概念学習における探索性能向上に関する研究

1997年11月

斉藤和巳

①

概念学習における探索性能向上に関する研究

１９９７年１１月

斉藤和巳

序文

概念学習は人工知能研究における最重要研究課題の1つであり，様々な学習アルゴリズムが提案されている．しかるに，大規模でノイズや不完全情報のある現実問題においても高い汎化能力を有する効率の良い学習アルゴリズムを構築すること，および，複数モデルの出力の統合法や多様な学習戦略の選択法を適切に学習するメタレベル学習機構を構築することが課題として挙げられる．本論文は，これらの課題解決に向けて著者が行なってきた研究について述べたもので，そこでの基本方針は，従来法よりもアルゴリズムを精緻にし，探索性能を向上させることにより，優れた概念学習アルゴリズムの構築を目指すことである．

全文は次の8章から構成される．

第1章は序論であり，本研究の歴史的背景および目的について概説するとともに従来の諸研究との関連について述べる．

第2章から第4章では，記号ベース概念学習 (ルール抽出) 法について述べる．第2章では，命題論理で記述される概念を対象に，ノイズを含まない事例からのルール抽出アルゴリズム RF2 について述べる．第3章では，ルール集合選択の新評価尺度を提案し，第2章で提案したアルゴリズムを用いた，ノイズを含む事例からのルール抽出アルゴリズム RF3 について述べる．第4章では，集約関数を含む1階述語論理で記述される概念を対象に，学習戦略を変更するメタレベル学習機構の構築に向けて，過去の問題解決の経験から自ら適応して概念学習効率を改善する適応学習アルゴリズム RF4 について述べる．

第5章から第7章では，ニューラルネットを用いた数値ベース概念学習法について述べる．第5章では，準ニュートン法に基づくニューラルネット高速学習アルゴリズム BPQ について述べる．第6章では，データに内在する数法則の発見をニューラルネットの学習問題として定式化し，第5章で提案したアルゴリズムと情報量基準 MDL を用いた，法則発見アルゴリズム RF5 について述べる．第7章では，複数モデルの出力を統合するメタレベル学習機構を有する専門家の階層混合モデル HME において，第5章で提案したアルゴリズムに基づく構成的学習アルゴリズムについて述べる．

第8章は結論であり，本研究で得られた成果をまとめ，今後の課題について述べる．

目次

1 序論 **1**

1.1 研究の背景および目的 1

1.2 論文の構成 3

2 事例からのルール抽出法: RF2 **7**

2.1 序言 7

2.2 フレームワーク 8

2.3 従来法とその問題点 8

2.4 RF2 アルゴリズム 10

2.4.1 アルゴリズムの概略 10

2.4.2 ルール候補の生成 11

2.4.3 ルールの精錬 12

2.5 実験による評価 13

2.5.1 目標概念 14

2.5.2 事例数 14

2.5.3 実験結果 15

2.5.4 汎化能力比較 16

2.6 医療診断問題への適用 16

2.7 結言 17

3 ノイズを含む事例からのルール抽出法: RF3 **19**

3.1 序言 19

3.2 フレームワーク 20

3.3 新評価尺度: MEF 20

3.3.1 MEF の導出 21

3.3.2 MEF の位置づけ 23

3.4 RF3 アルゴリズム . . . 23
3.4.1 訓練誤り . . . 24
3.4.2 ルール集合の複雑さ . . . 24
3.4.3 RF3 アルゴリズムの詳細 . . . 24
3.5 実験による評価 . . . 27
3.5.1 実験の設定 . . . 27
3.5.2 実験結果 . . . 28
3.6 結言 . . . 32

4 適応概念学習法: RF4 35
4.1 序言 . . . 35
4.2 フレームワーク . . . 36
4.3 RF4 アルゴリズム . . . 36
4.3.1 核機能 . . . 36
4.3.2 適応機能 . . . 38
4.4 チェス終盤戦への適用 . . . 41
4.5 ボンガルド問題への適用 . . . 41
4.5.1 ボンガルド問題 . . . 42
4.5.2 核機能の評価 . . . 44
4.5.3 適応機能の評価 . . . 45
4.6 タスク順序付け問題への適用 . . . 47
4.6.1 タスク順序付け問題 . . . 47
4.6.2 最尤推定法とベイズ推定法 . . . 49
4.6.3 事例実験 . . . 50
4.7 一般化タスク順序付け問題への適用 . . . 52
4.7.1 問題と解法の一般化 . . . 52
4.7.2 事例実験 . . . 53
4.8 結言 . . . 54

5 ニューラルネットの高速学習法: BPQ 57
5.1 序言 . . . 57
5.2 フレームワーク . . . 58
5.3 BPQ アルゴリズム . . . 59

5.3.1 準ニュートン法 . . . 59
5.3.2 探索方向の既存計算法 . . . 59
5.3.3 探索方向の新計算法 . . . 60
5.3.4 探索幅の既存計算法 . . . 61
5.3.5 探索幅の新計算法 . . . 62
5.4 計算量の考察 . . . 64
5.5 実験による評価 . . . 65
5.5.1 人工問題 . . . 65
5.5.2 パリティ問題 . . . 68
5.5.3 音声合成問題 . . . 68
5.6 結言 . . . 74

6 ニューラルネットを用いた法則発見法: RF5 75
6.1 序言 . . . 75
6.2 ニューラルネットを用いた法則の発見 . . . 76
6.3 RF5 アルゴリズム . . . 77
6.3.1 ネットワーク学習法 . . . 77
6.3.2 法則特定法 . . . 77
6.4 実験による評価 . . . 78
6.4.1 人工データ . . . 79
6.4.2 学習効率の評価 . . . 81
6.4.3 現実データ . . . 85
6.5 結言 . . . 85

7 HME の構成的学習法 87
7.1 序言 . . . 87
7.2 フレームワーク . . . 88
7.3 構成的学習アルゴリズム . . . 89
7.3.1 関係行列と目的関数 . . . 89
7.3.2 アルゴリズムの概要 . . . 89
7.3.3 初期化法 (Step 1) . . . 90
7.3.4 訓練法 (Step 2) . . . 92
7.3.5 結合重みの縮小 . . . 93

7.3.6 終了判定条件 (Step 3) 93
7.3.7 拡張法 (Step 4) 93
7.4 実験による評価 94
7.4.1 パリティ問題 94
7.4.2 関数近似問題 95
7.5 結言 96

8 結論 99

第 1 章

序論

1.1 研究の背景および目的

知能を定義付けるものとして，学習能力が挙げられる．ゆえに，機械学習は人工知能研究における最大の関心事の 1 つである．一般に，学習とは，与えられた事例や教師の指示などに基づいて，自らの内部状態を自動的に変化させ，その性能 (正答率や処理効率など) を改善できることと定義される．工学的な学習の有用性について言えば，学習機能のあるシステムでは，十分に学習を行うことにより，未知の環境や新たな事例に対処することも可能となるが，一方，学習機能のないシステムでは，設計者が意図しない状況において一般に極めて脆くなることが避けられない．

近年の計算機処理能力の向上により，大量データから複雑な知識を学習可能な状況にあり，特に，本論文で対象とする事例に基づく教師あり概念学習アルゴリズムは広く研究されている．概念学習とは，与えられた事例を適切に認識するための認識器を形成することであるが，いわゆるパタン認識との相違点を強調して定義すれば，学習という手段を用いて認識器を形成し，認識器の記述を概念として同定することである．概念学習は，論理結合を陽に用いて述語の組合せとして概念を表現する記号ベース概念学習と，数値パラメータで定義される関数を用いて概念を表現する数値ベース概念学習に分類できる．両者を探索アルゴリズムの観点で比較すれば，前者が離散空間の組合わせ探索を行うのに対して，後者では，勾配ベクトルなどを利用して連続空間の探索を行う．以下では，それぞれの歴史的背景について概説する．

記号ベース概念学習であるルール抽出の先駆的研究は Winston による “アーチ” の学習 [140] であり，さらに，質量分光器の予想ルールを学習する Buchanan らの META-DENDRAL システム [12]，大豆の病名診断ルールを学習する Michalski らの AQ11 システム [60]，また，決定木を学習する汎用学習アルゴリズムである Quinlan の ID3 [83] などが提案され，ルール抽出の有効性が実証された．これらの研究では，主として，命題論理で記述されるルールの学習を対象としたが，一階述語論理で記述されるルールを学習する汎用アルゴリズムとして，Michalski の INDUCE [58]，Quinlan の FOIL

[86], Muggleton らの GOLEM [75] などが提案されている．これら代表的なルール抽出アルゴリズムの探索には，正答率や簡潔性などに基づく様々なヒューリスティック評価関数が提案されている．しかし，与えられた問題に対して，予めどのような評価関数を用いるべきかは分らず，望ましいルール集合を学習できないケースもある．また，現実問題においては，事例がノイズを含むことを想定しなければならないので，全ての訓練事例を正しく分類するルール集合がベストとは限らない．よって，訓練事例に対する誤り許容率を適切に決定し，高い信頼性で良い精度を保証するルール抽出アルゴリズムが重要になる．

記号ベース学習では，学習により探索効率を改善する適応 (高速化) 学習アルゴリズムの研究も行われている．その先駆的研究は Samuel による checker プログラム [124] であるが，この技術は，主として，ゲーム探索の分野だけで利用されてきた．汎用アルゴリズムとしては，問題解決に有効なルールの組合せであるマクロオペレータ (チャンク) を学習する EBL (Explanation-Based Learning) アプローチ [68, 15] が広く研究され，その代表的システムとしては，Mitchell らの LEX [69], Laird らの SOAR [47], また，Minton らの PRODIGY [66] などが提案されている．しかし，1 つの事例から作成したマクロオペレータの一般的な有用性に関する utility 問題 [65] が課題として指摘されている．

ニューラルネットを用いた概念学習は数値ベース概念学習の典型であり，その先駆的研究は Rosenblatt のパーセプトロン [92] である．パーセプトロンで対象とする単層ネットワークの学習能力の限界は，Minsky らにより，数学的に明らかにされた [64]．しかるに，Rumelhart ら [93] により，多層ネットワークの学習アルゴリズムである BP (Back Propagation) が提案されると，BP は，英単語の発音問題 [127]，ソナーデータの識別問題 [33]，また，医療診断問題 [94] などの多様な現実問題に適用され，その有効性が実証されている．しかし，収束までには一般に多くの計算量が必要となり，さらに，性能に直結する学習定数などのパラメータの決定も課題となる．

記号ベース学習とニューラルネット学習を比較すれば，記号概念などの高次のレベルにおいて，簡潔なルール集合を生成するのに前者が適しているのに対し，後者では，信号情報などの低次のレベルで，複雑な写像を形成するのに適していると考えられる．なお，両者を統合するための試みも，多戦略学習法 [59] やハイブリッドシステム [31] などとして研究されている．一方，両者の代表的アルゴリズムを直接実験により比較した研究 [73, 138] では，ニューラルネット学習は，テスト事例に対する汎化能力の点では僅かに優るものの，学習完了までに多くの計算量を必要とすることが指摘された．ただし，これはニューラルネット学習の限界を示すものではなく，そこで採用された BP アルゴリズムの効率が悪いことを指摘しているのに他ならない．

今後の展望も含めて，既存の概念学習アルゴリズムの課題を考察すれば以下となる．第 1 に，幅広い現実問題への適用を可能にすることである．すなわち，ノイズを含む不完全な情報の事例しか得られないケースや，大量の背景知識などを含む大規模な問題でも適用可能にすることである．また，概念記述

の表現能力を向上させること，および，学習に重要な影響を及ぼすにも拘らず，問題に依存して，ユーザが試行錯誤で設定していた学習パラメータなどをできる限り自動設定可能にすることも重要である．第 2 に，複数モデルの出力の統合法や多様な学習戦略の選択法を学習するメタレベル学習機構を構築することである．具体的には，理論と実験の両面から，既存モデルやアルゴリズムの長所と短所を明確にすることを始めとし，それを踏まえて，記号ベースや数値ベースに基づく複数モデルの出力や学習結果を適切に統合する学習機構や，予め与えられたヒューリスティック評価関数だけを用いるのではなく，多様な学習戦略を問題に応じて適切に使い分けるための学習機構を構築することである．さらなる目標としては，現実環境との多様なインタラクションを可能にし，現実環境での学習を可能にすることである．インタラクションには，音声や画像などのマルチメディア情報を直接認識できることや，学習の重要なポイントをユーザが直接教示できることなどが挙げられる．

これらの状況を踏まえれば，まず．大規模でノイズや不完全情報のある現実問題においても高い汎化能力を有する効率の良い学習アルゴリズムを構築することが基本課題として挙げられる．これに対して本論文では，従来法よりもアルゴリズムを精緻にし，概念学習における探索性能を向上させることで解決を試みる．また，複数モデルの出力の統合法や多様な学習戦略の選択法を適切に学習するメタレベル学習機構を構築することも課題となる．一方，記号ベースと数値ベースの概念学習は，それぞれ互いを補完する形で，今後の概念学習アルゴリズムの重要な要素技術になると考えられる．よって，本論文では，それぞれのパラダイムで，大規模な問題でも高品質な学習結果をもたらす基本学習アルゴリズム，ノイズを含む事例からでも汎化精度の高い学習結果を得るための評価尺度を導入した学習アルゴリズム，および，メタレベル学習機構の構築に向けた試みとして開発した学習アルゴリズムについて述べる．すなわち，記号ベース概念学習では，厳密解を保証する IDA* 法 [44] を利用した基本学習アルゴリズム RF2，訓練誤り率の許容限界を適切に決定するための新評価尺度 MEF を RF2 に導入した学習アルゴリズム RF3，さらに，学習戦略を変更するメタレベル学習機構の構築に向けて，過去の問題解決の経験から自ら適応して概念学習効率を改善する適応学習アルゴリズム RF4 を提案する．一方，数値ベース概念学習では，準ニュートン法 [29] に基づくニューラルネット高速学習アルゴリズム BPQ，ニューラルネットの学習問題として定式化した数法則発見において BPQ と情報量基準 MDL を用いる学習アルゴリズム RF5，さらに，複数モデルの出力を統合するメタレベル学習機構を有する専門家の階層混合モデル HME の構成的学習アルゴリズムを提案する．

1.2 論文の構成

本論文は，序論と結論を含めて全体を 8 章で構成している．第 1 章の序論に続いて，まず，第 2 章から第 4 章では，記号ベース概念学習 (ルール抽出) 法について述べる．次に，第 5 章から第 7 章では，数値ベース概念学習法として，ニューラルネットを用いた学習アルゴリズムについて述べる．

第 2 章では，少ない事例からでも，簡潔で十分に汎化した分類ルールを抽出可能とする RF2 を提案する．RF2 はルール候補の生成と精錬の 2 フェーズからなり，正の事例の記述を逐次一般化することによりルール候補を生成し，IDA* と呼ばれる探索手法を用いてそれらを最適なルールの集合に精練する．人工問題への適用では，最新のルール抽出法でも抽出できなかったルールを，RF2 は少ない事例からでも抽出できたことを示す．また，医療診断問題への適用では，未知の事例に対する正答率が医者の知識を用いて作成したエキスパートシステムのものと匹敵したことを示す．

第 3 章では，ノイズを含む事例からでも，高い信頼性で良い正答率を保証する分類ルールを抽出可能とする RF3 を提案する．RF3 の特長は，ルール集合選択のための新評価尺度 MEF の導入にある．MEF 尺度は，任意の汎化誤り率の許容限界に対し，抽出したルール集合の汎化誤り率が許容限界より悪くなり，抽出失敗となる確率の期待値の最小化を意図するものである．また，MEF 尺度は，抽出したルール集合の複雑さと例外事例の個数の和を最小化する尺度としても解釈可能であることを示す．実験の範囲では，RF3 のルール抽出において，訓練誤り率の許容限界を順次変化させ，MEF 尺度と別途推定した汎化誤り率を比較したところ，両曲線が極めて類似したことを示す．

第 4 章では，集約関数を含む 1 階述語論理で記述される概念を対象に，適応学習する RF4 を提案する．RF4 は，望ましくない概念を枝刈りする学習バイアス，および，存在限量子や集約関数を用いた概念を生成する複合化ルールを用いて，事例の識別概念を深さ優先で探索する．また，論理式の探索順序は，過去に解いた問題を基に，それが識別概念の構成要素となる確率を推定することにより，動的に決定される．KRK チェス終盤戦問題では，ランダムに選んだ事例群の学習を数回繰り返せば，RF4 の探索効率が改善されただけでなく，未知の事例に対する正答率も向上したことを示す．図形の多彩な識別概念を求めるボンガルド問題では，問題を解くにつれて，推定確率の信頼性が高くなるので，ボンガルド問題を解くための平均時間が次第に短縮されたことを示す．適応学習の基本となるタスク順序付け問題では，ベイズ推定を用いる方法が，最尤推定してタスク列を求める方法と比較して，最小コストに速く近付くことを示す．

第 5 章では，3 層ネットワークにおける 2 次の高速学習アルゴリズム BPQ を提案する．BPQ は準ニュートン法をベースとし，探索方向を小記憶 BFGS 法で計算し，最適探索幅を 2 次近似の最小点として効率良く計算することを特徴とする．また，BPQ の 1 反復の計算時間については，一般的な状況で，標準的な BP とほぼ等しいことを示す．人工問題，パリティ問題，および音声合成問題を用いた実験では，他の代表的な学習アルゴリズムと比較して，BPQ は効率良く誤差を減少できたことを示す．さらに，この探索幅計算法は準ニュートン法の収束性に重要な役割果たし，一方，小記憶 BFGS 法は収束性を変えずに記憶容量を大幅に減少できたことを示す．

第 6 章では，数値データに内在する未知の法則を発見するアルゴリズムとして，ニューラルネットを用いた法則発見法 RF5 を提案する．RF5 では，法則の発見問題がニューラルネットの学習問題として定式化され，第 5 章で提案した BPQ アルゴリズムを利用して複数の法則候補を作り出し，その中から MDL 基準を利用して最良のものを法則として特定する．実験では，ある程度のノイズを含むデータからでも，RF5 は指数の値が整数に制限されない法則を効率良く発見できることを示す．

第 7 章では，複数ニューラルネットの出力の組合せを自己組織的に学習する専門家の階層混合モデル HME の構成的学習アルゴリズムを提案する．このアルゴリズムの特徴は，鞍点に捕らわれることを防ぐための結合重み初期化法，第 5 章で提案した BPQ アルゴリズムを利用したネットワーク学習法，および，“divide-and-conquer” アプローチに基づくネットワーク拡張法を有することである．パリティ問題と関数近似問題を用いた実験では，従来法では困難であるが，提案法を用いれば，最小規模の HME でも望ましい結果が得られることを示す．

第 8 章では，結論として以上についての成果のまとめを行う．

第 2 章

事例からのルール抽出法: RF2

2.1 序言

知識獲得はエキスパートシステム構築の最重要課題である. エキスパートからの知識 (ルール) 抽出は困難で, 可能な限りの自動化が強く望まれている.

