがあげられる。しかし、どちらかというと、従来から標準的に用いられているものが多く、データベース技術に始まり、人工知能の機械学習技術、推論技術などの、特徴的なデータマイニング技術を存分に活用しているとはいえない。

これに対し、仮説生成型の KDD は、データの操作により、データをより単純な表現形式に変換することを目的としていることから、このデータマイニング手法は、相関ルールの発見、分類ルールの発見、クラスタリングやデータ集約など、データマイニング特有の手法が幾つも見られる。

3. 提案するルール抽出アルゴリズム [文献 36]

本研究では、仮説生成型の KDD のデータマイニング手法である相関ルールの発見に注目し、過去の事例から有効なルールを抽出し、抽出したルールを定量的評価基準で評価して、ルールベースの構築、あるいはルールの追加を行うことを目的としている。

そこで、情報理論的な方法により平均圧縮情報量を用いて、ルールを定量的に評価する評価方法を提案する。

3.1 平均圧縮情報量 (Average Compressed Entropies) [文献 37]

データベース中には、有用性の有無に関わらず、極めて多くの知識が存在する。この中から有用である知識、つまり興味深い知識を見分けることは非常に困難であり、これを見分けるには、対象領域の背景知識や、対象領域固有の規準に基づいた興味深さの評価を行うのが通常である。これは、例えば 2 章、3 章で述べたルールベース洗練機能付推論システムでのルールの結合係数や確信度の付与の方法が、専門家による主観性に依存して行われていることが挙げられる。

よって、抽出した知識、あるいはルールをこういった背景知識によらず、客観的、定量的に評価する必要がある。1 節で述べたような知識獲得問題解決のために大きな意義があると考えられる。その理由としては、現在の専門家へのアンケートやインタビューによるルール抽出では、得られた知識が基本的に主観性を多く含む知識であることが挙げられる。また、アンケート調査などでは、アンケートの内容が複雑、大量になった時、回答をする専門家が適当な解を示すのではないかというヒューマンエラーや、アンケート内容自体が、対象領域全般を網羅し得ていないという問題も挙げられる。

これに対し、情報理論的に、データの持つ情報量を求めてルールを抽出することは、現存する事実の集合であるデータベースから客観的、定量的にルールを抽出し、それを基にルールベースを構築することができるため、プロダクションルールを適用したエキスパートシステムにおいて最も重要な機能であるルールベースに対するエラーを軽減することが可能であると考えられる。また、専門家にアンケート調査等を行ってルールベースを構築するよりも、データマイニング等による定量的評価によって構築したルールを専門家に提示して、検証してもらうほうが、専門家への労力軽減につながり、それにより、エキスパートシステム開発のボトルネックである知識獲得問題の解決につながると考えられる。

そこで、ルールを抽出することによって変化するデータベース内のデータが持つ情報量を用いて、定量的にルールの抽出を行い、抽出したルールの興味深さを評価することとした。このルールによって変化する情報量を平均圧縮情報量と呼ぶ。

3.2 データベースにおける事例とルール

データベース中に存在する各事例を e_i とし、事例の総数を N とする。つまり、データベース中には、以下のように N 個の事例が存在する。

$$\{ N: e_1, e_2, \dots, e_i, \dots, e_N \}$$

また、事例 e_i は、 m 個の要素 a_{im} によって構成されている。よって事例 e_i は、次のように表すことができる。

$$\{ e_i: a_{i1}, a_{i2}, \dots, a_{i1}, a_{im} \}$$

データベース内から、ルールを生成する際に、条件部になり得る要素 a_{i1} を y 、結論部となり得る要素 a_{im} を x とし、条件部と結論部になる要素を区別して表すことにする。なお、各要素は、条件部の要素にも結論部の要素にもなり得るものとする。

次に、データベース内の各事例 e_i において、条件部の要素 y を持つ事例の数を n_y とし、結論部の要素 x を持つ事例の数を n_x とする。また、条件部の要素 y と結論部の要素 x を共に持つ、すなわち、ルール $y \rightarrow x$ を抽出することのできる事例の数を $n_{y,x}$ とする。

3.3 平均圧縮情報量による評価アルゴリズム

はじめに、結論部の要素 x の生起確率 $p(x)$ を式(1)より求め、結論部の要素が x でない場合、 \bar{x} の生起確率 $p(\bar{x})$ を式(2)より求める。

$$p(x) = \frac{n_x}{N} \quad (1)$$

$$p(\bar{x}) = 1 - \frac{n_x}{N} \quad (2)$$

次に、結論部の要素 x と条件部の要素 y の結合確率 $p(x, y)$ を式(3)より求め、また、結論部の要素が \bar{x} で、条件部の要素が y の結合確率 $p(\bar{x}, y)$ を式(4)より求める。

$$p(x, y) = \frac{n_{y,x}}{N} \quad (3)$$

$$p(\bar{x} | y) = 1 - \frac{n_{y,x}}{N} \quad (4)$$

ここで、結合確率は、データベース内の事例 N 個のうち、ある事例 e_i について、条件部の要素 x と結論部の要素 y を同時に持つ事例の割合を示す。

また、条件部の要素 y が起こった時に結論部の要素 x が起こる結論部の要素 x についての条件付確率 $p(x|y)$ を式(5)より求め、条件部の要素 y が起こった時に、結論部が \bar{x} である条件付確率 $p(\bar{x}, y)$ を式(6)より求める。

$$p(x | y) = \frac{n_{y,x}}{n_y} \quad (5)$$

$$p(\bar{x}, y) = 1 - \frac{n_{y,x}}{n_y} \quad (6)$$

情報理論的には、通常、結論部の要素 x がデータベース内で持つ情報量を、 $-\log\{p(x)\}$ で示す。また、この時、結論部の要素 x を事例内の要素 $x = a_{im}$ として持つ事例が、同時に条件部の要素 y を $y = a_{il}$ として持つ場合があった時、 $y \rightarrow x$ というルールが事例中に内在していることになる。この要素 y と要素 x を同時に持つ時の結論部の要素 x がデータベース内で持つ情報量は、 $-\log\{p(x|y)\}$ で示すことができる。この2つの情報量により、事例中に内在する $y \rightarrow x$ というルールの情報量は、条件部の要素 y が起こる前の情報量 $-\log\{p(x)\}$ と、条件部の要素 y が起こった後の情報量 $-\log\{p(x|y)\}$ との差によって表すことができる。つまり、ルール $y \rightarrow x$ を生成することにより、結論部の要素 x についての情報量を圧縮したことになる。

また、式(3)に示したように、ルール $y \rightarrow x$ を生成することが可能な事例は、データベ

ース内に $N \cdot p(x, y)$ 個存在していることから、データベース内において結論部の要素 x が、条件部の要素 y によって圧縮される情報量は、 $N \cdot p(x, y)$ と各情報量との積により表すことができる。

以上に示したことは、同様に結論部の要素が x でない場合、 \bar{x} についてもいえる。

この $N \cdot p(x, y)$ あるいは $N \cdot p(\bar{x}, y)$ と、情報量の積を圧縮情報量 (Compressed Entropies : CE) と呼び、式(7)により $CE(x, y)$ で表す。

$$CE(x, y) = [-Np(x, y) \log\{p(x)\} - Np(\bar{x}, y) \log\{p(\bar{x})\}] \\ - [-Np(x, y) \log\{p(x|y)\} - Np(\bar{x}, y) \log\{p(\bar{x}|y)\}] \quad (7)$$

この圧縮された情報量を評価値とするため、データベース内の事例の総数 N により平均値をとって、評価値となる平均圧縮情報量 $ACE(x, y)$ を式(8)より求める。

$$ACE(x, y) = CE(x, y) / N \\ = \{[-Np(x, y) \log p(x) - Np(\bar{x}, y) \log p(\bar{x})] \\ - [Np(x, y) \log p(x|y) - Np(\bar{x}, y) \log p(\bar{x}|y)]\} / N \\ = \{-p(x, y) \log p(x) - p(\bar{x}, y) \log p(\bar{x})\} \\ - \{-p(x, y) \log p(x|y) - p(\bar{x}, y) \log p(\bar{x}|y)\} \quad (8) \\ = p(x, y) \log \frac{p(x|y)}{p(x)} + p(\bar{x}, y) \log \frac{p(\bar{x}, y)}{p(\bar{x})}$$

4 適用結果及び考察

4.3 節の平均圧縮情報量に基づく知識発見の適用例として、3 章と同様、鋼橋疲労損傷の補修方法選定問題を選び、既存の推論システムで得られる事例群を生データのデータベースと見なし、ここからルールを抽出した後、それをルールベースとする推論を実施して、抽出されたルールの有効性を検討する。