事例からのルール抽出は, 事例を用いて新らたな知識の生成を行う記号表現による概念学習であり, 機械学習の分野では, 近年最も活発に研究が行われている [131]. これらの手法の共通点の 1 つは, 可能な限り簡潔な知識の生成を目標とすることである [61]. 簡潔な結果は, 人間にとっても理解が容易であり, 知識獲得に望ましい性質である [61].

記号表現による概念学習の代表的なアルゴリズムには, ID3 [84], AQ [57] 等があり, 現実問題等にも適用され, 広く研究されている. また, これらの手法をベースに機能拡張や改良を加えたアルゴリズムの提案もある [14, 77]. しかしながら, 可能な限り簡潔な知識を生成すること, すなわち, 生成するルール (条件) の数を最少にすることは NP- 困難な問題であり, 一般に, 結果の良さと計算量のトレードオフが重要になる. エキスパートシステム構築の知識獲得という観点からは, ある程度の計算量を必要としても, より良いルールの抽出が望まれる. 本章では, 比較的規模の大きな問題 (例えば, 80 属性, 4000 事例) でも, ワークステーションを用いて妥当な計算時間 (数十分) で抽出が完了することを条件とし, より良いルールの抽出を可能とするアルゴリズムについて論じる.

従来の代表的なアルゴリズムの共通点は, 属性を逐次選択しながらルールを生成することにある. しかし, この属性選択方式では, 一般に, 正確にルール抽出できない問題が存在する. 例えば, 不要な属性を持つパリティー問題である. 理論的には, 任意の 1 つの属性での値 (オン / オフ) に着目しても, それぞれ, パリティー条件を満たす事例とそうでない事例は同数である. したがって, 属性選択方式では, 一般に, 終了条件が成立するまで, ランダムな属性の選択を繰り返さねばならない.

概念学習の結果 (ルール) の信頼性評価には, 一般に, さらに進んで未知の事例に対する正答率を調べる実験が必要である. しかし, 一定の確率分布に従って毎回独立に事例が現われ, 学習する概念の属

するクラスが予め分かっていれば, PAC (Probably Approximately Correct)- 学習の理論 [134, 9] を用いて, 確率的に汎化能力を保証することができる. すなわち, 学習する知識の複雑さから, 学習結果を保証するのに必要な事例数が分る. 現実の問題では, 前もって知識の複雑さが分からず, 適用できないことが多いが, 人工問題を用いれば, アルゴリズム評価のための重要な指標を得られる.

本章では, 応用の広い分類問題を対象に, 少ない事例からでも簡潔で汎化された分類ルールを抽出する RF2 [106] について述べる. まず, ルール抽出の枠組みについて説明し, 次に, 従来手法の問題点について述べる. 次いで, RF2 アルゴリズムの特徴とその詳細について述べる. 最後に, 人工問題, および現実問題への RF2 の適用結果について述べる.

2.2 フレームワーク

分類ルールの抽出を行う枠組みについて説明する. 概念 (concept) は, 既に学習した他の概念またはその概念の基本構成要素である属性 (attributes) によって表現される. 例えば, 病名という概念は {体温, 血圧, 性別, ···} という属性によって表現できる. 事例は分類したい概念に属す正の事例と属さない負の事例からなる. 例えば, 正の事例はその病名の患者であり, 負の事例は非患者である. そして, 各事例は属性空間内の点として位置付けられる.

一般に, 属性を用いて概念を表現する形式は論理式のクラスとして定式化される. ここではそのクラスとして, 人間にとって直感的で理解しやすいと考えられる選言標準形 (DNF: Disjunctive Normal Form) を扱う. DNF はターム (term) の論理和として表現され, タームはリテラル (literal) の論理積として表現される. 各リテラルは対応する 1 つの属性にのみ関係し, その属性の値によって, リテラルの値は真または偽となる. 本論文では, タームを 1 つのルール, DNF をルールの集合とみなす考え方をとる. なお, 本章では, ノイズを考慮しない範囲でのルール抽出について論じ, ノイズを含む事例からの抽出については, 次章で述べる.

2.3 従来法とその問題点

従来法とその問題点について述べる. 生成するルール (条件) の数を最少にするという尺度の下で, 厳密解を求めるアルゴリズムの計算量は莫大となり, 到底実行することはできないと考えられる. なぜなら, 単純に計算すれば, 属性数が N のとき, 各属性毎に 3 通り (肯定的, 否定的, 無関係) の条件の現われ方があるので, 可能なルール候補の個数は 3^N となる. さらに, 求めたルール候補の集合には一般に冗長性が存在するので, その中から適当な組み合わせを選択する必要がある. したがって, 効率的なルール抽出にはヒューリスティクスを用いたアプローチが必然になる. 以下に, 代表的な 3 種の概念学習アルゴリズムを概説し, それらの問題点を指摘する. 図 2.1 には, ここで取り上げるアルゴリズムの系譜を示す. なお, 本論文で提案する *RF2* (*Rule extraction from Facts version 2*) は, RF

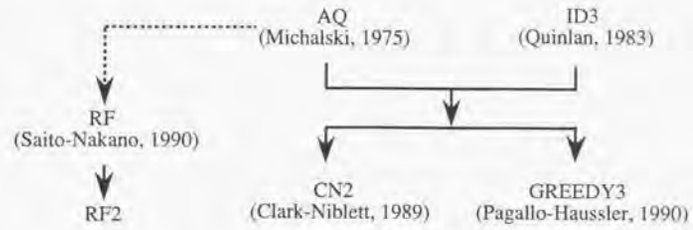


図 2.1: アルゴリズムの系譜

[104, 95] の改良アルゴリズムである.

第 1 の手法は事例から決定木 (decision tree) を構築する方法であり, ID3 [84] が代表的アルゴリズムである. ID3 は, 事例集合の分割を再帰的に実行し, すべての分割された事例集合が正または負の事例だけになるようにする (divide and conquer). ここで, 事例集合の分割には, 情報量の期待値に基づいて選択された属性の値が用いられる. 決定木学習の問題点は, 決定木そのものの表現能力が不十分なことである. 例えば, 和結合した概念 $x_1x_2 \vee x_3x_4 \vee x_5x_6$ を簡潔な決定木として学習できない. なぜなら, x_1 で最初の分割が行われた場合, 2 つの分割された事例集合のそれぞれが $x_3x_4 \vee x_5x_6$ を分類しなければならない. さらに, それぞれの部分集合で x_3 が選択された場合には, x_5x_6 を分類する部分木は全体で 4 つ必要になる. したがって, x_5x_6 を満たす事例も適当に 4 つに分割され, 特に, 元の事例数が少ない場合には, 決定木の細部まで完全に学習することは極めて困難になる [77].

第 2 の手法は, AQ システム [62], 概念クラスタリング [63] の基本アルゴリズムとして用いられる AQ [57] である. AQ の考え方に基づいたルール抽出アルゴリズム AQR [14] について説明する. AQR では, すべての正の事例が少なくとも 1 つのルール (スター) にカバーされるまで, ルールの生成を繰り返す. ルールの生成では, まず, どのルールにもカバーされていない正の事例を seed としてランダムに選択する. 次に, ヒューリスティク評価関数に基づき, seed をカバーして, すべての負の事例が排除できるルールを Beam サーチにより生成する. ヒューリスティクスでは, より多くの正の事例をカバーして, より多くの負の事例を排除できる属性の組み合わせを優先して探索する.

AQR アルゴリズムを一部修正して, すべての正の事例を seed としてルール生成を行うとすれば, 結果はかなり冗長なルール集合になってしまう. 冗長なルールを消去することは, 後述するように, ルールを集合, それに含まれる正の事例を集合の要素と考えることにより, SC 問題 (set covering) [42] として定式化できる. ところが, AQR のようにランダムに seed を 選択することは, SC 問題において, すべての要素をカバーするまで, ランダムに集合 (ルール) の選択を繰り返すことに他ならない. よっ

て, SC 問題の解法を追及することにより, より良いルール集合の抽出が可能になると考える.

第 3 の手法は決定リスト (decision list) [91] と呼ばれる if-then-else 形式のルールリストを構築する方法であり, CN2 [14], GREEDY3 [77] などのアルゴリズムが提案されている. 2 つの方法では, すべての正の事例が消去されるまで, ルールの生成とそこでカバーされた正の事例の消去を繰り返す (separate and conquer). ルールの生成では, ヒューリスティク評価関数に基づき, より多くの正の事例をカバーできる属性の組み合わせを逐次選択する. なお, CN2 では, AQ と同様に Beam サーチが用いられる. ヒューリスティクスは, CN2 では情報量の期待値が用いられるのに対し, GREEDY3 ではベイズ規則により評価される. つまり, 抽出ルールの表現形式を決定木から決定リストに変えることにより, 第 1 の手法と比較して, 表現能力を向上させ, SC 問題として定式化できるルール集合の精練を欲張り法 (greedy algorithm) [42] に基づいて実行することにより, 第 2 の手法と比較して, その精練能力を向上させた. ここで, 欲張り法とは, すべての点 (事例) が消去されるまで, 最も多くの点をカバーする集合 (ルール) の選択とその集合にカバーされた点の消去を繰り返す方法である.

しかし, 欲張り法は近似解法であり, さらに SC 問題の解法を追及する余地がある. すなわち, CN2 と GREEDY3 により順番に生成されるルールが, 一般に, 最終的なルール集合として適切なものを生成しているとは限らない. 例えば, 初めに抽出したルールが, 後から抽出したルール集合の和で完全に包含されるケースも考えられる. したがって, ルール生成段階で, ルール候補としてルールを冗長に生成し, 次のフェーズとして, それらを簡潔なルール集合に精練することにより, より良いルール集合の抽出が可能になると考える.

さらに, すでに指摘したように, 以上説明した手法に基づいた 4 種のアルゴリズムの共通点は, 属性を逐次選択しながらルール生成をすること, すなわち, 属性選択方式を採用していることにある. したがって, 不要な属性を持つパリティー問題などでは, 極めてルール抽出が困難となる. 一方, RF2 では, 属性選択方式を採用せず, 事例の記述を逐次一般化してルール抽出することにより, この問題点の克服を試みている.

2.4 RF2 アルゴリズム

ここでは, まず RF2 アルゴリズムの概略について説明し, 続いてその詳細を述べる.

2.4.1 アルゴリズムの概略

RF2 の 1 つの特長は, 正の事例の記述を一般化することによりルール生成をすることである. このため, 属性選択方式では適当なヒューリスティク評価尺度が得られない問題でも, 類似した事例の記述を逐次一般化することにより, 良いルールを得ることが期待できる. しかし, 一般に, この方法だけでは, 冗長なルール集合が生じる場合も考えられる. すなわち, あるルールが他のルールの和集合で包含

される場合である. この問題に対処するため, RF2 のもう 1 つの特長として, ルール候補の集合を最適なルール集合に精練するフェーズを備えている. さらに, このフェーズは, IDA* と呼ばれる探索手法 [44] を用いることにより, 効率良く解を得ることができる.

2.4.2 ルール候補の生成

RF2 によるルール候補生成の特徴は, 各正の事例の記述を逐次一般化することにより, special-to-general 探索でルール候補を生成することである. すなわち, 正の事例を seed として選択し, ヒューリスティクスにより選択する正の事例を順次用いて, seed の記述の最小限の一般化を繰り返し, ルールを生成する. 一方, 前節で説明した従来の方法では, ヒューリスティク評価関数等を利用して属性を選択することにより, general-to-special 探索でルールを生成する.

seed の記述の一般化には, 参照統合オペレータ (Ref-Union operator) [63] を用いる. 今後, このオペレータを *RU* と略記する. RU では, seed の記述, 正の事例, および, すべての負の事例から, 以下の手続きにより, seed の記述を出力する.

- *RU* オペレータ

  1. 事例の記述と正の事例の属性値の論理和を各属性毎にとり, 中間記述を生成する.
  2. もし, 中間記述が負の事例を 1 つでもカバーすれば, 元の記述を結果とする. さもなければ, 中間記述を結果とする.

例えば, D を seed の記述, p_i を正の事例, n_j を負の事例とし, $D = (1,0,0 \vee 1)$, $p_1 = (1,0,0)$, $p_2 = (1,1,1)$, $p_3 = (0,0,0)$, $n_1 = (0,0,1)$, $n_2 = (0,1,0)$, とすれば,

$$
\begin{aligned}
RU(D,p_1) &= (1,0,0 \vee 1),\\
RU(D,p_2) &= (1,0 \vee 1,0 \vee 1),\\
RU(D,p_3) &= (1,0,0 \vee 1),
\end{aligned}
$$

が結果となる.

正の事例を選択するヒューリスティクスとして, 以下の 2 つを考案した. 第 1 に, 既に得られているルール候補集合にカバーされた事例の優先度を低くする. 第 2 に, seed とした事例との距離が小さい事例の優先度を高くする. すなわち, 前者は, 類似したルールを繰り返し生成することを極力避けるためであり, 後者は, 類似する事例を同じルールにカバーさせるためのヒューリスティクスである.

ルール候補の生成アルゴリズムの詳細を以下に示す. ここで, seed の記述は D_0 から D_f へ一般化される. ALPHA は最終的なルール候補の集合となり, BETA はアルゴリズムを制御するためのルール候補の集合である. また, ユーザの定義する MC は, 一般化された seed の記述を何個ルール候補と

するかを制御するパラメータである．ただし，以下での実験では，$MC = 1$ とした最もシンプルな場合の結果である．

- ルール生成アルゴリズム

  1. 各正の事例を seed として，以下のヒューリスティクスに基づき，各 seed の記述 D_0 を逐次一般化する．
  2. BETA に含まれるルール候補集合にカバーされていない正の事例を対象に，最も多くの事例をカバーする seed の記述 D_f を BETA に加え，多くの事例をカバーする上位 MC 個の seed の記述 D_f を ALPHA に加える．
  3. すべての正の事例が BETA に含まれるルール候補でカバーされたならば処理を終了する．さもなければ，各 seed の記述 D_f を D_0 に戻し，1. に戻って処理を繰り返す．

- ヒューリスティクス

  1. BETA に含まれるルール候補にカバーされる回数が少ない事例の優先度を高くする．
  2. 前項が同じ事例に対しては，seed の記述 D_0 との距離が小さい事例の優先度を高くする．

2.4.3 ルールの精錬

ルールの精錬には，冗長に生成したルールを除去するルールレベル精錬と，各ルールに現われる冗長なリテラルを除去するリテラルレベル精錬がある．まず，SC 問題とは，集合の族 $F = \{S_1, \cdots, S_n\}$ と各集合 $S_i = \{p_{i1}, \cdots, p_{ik_i}\}$ が与えられたとき，

$$\{F' \subset F : \bigcup_{S \in F'} S = \bigcup_{S \in F} S\}$$

から $|F'|$ を最小にするものを求める問題である．2 つのタイプの精錬は，次のように対応させればSC 問題として定式化できる．ルールレベル精錬はルール候補を集合 S_i，それに含まれる正の事例 p_{ij} を要素と考える．また，リテラルレベル精錬では，各ルールをそれぞれ独立に扱い，リテラルを集合 S_i，それにより排除される負の事例 p_{ij} を要素と考える．

しかし，SC 問題は NP 困難な問題なので [27]，一般には，厳密解を求めることは大変難しく，近似解を求めるアプローチが採用されている．解法としては，最も多くの点をカバーする集合の選択とその集合にカバーされた点の消去を繰り返す欲張り法がよく用いられている．この方法の特徴は高速性と解の高品質にある．すなわち，解の最悪ケースが良い精度で保証され [42]，事例数を N，最適解の集合の個数を K，欲張り法で求めた集合の個数を S とすれば次式が成り立つ．

$$S \leq K + (\log_e N + 1)$$

ところが，欲張り法の弱点は，例えば次のようなケースに現われる： $A = \{1,2,3\}$, $B = \{2,3,4,5\}$, $C = \{4,5,6\}$．すなわち，まず，B が選ばれ，続いて A と C がそれぞれ選ばれる．このケースでは，明らかに A と C の 2 つの集合で十分である．RF2 でのルール候補の生成でも，多くの正の事例をカバーする seed の記述をルール候補とするため，まさにこのケースが起こると考えられる．したがって，この方法では高精度のルールの精錬ができない．

既述のように RF2 では，第一フェーズにおいてルール候補を絞り込んで生成し，RU オペレータの働きにより，各ルール候補の不要なリテラルを消去しているため，厳密解を求めるアプローチも現実的である．すなわち，A* 法 [44] を用いた探索が可能である．まず，集合が選択されていない状態をルートとする．そして，次に展開するノード (選択する集合) は以下の目的関数 $f(n)$ を最小にするものである．この評価尺度 $f(n)$ は展開されたノードに対しても再帰的に用いられる．最終的に，最初にすべての点をカバーしたノードをルートまで手繰ることにより結果を得ることができる．

$$\begin{aligned} f(n) &= g(n) + h(n), \\ g(n) &= \text{親ノードまでの探索木の深さ}, \\ h(n) &= |\,\text{親ノード以前でカバーされていない点}\,| \\ &\quad \div |\,\text{ノード (集合) がカバーする点}\,|\,. \end{aligned}$$

しかし，A* 法は最良優先探索であり，現在の計算機アーキテクチャでは，メモリ容量の限界からすぐに計算が不可能となる．そこで，RF2 では，探索の閾値を更新しながら繰り返し深さ優先探索を行う IDA* 法 [44] を用いてルールを精錬する．また，処理の効率化のため，任意の探索段階で一般に用いられる以下の規則を適用した．

- 他の集合の部分集合となる集合は選択しない．
- 点を固有にカバーする集合を優先して選択する．
- バックトラックの結果，カバーできない点が現われた場合，さらにバックトラックする．

なお，RF2 で用いたヒューリスティク関数は最も単純なものである．SC 問題の効率的な解法を追及した研究 [24, 3] では，工夫を凝らしたヒューリスティク関数が提案され，これらの関数を導入することにより，RF2 の効率をさらに改善することができると考える．

2.5 実験による評価

ここでは，ベンチマーク問題を用いた実験により，RF2 の性能を評価する．

表 2.1: 目標概念の要約

概念名	属性数	ターム数	平均リテラル数	学習事例数
dnf1	80	9	5.8	3,292
dnf2	40	8	4.5	2,188
dnf3	32	6	5.5	1,650
dnf4	64	10	4.1	2,640
mx6	16	4	3	720
mx11	32	8	4	1,600
par4	16	8	4	1,280
par5	32	16	5	4,000

2.5.1 目標概念

RF2 の汎化能力を評価するため, 以下に示す 3 タイプの目標概念を考える. これらの概念の要約を表 2.1 に示す. なお, これらの目標概念は GREEDY3 に対して行われた実験と同じものである [77].