4.1 データベースの作成

ここでは、実際に式(8)を用いることで、データベースから有効な知識を抽出することが可能であるかを検討するため、3 章で適用した、田中らによる鋼橋疲労損傷の補修方法選定システムのルールベース[文献 34]を基に、総数 2695 の仮想事例を作成し、これをデータベースとした。

仮想事例の作成は、表 3.1 の亀裂の内的要因、外的要因、継手の作用力、亀裂様式の 30 個の入力情報項目について、起こり得る可能性のある組み合わせ、2695 通りを仮に発生した事実とし、これを仮想事実とした。この仮想事実に対して、3 章で用いたルールベース洗練機能付推論システムより、補修方法を選定し、仮想事実と組み合わせで仮想事例とした。

4.2 データベースからのルール抽出

平均圧縮情報量である式(8)を用いたルール抽出アルゴリズムを 3.4.1 のデータベースに適用してルール抽出を試みた。この結果、620 のルールがデータベースから抽出された。図 4.2 には、横軸に抽出された各ルールをとり、縦軸に評価値である平均圧縮情報量(ACE 値)をとって示す。

この抽出されたルールを基に新たなルールベースを構築し、鋼橋疲労損傷の補修方法の選定を行って、抽出されたルールの有効性を検討した。

抽出されたルールのうち、ACE 値の高いルールから順に、次に示す 3 通りの規準で結合係数を付与し、それぞれ新たなルールベースを構築した。

- a) 全ルールを 5 分割し、上位 40%を 0.7、次の 40%を 0.4 とした場合 (表 4.2)
- b) 全ルールを 5 分割し、上位から 20%毎に 0.8, 0.6, 0.4, 0.2 とした場合 (表 4.3)
- c) 全ルールを 10 分割し、10%毎に、0.8 から 0.1 までとした場合 (表 4.4)

なお、3 通り全てにおいて、下位 20%のルールについては切り捨てとした。また、表中の番号 1~30 は、表 3.1 の入力情報項目の番号に対応しており、31~54 は、表 3.2 の補修方法項目の番号に対応している。

4.3 補修方法選定結果

はじめに、3章の鋼橋疲労損傷の補修方法選定を対象に、新たに構築したルールベースを適用して、補修方法の選定を行った。これにより、抽出されたルールを用いて構築したルールベースと、3章で適用したルールベースをそれぞれ適用した場合での補修方法選定の結果を比較することで、抽出されたルールの妥当性を検討することができる。

推論に適用した事例は、表 3.3 で提示した事例 1, 2 である。

図 4.3, 図 4.4 に、a) の場合における補修方法選定結果をそれぞれ事例 1, 事例 2 について、横軸に補修方法項目の番号をとり、縦軸に各項目の可能性であるノード値をとって示す。比較のため、田中らによるルールベースで推論した推論結果、及び教師データを併せて示す。以下、図 4.5, 図 4.6 には b) の場合の結果を示し、図 4.7, 図 4.8 には c) の場合の結果を示す。

この結果、a) の場合、図 4.2 のように、ACE 値による評価によって幅広く分布するルールを 0.7 と 0.4 の 2 段階で分類したため、ノード値による可能性の分類が十分になされていないことがわかる。これに対し、b), c) の場合は、それぞれ 5 段階、10 段階に分類したことから、共に十分な可能性の分類を行って推論結果を提示している。図 4.5, 図 4.7 の事例 1 の結果では、教師データとの間に大きな違いが見られる。特に、補修方法項目の番号 51, 54 で、田中らのルールベースでの推論結果、及び教師データ共にノード値 0.4, 0.2 と低い可能性を示しているのに対し、ノード値 0.8 と高い可能性を示しているなど、ノード値が 0.8 を示す項目に関しての違いが目立つ。しかし、このことは、田中らによるルールベースを用いた場合の推論結果についても同様である。また、図 4.6, 図 4.8 の事例 2 の結果では、ノード値 0.8 を示す補修方法項目は、ほぼ教師データと一致しており、b) の場合で比較すると、田中らのルールベースによる推論結果よりも精度の良い結果を示している。また、推論結果全体の傾向としても、可能性の値に違いは見られるが、田中らによるルールベースでの推論結果、及び教師データの傾向と同様な傾向を示している。

しかし、c) の場合、結合係数を細分化したことにより、ノード値 0.8 を示す可能性の高い項目の提示が、特に事例 1 については 3 項目と少数であることから、本来システムに要求されている技術者支援システムとしての可能性の分類の観点からすると、利用者が提示された解を参照する以前に、過度に可能性が分類される恐れがある。

4.4 抽出されたルールと既存ルールの比較

ここでは、4.3において事例1と事例2の間です異論結果にさが生じていることから、この原因を検証するため、抽出したルールが事例1,2の入力情報と補修方法との間の因果関係に及ぼした影響について検討する。

事例1については図4.9から図4.12に、事例2については図4.13から図4.16に、それぞれ入力情報項目と補修方法項目との結合係数を、横軸に補修方法項目（番号34-54）をとり、縦軸に各項目の結合係数をとった図を示す。なお、図4.9から図4.16は、それぞれ次に示す入力情報についての補修方法との結合関係を示した図である。

図4.9：事例1の亀裂の外的要因 [活荷重の作用（入力情報項目2）]

図4.10：事例1の亀裂の内的要因 [応力集中（入力情報項目12）]

図4.11：事例1の継手の作用力 [重ね継手の作用力②（入力情報項目23）]

図4.12：事例1の亀裂様式 [重ね継手の亀裂様式（i）（入力情報項目29）]

図4.13：事例2の亀裂の外的要因 [輸送架設荷重（入力情報項目11）]

図4.14：事例2の亀裂の内的要因 [2次応力（入力情報項目14）]

図4.15：事例2の継手の作用力 [T継手の作用力③（入力情報項目18）]

図4.16：事例2の亀裂様式 [T継手の亀裂様式d（入力情報項目27）]

この結果、事例1では、図4.5,図4.7において補修方法項目51,54がノード値0.8を示しているのは、図4.12の亀裂様式と補修方法との関係で、結合係数が、3章のルールベースでは補修方法項目51に弱い因果関係を持たせているだけであるのに対し、抽出したルールでは、51,54共に強い因果関係を持っていることが示されている。このように、事例1では、3章でのルールベースでは弱い因果関係であったり、全く因果関係を持たないルールであるのに対し、ルール抽出アルゴリズムでは強い因果関係を持ってルールを抽出していることが、図4.11や図4.12の他の項目に関しても示されており、この結果、事例1の推論では、3章での推論結果、及び教師データの値との間で違いが生じたと考えられる。

事例2では、3章でのルールベースで強い因果関係を持つルールのほとんどに関して、同様に強い因果関係を持ってルールを抽出していることがわかる。そのため、事例2の推論では、低い可能性のノード値しか示していない補修方法項目番号36,37において、3章のルールベースでは、関係は弱い因果関係を持っているルールに対し、ルール抽出アルゴリズムではルールを抽出していない程度であり、推論全体の傾向も3章での推論結果、及び教師データと同様の傾向を示していることから、抽出したルール、ACE値による及びルールの評価は十分有効に機能していると考えられる。

5. おわりに

4 章では、知識獲得の初期における知識獲得問題解決のため、データベースからの知識発見 (KDD) の考え方をを用いて、平均圧縮情報量 (ACE 値) によるデータマイニングの手法を提案し、これによるルール抽出を試みた。また、抽出したルールの有用性を検討するため、抽出したルールを用いて鋼橋疲労損傷の補修方法選定のためのルールベースを構築し、実際の事例を適用して補修方法選定を試みた。

この結果、推論結果は、3 章において適用した田中らによるルールベースでの推論結果と同等の結果を示していることから、データマイニングによるルール抽出の手法に ACE 値を適用したことは十分有効であると考えられる。

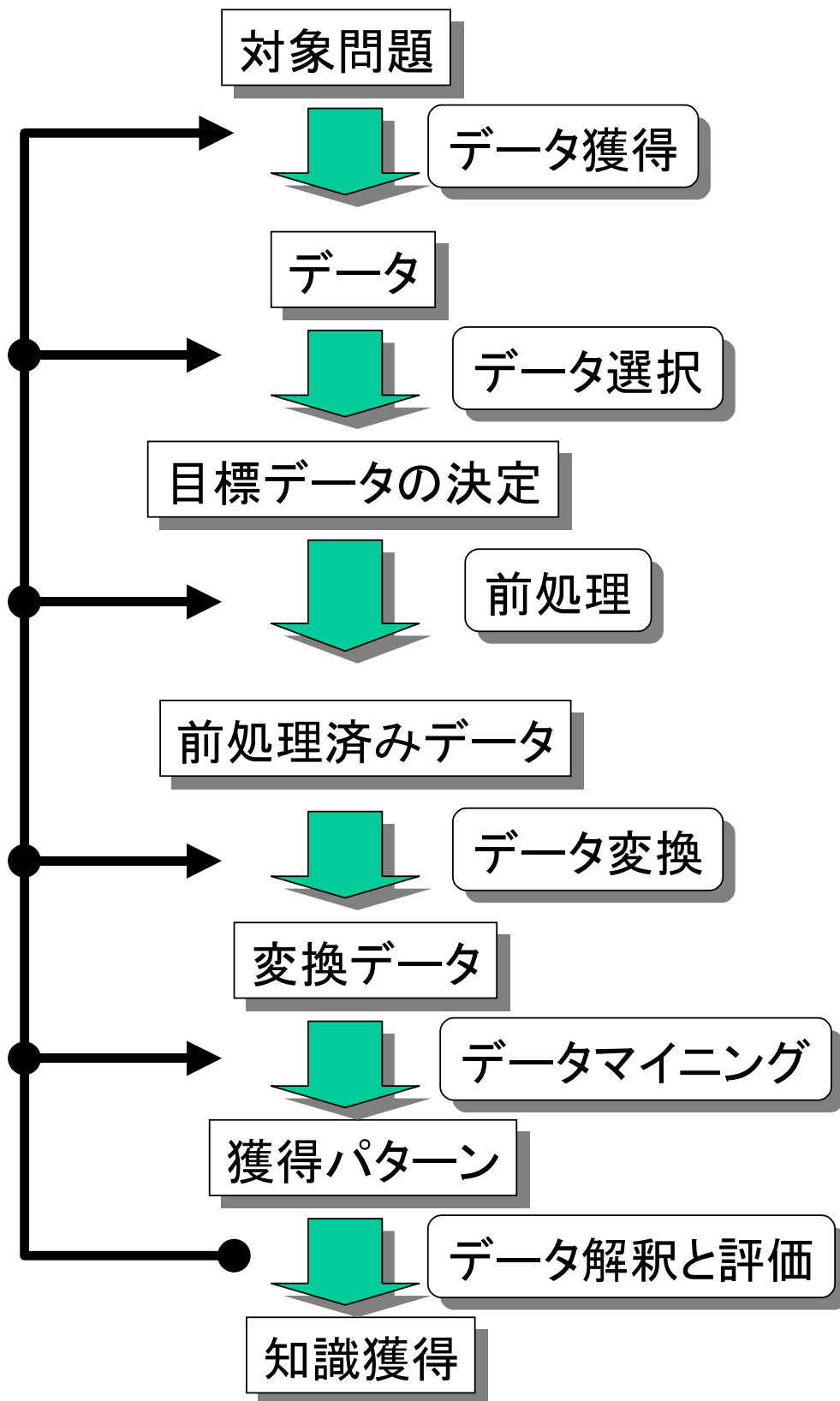


図 4.1 知識発見プロセスの構成 [文献 36]

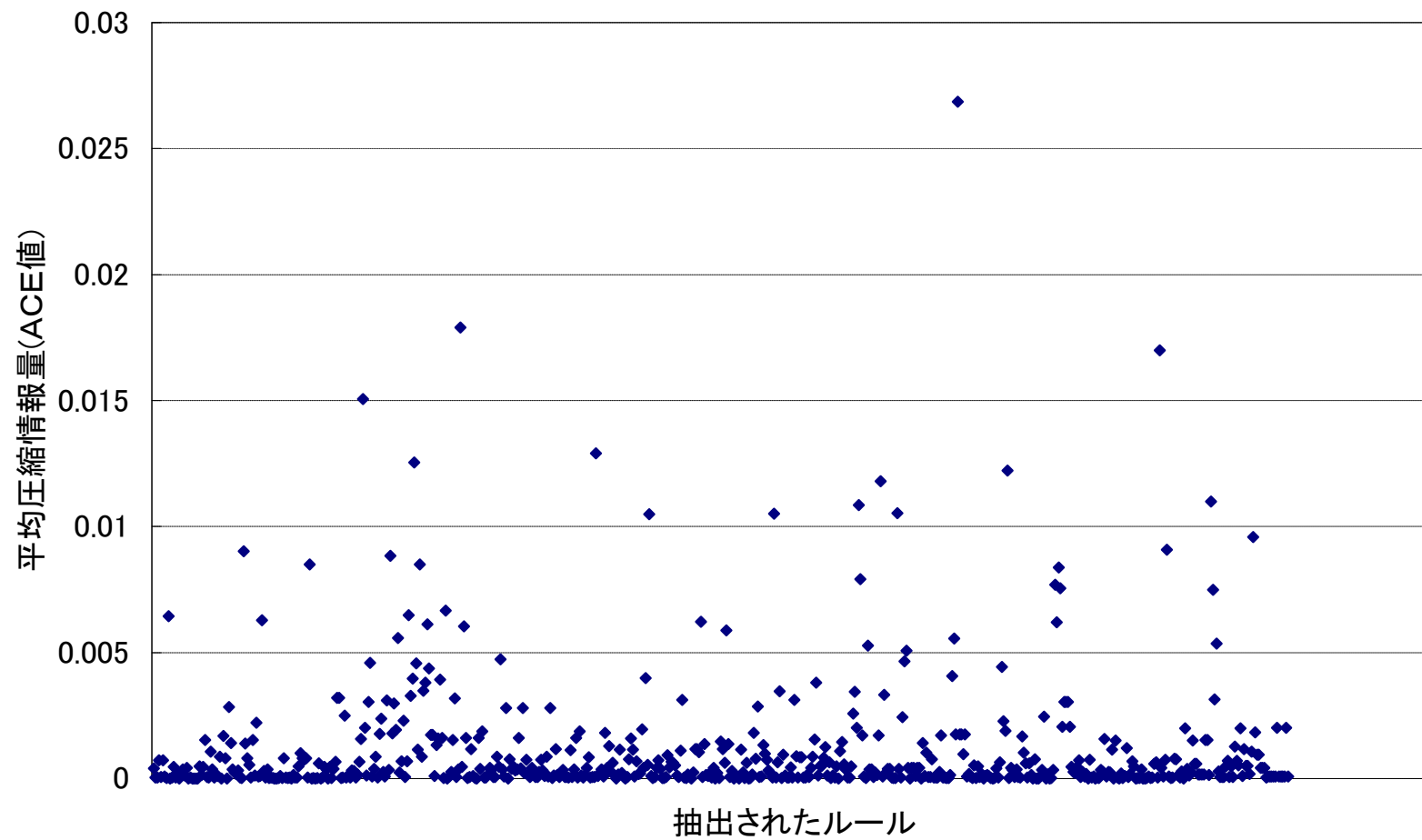


図 4.2 抽出されたルール

表 4.1 3章で適用したルールベース

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1								0.4							0.6									
2	0.8		0.8	0.4				0.8		0.4		0.8	0.8								0.4			
3																								
4	0.4						0.4							0.4										
5	0.8	0.8			0.4			0.8				0.8				0.2	0.8					0.4		
6	0.8	0.8			0.8	0.4		0.8	0.8			0.8	0.8			0.4	0.8	0.4		0.4				
7	0.6											0.4												
8		0.2						0.2		0.4	0.4													
9	0.8	0.4	0.8	0.8			0.6	0.8		0.4		0.8	0.8	0.4							0.4			
10																						0.4	0.4	
11	0.8		0.8									0.8												
12		0.8	0.8	0.8			0.6	0.8		0.4		0.8	0.8	0.4	0.6					0.4	0.4	0.4	0.4	0.4
13	0.8	0.8	0.8		0.8			0.8	0.8			0.8				0.4	0.8	0.4	0.2					
14	0.8		0.8					0.8				0.8	0.8											
15	0.8	0.4			0.4	0.4		0.8		0.4	0.8	0.8			0.2	0.2						0.4		
16	0.2					0.2				0.2	0.2	0.2												
17	0.8	0.8	0.8	0.8				0.8	0.8			0.8	0.8		0.6	0.4	0.6			0.4				
18	0.8	0.8	0.8		0.8	0.4	0.6	0.8	0.8		0.4	0.8	0.8			0.4	0.4	0.4	0.2					
19																								
20	0.8		0.8	0.4				0.8				0.8	0.6									0.4	0.4	
21	0.4	0.2						0.2				0.4	0.4											
22							0.4								0.4									
23				0.4				0.2		0.4		0.4									0.4	0.4		
24	0.8	0.6						0.6																
25	0.8		0.6					0.6	0.4			0.8	0.4											
26	0.8	0.6	0.4		0.4	0.4	0.4	0.8	0.4		0.4	0.8	0.8		0.4	0.2	0.6					0.4	0.4	
27	0.8	0.6	0.8	0.4	0.6			0.8	0.4			0.8	0.4			0.4		0.4	0.2	0.4				
28	0.8		0.6	0.4				0.6	0.4			0.4	0.6		0.4									
29				0.4				0.2		0.4		0.4									0.4			
30																						0.4		