ランダム DNF 概念 ランダムに生成した比較的小規模な DNF により記述された概念である. ただし, ターム数は予め指定され, 各タームに現われるリテラル数は正規分布 (平均と標準偏差を指定) に従って決定された. 実験では 4 つのランダム DNF を扱う. その中で, 2 つの DNF はモノトーン (負のリテラルが現われない) である.

MX (multiplexor) 概念 $k+2k$ 属性 (ビット) の始めの k 属性がアドレス, それに続く $2k$ 属性がデータとなる概念である. 実験では, $k=2,3$ の場合を扱い, それぞれ 10, 21 個の不要な属性を付加した.

パリティ概念 k 属性だけに着目したとき, それが k ビットの偶パリティとなる概念である. 実験では, 4, 5 パリティを扱い, それぞれ 12, 27 個の不要な属性を付加した.

2.5.2 事例数

事例の出現には, 未知だが一定の分布があり, 各事例がその分布に従って, 毎回独立に出現すると仮定できるとき, 目標概念の複雑さ (complexity) が分かっていれば, PAC-学習の理論を用いて, テスト事例に対する正答率を保証するのに必要な抽出事例数を計算できる. すなわち, 概念記述に必要な属性数を n, 概念記述に現われるリテラル数を k, テスト事例を正答できない確率を ϵ とすれば, 抽出に必

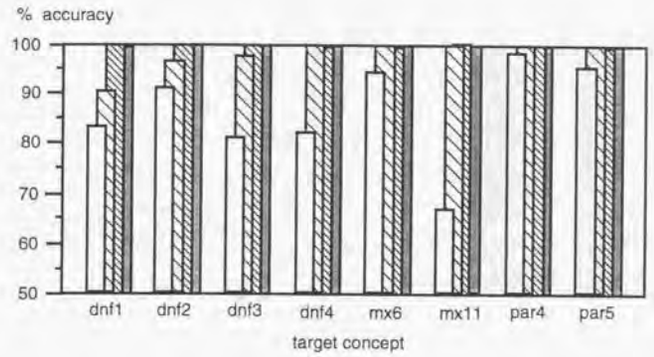


図 2.2: RF2 法の学習結果

要な事例数 m は以下の式で近似できる [77].

$$m = \frac{k \times \log n}{\epsilon} \tag{2.1}$$

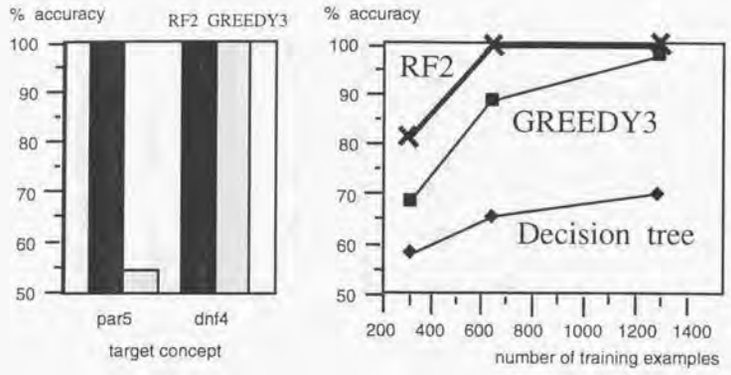
本章の実験ではすべて, $\epsilon = 10\%$, テスト事例数を 2000 とし, すべての事例の属性値には, ランダムに 1 または 0 を与えた (一様分布).

2.5.3 実験結果

図 2.2 に RF2 を用いて抽出したルールのテスト事例に対する正答率を示す. 各目標概念における 4 つの棒グラフは, 一番奥が (2.1) 式で求めた値を抽出事例数とした結果であり, 手前になるに従って抽出事例数が半分となる. すなわち, 一番手前の棒グラフは, (2.1) 式の 8 分の 1 の事例数で抽出した結果である.

図に示してある正答率は, ランダムに 10 回抽出事例を生成して実験を行った結果の平均値である. 抽出の条件 ($\epsilon = 10\%$) より, テスト事例に対して 90% 以上正答できていれば, 抽出が成功したと言える. したがって, すべての目標概念において, (2.1) 式の 4 分の 1 の事例数で十分に抽出が成功したと言える. このことは, 分布が一様という条件付ではあるが, RF2 は少ない事例からでも汎化されたルールを抽出できることが実験的に明らかになった.

3 つの目標概念 (dnf1,dnf2,dnf3) では, (2.1) 式の 4 分の 1 の事例数のとき, 完全に正しいルールを抽出できなかった. その理由として, 正の事例の全体の事例に対する割合が異なるからであると考える. すなわち, 3 つの目標概念では, その割合が 15% ~ 25% 程度であるのに対し, それ以外の目標概



(a) 2 概念の学習結果　　(b) dnf4 の学習結果

図 2.3: RF2 対 GREEDY3

念での割合は 50% 程度となっていた. したがって, RF2 法では, 正の事例数の割合が 50% に近いほうが, 抽出が容易になると予想される. 従来のアルゴリズムにおいてどのような性質があるかは, 興味深い問題である.

2.5.4 汎化能力比較

図 2.3 に RF2 法と GREEDY3 との汎化能力比較を示す. 図 2.3(a) には, 2 つの概念 (par5, dnf4) において, (2.1) 式で求めた値を抽出事例数とした結果を示す. パリティ概念は, GREEDY3 などの属性選択方式のアルゴリズムでは, 抽出するのが一般に困難な概念だと言われる [77]. ところが, RF2 を用いれば, さらに少ない事例数からでも完全なルールの集合が抽出できた. ランダム DNF 概念では, 十分な数の事例が与えられれば, GREEDY3 でも完全なルールの集合が抽出できる. しかし, 図 2.3(b) に示すように, 事例数が少ない場合には, GREEDY3 と比較して, RF2 が常に高い汎化能力を示している. なお, RF2 を決定木による方法と比較すれば, RF2 では決定木による方法の 4 分の 1 の事例数から抽出を行った場合でも, より高い汎化能力を示していることが分かる.

2.6 医療診断問題への適用

現実問題での RF2 の能力を評価するため, 患者への問診から患者の病名を判定する医療診断問題 [56] へ RF2 を適用した. ただし, この問題では患者の頭痛に関する症状と病名だけを扱っている. 問診例は, 「あなたの頭痛はいつから始まりましたか?」という質問に対し, 患者が「本日」「3 日前」等の選択肢に応える形式である. これらの選択肢の合計数は 216 (各属性値は 2 値) となる. 本実験では, 約 400 人の患者データを利用し, 患者が筋収縮性頭痛であるか判定するルールの抽出を行った.

抽出の結果, すべての患者データから, RF2 を用いて 7 つのルールが抽出できた. これらのルールの条件 (リテラル) の個数は, 平均 22 個であり, 人間にとっても理解容易なものであると考える. 抽出したルールは, 例えば, 「頭痛の場所が一定で, 痛みが 4 日以上前に始まっている, …, ならば, 筋収縮性頭痛である」というものである.

すべてのデータから, 300 事例をランダムに選択し, ルール抽出を行い, 残りの約 100 事例に対する正答率を調べた. この実験を 10 回行った結果の平均正答率は 73% であり, この値は医者の知識を用いて作成したエキスパートシステムの正答率と匹敵するものである. したがって, RF2 法を用いれば, 簡潔で汎化したルールの抽出が可能であることを実証できたと考える. さらに, 抽出に要した時間はワークステーションを用いてわずか 2 分程度であり, 実用規模の問題に十分適用可能であることも明らかとなった.

2.7 結言

本章では, 知識獲得ボトルネック解決のための試みとして, 少ない事例からでも簡潔で十分に汎化した分類ルールを抽出する RF2 を考案した. RF2 はルール候補の生成と精錬の 2 フェーズからなり, 正の事例の記述を逐次一般化することによりルール候補を生成し, IDA* と呼ばれる探索手法を用いてそれらを最適なルールの集合に精練する. RF2 を人工問題へ適用した結果, PAC- 学習で必要とされる 4 分の 1 の数の事例から, 十分に正確なルールを抽出できた. また, 最新のルール抽出法である GREEDY3 でも抽出できなかったルールを少ない事例からでも抽出できた. RF2 を医療診断問題へ適用した結果, 未知の事例に対する正答率が医者の知識を用いて作成したエキスパートシステムのものと匹敵した. 抽出した 7 個のルールは, 人間にとっても容易に理解できる程度に簡潔であった. さらに, 抽出に要した時間はワークステーションを用いて 2 分程度であり, 実用規模の問題にも十分適用可能であることが分かった.

第 3 章

ノイズを含む事例からのルール抽出法: RF3

3.1 序言

現実問題におけるルール抽出では, 事例がノイズを含むことを想定しなければならない. したがって, ノイズを許容しつつ, 高い信頼性で良い正答率を保証する抽出法が重要になる.

ノイズを含む事例からのルール抽出 (仮説選択) の理論的な枠組みとしては, 少なくとも, 以下の 3 種の概念学習へのアプローチが知られている.

第 1 に, PAC (Probably Approximately Correct) 学習に基づく方法がある [134]. PAC- 学習では, 事例が一定の確率分布に従って毎回独立に出現すること, 及び, 仮説の属する概念クラスが仮定される. そして, 汎化誤り率とアルゴリズムの失敗確率の各許容限界を設定すれば, 仮定した概念クラスの複雑さと 2 つの許容限界値から, 学習結果を保証するのに必要な訓練事例数を求めることができる. しかし, PAC- 学習の評価尺度は, 最悪ケースの解析に基づいたものであり, 現実的には, 平均ケースの解析の方がより適していると考える. さらに, 一般に, ルール抽出では, 与えられた訓練事例数に依存した評価尺度が望まれるが, PAC- 学習では, 学習に必要な訓練事例数が得られるだけである [13].

第 2 に, MDL (Minimum Description Length) 基準に基づく方法がある [90]. MDL 基準の分類問題への適用では, 仮説の属する概念クラスが仮定され, 仮説と例外事例をコード化する方法が与えられる. 各コードに対しては, コード長が短くなるにつれ大きな確率を割り当てる. そして, 仮説とその仮説の下での例外事例に割り当てた確率の和を求め, その値を最大にする仮説を探索し, 最終結果とする. なお, 仮説と例外事例に割り当てる確率は, Bayesian の立場から, 事後確率を求めるための事前確率と条件付き確率と解釈できる. ところが, MDL 基準では, 各コードへ確率を割り当てるのに必要なパラメータの値を定めねばならない. 一般に, その値の決定は困難な問題として残されている.

第 3 に, 再サンプリング (resampling) 法に基づく bootstrap, cross-validation などの方法がある [20]. 訓練事例を 2 つに分割 (再サンプリング) して, 一方を仮説の選択, 他方を仮説の評価に用いれば, すべての訓練事例を用いて選択する仮説の汎化誤り率を推定できる. しかし, 1 回の実験では, 高い

信頼性で推定できないので，再サンプリングを繰り返して仮説選択を行なうことにより，推定値の信頼性を高めて行く．すなわち，1 つの仮説の評価に，数十から数百回の再サンプリングが必要である [20]．最終的にルール抽出するには，仮説の生成だけでなく，複数の仮説からの仮説選択という負荷の重い処理が必要になる．各仮説に対して，再サンプリングによる評価を行なえば，全体ではかなりの計算量が必要である．

これら 3 つのアプローチに基づいて，ノイズを含む事例からルール抽出するいくつかのアルゴリズムが提案されている [2, 87, 85]．いずれも，理論的な課題の解明，または，決定木 (decision trees) の構築に関するものである．決定木の構築では，プルーニング (pruning) と呼ばれる幾つかの効率的なアルゴリズムが提案されている [85]．しかし，決定木は，前章で述べたように，$x_1x_2 \vee x_3x_4x_5$ のような和結合した概念の表現が困難であり，それ自身の表現能力が不十分である．したがって，より実用的なルール抽出の枠組みを設定し，高い信頼性で良い正答率を保証するルール抽出法の提案が望まれている．

本章では，ノイズを含む事例から，高い信頼性で良い正答率を保証する簡潔な分類ルールの抽出を目標とする RF3 [105] について述べる．まず，ノイズを含む事例からの分類ルール抽出の枠組みについて説明する．次に，複数のルール集合から適当なルール集合を選択するための新評価尺度 MEF を提案し，その位置付けについて述べる．次いで，RF3 アルゴリズムの特徴とその詳細について述べる．最後に，RF3 の実験結果について述べる．

3.2 フレームワーク

本章でも，前章と同様に，ルールを表現する論理式のクラスとして選言標準形を仮定する．概念クラスを仮定すれば，そのクラスに属す任意の概念は，属性を用いた論理式として表現できる．したがって，それら各論理式を予め仮説の集合 $\{f_1, \cdots, f_h\}$ として生成することにより，概念学習は仮説の選択問題として定式化できる．しかし，一般に，生成し得る仮説の個数は莫大であり，到底すべてを実際に数え上げることはできない．以下では，各事例は任意の分布 P_D に従って毎回独立に出現すると仮定する．事例 e_i は $(\mathbf{x}_i, y_i)$ で表せる．ここで，x_i は属性値からなるベクトル，y_i は事例の正負を示すラベルを表す．事例の集合 $\{e_1, \cdots, e_m\}$ を訓練事例と呼ぶ．なお，ノイズを含む事例からのルールの抽出では，各事例の出現頻度が重要な意味を持ち，重複した事例の存在を許容しなければならない．したがって，厳密には，訓練事例の集合は，重複を許した多重集合 (multi-set) として定義される．

3.3 新評価尺度: MEF

ここでは，複数のルール集合から適当なルール集合を選択するための新評価尺度を提案し，その位置付けについて述べる．

3.3.1 MEF の導出

以下の議論で用いるいくつかの表記法を導入する [20]．まず，任意の事例 e_i と仮説 f_n に対して損失関数 L を以下のように定める．

$$L(e_i, f_n) = \begin{cases} 0 & y_i = f_n(\mathbf{x}_i) \\ 1 & \text{otherwise.} \end{cases}$$

訓練事例 $\{e_1, e_2, \cdots, e_m\}$ に対する仮説 f_n の誤り率 err は以下のように表現できる．

$$err(\{e_i\}_{1 \le i \le m}, f_n) = \frac{1}{m} \sum_{i=1}^{m} L(e_i, f_n).$$

今後，この値を訓練誤り率と呼ぶ．確率分布 P_D に従って新たに出現する事例 e に対する仮説 f_n の平均誤り率 Err を以下とする．

$$Err(f_n) = E(L(e, f_n)).$$

今後，この値を汎化誤り率と呼ぶ．

複数のルール集合から適当なルール集合を選択するための *MEF* (*Minimum Expected Failure*) 尺度を導く．ルール抽出では，高い信頼性で汎化誤り率をできる限り小さくすることが望まれる．そこで，これらを定量的に表現するため，汎化誤り率の許容限界を ϵ，実際の汎化誤り率が ϵ 以上となり，アルゴリズムが失敗となる確率を δ とする．望ましい評価尺度の 1 つは，任意の汎化誤り率の許容限界 ϵ に対して，アルゴリズムの失敗確率 δ の期待値を最小にすることである．この期待値を直接計算することは困難であるが，期待値の良い近似としてその上限を次の命題より求めることができる．なお，この結果は Devroye が証明した定理 [17] の 1 つの系として示すことも可能である．

定理 3.1 h を仮説 $\{f_i\}_{1 \le i \le h}$ の個数，m を訓練事例 $\{e_i\}_{1 \le i \le m}$ の個数，ϵ を選択した仮説の汎化誤り率の許容限界，ϵ' を選択した仮説の訓練誤り率の許容限界とする．各事例が任意の分布 P_D に従って毎回独立に出現する場合，もし，選択した仮説 $f \in \{f_i\}$ が，少なくとも，$(1-\epsilon')m$ 個の訓練事例を正答すれば，f の汎化誤り率が ϵ より大となるアルゴリズムの失敗確率 δ の上限は，

$$\delta \le h \exp(-2m(\epsilon' - \epsilon)^2)$$

となる．また，任意の汎化誤り率の許容限界 ϵ に対するアルゴリズムの失敗確率 δ の期待値の上限は，

$$E(\delta) \le \epsilon' + \sqrt{\frac{\log h}{2m}} + \sqrt{\frac{1}{8m \log h}} \tag{3.1}$$

となる．

証明 汎化誤り率が ϵ より大である仮説 f_i が, 少なくとも, $(1-\epsilon')m$ 個の訓練事例を正答する確率は, 二項展開式の $\epsilon' m$ までの和でおさえられる.

$$P(Err(f_i) > \epsilon) \leq \sum_{k=0}^{\epsilon' m} \binom{m}{k} \epsilon^k (1-\epsilon)^{m-k}$$

したがって, 選択した任意の仮説 f に対する失敗確率 ϵ の上限は, Hoeffding の不等式 [36] を用いて求められる.

$$\begin{aligned} \delta &= P(Err(f) > \epsilon) \\ &\leq h \sum_{k=0}^{\epsilon' m} \binom{m}{k} \epsilon^k (1-\epsilon)^{m-k} \\ &\leq h \exp(-2m(\epsilon' - \epsilon)^2) \end{aligned}$$

失敗確率 δ の期待値は, ϵ で定積分することにより求められる.

$$E(\delta) = \int_0^1 P(Err(f) > t) dt$$

したがって, 上記の結果と Gordon の不等式 [32] を用いれば以下の結果を得る.

$$\begin{aligned} E(\delta) &\leq \int_0^u P(Err(f) > t) dt + \int_u^\infty h \exp(-2m(\epsilon' - t)^2) dt \\ &\leq u + \sqrt{\frac{1}{8m \log h}} \end{aligned}$$

ここで,

$$u = \epsilon' + \sqrt{\frac{\log h}{2m}}.$$

□

(3.1) 式で, 第 2 項に対する第 3 項の比を計算すれば $1/2 \log h$ となる. ここで, $h = 100$ としても, $1/2 \log h < 0.11$ なので, 現実的なルール抽出では, 安全に第 3 項を無視できる. したがって, 本稿で扱う MEF 尺度としては以下を用いる.

$$\text{MEF} \stackrel{\text{def}}{=} \epsilon' + \sqrt{\frac{\log h}{2m}} \tag{3.2}$$

MEF 尺度が過剰評価となる可能性について考察する. 例えば,

$$\epsilon' + \sqrt{\frac{\log h}{2m}} \ll \frac{1}{4}$$

の場合, Hoeffding の不等式は $\epsilon < 1/4$ の範囲で, Chernoff の不等式に置き換えることがより厳密な評価となる [26]. また, もし,

$$\epsilon' \gg \sqrt{\frac{\log h}{2m}}$$

である場合には, 0 から

$$\epsilon' - \sqrt{\frac{\log h}{2m}}$$

までの積分値を考慮すべきである. しかし, 現実には, 仮説の数に比して十分に訓練事例が得られないことが一般的であり, 実用的にはこれらのケースを安全に無視できる.

3.3.2 MEF の位置づけ

MEF 尺度の仮定の 1 つは, 機械学習の分野では標準と見なされてきた PAC- 学習と同じである. すなわち, 各事例が任意の一定の分布に従って毎回独立に出現することを仮定している (distribution-free learning). ここで, 既に指摘したように, PAC- 学習の問題点の 1 つは, その評価が最悪ケースの解析に基づくものであり, 十分な数の訓練事例を得られず適用困難な場合が起こることであった. 一方, MEF 尺度では, 任意の汎化誤り率の許容限界に対して, 抽出アルゴリズムが失敗する確率の期待値を最少にすることを試みている. したがって, より現実の問題に適した評価尺度を構築していると考える.

MEF 尺度は, 例外事例の個数とルール集合の複雑さの和を最小化する MDL 基準から見ると, 次のように解釈できる. 訓練事例数 m が決まれば, その値を (3.2) 式に乗ずることにより,

$$m \times \text{MEF} = m\epsilon' + \sqrt{\frac{m \log h}{2}} \tag{3.3}$$

を得る. ここで, 第 1 項は例外事例の個数の上限に他ならず, 第 2 項はルール集合の複雑さに関する項となる. なぜなら, 第 2 項の変数は h だけであり, 仮説の個数が多いということは, 必然的に複雑なルール集合を仮定していることになる.

計算量の観点から MEF 尺度を考えると, 1 つの仮説の評価には, (3.3) 式の計算だけで十分である. したがって, 再サンプリング法と比較すれば, 各仮説の評価に必要な計算はかなり効率的なものである.

3.4 RF3 アルゴリズム

ここでは, RF3 アルゴリズムの説明のための準備をし, 続いてその詳細について述べる.

3.4.1 訓練誤り

ノイズを含む事例からルール抽出を行なう場合, 一般に, すべての訓練事例を正答させることにはあまり意味がない. 抽出アルゴリズムが例外事例をも説明することを試みて, 必要以上に複雑なルール集合を生成してしまうからである. そこで, 抽出アルゴリズムに対して, 訓練誤りを許容するため, 訓練誤り率の許容限界 (MPE: Maximum Permissable Error) を導入する必要がある.

訓練誤りは, FP (false-positive) 誤りと FN (false-negative) 誤りに分類できる [133]. ここで, FP 誤り率とは, 訓練に用いたすべての負の事例のうち, 少なくとも 1 つ以上のルールにカバーされてしまう負の事例の割合である. 一方, FN 誤り率とは, 訓練に用いたすべての正の事例のうち, どのルールにもカバーされない正の事例の割合である. 一般に, ルール集合の複雑さを一定にして, 訓練誤り率を最少にすることを試みれば, この 2 種の誤りはトレードオフの関係にある.

この 2 種の誤りに対しても, それぞれ, FP 許容限界と FN 許容限界が導入できる. ところが, FP 許容限界では, 最終的に抽出されるルールがすべて確定するまで, 実際の FP 誤りを計算できないので, RF3 にとっては扱いにくいものとなる. そこで, 各ルール毎にカバーする負の事例の割合を RFP (rule false-positive) 誤り率として定義し, これに対して RFP 許容限界を導入する. なお, RFP 誤り率の逆数はルールの確信度と考えることもできる.

3.4.2 ルール集合の複雑さ

ルールの集合として DNF を仮定した. しかし, DNF で表現可能なルール集合の個数は莫大であり, その数を仮説の個数として採用することは現実的でない. そこで, 抽出を試みる任意の仮説の複雑さは, 抽出したルール集合のものと同程度以下であると仮定し, アルゴリズムにより抽出したルール集合の複雑さから仮説の個数を推定することを考える. まず, n を属性数とし, k をルール集合に現われる条件 (リテラル) の総数とする. すると, そのルール集合を表現するのに必要なビット数の近似は $k \log_2 n$ となる [77]. したがって, 新評価尺度を求めるのに使われる仮説の個数 h は

$$\log h \quad = \quad k \log_2 n$$

として求めることができる.

3.4.3 RF3 アルゴリズムの詳細

RF3 は, 訓練誤り率の許容限界 MPE を順次変化させ, ルール集合を抽出し, その中で MEF 尺度を最小にするルール集合を最終結果とする. ルール集合の抽出には, 核アルゴリズムとして, 前章で述べた RF2 を用いる. すなわち, 正の事例の記述を逐次一般化することによりルール候補の集合を生成し, IDA* と呼ばれる探索手法 [44] を用いてそれらを最適なルール集合に精練する. 図 3.1 には, RF3 のフローチャートを示す.

訓練事例からのルール候補の生成

第 1 フェーズでは, すべての訓練事例と RFP 許容限界が入力され, ルール候補の集合が出力される. まず, 正の事例を順番に seed として選択する. 次に, ヒューリスティクスを用いて正の事例を順次選択することにより, seed の記述を最小限に一般化する. すなわち, 選択した正の事例をカバーし, カバーしてしまう負の事例の割合が RFP 許容限界以下となるよう一般化する. ただし, 条件を満たす一般化が存在しない場合には, seed の記述はそのままとなる.

ルール候補の最適ルールへの精練

第 2 フェーズでは, 第 1 フェーズで生成したルール候補の集合と FN 許容限界が入力され, 最適に精練されたルール集合が出力される. 第 1 フェーズ完了後, 一般に, ルール候補の集合には冗長なタームやリテラルが存在する. それらを除去することは, カバーしない正の事例の割合を FN 許容限界以下にする SC (set covering) 問題として定式化できる.

パラメータ

RF3 を用いてルール抽出するには, ユーザに入力させる 3 種のパラメータが必要である. まず, 訓練誤りの許容限界 MPE は, 0 を初期値とし, 一定の値を加えることにより変化させる. なぜなら, MPE を入力, MEF を出力と考えれば, RF3 は, 一次元区間を定義域とするシステムの最小値探索となる. したがって, システムの特別な性質が明らかでないので, ユーザの指定した一定の探索幅で入力サンプル点を選び, システムの出力を調べればよいと考える. 一般に, この探索幅がシステムの性能と効率のトレードオフになると考える.

次に, MPE の RFP 許容限界と FN 許容限界への分配比である. 一般に, RFP 誤りと FN 誤りの深刻度は適用問題に依存して異なっている. 例えば, 医療診断問題では, 初期診断で真の患者を病気でないと誤る FN 誤りは, RFP 誤りよりも深刻な問題になると思われる. したがって, この分配比の決定もユーザに任せることが望ましい.

最後に, MPE を変化させる区間の上限である. 一般に, 上限の設定は正負の事例の割合と関係がある. すなわち, この割合が偏っている場合, 新たに出現する事例に対して, 事例の多い方のクラスを常にその分類結果とすれば, 一般に, ある程度の正答率を得ることができる. ここで, 抽出したルールの正答率がそれ以下では意味がない. したがって, 正負の事例の割合から, 訓練誤りの許容限界 MPE を変化させる区間の上限の目安を得ることができると考える. 例えば, 正負の事例の割合が 1:2 の場合

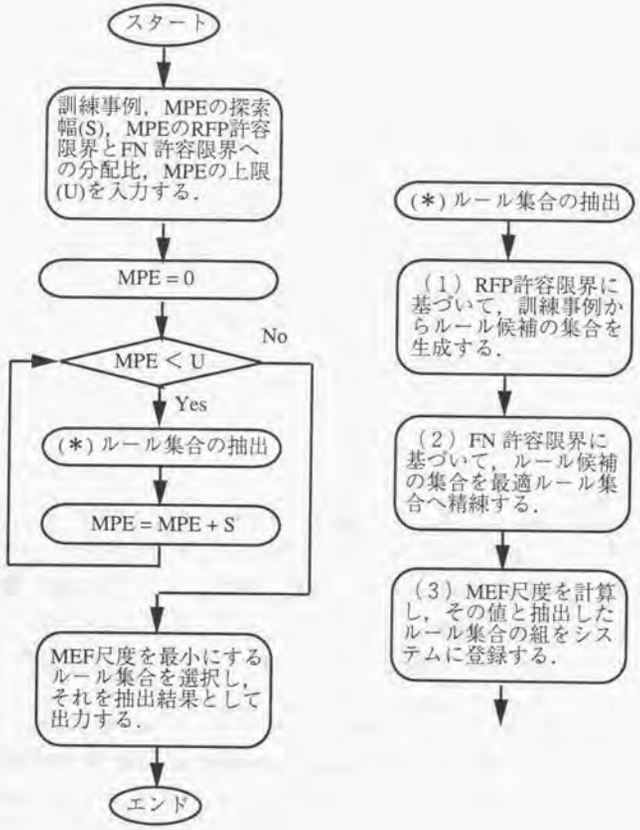


図 3.1: RF3 法のフローチャート

表 3.1: 実験設定の要約

事例を発生させる分布	一様分布		
各属性へのノイズの与え方	5% の割合で属性値を反転		
目標概念	mx6 概念	par4 概念	dnf4 概念
訓練事例数	360	640	1,320
テスト事例数	5,000		
MPE を変化させる区間	0% ～ 49%		
MPE を変化させる定数	1%		
RFP 許容限界と FN 許容限界	1:1 の分配比		

には，上限の目安は 66% となる．

3.5 実験による評価

ここでは，MEF 尺度を導入した RF3 の能力を検証するための実験とその結果について述べる．

3.5.1 実験の設定

表 3.1 に，以下で説明する実験設定の要約を示す．

実験では，前章の実験で用いた概念より mx6, par4, dnf4 を採用した．図 3.2 には，3 つの概念の属性数と完全な DNF 記述を示す．事例は，一様分布に基づき毎回独立に生成した．すなわち，各事例の属性値には，ランダムに 0 または 1 を与え，そして，その属性値ベクトルがそれぞれのルール (DNF 記述) を満足するか否かにより，正または負のラベルを与えた．ただし，ラベルの付与された各事例に対しては，各属性毎に γ% の確率でその値を反転させることにより，ノイズを与えた．したがって，矛盾する事例が出現する可能性もある．ノイズがあまりに大きいとアルゴリズムの優位性が分らなくなるので，ベンチマークの LED (LED display) [11] 等と比較して，γ を半分の 5% に設定した．

各概念における訓練事例数は，PAC-学習の計算式 ((2.1) 式) に基づいて求めた．ここでは，PAC-学習は最悪ケースの解析がベースであることを考慮し，同じ概念に対して行なわれた実験 [77] と比較して，半分の訓練事例からでも有効なルール抽出ができることを確認するため，ϵ を 2 倍の 20% に設定した．

汎化誤り率の推定には，訓練事例とは独立にランダムに生成した 5000 のテスト事例を用いた．但し，テスト事例に対してもノイズを与えた．RF3 のパラメータに関しては，RFP 許容限界と FN 許容

mx6 (16-attribute)

$$\bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_4 \vee x_1\bar{x}_2x_5 \vee x_1x_2x_6$$

par4 (16-attribute)

$$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3\bar{x}_4 \vee$$
$$x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2x_3x_4$$

dnf4 (64-attribute)

$$x_1x_4x_{13}x_{57}\bar{x}_{59} \vee x_{18}\bar{x}_{22}\bar{x}_{24} \vee x_{30}\bar{x}_{46}x_{48}\bar{x}_{58} \vee \bar{x}_9x_{12}\bar{x}_{38}x_{55} \vee \bar{x}_5x_{29}\bar{x}_{48} \vee$$
$$x_{23}x_{33}x_{40}x_{52} \vee x_4\bar{x}_{26}\bar{x}_{38}\bar{x}_{52} \vee x_6x_{11}x_{36}\bar{x}_{53} \vee \bar{x}_6\bar{x}_9\bar{x}_{10}x_{39}\bar{x}_{46} \vee x_3x_4x_{21}\bar{x}_{37}\bar{x}_{55}$$

図 3.2: 目標概念の記述

表 3.2: ルール集合の比較

概念名	元のルール集合		抽出したルール集合	
	汎化誤り (%)	リテラル数	汎化誤り (%)	リテラル数
mx6	9.9	12	9.9	12
par4	16.6	32	18.5	33
dnf4	11.6	41	15.9	30

限界の分配比を等しくし, 訓練誤りの許容限界 MPE を 0% から 49% まで 1% 刻みで変化させた. なお, 実験に用いた概念では, 正負の事例の割合がほぼ等しいので, MPE の上限を 50% とした.

3.5.2 実験結果

表 3.2 には, 3 つの概念に対して, 元のルール集合と RF3 が抽出したルール集合における汎化誤り率とルール集合のサイズの比較を示す. ここで, 元のルール集合 とは, 図 3.2 に示した元々の DNF 記述のことである. RF3 が抽出したルール集合は, MEF 尺度を最少にするものであり, 表からも分るように, 最適なルール集合とほぼ同等なものと考える.

図 3.3 から図 3.5 には, 訓練誤り率の許容限界 MPE (x 軸) を順次変化させ, MEF 尺度 (MEF), 汎化誤り率 (Err) がそれぞれどのように変化したかを示す. なお, y 軸には, アルゴリズムの失敗確率の期待値 (MEF 尺度) と汎化誤り率の 2 つの意味が当てられている.

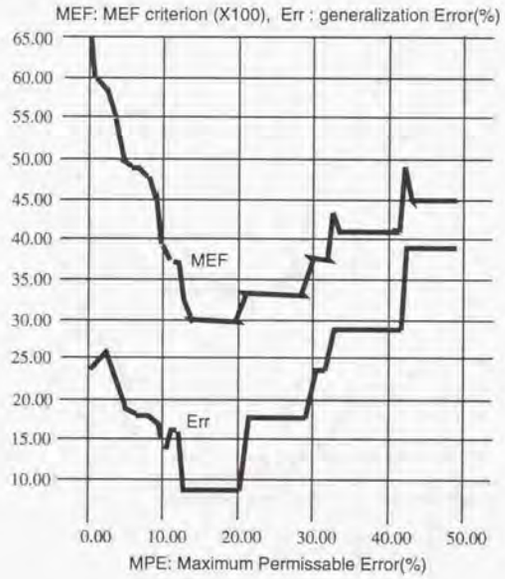


図 3.3: mx6 概念に対する MEF 曲線と Err 曲線

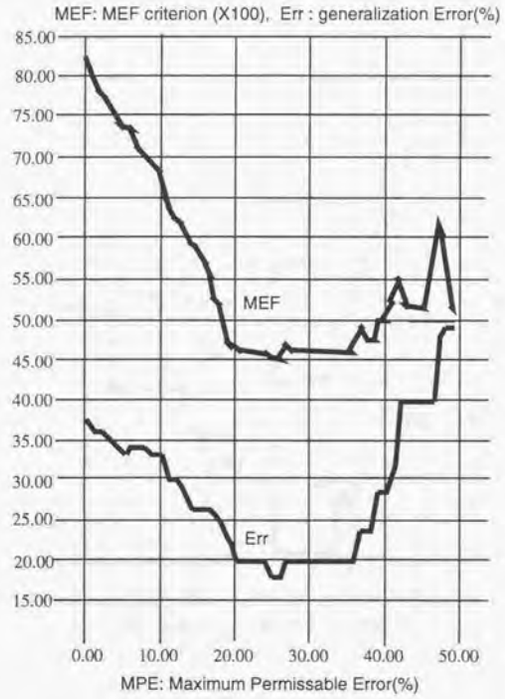


図 3.4: par4 概念に対する MEF 曲線と Err 曲線

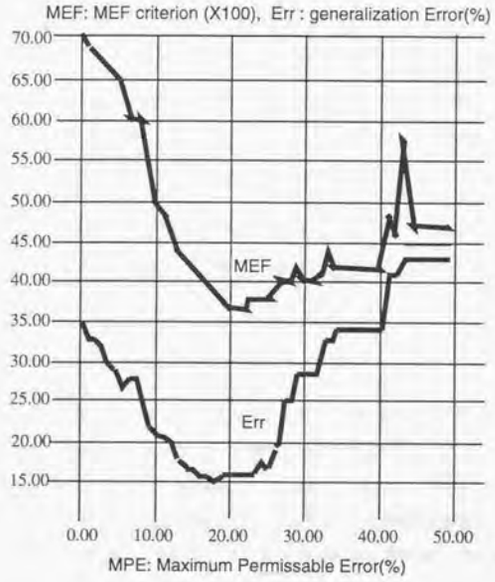


図 3.5: dnf4 概念に対する MEF 曲線と Err 曲線

mx6 概念

図 3.3 には, 360 事例から mx6 概念のルールを抽出したときの曲線を示す. この概念では, RF3 は元のルール集合を完全に復元した. ノイズを含む事例からの抽出であることを考えると, このことは注目できる結果である. さらに, 汎化誤り率が最少になる部分は, MEF 尺度が最少になる場所と幅広く一致し, その範囲では, すべて元のルール集合を完全に復元していた. ここで, 同程度の複雑さを持つ他の概念 (ルール集合) と比較して, mx6 概念には, 抽出が容易にできる特別な理由はないと考える. むしろ, 多くの属性選択方式のアルゴリズムを用いた場合には, 有効にヒューリスティック評価関数が働かず, 抽出が困難となる [77]. したがって, RF3 を用いれば, mx6 概念と同程度の複雑さを持つシンプルなルールの場合には, ノイズを含む事例からでもかなり良いルールの抽出が期待できる.

par4 概念

図 3.4 には, 640 事例から par4 概念のルールを抽出したときの曲線を示す. この概念では, 1 つのタームに 1 つの余計なリテラルが現われたことを除き, 元のルール集合を復元できた. そして, 汎化誤り率が最少になる点は, まさに, MEF 尺度が最少になる点と一致した. なお, パリティ概念に関しても, 多くの属性選択法のアルゴリズムによる抽出が困難であることを考えると, RF3 の能力は注目に値する.

dnf4 概念

図 3.5 には, 1320 事例から dnf4 概念のルールを抽出したときの曲線を示す. この概念では, 元のルール集合より, かなりシンプルなルール集合を抽出した. 図 3.6 には, 実際に抽出したルール集合を示す. 元のルール集合と比較すれば, 5 つのリテラルからなるターム (ルール) を抽出できなかった. すなわち, 2 つのタームは完全に無視され, もう 1 つのタームでは, 1 つのリテラルが欠落してしまった. ここで, 実験では, 事例を一様分布に基づいて生成しているので, 一般に, リテラルが 1 つ増えればカバーする事例は半分となる. したがって, カバーする事例数が少ないターム (small disjuncts) を抽出できなかったことになる. ところが, small disjuncts は, 一般に汎化誤りを大きくする問題の 1 つとして指摘されている [38]. したがって, 元のルールを完全に抽出できない場合には, small disjuncts を無視することは有効な 1 つの戦略なので, MEF 尺度は望ましい性質を備えていると考える. また, dnf4 概念でも, 汎化誤り率と MEF 尺度が最少になる点は, ほぼ一致した.