	12	13	14	15	16	17	18	19	20	21	22	23
1	0.2					0.2						
2	0.6		0.2			0.6	0.6		0.6			0.2
3							0.2					
4	0.2					0.2					0.2	
5	0.8	0.2	0.4			0.6	0.2					0.2
6	0.8	0.4	0.8	0.8	0.2	0.6	0.4	0.2	0.4			
7		0.2	0.2			0.2	0.2					
8	0.2			0.2			0.2			0.2		0.2
9	0.2	0.8			0.2	0.4	0.6		0.2	0.2	0.2	0.2
10	0.2								0.2			
11			0.2				0.2					
12						0.8	0.8		0.8	0.2	0.2	0.2
13						0.8	0.6					
14						0.6	0.6	0.2	0.6			
15						0.2	0.2					
16									0.2			

表 4.2 a)の規準で構築したルールベース

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1	0.4	0.4			0.7			0.4	0.7			0.4			0.7									
2		0.4	0.7					0.4				0.4	0.4											
3	0.4	0.4										0.7		0.7			0.4				0.7	0.7		
4	0.4	0.4	0.7	0.4				0.4	0.4			0.7									0.7	0.7		
5		0.8	0.7	0.4	0.4			0.4	0.4								0.7							
6	0.7	0.6	0.7	0.4	0.7			0.4	0.7			0.4	0.4				0.7							
7	0.4	0.4	0.4																		0.4			
8		0.6							0.4				0.4	0.7							0.7	0.7		
9	0.7	0.6		0.7									0.7											
10			0.7	0.4				0.7				0.4	0.7										0.7	0.7
11		0.4	0.7	0.4	0.4			0.4	0.4			0.4												
12	0.4							0.7				0.7	0.7											
13	0.7	0.4	0.7					0.7				0.4	0.4											
14	0.7	0.4	0.7					0.7				0.4	0.4											
15	0.7	0.4	0.7	0.7	0.7			0.7	0.7			0.4	0.7				0.7							
16	0.4	0.6	0.7	0.7	0.7			0.7	0.7			0.7	0.7	0.7	0.7		0.7				0.7	0.7	0.7	0.7
17	0.4	0.8	0.7	0.7	0.7				0.7				0.4				0.4							
18	0.7	0.8	0.4	0.4	0.7			0.7	0.7			0.7					0.4							
19	0.7	0.4			0.7			0.4	0.7			0.4	0.4		0.4						0.7	0.4	0.4	0.4
20	0.7	0.6	0.7	0.4				0.4	0.7				0.7											
21	0.4	0.6	0.4	0.4	0.7			0.4	0.7			0.4	0.7		0.4		0.4				0.7	0.4	0.4	0.4
22	0.4	0.6	0.4	0.4	0.7			0.4	0.7				0.4	0.7	0.4		0.4				0.4	0.4	0.4	0.4
23	0.4	0.6	0.4	0.4	0.7			0.4	0.7			0.4			0.4		0.7				0.7	0.4	0.4	0.4
24	0.7	0.4	0.4	0.4	0.4	0.7		0.7	0.7			0.7	0.4				0.7							
25			0.4		0.4			0.7	0.4			0.4					0.4							
26	0.7		0.7	0.7	0.4			0.7	0.4			0.4	0.7				0.7							
27	0.7	0.6	0.7	0.4				0.4	0.7			0.7	0.4											
28			0.4	0.7	0.7			0.7	0.7			0.7	0.4				0.4							
29	0.7		0.7	0.4				0.4	0.4				0.4	0.7	0.7		0.7				0.7		0.4	0.7
30	0.4	0.4	0.4	0.4	0.7				0.4			0.7	0.7		0.7		0.7					0.7	0.7	0.4

表 4.3 b)の規準で構築したルールベース

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1	0.4	0.2			0.6			0.2	0.6			0.2			0.8									
2		0.2	0.6					0.4				0.4	0.4											
3	0.2	0.4										0.8		0.6			0.4				0.6	0.8		
4	0.2	0.2	0.6	0.4				0.2	0.2			0.6									0.8	0.6		
5		0.8	0.6	0.4	0.4			0.4	0.4								0.8							
6	0.6	0.6	0.6	0.2	0.8			0.2	0.8			0.2	0.2				0.8							
7	0.4	0.2	0.4																		0.2			
8		0.6							0.4				0.2	0.6							0.6	0.6		
9	0.6	0.6		0.8									0.6											
10			0.6	0.4				0.6				0.4	0.6										0.8	0.8
11		0.2	0.8	0.2	0.4			0.2	0.4			0.4												
12	0.4							0.6				0.8	0.8											
13	0.8	0.4	0.8					0.8				0.2	0.4											
14	0.6	0.2	0.8					0.8				0.4	0.2											
15	0.8	0.4	0.8	0.8	0.8			0.8	0.8			0.4	0.6				0.8							
16	0.2	0.6	0.8	0.8	0.8			0.8	0.8			0.6	0.8	0.6	0.8		0.8				0.8	0.8	0.8	0.8
17	0.2	0.8	0.8	0.8	0.8				0.8				0.2				0.4							
18	0.8	0.8	0.2	0.4	0.8			0.6	0.8			0.8					0.2							
19	0.6	0.2			0.8			0.4	0.8			0.2	0.2		0.4						0.6	0.4	0.2	0.2
20	0.6	0.6	0.8	0.4				0.2	0.8				0.6											
21	0.4	0.6	0.4	0.4	0.8			0.4	0.8			0.4	0.6		0.4		0.2				0.6	0.4	0.2	0.2
22	0.4	0.6	0.4	0.2	0.6			0.2	0.8				0.2	0.6	0.4		0.2				0.4	0.4	0.2	0.2
23	0.2	0.6	0.2	0.4	0.8			0.2	0.8			0.2			0.4		0.6				0.6	0.2	0.2	0.2
24	0.8	0.2	0.4	0.4	0.2	0.6		0.8	0.6			0.6	0.4				0.6							
25			0.4		0.4			0.6	0.4			0.2					0.2							
26	0.6		0.8	0.6	0.2			0.6	0.4			0.4	0.8				0.6							
27	0.8	0.6	0.8	0.2				0.2	0.6			0.6	0.4											
28			0.2	0.6	0.6			0.6	0.6			0.6	0.2				0.2							
29	0.6		0.8	0.2				0.4	0.2				0.4	0.6	0.8		0.6				0.8		0.2	0.8
30	0.4	0.2	0.4	0.2	0.6				0.4			0.6	0.6		0.8		0.6					0.8	0.8	0.2

表 4.4 c)の規準で構築したルールベース

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
1	0.4	0.1			0.6			0.1	0.6			0.2			0.8									
2		0.1	0.5					0.3				0.3	0.4											
3	0.2	0.4										0.7		0.5			0.3				0.5	0.7		
4	0.1	0.1	0.6	0.4				0.2	0.2			0.6									0.7	0.6		
5		0.8	0.6	0.4	0.3			0.3	0.4								0.8							
6	0.6	0.6	0.5	0.1	0.7			0.2	0.7			0.2	0.2				0.8							
7	0.4	0.1	0.4																		0.2			
8		0.6							0.3				0.1	0.5							0.6	0.6		
9	0.5	0.6		0.8									0.5											
10			0.5	0.4				0.5				0.4	0.5										0.8	0.8
11		0.1	0.7	0.1	0.3			0.1	0.4			0.3												
12	0.3							0.5				0.7	0.8											
13	0.7	0.3	0.8					0.8				0.2	0.4											
14	0.6	0.1	0.7					0.7				0.3	0.2											
15	0.8	0.4	0.8	0.7	0.8			0.7	0.8			0.3	0.5				0.8							
16	0.1	0.6	0.8	0.8	0.8			0.8	0.8			0.6	0.8	0.6	0.8		0.8				0.8	0.8	0.7	0.7
17	0.2	0.7	0.7	0.8	0.7				0.8				0.2				0.3							
18	0.7	0.8	0.2	0.4	0.8			0.5	0.8			0.7					0.1							
19	0.6	0.2			0.7			0.4	0.7			0.1	0.1		0.3						0.5	0.4	0.2	0.2
20	0.6	0.5	0.8	0.4				0.2	0.7				0.6											
21	0.4	0.5	0.4	0.4	0.7			0.4	0.7			0.3	0.6		0.3		0.1				0.5	0.4	0.2	0.2
22	0.3	0.6	0.4	0.2	0.6			0.2	0.7				0.2	0.6	0.3		0.1				0.3	0.4	0.2	0.2
23	0.1	0.6	0.1	0.4	0.7			0.1	0.7			0.2			0.3		0.5				0.6	0.1	0.2	0.2
24	0.8	0.2	0.3	0.4	0.2	0.6		0.7	0.6			0.6	0.3				0.5							
25			0.3		0.3			0.6	0.3			0.1					0.1							
26	0.5		0.7	0.6	0.2			0.5	0.3			0.3	0.7				0.5							
27	0.8	0.5	0.8	0.2				0.1	0.5			0.5	0.4											
28			0.2	0.6	0.5			0.5	0.5			0.5	0.2				0.1							
29	0.6		0.8	0.1				0.3	0.1				0.3	0.6	0.7		0.6				0.8		0.2	0.7
30	0.3	0.2	0.3	0.1	0.5				0.3			0.6	0.6		0.7		0.5					0.8	0.7	0.2