3.6 結言

ノイズを含む事例から, 高い信頼性で良い正答率を保証する簡潔な分類ルールの抽出を目標とする RF3 の提案を行った. RF3 の特長は, ルール集合選択のための新評価尺度 MEF の導入にある. MEF

$$x_{18}\bar{x}_{22}\bar{x}_{24} \vee x_{30}\bar{x}_{46}x_{48}\bar{x}_{58} \vee \bar{x}_{9}x_{12}\bar{x}_{38}x_{55} \vee \bar{x}_{5}x_{29}\bar{x}_{48} \vee$$
$$x_{23}x_{33}x_{40}x_{52} \vee x_{4}\bar{x}_{26}\bar{x}_{38}\bar{x}_{52} \vee x_{6}x_{11}x_{36}\bar{x}_{53} \vee \bar{x}_{6}\bar{x}_{9}x_{39}\bar{x}_{46}$$

図 3.6: dnf4 概念に対して抽出したルール集合

尺度では, 任意の汎化誤り率の許容限界に対し, アルゴリズムの失敗確率の期待値が近似的に最小化される. また, MEF 尺度は, 抽出したルール集合の複雑さと例外事例の個数の和を最小化する尺度としても解釈できた. 実験の範囲では, RF3 を用いて, 元のルール集合とほぼ同等なルール集合を抽出できた. また, 訓練誤り許容限界を変化させ, MEF 尺度と汎化誤り率を比較したところ, 両者の傾向が酷似することが分かった. したがって, MEF 尺度を用いることにより, 高い信頼性で汎化誤り率を最小にするルール集合の選択が可能になると期待できる.

第 4 章

適応概念学習法: RF4

4.1 序言

知能の基本的な特徴の 1 つは, 与えられた事例 (問題) 集合から, 実質的に新たな知識を学習 (発見) できることである. その知識には, 新たな事例を適切に識別するための概念や, 問題を高速に解決するための手続きなどがある.

概念学習は, 与えられた論理空間において, 適切な論理 (識別) 式を探索する問題として定式化できる [67]. 代表的な概念学習アルゴリズム [58, 86, 75] では, 正答率や簡潔性などに基づいた 1 つのヒューリスティック評価関数に基づいて, 評価値の高い論理式を選択し, それらの組合せにより概念を探索する. しかし, この戦略では, 効率は良いが, 評価値が十分に高くない論理式の組合せからな識別概念を見逃す危険がある. 一般に, このような状況は少なからず起こるので, ある論理式を枝刈る積極的な根拠がなければ, その論理式を概念の探索に用いるべきである. よって, 多数の枝刈り尺度を採用した深さ優先探索は, 概念学習において有望なアプローチであると考える.

多くの概念学習アルゴリズムでは, 同じ問題が繰り返して与えられても, 常に同じ処理ステップを実行し, 過去の問題解決の経験に基づいて, 概念学習効率を改善することはない. 問題解決 (概念学習) の高速化には, 問題解決に有効なルールの組合せであるマクロオペレータ (チャンク) を学習する EBL [68, 47] アプローチが広く研究されているが, 1 つの事例から作成したマクロオペレータの一般的な有用性に関する utility 問題 [65] が課題として指摘されている. ここでは, 問題集合からヒューリスティック評価関数を学習するアプローチ [124] を採用する. この技術は, 主として, ゲーム探索の分野だけで利用されているので, より広い応用においての検討が望まれる.

本章では, 過去に解いた問題を用いて適応的に探索効率を改善する概念学習法 RF4 [107, 97, 96, 113] について述べる . まず, 概念が一回述語論理で記述されるときの学習の枠組みについて説明する. 次に, RF4 アルゴリズムの詳細について述べる. 最後に, チェスの終盤戦問題, および, ボンガルド問題を用いて, RF4 の能力を評価する.

4.2 フレームワーク

事例を識別する概念は, 論理式で表現されるとする. 各事例は, 未知の概念 (論理式) によりクラス 1 (正の事例) かクラス 2 (負の事例) に分類されているとする. 概念学習とは, 分類された事例集合から, その識別概念を求める問題である.

論理式の表現には, 記述力と理解容易性を考慮して, 集約関数 (aggregate function) を含む一階述語論理 (first-order predicate calculus) を採用する. 以下, 論理式の再帰的な定義を示す. ターム (term) は, 定数 / 変数であるか, または, 論理式に集約関数 (count, sum, average, max, min) を施したものである. 原子式 (atomic formula) は, 2 つのタームに比較演算子 ($>, \geq, <, \leq$) を施したものである. 論理式は, 原子式であるか, または, 論理式に論理結合 ($\wedge, \vee$) や限量子 ($\forall, \exists$) を施したものである.

4.3 RF4 アルゴリズム

RF4 (*Rule extraction from Facts version 4*) は, 望ましくない論理式を枝刈りする学習バイアス, および, 存在限量子や集約関数を含む論理式を生成する複合化ルールを用いて, 事例の識別概念を深さ優先で探索する. 論理式の探索順序は, 過去の経験に基づいて, 原始式が識別概念の構成要素となる確率を推定することにより, 動的に決定される.

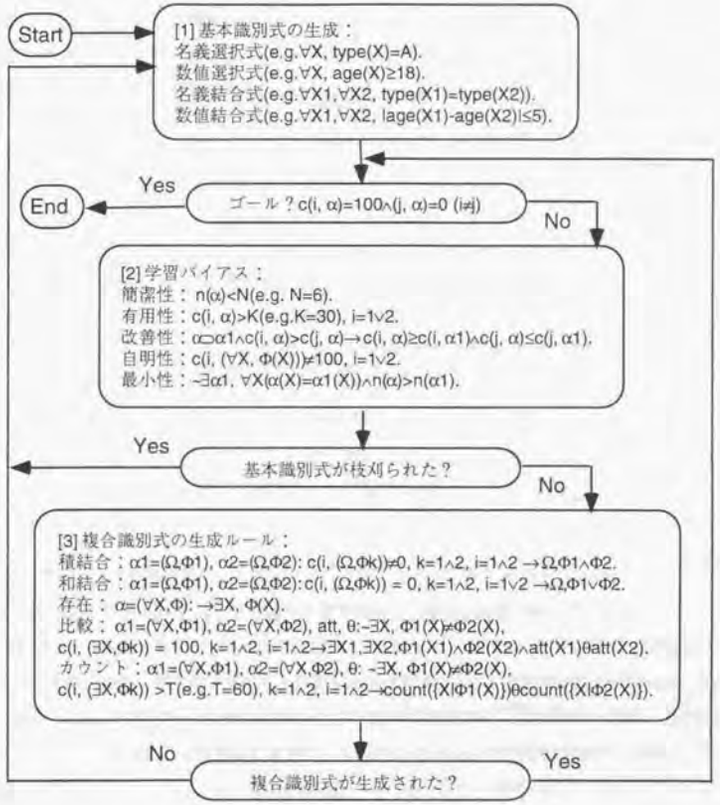
4.3.1 核機能

基本フロー

図 4.1 に, RF4 のフローチャートを示す. 基本フローは, 以下の処理を繰り返す. まず, 入力記述より基本識別式を生成する. もし, その識別式がゴール条件を満たさず, 学習バイアスにより枝刈られるならば, 別の基本識別式を生成する. さもなければ, その識別式を格納するとともに, 既に格納されている識別式を用いて複合識別式を生成する. ここで, 新たに生成された識別式が優先して処理される. もし, 生成した識別式がゴール条件を満たさず, 学習バイアスにより枝刈られるならば, 別の複合識別式を生成する. もし, 新たな複合識別式を生成できなければ, 別の基本識別式を生成する. また, 新たな基本識別式も生成できなければ, 解答不能としてアルゴリズムが終了する. ここで, カバー率 $c(i, \alpha)$ は, クラス i で論理式 α にカバーされる事例数とクラス i の事例数の比である.

基本識別式の生成

前項で述べたように, 事例から直接生成される最も単純な識別式として, 4 種の基本識別式を生成する. 図 4.1-[1] に, 各識別式の例を示す. 名義属性からは, 属性とその値のペアより, 名義選択式を生成する. 数値属性に対しては, 簡単なクラスタリングを施す. すなわち, まず, 数値をソートして昇順に配



注) 変数 X はすべて ∀F∈facts, X∈F で規定；θ：比較演算子；
F,X：変数；Ω：限量子；Φ：条件；α：論理式；n(α)：原子式数；c(i,α)：
カバー率；c(i,k), k=1∧2, i=1∨2≡(c(1,1)∧c(1,2))∨(c(2,1)∧c(2,2)) .

図 4.1: RF4 の基本フロー

置し, 隣合う値のギャップが最大となる箇所で "以上", "以下" の 2 つの識別式を生成する. 例えば, 属性値が $att(X) = \{8, 9, 10, 12, 14, 18, 19, 21, 23, 25\}$ であるとき, 結果の 2 つの条件は $att(X) \leq 16$, $att(X) > 16$ となる. また, この手続きは, さらに別の条件を得るため, 分割した値の集合に対して再帰的に適用できる. RF4 では, 最も大きい 2 つのギャップを選択し, 条件を生成する.

複数のオブジェクトがあるときには, 2 つのオブジェクトの属性値を比較するケースが考えられる. 名義属性からは, 存在限量子を用いて, 名義結合式を生成する. 数値属性に対しては, 2 つの値の差にクラスタリングを施して, 数値結合式を生成する.

学習バイアス

新たに生成される識別式に対して, 5 種の学習バイアス (簡潔性, 有用性, 改善性, 自明性, 最小性) を用いて, 不要な識別式を枝刈りする. 図 4.1-[2] に, 各バイアスの定義を示す. これらは, 複合式に含まれる原子式数が一定値 (e.g. $N = 5$) を越えないこと, 識別式のカバー率が一定値 (e.g. $K = 30\%$) を越えること, 複合化前の識別式より複合化後の識別式のカバー率が改善されること, 識別式が全てのオブジェクトをカバーしないこと, および, 同じ内容を複雑な識別式で表現しないことを要求する. なお, パラメータ値の設定については, ユーザが許容する識別概念の複雑さを N とし, N 個の原子式から成る識別式を考慮して, 1 個あたり少なくとも $K \approx 100/N$ を目安とする. また, 一般に, N が大きく K が小さければ, 複雑な識別式の生成も可能になるが, 多くの学習時間が必要になる.

複合識別式の生成

図 4.1-[3] に, 各複合化ルールの定義を示す. 識別式が 2 つのクラスのオブジェクトをカバーするときには, 論理積結合を用いて, 識別式を特殊化する. 逆に, 識別式が特殊すぎるときには, 論理和結合を用いて, 識別式を一般化する. また, 全称限量子を存在限量子に置き換えて, 識別式の条件を弱めるルールもある. 図形オブジェクトが複数ある場合, 適当な条件の下で, 2 つのオブジェクトの属性を比較できる. さらに, 集約関数の導入により, 数に関してより豊富な識別概念が生成可能となる. RF4 では, count だけを扱うが, 他の関数の実現は簡単な拡張で行なえる.

4.3.2 適応機能

識別概念は原始式の組合わせであり, 探索の効率の点では, どのような順番で原始式の組合わせを探索するかが, 非常に重要な問題となる. その順番は問題ごとに異なると考え, 問題の特徴を表現する問題特徴 $\mathbf{f}$ を導入する. また, その順番はこれまでの探索でどの原始式を使って来たかにも依存するので, それを探索状態 $\mathbf{r}$ として管理する. 詳細には, $\mathbf{f}$ の各要素は, その問題において, 予め定義したある論理式が真となるかどうかを示すブール値であり, 一方, $\mathbf{r}$ の各要素は, 1 対 1 で 1 つの原始式に対応し, 現時点での探索に利用されているかどうかを示すブール値である. 以下では, $\mathbf{f}$ の要素と $\mathbf{r}$ の要素をそのまま並べた 1 つのベクトルを考え, 状態ベクトル $\mathbf{s}$ と呼ぶ.

ある状況 $\mathbf{s}$ において, 現時点の探索で利用されていない各原始式が識別概念の構成要素となる条件付き確率が分るならば, その確率が最大となる原始式を次に用いて探索を実行することを繰返すことにより, 探索の期待コストを最小にすることができる. RF4 では, 以下で述べるように, 過去の問題解決の経験に基づいて, その近似確率を推定する.

$P(\mathbf{s})$ の 2 次近似

$\mathbf{s} = (s_1, \cdots, s_d)$ を状況ベクトル, 確率 $P(\mathbf{s})$ を状況ベクトル $\mathbf{s}$ の生起確率とする. まず, Bahadur-Lazarsfeld 展開 [18] の 1 次項だけで $P(\mathbf{s})$ を近似すれば,

$$P_1(\mathbf{s}) = \prod_{i=1}^{d+1} p_i^{x_i}(1-p_i)^{1-x_i} \tag{4.1}$$

となる. ただし, $p_i = P(s_i = 1)$ である. この近似は s_i の値が互いに独立に定まるときに他ならない. 次に, 2 次項まで用いて $P(\mathbf{s})$ を近似すれば,

$$P_2(\mathbf{s}) = P_1(\mathbf{s})\left(1 + \sum_{i=2}^{d+1}\sum_{j=1}^{i-1}(p_{ij} - p_i p_j)y_i y_j\right) \tag{4.2}$$

となる. ただし, $p_{ij} = P(s_i = 1, s_j = 1)$, $y_i = (s_i - p_i)/(p_i(1-p_i))$ である. $P(\mathbf{s})$ を k 次項まで展開すれば, 推定すべき確率の数は $O(n^k)$ となり, 一般には多くの事例を得られないので, それらを高い信頼性で推定できなくなる. 一方, 1 次近似を採用すれば, 出現した事例の確率を十分に反映できない場合がある. 例えば, 表 4.1 に示すように, 2 つの 3 次元ベクトルが現れた後, 1 次近似を採用すれば, すべてのベクトルは同じ確率で出現すると推定されるが, 2 次近似を採用すれば, 出現した 2 つのベクトルは他よりも高い確率で出現すると推定される. つまり, RF4 では, 推定確率の大きい順に探索が行われるので, 1 次近似を採用すれば, その探索はランダムになるが, 2 次近似を採用すれば, 確率の大きいものから探索できる. よって, RF4 では 2 次近似を採用する.

p_i, p_{ij} の推定

2 次近似の要素確率 p_i, p_{ij} の値は, 最尤推定 [18] に基づき, 1 つの問題解決ごとに再推定される. つまり, これらの値は, 状況ベクトル $\mathbf{s} = (\mathbf{f}, \mathbf{r})$ 要素の値が真となる問題数とこれまでに解いた問題数との比である. ここで, 問題特徴 $\mathbf{f}$ の各要素が真となるのは, 対応する論理式が少なくとも一方のクラスの全事例をカバーするときとし, さもなければ, 偽とする. 一方, 探索状態 $\mathbf{r}$ の各要素が真となるのは, 対応する原始式が識別概念の構成要素となるときとし, さもなければ, 偽とする.

表 4.1: 観測事例と推定確率

観測事例	推定確率					
(s_1,s_2,s_3)	(s_1,s_2,s_3)	$P_1(\mathbf{s})$	$P_2(\mathbf{s})$	(s_1,s_2,s_3)	$P_1(\mathbf{s})$	$P_2(\mathbf{s})$
	(0, 0, 0)	1/8	0	(0, 1, 0)	1/8	1/2
(1, 0, 1)	(0, 0, 1)	1/8	0	(0, 1, 1)	1/8	0
(0, 1, 0)	(1, 0, 0)	1/8	0	(1, 1, 0)	1/8	0
	(1, 0, 1)	1/8	1/2	(1, 1, 1)	1/8	0

確率推定

現時点で, 探索に利用していない原始式に対応する探索状態要素を s_k とすれば, 値が既知の状況ベクトル要素群 $\{s_1,\cdots,s_h\}$ に対して, s_k が真となる (識別概念の構成要素となる) 条件付き確率は

$$P_2(s_k=1 \mid s_1,\cdots,s_h) = p_k + \frac{\alpha^{(h)}}{\beta^{(h)}} \tag{4.3}$$

となる. ただし,

$$\alpha^{(h)} = \sum_{i=1}^{h}(p_{ik}-p_ip_k)y_i, \tag{4.4}$$

$$\beta^{(h)} = 1+\sum_{i=2}^{h}\sum_{j=1}^{i-1}(p_{ij}-p_ip_j)y_iy_j \tag{4.5}$$

である. 以下 (4.3) 式の導出について述べる. 近似確率の定義 (4.1) 式と (4.2) 式, および, $\beta(h)$ の定義 (4.5) 式を考慮すれば,

$$P_2(s_1,\cdots,s_h) = P_1(s_1,\cdots,s_h)\beta^{(h)}$$

であることを確認する. ここで,

$$P_2(s_k,s_1,\cdots,s_h) = p_k^{s_k}(1-p_k)^{(1-s_k)}P_1(s_1,\cdots,s_h)\left(\beta^{(h)}+\sum_{j=1}^{h}(p_{kj}-p_kp_j)y_jy_k\right)$$

であり, $s_k=1$ のとき, $y_k=1/p_k$ となるので,

$$P_2(s_k=1,s_1,\cdots,s_h) = p_kP_1(s_1,\cdots,s_h)\beta^{(h)}+P_1(s_1,\cdots,s_h)\sum_{j=1}^{h}(p_{kj}-p_kp_j)y_j$$

である. したがって, $\alpha^{(h)}$ の定義 (4.4) 式を考慮すれば,

$$P_2(s_k=1 \mid s_1,\cdots,s_h) = \frac{P_2(s_k=1,s_1,\cdots,s_h)}{P_2(s_1,\cdots,s_h)}$$

$$= p_k+\frac{\sum_{j=1}^{h}(p_{kj}-p_kp_j)y_j}{\beta^{(h)}}$$

$$= p_k+\frac{\alpha^{(h)}}{\beta^{(h)}}$$

となり, (4.3) 式が導出できる. RF4 では, この条件付き確率を最大にする原始式を次に用いて探索を実行する. なお, (4.3) 式の右辺は, s_k に対応する原始式が識別概念の構成要素となる頻度 p_k と 2 次近似に基づく補正値 $\alpha^{(h)}/\beta^{(h)}$ を加えた値と解釈できる.

RF4 では, 探索の過程で, 問題特徴要素の真理値を求める. すなわち, 探索を開始する前には, 問題特徴要素は未知であり, 探索が進むにつれて, それらの値を次第に明らかにする. また, 探索状態要素の値も探索の過程で変化するので, RF4 では, 条件付き確率を再帰的に求める計算法を採用する. すなわち, 状況ベクトル要素群 $\{s_1,\cdots,s_{h-1}\}$ の値を既知とし, さらなる探索での状況ベクトル要素 s_h の値が確定したとすれば, $\alpha^{(h)}$, $\beta^{(h)}$ の値は, 次の漸化式を用いて効率良く計算できる.

$$\alpha^{(h)} = \alpha^{(h-1)}+(p_{h(d+1)}-p_hp_{d+1})y_h,$$

$$\beta^{(h)} = \beta^{(h-1)}+\alpha^{(h-1)}y_h.$$

ただし, 初期値は $\alpha^{(0)}=0$, $\beta^{(0)}=1$ である.

4.4 チェス終盤戦への適用

KRK 問題 [74] と呼ばれるチェス終盤戦の概念学習問題を用いて, RF4 の性能を評価した. KRK 問題とは, 白の King と Rook 対黒の King の戦いであり, 与えられた局面において, 黒の King が白のどちらかの駒で直接攻撃されるか判定する概念を求める問題である. 可能な駒の配置は, $64\times63\times62$ であり, このうち約 1/3 がこの概念を満たす配置となる. 実験での問題表現には, 行 / 列について, 各駒のペアの位置が等しい / 隣接 / 小さいかを示す 6 述語を利用した. ただし, KRK 問題には 1 つの概念しか現れないので, 問題特徴 f は考えなかった.

実験では, ランダムに 50 事例を選択し, その識別概念を学習することにより, 確率 p_i, p_{ij} を推定した. また, この学習を続けて 15 回繰り返すことにより, 各段階において, 探索した論理式数, および, 訓練事例とは独立な 500 事例に対する正答率を評価した. 図 4.2 に, これらの一連の試行を 100 回繰り返した平均値を示す. RF4 を用いて学習を繰り返せば, 探索効率が改善されるだけでなく, 未知の事例に対する正答率も向上する傾向にあることが, KRK 問題において確認できた.

4.5 ボンガルド問題への適用

ここでは, ボンガルド問題を用いて, RF4 の性能を評価する.

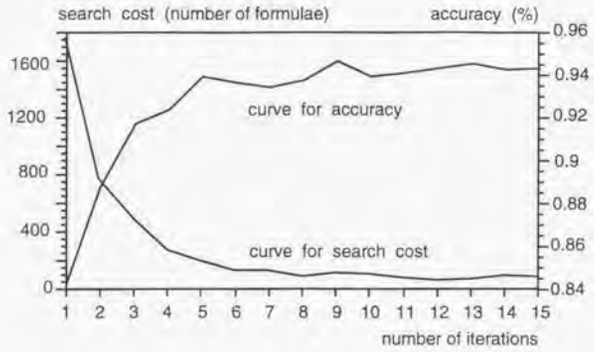


図 4.2: KRK 問題での学習曲線

4.5.1 ボンガルド問題

ボンガルド問題 [10, 37] とは, M. Bongard が考案した図形に関する多彩な識別概念を如何に発見するかという問題である. そこには, さまざまな識別問題の型が存在し, 識別の困難さも多様なレベルで現われる [61]. したがって, 機械学習の研究において, 興味深く有用なテストケースであると考えられる. しかし, その解決に向けた試みはほとんどなされていない. ボンガルド問題の例を図 4.3 に示す. 各問題は, 図形オブジェクトが描かれた 12 個のボックス (事例) からなり, 左側の 6 個のボックスはクラス 1 に属し, 右側の 6 個はクラス 2 に属すとする. 問題はクラス 1 とクラス 2 の識別概念を見つけることである. なお, ボンガルド問題は, 例えば, 概念学習アルゴリズムの評価に広く利用されている "東行き / 西行き列車の識別問題" と基本構造が同じであり, 同様な現実問題は少なくないと考える.

入力インタフェース

各問題は, 図形オブジェクトの作図過程から得られる情報を利用して表現する. 例として, 黒い楕円を描く場合を考える. まず, アイテム "oval" を図形オブジェクトの "shape" メニューから選択し, 次に, 楕円の左上と右下の座標をマウスで指定し, そして, 値 "black" を "texture" メニューから選択する. このように設定された値を用いれば, 簡潔な記述が得られる. ここでは, オリジナル属性 (shape name, texture name, shade direction, rotation angle, line width) の値はユーザが直接設定し, 派生属性 (size, convexity, relation, number of angles, roundness, aspect ratio, gravity) の値はオリジナル属性値から計算される.

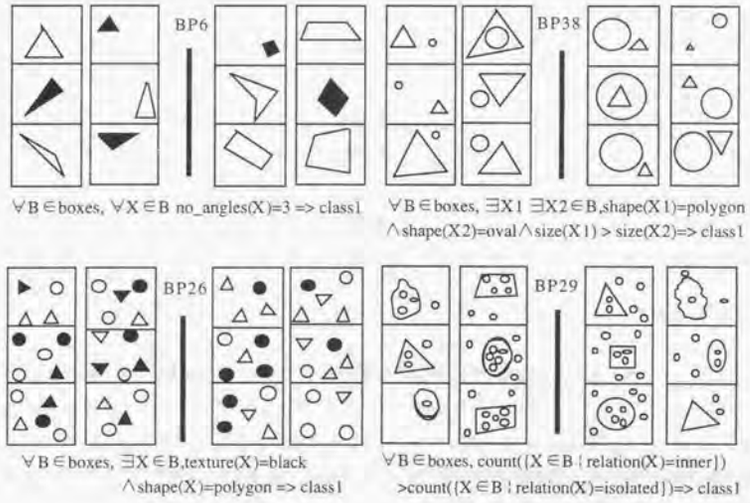


図 4.3: ボンガルド問題の例

表 4.2: 各アルゴリズムの主な特徴

手法	探索法	枝刈り尺度	生成概念の特徴
INDUCE	ビーム	正答率 & 簡潔性	count(limited)
GOLEM	ランダム	determinate literals	再帰
FOIL	欲張り	情報量の期待値	再帰
RF4	深さ優先	5 種の学習バイアス	比較 & count

4.5.2 核機能の評価

RF4 の結果

パソコン上に C で実現した RF4 のプログラムは，100 のボンガルド問題のうち 41 問に対して，各々数秒で正答できた．正答できた問題のタイプは，概念の生成に必要な操作で分類できる．すなわち，単一の図形オブジェクトの条件に関する問題 (Single)，ある条件を満たす図形オブジェクトの存在に関する問題 (Existance)，図形オブジェクト間の属性比較をする問題 (Comparison)，図形オブジェクトを数える問題 (Counting) である．図 4.3 に，各問題タイプの典型例と RF4 の出力結果を示す．

既存法との比較

RF4 の能力を評価するため，代表的な概念学習アルゴリズムである GOLEM [75]，INDUCE [58]，FOIL [86] による，ボンガルド問題解決を試みた．ここで，実験に用いたプログラムのバージョンは，"Golem alpha version"，"induce 3 - version as of feb 10 1984"，"FOIL.2" である．なお，今回の実験では，各プログラムにはその初期設定のパラメータを用いた．表 4.2 に，各アルゴリズムの主な特徴を示す．

各アルゴリズムを同一条件で比較するため，既存法には，RF4 が生成する基本識別式を問題表現として与えた．例えば，RF4 と同じクラスタリング手法を用いて，2 つの図形オブジェクトのサイズがほとんど等しいときには，$eqsize(X_1, X_2)$ のような述語を与えた．

実験では，既存法が正答できた問題は，いずれも RF4 で正答していた．図 4.4 に，問題タイプ別の各アルゴリズムの正答率を示す．以下に，既存法が正答できなかった主な理由を考察する．

- 各ボックスに 2 つ以上の図形オブジェクトがある場合，GOLEM はどの問題も正答できなかった．なぜなら，GOLEM において本質である "determinate literal" と呼ばれるヒューリスティックス (新リテラルの変数の値は，ホーン節で既に出現した変数の値を与えることにより，一意に決定されねばならない) を採用しているためである．すなわち，ボンガルド問題では，任意の解の形式は，$class1(B) :- contain(B, X), \cdots$ と表現される．しかし，2 つ以上の図形オブジェクトがある場合，変数 B の値を与えても，変数 X の値は一意には決定されない．

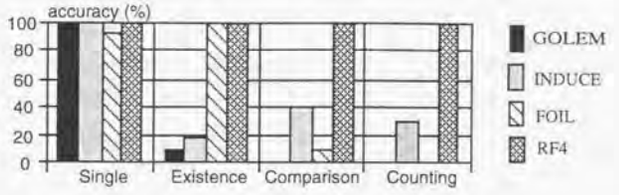


図 4.4: 問題タイプ別の正答率

- FOIL は比較型をほとんど正答できず，INDUCE はこの型の半分以上の問題を正答できなかった．なぜなら，両者はヒューリスティック探索 (前者は欲張り探索，後者はビーム探索) を行なうためである．例として，FOIL による，図 4.3 に示した BP38 の問題解決について考える．正答の概念は，$class1(B) :- contain(B, X_1)$, $contain(B, X_2)$, $polygon(X_1)$, $oval(X_2)$, $gtsize(X_1, X_2)$ と表現できる．FOIL では，事例のクラス分布に関する情報量を評価尺度とするので，$polygon(X_1)$ はすべての事例に現れるため，情報量は 0 (分類に関する情報量は増えない) となり，それを探索に用いることができない．なお，探索法を変えても，FOIL の評価尺度に基づけば，情報量は 0 なので，識別概念を求めるには，ほとんど全探索をすることになり，現実的ではない．

- count 型では，GOLEM と FOIL はどの問題も正答できず，INDUCE は一部の問題しか正答できなかった．なぜなら，FOIL と GOLEM にはオブジェクトの count 機能がなく，一方，INDUCE は count できるが，count した値を比較する機能がないためである．ここで，他のアルゴリズムに対して，単純に count 機能を追加しても，組合せの数が極端に増大し，妥当な時間で問題を解くことができなくなると考えられる．

4.5.3 適応機能の評価

RF4 の結果

RF4 の適応機能がどれだけ概念学習効率を改善したかを評価するため，原子式の利用頻度だけに基づいて探索順序を決定する方法との比較を行なった．ここで，問題特徴 $\mathbf{f}$ は，名義属性からなる原子式に存在限量子を施した論理式の集合とした．実験では，訓練問題は，RF4 が解決した 41 のボンガルド

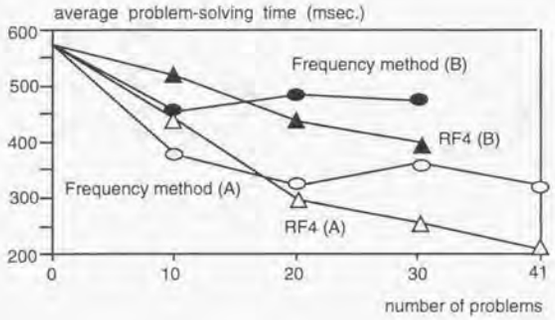


図 4.5: ボンガルド問題での学習曲線

問題からランダムに選択し，テスト問題には，41 問全てをテスト問題とするタイプ (Test A)，および，訓練に利用しなかった残りの問題をテスト問題とするタイプ (Test B) を用いた．また，訓練問題数は 10 単位で増やし，これらの試行をそれぞれ 40 回繰り返して評価した．

図 4.5 に，テスト問題を解くのに要した平均 CPU 時間を示す．ただし，2 つの方法には，同じ順番で訓練問題を与えた．以下に，実験結果を考察する．

- RF4 では，41 の訓練問題を解いた後，同じ訓練問題を解くのに要した平均時間は 1/3 に減少した．また，30 の訓練問題を解いた後，残りのテスト問題を解くのに要した平均時間は 2/3 に減少した．ゆえに，RF4 の適応機能はうまく働いたと考える．
- 訓練問題数が少い段階では，頻度に基づく方法の問題解決効率は RF4 の効率よりも優れていた．その理由は，RF4 には，確率計算と原子式選択の 2 つの付加タスクが必要であり，もし，推定確率の信頼性があまり高くなければ，これらのタスクが RF4 への単なるオーバーヘッドになるためと考える．
- 訓練問題数が増えても，頻度に基づく方法の問題解決効率は，早い段階で改善されなかった．その理由は，頻度に基づく方法が，RF4 と比較して，かなり速い段階でその限界に達したためと考える．

知識の検証

RF4 の適応機能が探索効率を改善したのは，推定確率を利用したからであるが，確率は多数の数値集合により表現されるので，実際に獲得した知識が何を意味するかを明確に知ることは困難である．ここでは，χ^2- 検定を用いて，一部の顕著な知識 (予め定義した状況ベクトルと識別概念の因果関係) の抽出を試みる．

状況ベクトルのある要素が真となる事象を e_1，ある属性が識別概念の構成要素となる事象を e_2，2 つの事象 e_1, e_2 が同時に起こる事象を e_{12} とする．また，推定に利用した事例数を m，3 つの事象 e_1, e_2, e_{12} が起こる確率の推定値をそれぞれ $\hat{p_1}$, $\hat{p_2}$, $\hat{p_{12}}$ とする．もし，2 つの事象 e_1, e_2 が独立ならば，

$$\chi^2 = m\frac{(\hat{p_{12}} - \hat{p_1}\hat{p_2})^2}{\hat{p_1}(1-\hat{p_1})\hat{p_2}(1-\hat{p_2})}$$

は自由度 1 の χ^2- 分布に従う．もし，χ^2 の値がある値より大きければ，指定した信頼率で，2 つの事象 e_1, e_2 が独立であるという仮説を棄却できる．実験では，信頼率 95% ($\chi^2 \leq 3.84$) で棄却できる仮説を求めた．

得られた結果をルール形式で表せば，「*oval* が存在するならば，図形オブジェクトの内部 / 外部の関係を調べろ ($\chi^2 = 5.45$)」，「*polygon* が存在するならば，図形オブジェクトの角数を調べろ ($\chi^2 = 4.97$)」，および，「*rectangle* が存在するならば，図形オブジェクトの内部 / 外部の関係を調べろ ($\chi^2 = 4.14$)」である．第 2 のルールについては，図形オブジェクトが *polygon* であれば，その角数を比較するという，人間の直感的な知識と符合する．一方，第 1 と第 3 のルールについては，*oval* や *rectangle* は，他の図形オブジェクトの内部に描かれる小さな図形オブジェクトとして，しばしば利用されることに符合する．よって，ボンガルド問題において，問題解決の高速化に寄与した知識を抽出できたと考える．なお，著者らは，第 1 と第 3 のルールについては，全く気付いていなかった．

4.6 タスク順序付け問題への適用

4.6.1 タスク順序付け問題

確率推定しながら問題解決する基本的なやり方の 1 つに，タスク順序付け (task sequencing) 問題 [28] がある．タスク集合を $\{T_1, \ldots, T_n\}$，タスク T_i の実行コストを c_i，任意のタスク列を $\sigma = (T_{\sigma(1)}, \ldots, T_{\sigma(n)})$ とする．また，各事例に対しては，$T_{\sigma(1)}$ から順に，どれかのタスクが成功するまで，処理が試みられるとする．タスク T_i が成功する事象を s_i，失敗する事象を $\neg s_i$ とすれば，タスク $T_{\sigma(i)}$ が実行されるのは，それ以前に実行したタスクが全て失敗したときであり，事例の処理が完了するまでの期待コスト $E(\sigma)$ は

$$E(\sigma) = \sum_{i=1}^{n} q_{\sigma(i)} c_{\sigma(i)}$$

で定義できる．ここで，確率 $q_{\sigma(i)}$ は

$$q_{\sigma(i)} = \begin{cases} 1 & i = 1 \\ P(\neg s_{\sigma(1)} \wedge \cdots \wedge \neg s_{\sigma(i-1)}) & i > 1 \end{cases}$$

で定義され，タスク $T_{\sigma(i)}$ が実行される確率である．タスク順序付け問題とは，期待コスト $E(\sigma)$ を最小にするタスク列 σ を求める問題である．