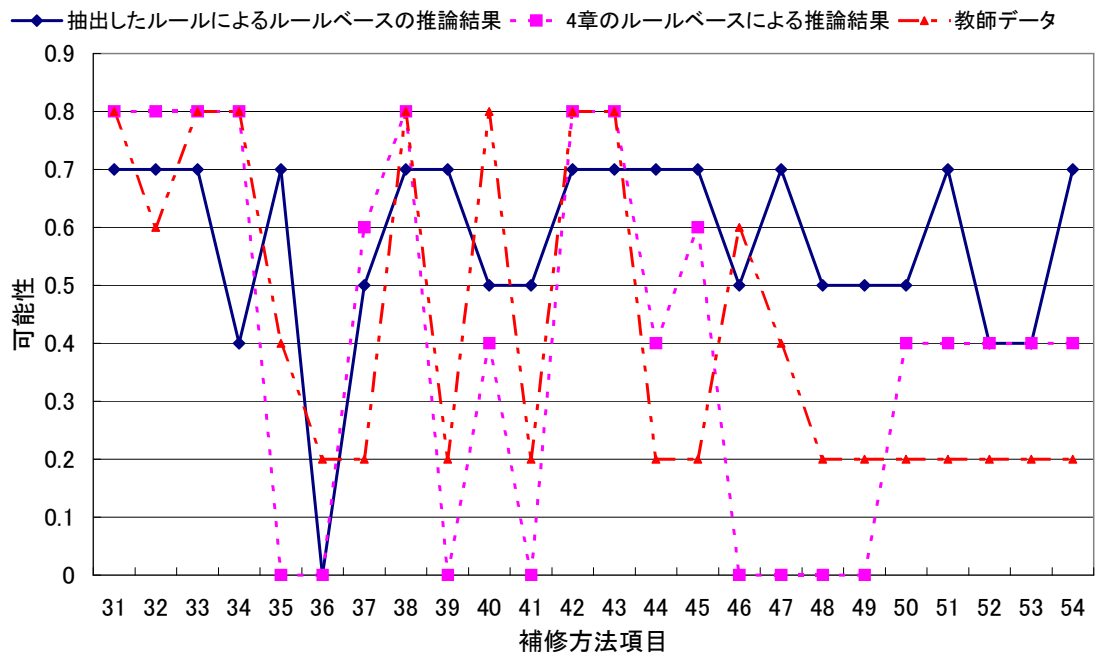


図 4.3 a)の場合の補修方法選定結果 (事例 1)

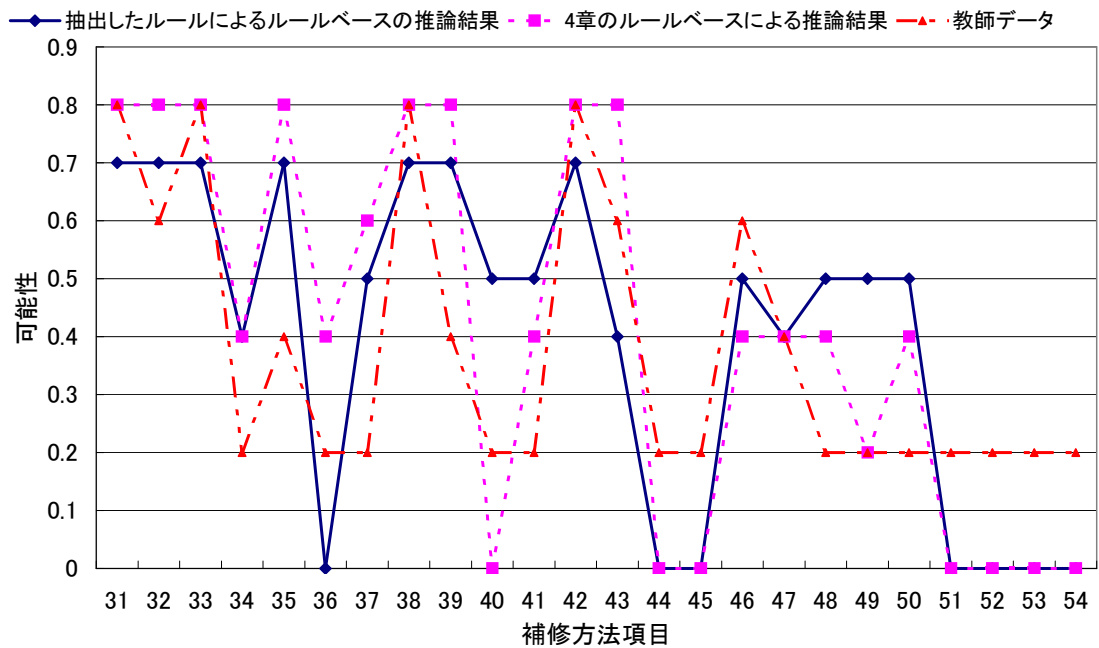


図 4.4 a)の場合の補修方法選定結果 (事例 2)

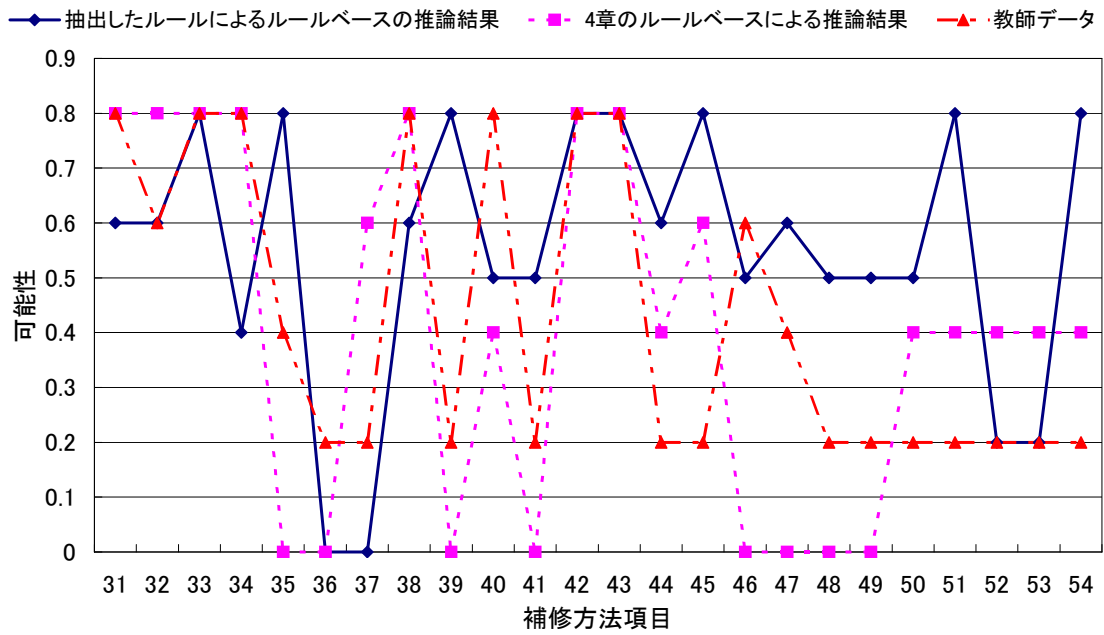


図 4.5 b)の場合の補修方法選定結果 (事例 1)