確率 $q_{\sigma(i)}$ を求めるには，一般に，任意のタスクに関する条件付き確率を考慮しなければならないが，多くの場合，現実的な仮定を導入できる．すなわち，各タスクの成功事象は，互いに排反 (exclusive) であるか，または，互いに独立 (independent) であると仮定する．ここで，事象 $e_1, \cdots, e_n$ が排反とは，

$$P(e_1 \vee \cdots \vee e_n) \;=\; P(e_1) + \cdots + P(e_n)$$

であり，事象 $e_1, \cdots, e_n$ が独立とは，

$$P(e_1 \wedge \cdots \wedge e_n) \;=\; P(e_1) \times \cdots \times P(e_n)$$

である．独立ケースについては，従来より広く研究され，さまざまな応用が知られている [130, 28, 129]．一方，排反ケースについては，あまり研究されていないが，この例には，多重故障確率が極めて小さい故障診断問題があり，応用分野の広い重要なケースである．以下では，タスク T_i の成功確率を $P(s_i) = p_i$ とする．

タスクの成功事象が互いに排反であり，全成功確率の和が 1 であるときには，確率 $q_{\sigma(i)}$ は

$$q_{\sigma(i)} \;=\; 1 - \sum_{j=1}^{i-1} p_{\sigma(j)} \;=\; \sum_{j=i}^{n} p_{\sigma(j)}$$

となる．したがって，期待コスト $E(\sigma)$ は

$$E(\sigma) \;=\; \sum_{i=1}^{n} (\sum_{j=i}^{n} p_{\sigma(j)}) c_{\sigma(i)}$$

となる．一方，タスクの成功事象が互いに独立であるときには，その否定も互いに独立となるので，$i > 1$ ならば，確率 $q_{\sigma(i)}$ は

$$q_{\sigma(i)} \;=\; \prod_{j=1}^{i-1} (1 - p_{\sigma(j)})$$

となる．したがって，期待コスト $E(\sigma)$ は

$$E(\sigma) \;=\; \sum_{i=1}^{n} (\prod_{j=0}^{i-1} (1 - p_{\sigma(j)})) c_{\sigma(i)}$$

となる．但し，$p_{\sigma(0)} = p_0 = 0$ とする．

各タスクの成功確率 p_i を既知とすれば，最適タスク列は容易に得られる．すなわち，タスクの成功事象が排反か独立であるとき p_i 既知の場合は，タスクを p_i/c_i の大きい順に並べれば，期待コストが最小となる [112]．ただし，タスクの成功事象が互いに排反，または，独立であると仮定しても，最適タスク列を生成する評価尺度は同じものとなるが，期待コストの差に関しては，異なる性質を持つ．すなわち，前者の場合には，期待コストの差は，入れ換える 2 つのタスクの成功確率とコストだけに依存する．一方，後者の場合には，$k \geq 2$ では，期待コストの差は，入れ換える 2 つのタスクの成功確率とコストだけでなく，k 以前に現れるタスクの成功確率にも依存する．なお，後者の結果は，多くの研究者により，独立に発見されている [130, 28, 129]．

4.6.2 最尤推定法とベイズ推定法

一般に，タスク成功確率 p_i は未知なので，事例より推定しなければならない．代表的な確率推定法には，最尤推定とベイズ推定がある．以下では，それぞれの方法について述べる．

まず，タスクの成功事象が互いに排反であるときには，タスクの成功は多項分布に従うので，総事例数を m，タスク T_i が成功した事例数を m_i とすれば，タスク成功確率は

$$\hat{p}_i \;=\; \frac{m_i}{m} \tag{4.6}$$

により最尤推定できる．一方，タスクの成功事象が互いに独立であるときには，n 個の二項分布の集まりとみなせるので，同様に，(4.6) 式により，最尤推定できる．したがって，タスクを m_i/c_i の大きい順に並べれば，タスク成功確率の最尤推定値 $\{\hat{p}_i\}$ に対する期待コストが最小になる．以下では，この評価尺度に基づいてタスク列を生成する方法を最尤推定法と呼ぶ．

大数の法則より，非常に多くの事例が与えられれば，$p_i \approx \hat{p}_i$ となるので，最小に近い期待コストで新たに現れる事例を処理できる．しかし，医療などへの応用では，タスク実行コストが非常に高いケースが考えられ，利用できる事例が少ない段階でも，できるだけ期待コストが小さいタスク列を生成する方法が望まれる．しかるに，最尤推定法は少い事例に対して優れたタスク列を与える保証はない．

事例が少いときにも有効な推定法として，事前確率に関する知識がなければ一様分布を仮定するラプラスの法則 (Laplace's law) に基づいたベイズ推定法が提案されている [30]．ラプラスの法則とは，タスク成功確率に関する事前知識がなければ，それらに一様分布を仮定することである．次の定理より，ラプラスの法則に基づくベイズ推定 (以下では，単にベイズ推定と呼ぶ) には，望ましい性質を示すことができる．すなわち，ベイズ推定に基づいて，タスクを $(m_i+1)/c_i$ の大きい順に並べれば，生成したタスク列の平均期待コストが最小になる [112]．以下では，この評価尺度に基づいてタスク列を生成する方法をベイズ推定法と呼ぶ．

最尤推定法の評価尺度 m_i/c_i の代わりに，ベイズ推定法では $(m_i+1)/c_i$ を用いることの妥当性について考察する．タスクの成功事象が互いに排反の場合には，ベイズ推定に基づけば，タスク成功確率は

$$\hat{p}_i \;=\; \frac{m_i + 1}{m + n}$$

により推定されることになる．まず，初期タスク列に関しては，$m = m_i = 0$ なので，$\hat{p}_i = 1/n$ となり，等確率になる．一方，最尤推定法では確率が定まらない．タスク成功確率が未知のときには，それらをすべて等しいと仮定するのは極めて自然であり，ベイズ推定法により合理的な推定値が得られる．

次に，事例が少い段階では，最尤推定法では，望ましくないタスクのペアの入れ換えが起こり得ることを示す．いま，$p_i/c_i > p_j/c_j$ であり，c_i は c_j より十分小さいが，p_j は p_i より若干大きいとする．まず，初期タスク列を $\hat{p}_i = \hat{p}_j$ として定めると，それは望ましい順番となっている．その後，$m_i = 0$, $m_j = 1$ となる確率は，$m_i = 1$, $m_j = 0$ が起こる確率よりも大きい．$m_i = 0$, $m_j = 1$ が起こると，最尤推定法では，望ましくないタスクのペアの入れ換えが起こる．従って，c_i が c_j より十分小さく，事例が少いときは，タスクの入れ換えを行なうべきではない．この問題に対処するには，コストだけに基づいた尺度 $(1/c_i)$，及び，推定確率とコストを用いた尺度 (m_i/c_i) の間でのトレードオフを考えなければならない．ベイズ推定法では，両者の単純な和を尺度にすべきことを示している．

最後に，事例が十分に多い段階になれば，最尤推定値の信頼性は高くなる．一方，一般に m_i と m の値は 1 に比較して十分大きな値となる．したがって，この段階では，1 を加えることが無視できるようになり，最尤推定法とベイズ推定法は，ほぼ同じタスク列を生成することになる．

一方，タスクの成功事象が互いに独立の場合には，n 個の二項分布の集まりなので，ベイズ推定に基づけば，タスク成功確率は

$$\hat{p}_i = \frac{m_i + 1}{m + 2}$$

により推定されることになる．そして，このときにも上記と同じ議論が成り立つ．

4.6.3 事例実験

最尤推定法とベイズ推定法を比較するため，10 個のタスクから成る問題を作って実験を行なった．タスクの成功事象が互いに排反の場合には，ランダムな成功確率 (和は 1) と 0 から 100 までのランダムなコストを与えた (表 4.3)．タスクの成功事象が互いに独立の場合には，0.5 以下のランダムな成功確率と表 1 と同じコストを与えた (表 4.4)．ここで，後者の確率を 0.5 以下に制限したのは，成功確率の高いタスクが先頭に存在し，容易に最適に近いタスク列が生成されるのを避けるためである．実験では，4 種の方法を比較した．即ち，ランダムなタスク列を用いる方法，ランダムに初期タスク列を設定した最尤推定法，等確率を仮定してコストの小さい順に初期タスク列を設定した最尤推定法，および，ベイズ推定法である．真の成功確率 (表 4.3 または 4.4 の値) に基づいてランダムに 50 個までの事例を生成して，それぞれの方法によりタスク列を逐次更新し，期待コストを計算した．

図 4.6 に，タスクの成功事象が互いに排反のケース，図 4.7 に，タスクの成功事象が互いに独立のケースの学習曲線を示す．ただし，結果は，試行を 100 回繰り返した平均値である．実験結果より，ベイズ推定法を用いれば，ランダムなタスク列や最尤推定法と比較して，任意の事例数の段階において，

表 4.3: 成功確率とコスト (排反ケース)

タスク	1	2	3	4	5	6	7	8	9	10
成功確率 p_i	.15	.22	.14	.4	.3	.17	.12	.3	.3	.7
コスト c_i	32	95	24	6	80	26	91	5	55	19
$p_i/c_i \times 10^4$	46	23	60	71	4	66	13	57	5	38

表 4.4: 成功確率とコスト (独立ケース)

タスク	1	2	3	4	5	6	7	8	9	10
成功確率 p_i	.12	.42	.11	.18	.14	.23	.49	.12	.12	.31
コスト c_i	32	95	24	6	80	26	91	5	55	19
$p_i/c_i \times 10^4$	38	44	46	300	17	88	54	240	22	163

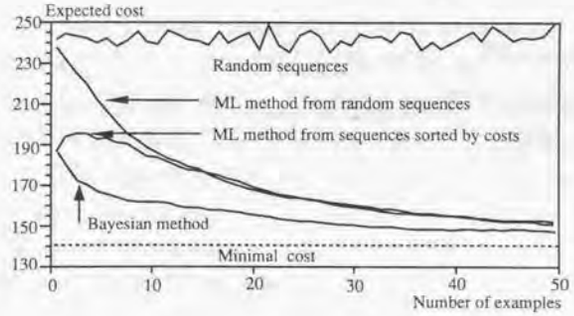


図 4.6: 学習曲線 (排反ケース)

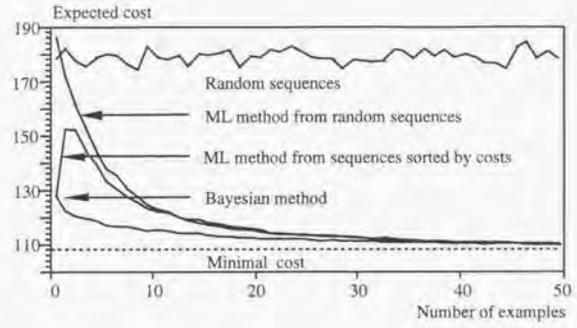


図 4.7: 学習曲線 (独立ケース)

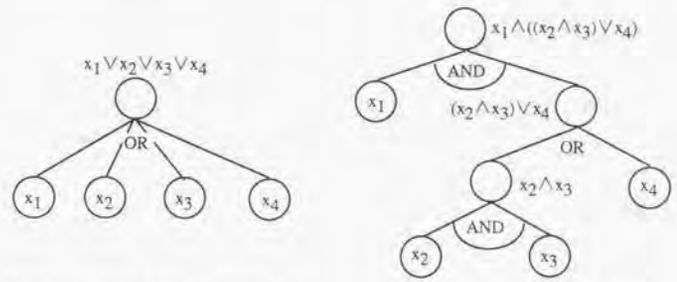
期待コストが最も小さいタスク列を生成できた. また, 最小コストにも速く近付くことが分かった. 最尤推定法では, 初期タスク列をコストの小さい順に設定しても, 初期タスク列をランダムに設定したときの曲線へ急速に近付くことが分かった. つまり, 事例が少ない段階では, 期待コストを改悪するのである. その理由は, 既に指摘したように, 事例が少い段階では, 最尤推定値の信頼性は低く, 望ましくないタスクのペアの入れ換えが起こるためと考えられる.

4.7 一般化タスク順序付け問題への適用

4.7.1 問題と解法の一般化

タスク順序付け問題の主たる応用は, μ-式 (μ-formulae) [80] に属す論理式の真偽値を高速に判定する問題として一般化できる. ここで, μ-式とは, ブール木 (Boolean trees) とも呼ばれ, 各原子式 (atomic fomula) が式に高々 1 回しか現れない論理式のクラスである. μ-式には, 任意の論理積 / 和結合が許されるので, 多くの自然な論理式をこのクラスで表現することができる. なお, これまでに述べたオリジナルなタスク順序づけ問題は, 任意の原子式が論理和結合する特殊な場合に対応する. 図 4.8に, オリジナルと一般化したタスク順序づけ問題の AND/OR 木の例を示す. また, 木が μ-式で表現できるので, ゲームの MINIMAX 探索への応用も可能である [78].

一般化した問題の解法を説明する. ここで, 原子式 x_i の真偽値を判定するためのコストを c_i, x_i が真となる確率を p_i とする. ここでは, 各原子式が真となる事象は互いに独立であると仮定する. まず, $f = x_1 \vee \cdots \vee x_n$ ならば, 少くとも 1 つの x_i が真となる成功確率 p_f, および, 原子式列 $\sigma(i)$ の順番で



(a) オリジナルなタスク順序付け問題　(b) 一般化したタスク順序付け問題

図 4.8: タスク順序付け問題の一般化

式を評価したときの判定に要する期待コスト c_f は

$$p_f = 1 - \prod_{i=1}^{n}(1-p_i), \qquad c_f = c_{\sigma(1)} + \sum_{i=2}^{n}\left(\prod_{j=1}^{i-1}(1-p_{\sigma(j)})\right) c_{\sigma(i)}$$

となる. 一方, $f = x_1 \wedge \cdots \wedge x_n$ ならば, すべての x_i が真となる成功確率 p_f, および, その判定に要する期待コスト c_f は

$$p_f = \prod_{i=1}^{n} p_i \qquad c_f = c_{\sigma(1)} + \sum_{i=2}^{n}\left(\prod_{j=1}^{i-1} p_{\sigma(j)}\right) c_{\sigma(i)}$$

となる. したがって, 任意の μ-式に対して, 葉から根に向けてこれらを再帰的に適用すれば, 成功確率とコストが順番に定義できるので, μ-式の真偽値を高速に判定する問題は, タスク順序付け問題により解決することができる. なお, 論理積結合した式については, 速い段階で偽となる式を見つければ, 後の処理を実行する必要がなくなるので, 論理和結合の場合とは逆に, タスクを確率とコストの比の小さい順に並べれば, 期待コストが最小になる.

4.7.2 事例実験

以下に, 実験で用いる μ-式 f を示す.

$$f = ((x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6 \vee x_7)) \vee ((x_8 \vee x_9 \vee x_{10} \vee x_{11}) \wedge (x_{12} \vee x_{13} \vee x_{14}))$$

ここで, f の真偽値を判定する問題は, 7 個のタスク順序付け問題に分解できる. すなわち, まず, 原子式が論理和結合した 4 つの式の評価, 次のレベルで論理積結合した 2 つの式の評価, 及び, トップレベ

表 4.5: 原子式の確率とコスト

原子式	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
確率 p_i	.68	.96	.35	.18	.73	.34	.79	.66	.14	.54	.75	.85	.22	.04
コスト c_i	20	41	41	22	47	34	90	87	24	41	75	49	86	3

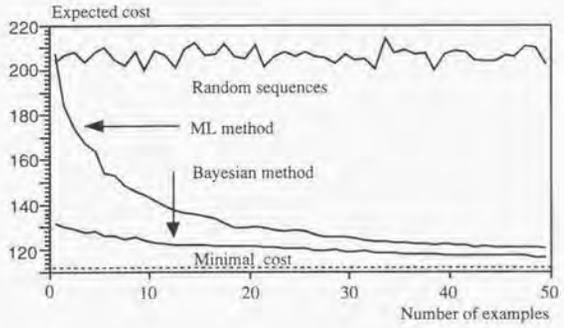


図 4.9: 論理式に対する学習曲線

ルで論理和結合した式の評価で, 合計 7 個である. 実験では, 3 種の方法を比較した. すなわち, ランダムなタスク列に基づく方法, 最尤推定法, および, ベイズ推定法である. なお, 実験の枠組は, 前回と同じ設定を用いた. 表 4.5 に, 実験で用いた成功確率とコスト, 図 4.9 に, 結果の学習曲線を示す. ベイズ推定法を用いれば, ランダムなタスク列や最尤推定法と比較して, 任意の事例数の段階において, 最も期待コストが小さくなった. また, 事例が増えても最尤推定法との差は明らかであり, その理由は, 複数のタスク列最適化問題の効果が相乗しているためと考えられる. したがって, 問題の規模が大きく複雑になれば, ベイズ推定法を用いるメリットは, さらに大きくなると期待される.

4.8 結言

過去に解いた問題を用いて, 自ら適応して探索効率を改善する概念学習法 RF4 を開発した. すなわち, 複数の枝刈り尺度と概念の複合化ルールを利用した深さ優先探索において, 状況ベクトルから, 原子式が識別概念の構成要素となる確率を計算することにより, 適応して探索を高速化する. KRK 問題では, RF4 の探索効率が改善されるだけでなく, 未知の事例に対する正答率も向上することを確認し

た. ボンガルド問題では, 100 のボンガルド問題に対して, RF4 は 41 問を正答できた. また, RF4 が 41 問を解答した後では, その問題解決時間は平均して 1/3 に短縮され, 問題解決を高速化する知識の一部も抽出できた. さらに, タスク順序付け問題, および, この問題を一般化した論理式の真偽値判定問題では, ベイズ推定を用いる方法が, ランダムなタスク列や最尤推定してタスク列を求める方法と比較して, 任意の事例数において, 期待コストが最小のタスク列を生成し, 最小コストにも速く近付くことを確認した.