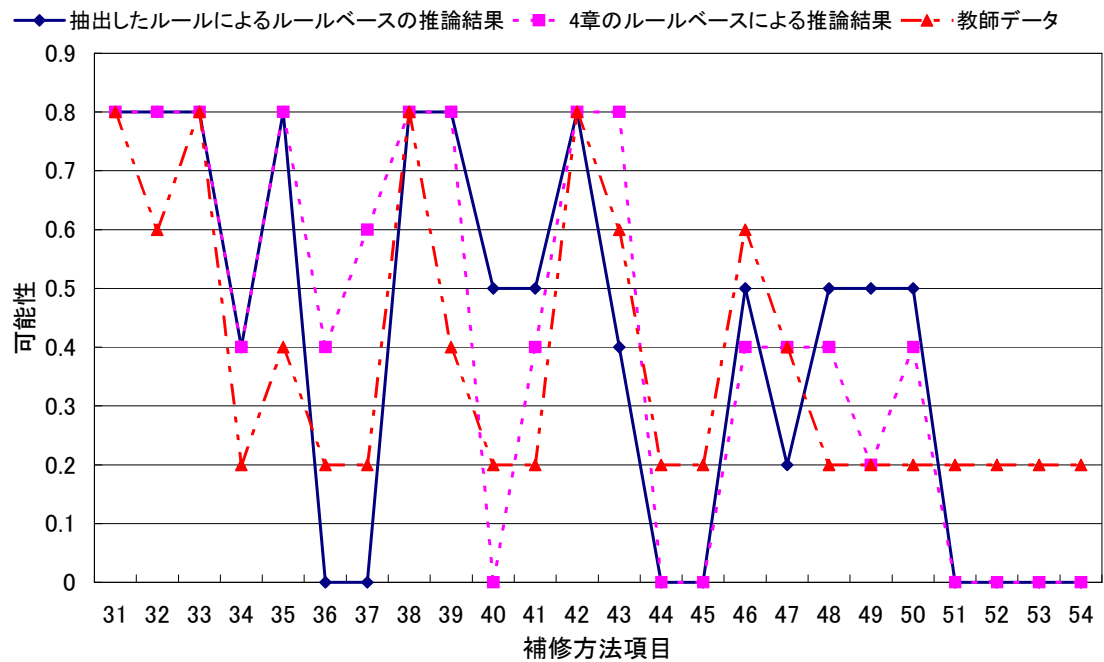


図 4.6 b)の場合の補修方法選定結果 (事例 2)

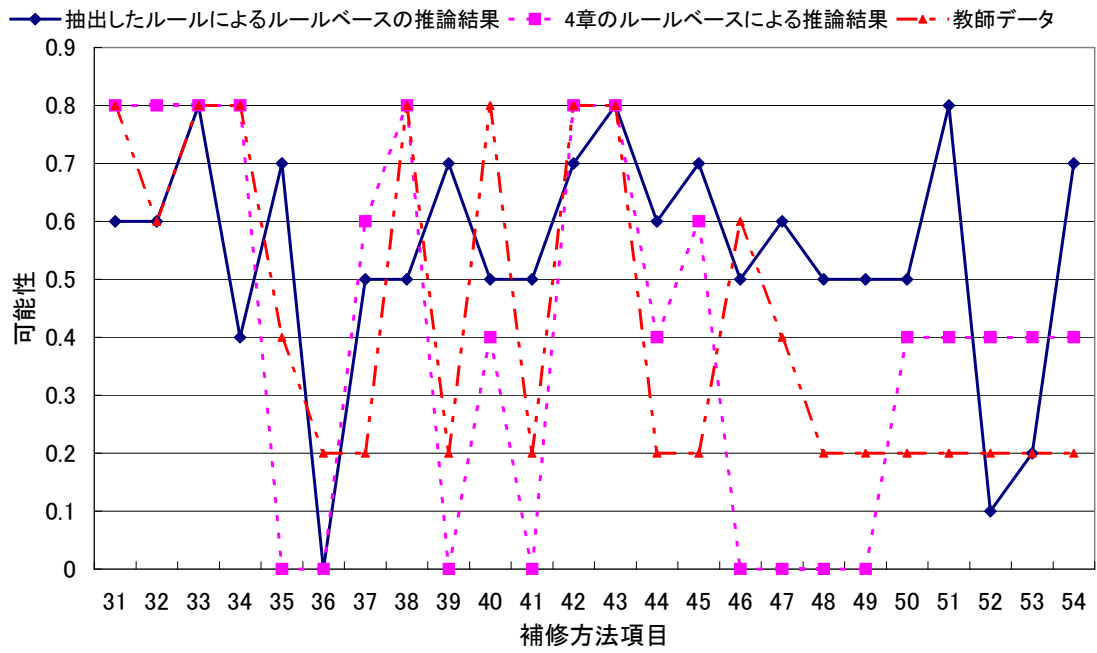


図 4.7 c)の場合の補修方法選定結果 (事例 1)

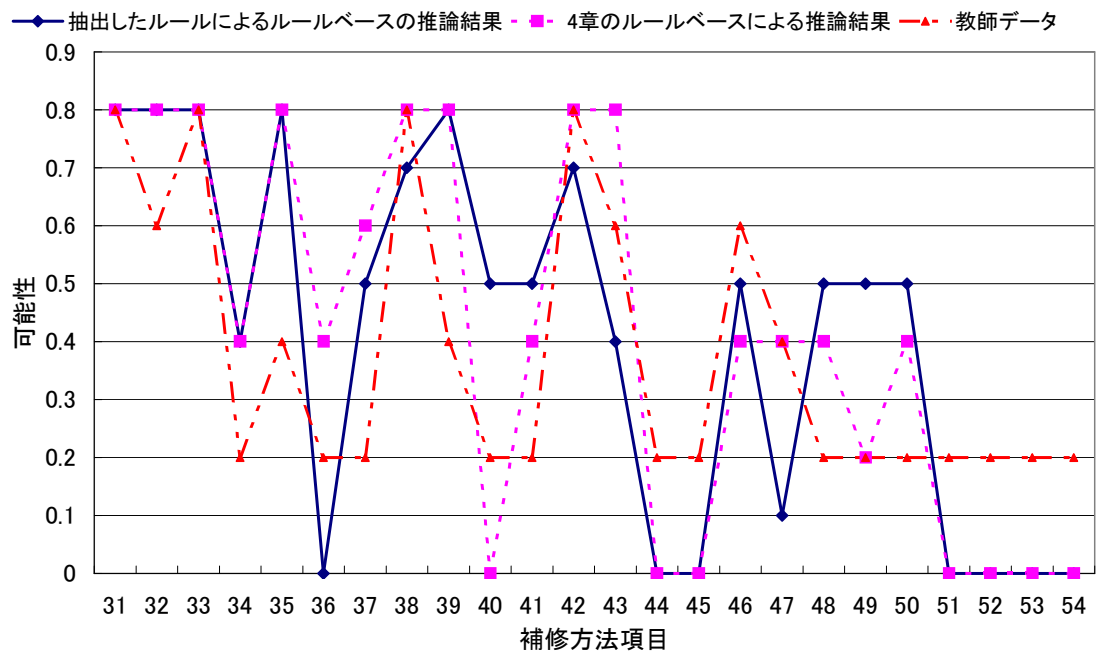


図 4.8 c)の場合の補修方法選定結果 (事例 2)

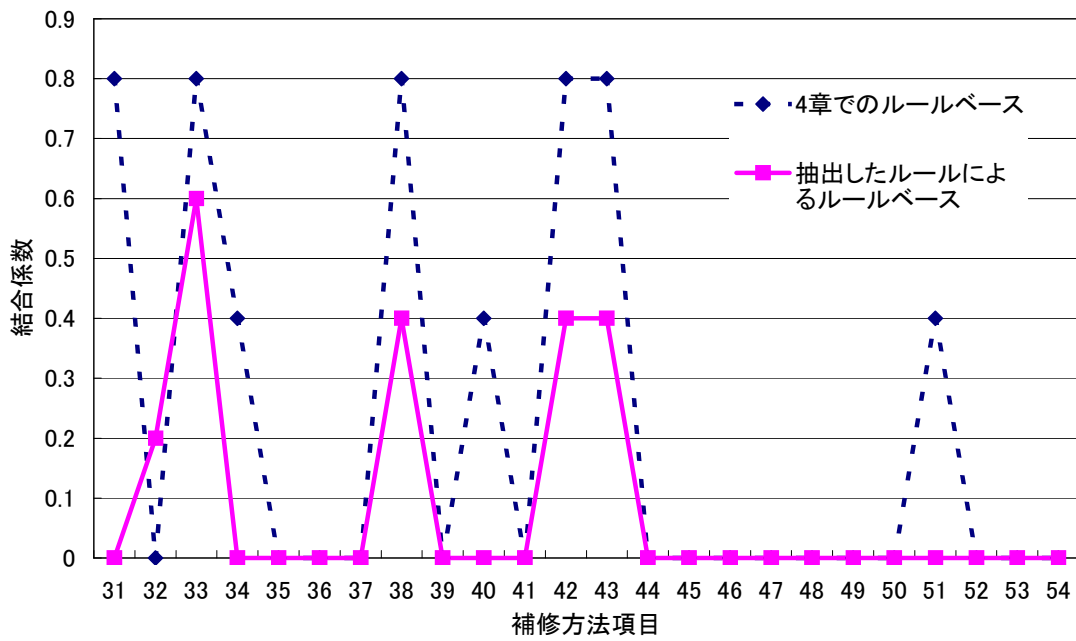


図 4.9 外的要因 (2) と補修方法の因果関係 (事例 1)

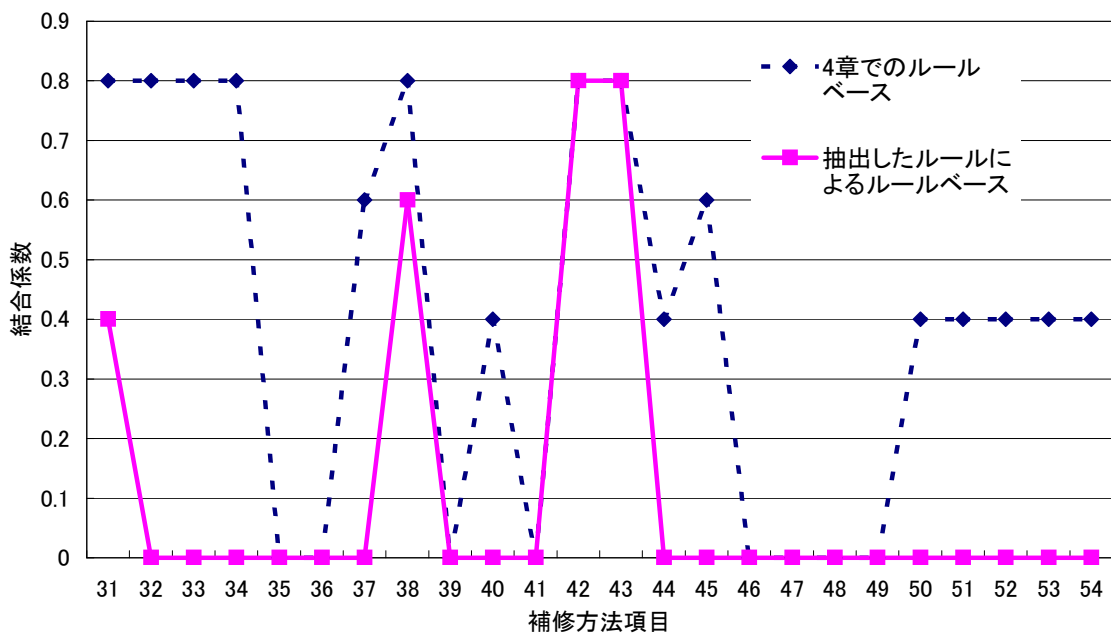


図 4.10 内的要因 (12) と補修方法の因果関係 (事例 1)

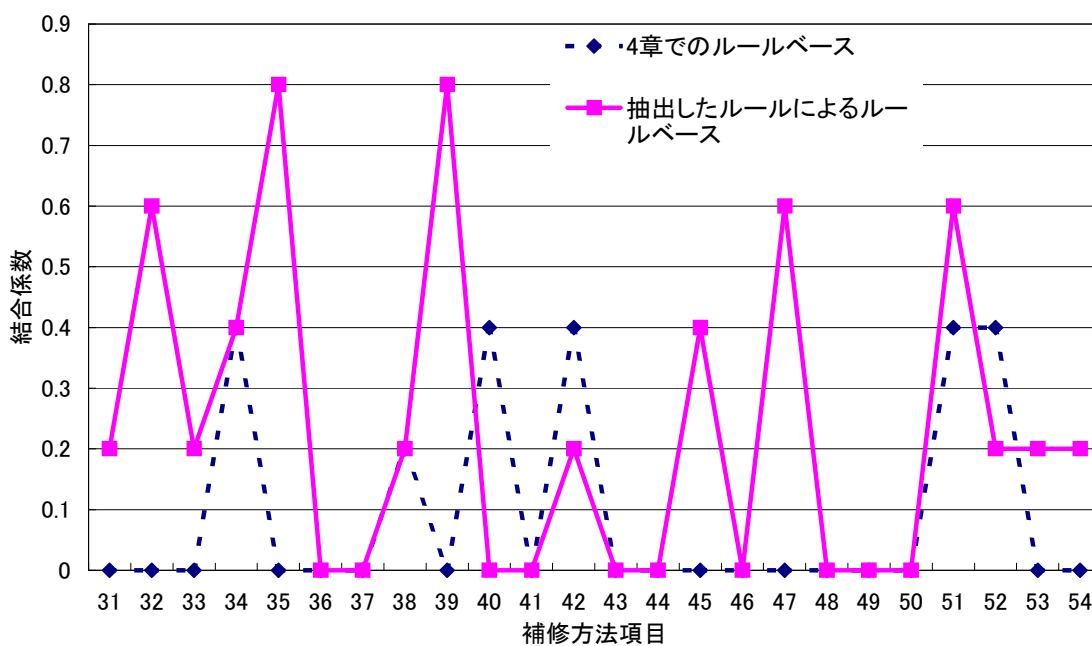


図 4.11 継手の作用力 (23) と補修方法の因果関係 (事例 1)

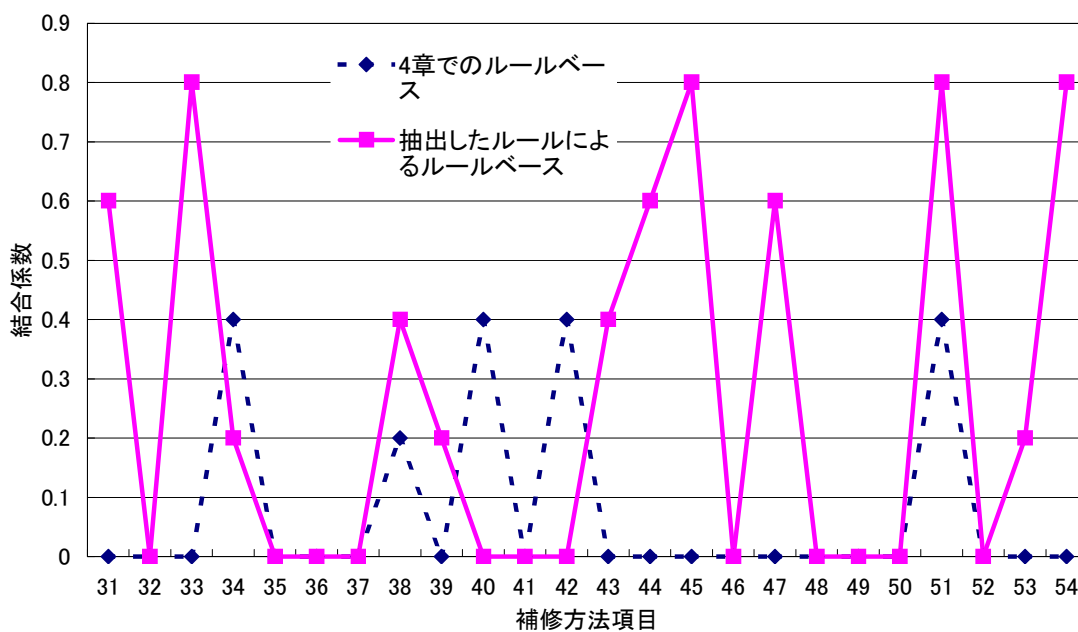


図 4.12 亀裂様式 (29) と補修方法の因果関係 (事例 1)

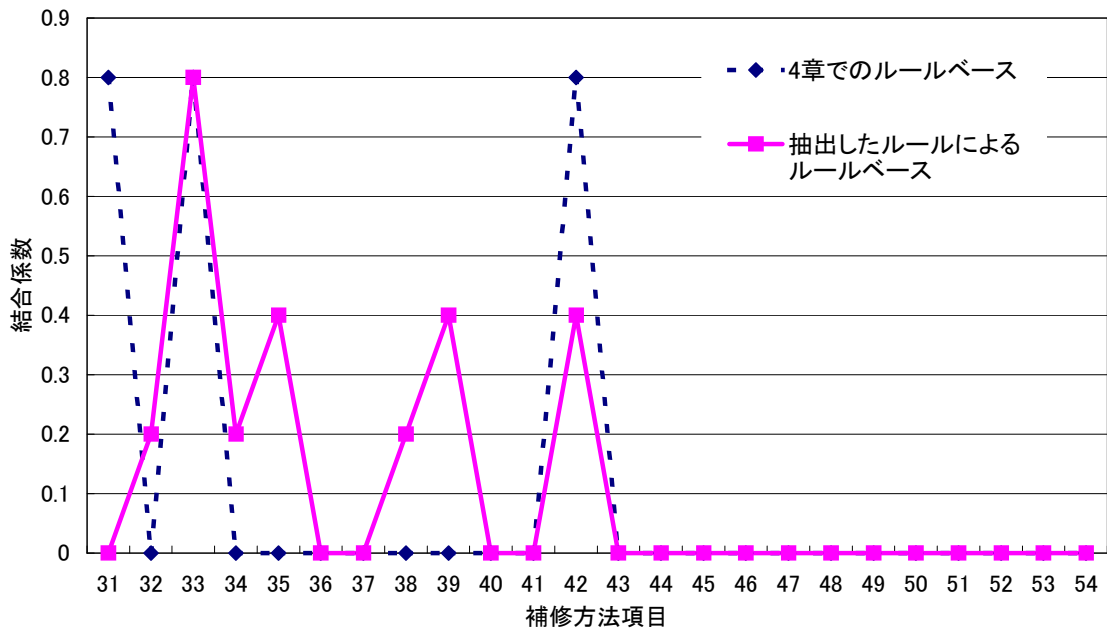


図 4.13 外的要因 (11) と補修方法の因果関係(事例 2)

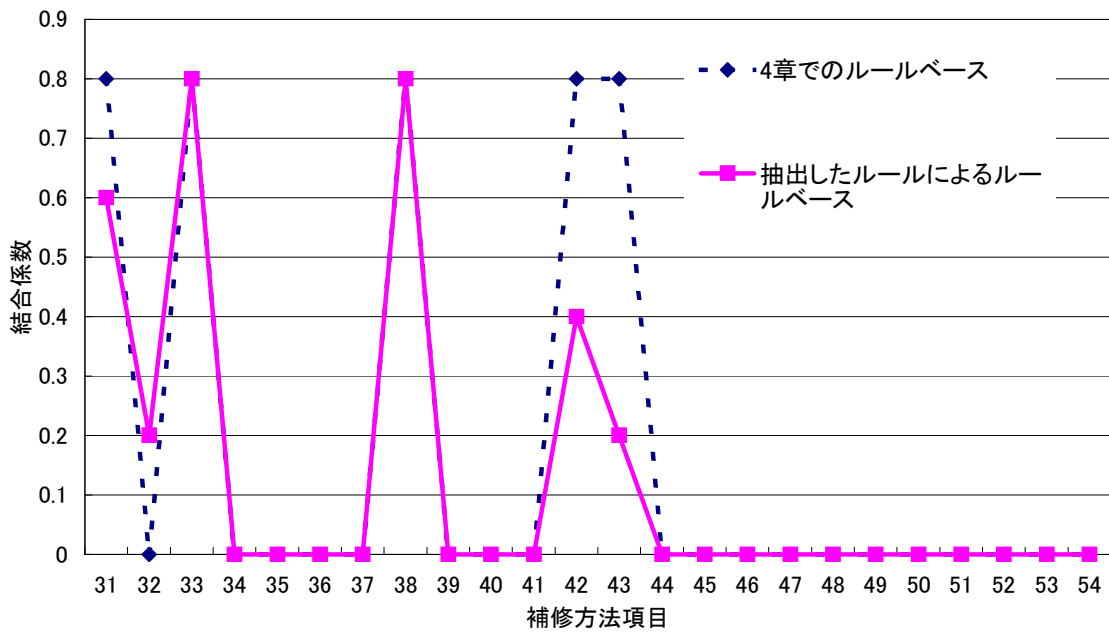


図 4.14 内的要因 (14) と補修方法の因果関係(事例 2)

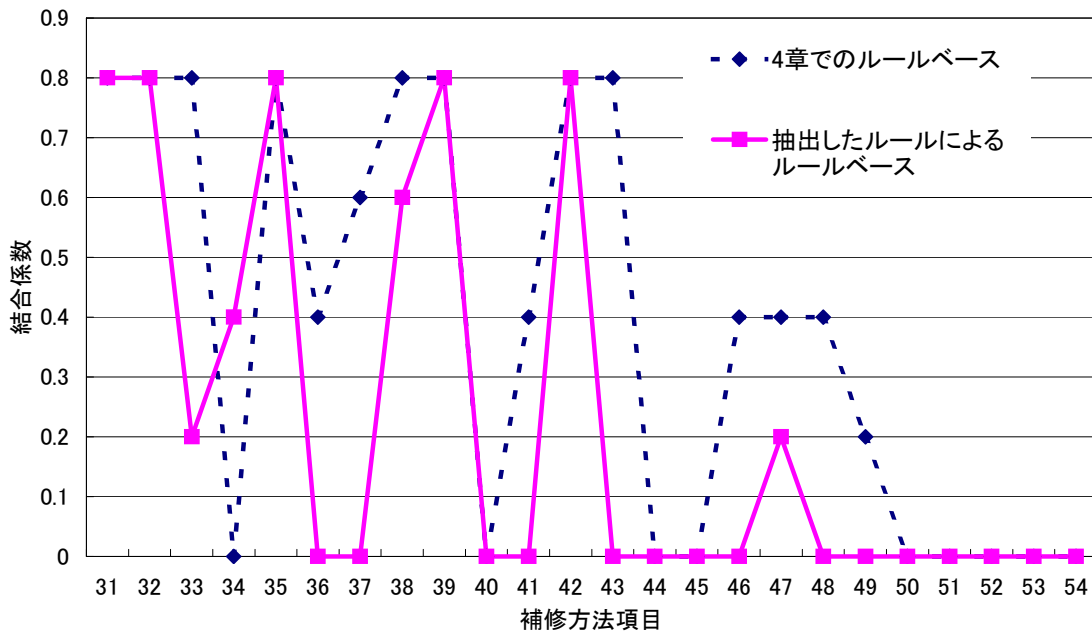


図 4.15 継手の作用力 (18) と補修方法の因果関係 (事例 2)

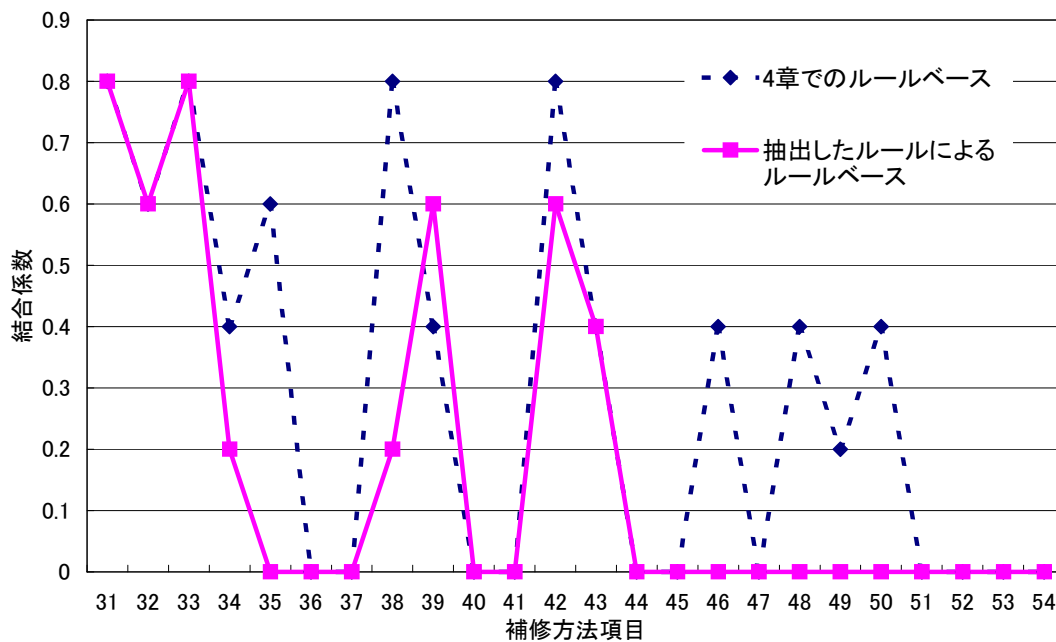


図 4.16 亀裂様式 (27) と補修方法の因果関係 (事例 2)