第 5 章

ニューラルネットの高速学習法: BPQ

5.1 序言

多層ニューラルネット [93] はさまざまな分野に適用され，その有効性が実証されている [127, 94]. しかし，その代表的学習法の BP (Back Propagation) アルゴリズムでは，たとえ慣性項 (momentum term) を導入しても，収束までには一般に多くの反復が必要となり，さらに，性能に直結する学習定数 (learning rate) などのパラメータは，ユーザが試行錯誤により決定しなければならない．これらの課題の解決に向けて，学習定数最大化法 [52]，学習定数適合化規則 [40, 128], QuickProp [21] や RPROP [89] などの工夫を施したアルゴリズム，および，非線型最適化手法 [29] に基づく 2 次学習アルゴリズム [137, 4, 6, 71] などが提案され，それぞれはある程度の成功を収めている．しかるに，これらのアプローチの中では，理論的に優れた収束性が保証されるので，2 次学習アルゴリズムについて，さらに進んで研究すべきと考える．しかし，現時点では，解決すべき以下の 2 つの課題がある．

第 1 は大規模問題への適用性である．すなわち，Levenberg-Marquardt 法や準ニュートン (quasi-Newton) 法では，問題規模が大きくなれば適用が困難になる．Levenberg-Marquardt 法に基づくアルゴリズムは，各反復において探索方向を求めるのに $O(N^2m)$ の計算量が必要となるので，数百の結合重みからなるネットワークでも，一般に，収束までには多くの計算量が必要となる．ただし，N は結合重みの総数，m は事例数を表す．標準的な準ニュートン法に基づくアルゴリズム [137, 4] は，探索方向を求めるのに N^2 の記憶容量が必要となるので，N が数千以上のネットワークでは実用的でない．この問題に対処するため，OSS アルゴリズム [5] では，記憶なし更新法 [29] を採用しているが，探索方向を近似計算するので，優れた性能を示す保証はない．

第 2 は最適探索幅計算の処理負荷である． 適切な探索幅 (step-length) を求める直線探索 (line search) は，準ニュートン法や共役勾配 (conjugate gradient) 法に基づく学習アルゴリズムにおいて不可欠であり，不正確な直線探索では望ましい性能を得られないので，ある程度正確な直線探索を実行しなければならず，結果として多くの計算量が必要となる．なお，厳密な直線探索には，最小値を求

めるために多くの反復が必要となり，この反復処理が計算量を増大させる．ところが，共役勾配法に基づくアルゴリズムで優れた収束性を実現するには，かなり正確な直線探索が必要であり，よって，この種のアルゴリズムの効率を改善するのは困難であると考える．しかるに，SCG アルゴリズムでは [71]，差分近似に基づく効率の良い直線探索法が提案されている．一方，準ニュートン法に基づくアルゴリズムでは，直線探索が厳密でなくても，適当な条件が満たされれば，理論的にその収束が保証される [82]．ただし，このときに収束効率が良いとは限らない．よって，最適探索幅を妥当な精度で効率良く求めることができれば，準ニュートン法に基づくアルゴリズムは収束性と効率の両面で優れた方法になる．

多くの類似した事例を含む大規模問題では，各事例毎に結合重みの更新を行うオンライン (on-line) 法に基づくアルゴリズム [52] は，オフライン (off-line) 法に基づくアルゴリズムと比較して，一般に効率よく働くと考えられる．すなわち，2 次学習アルゴリズムはオフライン更新するのが前提であるが，1 回の更新を行うのに類似した多数の事例の情報を用いるので，効率的とは言えない．この問題に対処するため，2 次学習アルゴリズムにおいて，事例集合の部分集合に対して更新を行う疑似オンライン法の研究が行われている [72, 45]．よって，優れた 2 次学習アルゴリズムを開発できれば，それを用いた疑似オンライン法の効率も改善できる．

本章では，ニューラルネットの高速学習アルゴリズム BPQ [108, 99] を提案する．まず，ニューラルネットの学習問題について説明する．次に，探索方向を小記憶 BFGS 法で計算し，最適探索幅を 2 次近似の最小点として計算する 2 次学習アルゴリズム BPQ について述べ，続いて，BPQ と既存法の計算量について考察する．最後に，既存アルゴリズムとの比較実験により，BPQ の性能を評価する．

5.2 フレームワーク

$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ を訓練事例集合とする．ただし，$\mathbf{x}_t$ は n 次元入力ベクトル，y_t は $\mathbf{x}_t$ に対する目標出力値である．3 層ニューラルネットワークにおいて，h を中間ユニット数，$\mathbf{w}_i\,(i = 1, \cdots, h)$ を全入力ユニットと中間ユニット i 間の結合重み，$\mathbf{w}_0 = (w_{00}, \cdots, w_{0h})^T$ を全中間ユニットと出力ユニット間の結合重みとする．ただし，w_{i0} はバイアス項を表し，x_{t0} を 1 に設定する．以下では，全ての結合重みからなる 1 つのベクトルを $\mathbf{\Phi} = (\mathbf{w}_0^{\mathrm{T}}, \cdots, \mathbf{w}_{\mathrm{h}}^{\mathrm{T}})^{\mathrm{T}}$ で表し，その次元を $N = nh + 2h + 1$ とする．よって，3 層ニューラルネットの学習問題は以下の目的関数を最小にする $\mathbf{\Phi}$ を求める問題として定式化される．

$$f(\mathbf{\Phi}) \;=\; \frac{1}{2}\sum_{t=1}^{m}(y_t - z_t)^2 \tag{5.1}$$

ただし，$z_t = z(\mathbf{x}_t; \mathbf{\Phi}) = w_{00} + \sum_{i=1}^{h} w_{0i}\sigma(\mathbf{w}_i^T\mathbf{x}_t)$ であり，$\sigma(u)$ はシグモイド関数 $\sigma(u) = 1/(1+e^{-u})$ を表す．なお，出力ユニットに非線形がないモデルを扱うのは，それが関数近似の観点で本質ではないからである．

5.3 BPQ アルゴリズム

5.3.1 準ニュートン法

$f(\mathbf{\Phi} + \Delta\mathbf{\Phi})$ の $\Delta\mathbf{\Phi}$ での 2 次 Talor 展開式は $f(\mathbf{\Phi}) + (\nabla f(\mathbf{\Phi}))^T\Delta\mathbf{\Phi} + \frac{1}{2}(\Delta\mathbf{\Phi})^T\nabla^2 f(\mathbf{\Phi})\Delta\mathbf{\Phi}$ であり，ヘス行列 $\nabla^2 f(\mathbf{\Phi})$ が正定値ならば，この式の最小値は $\Delta\mathbf{\Phi} = -(\nabla^2 f(\mathbf{\Phi}))^{-1}\nabla f(\mathbf{\Phi})$ で与えられる．ニュートン法では，各反復でこの修正ベクトル $\Delta\mathbf{\Phi}$ を求めることにより，目的関数 $f(\mathbf{\Phi})$ を最小化する [29]．しかし，$(\nabla^2 f(\mathbf{\Phi}))^{-1}$ を求めるには $O(N^3)$ の計算量が必要であり，ニュートン法を大規模問題へ適用することは困難である．一方，準ニュートン法 [29] は，探索の過程で反復により，ヘス逆行列 $(\nabla^2 f(\mathbf{\Phi}))^{-1}$ の近似行列 $(\mathbf{H})$ を各ステップで求めることを特徴とする．基本アルゴリズムは以下である．

step 1: $\mathbf{\Phi}^1$ を初期化し，$\mathbf{H}^1 = \mathbf{I}$, $k = 1$ とする．
step 2: 探索方向を求める: $\Delta\mathbf{\Phi}^k = -\mathbf{H}^k\nabla f(\mathbf{\Phi}^k)$.
step 3: 停止条件を満たせば，反復を終了させる．
step 4: $f(\mathbf{\Phi}^k + \lambda\Delta\mathbf{\Phi}^k)$ を最小にする λ^k を求める．
step 5: 結合重みを修正する: $\mathbf{\Phi}^{k+1} = \mathbf{\Phi}^k + \lambda^k\Delta\mathbf{\Phi}^k$.
step 6: $k \equiv 0 \pmod N$ ならば，$\mathbf{H}^{k+1} = \mathbf{I}$ とし，さもなければ，$\mathbf{H}^{k+1}$ を更新する．
step 7: $k = k+1$ とし，**step 2** に戻る．

5.3.2 探索方向の既存計算法

準ニュートン法において，近似行列 $\mathbf{H}$ の計算法にはさまざまな提案があるが，Broydon-Fletcher-Goldfarb-Shanno (BFGS) 法 [29] は，理論と数値実験の両面で最も優れた方法とである．ここで，$\mathbf{p}_k = \lambda_k\Delta\mathbf{\Phi}_k$, $\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ とおけば，BFGS 法による更新公式は

$$\mathbf{H}_{k+1} \;=\; \mathbf{H}_k - \frac{\mathbf{p}_k\mathbf{q}_k^T\mathbf{H}_k + \mathbf{H}_k\mathbf{q}_k\mathbf{p}_k^T}{\mathbf{p}_k^T\mathbf{q}_k} + \left(1 + \frac{\mathbf{q}_k^T\mathbf{H}_k\mathbf{q}_k}{\mathbf{p}_k^T\mathbf{q}_k}\right)\frac{\mathbf{p}_k\mathbf{p}_k^T}{\mathbf{p}_k^T\mathbf{q}_k}. \tag{5.2}$$

となる．しかし，数千以上の結合重みのあるネットワークでは，N^2 の記憶容量が必要となるので，近似行列 $\mathbf{H}$ を保持することは実用的ではない．この問題に対処するため，OSS (One-Step Secant) アルゴリズム [5] では，記憶なし BFGS 法 [29] を採用している．すなわち，前回の近似行列 $\mathbf{H}_k$ を常に単位行列とし $(\mathbf{H}_k = \mathbf{I})$, **Step 2** で計算する探索方向を

$$\Delta\mathbf{\Phi}_{k+1} \;=\; -\mathbf{g}_{k+1} + \frac{\mathbf{p}_k\mathbf{q}_k^T\mathbf{g}_{k+1} + \mathbf{q}_k\mathbf{p}_k^T\mathbf{g}_{k+1}}{\mathbf{p}_k^T\mathbf{q}_k} - \left(1 + \frac{\mathbf{q}_k^T\mathbf{q}_k}{\mathbf{p}_k^T\mathbf{q}_k}\right)\frac{\mathbf{p}_k\mathbf{p}_k^T\mathbf{g}_{k+1}}{\mathbf{p}_k^T\mathbf{q}_k} \tag{5.3}$$

で求める．明らかに，(5.3) 式は $O(N)$ の計算量と $O(N)$ の記憶容量で計算できる．しかし，探索方向をこのような近似で計算するので，OSS が優れた性能を示す保証はない．我々の実験では，記憶なし BFGS 法は，次に示す小記憶 BFGS 法と比較して，低い性能であった．

5.3.3 探索方向の新計算法

ここでは, 記憶容量が $2Ns$ ($s \ll N$) となる小記憶 BFGS 法を提案する. その探索方向は, 始めの $s+1$ 反復において, オリジナル BFGS 法と完全に一致し, s は以下で述べる履歴の長さを表す局部性 (partiality) パラメータである. $\mathbf{r}_k = \mathbf{H}_k\mathbf{q}_k$ とおけば,

$$\mathbf{H}_k\mathbf{g}_{k+1} \;=\; \mathbf{H}_k\mathbf{q}_k + \mathbf{H}_k\mathbf{g}_k \;=\; \mathbf{r}_k - \frac{\mathbf{p}_k}{\lambda_k}.$$

である. よって, (5.2) 式を用いて計算する探索方向は

$$\begin{aligned}\Delta\mathbf{\Phi}_{k+1} &= -\mathbf{H}_{k+1}\mathbf{g}_{k+1} \\ &= -\mathbf{r}_k + \frac{\mathbf{p}_k}{\lambda_k} + \frac{\mathbf{p}_k\mathbf{r}_k^T\mathbf{g}_{k+1} + \mathbf{r}_k\mathbf{p}_k^T\mathbf{g}_{k+1}}{\mathbf{p}_k^T\mathbf{q}_k} - \left(1 + \frac{\mathbf{q}_k^T\mathbf{r}_k}{\mathbf{p}_k^T\mathbf{q}_k}\right)\frac{\mathbf{p}_k\mathbf{p}_k^T\mathbf{g}_{k+1}}{\mathbf{p}_k^T\mathbf{q}_k}. \qquad (5.4)\end{aligned}$$

となる. $\mathbf{r}_k$ が求まれば, 記憶なし BFGS 法と同様に, (5.4) 式は $O(N)$ の計算量と $O(N)$ の記憶容量で計算できる. そこで, $\mathbf{r}_k$ が $O(Ns)$ の計算量と $2Ns$ の記憶容量で計算できることを以下に示す.

まず, $k \le s$ を仮定する. $k=1$ のとき, $\mathbf{r}_1$ $(=\mathbf{H}_1\mathbf{q}_1 = \mathbf{g}_2 - \mathbf{g}_1)$ は引き算だけで計算できる. $k>1$ のとき, $\mathbf{r}_1, \cdots, \mathbf{r}_{k-1}$ の各要素は既に計算されているとする. ここで, $i<k$ では,

$$\alpha_i = \frac{1}{\mathbf{p}_i^T\mathbf{q}_i} \quad \text{and} \quad \beta_i = \alpha_i\left(1 + \alpha_i\mathbf{q}_i^T\mathbf{r}_i\right)$$

は各反復で既に計算されていることを確認する. よって, (5.2) 式を再帰的に適用すれば, $\mathbf{r}_k$ は以下のように $O(Ns)$ の計算量と $2Ns$ の記憶容量で計算できる.

$$\begin{aligned}\mathbf{r}_k &= \mathbf{H}_k\mathbf{q}_k \\ &= \mathbf{H}_{k-1}\mathbf{q}_k - \alpha_{k-1}\mathbf{p}_{k-1}\mathbf{r}_{k-1}^T\mathbf{q}_k - \alpha_{k-1}\mathbf{r}_{k-1}\mathbf{p}_{k-1}^T\mathbf{q}_k + \beta_{k-1}\mathbf{p}_{k-1}\mathbf{p}_{k-1}^T\mathbf{q}_k \\ &= \mathbf{q}_k + \sum_{i=1}^{k-1}\left(-\alpha_i\mathbf{p}_i\mathbf{r}_i^T\mathbf{q}_k - \alpha_i\mathbf{r}_i\mathbf{p}_i^T\mathbf{q}_k + \beta_i\mathbf{p}_i\mathbf{p}_i^T\mathbf{q}_k\right) \qquad (5.5)\end{aligned}$$

次に, $k=s+1$ のとき, 2 つの更新法が考えられる. すなわち, これまでに蓄えた探索情報ベクトル $(\mathbf{p}, \mathbf{r})$ をすべて消去して更新を再開する方法, または, 最新の探索情報ベクトルで更新を続ける方法である. どちらの場合も, (5.5) 式は $O(Ns)$ の計算量と $2Ns$ の記憶容量で計算できる. したがって, (5.4) 式は $O(Ns)$ の計算量と $2Ns$ の記憶容量で計算できることが示された. 我々の実験では, s が小さいとき, 後者の更新法は効率良く働かなかったので, 前者の更新法が採用された.

小記憶 BFGS 法のアイデアについては, 既に知られている [54]. しかし, 厳密に言えば, それはここでの方法とは異なるものである. つまり, 初期の提案では, ベクトル $\mathbf{p}_i, \mathbf{q}_i$ を保持することが意図されたが, 本論文での提案法はベクトル $\mathbf{p}_i, \mathbf{r}_i$ を保持する. オリジナル BFGS 法に対する小記憶 BFGS 法の直接の利点は, 大規模問題にも適用可能なことにある. すなわち, たとえ N が非常に大きくても,

システムの記憶容量を考慮して s を適切に設定すれば, 小記憶 BFGS 法を適用することができる. 一方, 記憶なし BFGS 法に対する利点は, 優れた収束性が期待できることにある. もちろん, この点については, 幅広い問題で実証すべきである. なお, $s=0$ ならば, 小記憶 BFGS 法は常に勾配方向を計算し, $s=1$ のときは, 記憶なし BFGS 法と対応する.

5.3.4 探索幅の既存計算法

Step 4 では, λ が $f(\cdot)$ の唯一の変数なので, $f(\mathbf{\Phi}+\lambda\Delta\mathbf{\Phi})$ を $\zeta(\lambda)$ で表す. $\zeta(\lambda)$ を最小にする λ をもとめる処理は直線探索と呼ばれる. 以下では, さまざまな直線探索法の中で, 典型と考えられる 3 つの方法について考える. すなわち, 速いが不正確な方法, 妥当な精度の方法, および, 厳密だが遅い方法である.

まず, 速いが不正確な方法について説明する. 2 次補完法 [29, 5] を用いれば, $\zeta(\lambda) < \zeta(0)$ だけを保証する効率の良い方法を導ける. λ_0 を探索幅の初期値とする. もし $\zeta(\lambda_0) < \zeta(0)$ ならば, λ_0 を結果の探索幅とする. さもなければ, 条件 $h(0)=\zeta(0)$, $h(\lambda_0)=\zeta(\lambda_0)$, $h'(0)=\zeta'(0)$ を満たす次の 2 次近似式 $h(\lambda)$ を考える.

$$\zeta(\lambda) \;\approx\; h(\lambda) \;=\; \zeta(0) + \zeta'(0)\lambda + \frac{\zeta(\lambda_0) - \zeta(0) - \zeta'(0)\lambda_0}{\lambda_0^2}\lambda^2$$

$\zeta(\lambda_0) \ge \zeta(0)$ かつ $\zeta'(0) < 0$ より, $h(\lambda)$ の最小点は

$$\lambda \;=\; -\frac{\zeta'(0)\lambda_0^2}{2(\zeta(\lambda_0) - \zeta(0) - \zeta'(0)\lambda_0)}. \qquad (5.6)$$

で与えられる. このとき, (5.6) 式では $0 < \lambda < \lambda_0$ が保証される. よって, この処理を $\zeta(\lambda) < \zeta(0)$ となるまで繰り返せば, $\zeta(\lambda) < \zeta(0)$ となる λ を常に求めることができる. ここで, $\mathbf{H}$ が $(\nabla^2 f(\mathbf{\Phi}))^{-1}$ を十分に良く近似していれば, 最適探索幅はほぼ 1 なので, λ_0 の初期値を 1 に設定する. 以下では, オリジナル BFGS 法に, 速いが不正確な直線探索法を組み込んだ準ニュートン法のアルゴリズムを *BFGS1* と呼ぶ.

2 次外挿法 [25] を用いれば, $\zeta'(\lambda_\nu) > \gamma_1\zeta'(\lambda_{\nu-1})$ と $\zeta(\lambda_\nu) < \zeta(\lambda_{\nu-1})$ を保証する, 妥当な精度の直線探索法が導かれる. ただし, γ_1 は適当な定数である (e.g. 0.1). つまり, λ_ν が終了条件を満足しないとき, もし $\zeta(\lambda_\nu) \ge \zeta(\lambda_{\nu-1})$ ならば, (5.6) 式で $\lambda_{\nu+1}$ を計算し, さもなければ, 傾き $\zeta'(\lambda_{\nu-1})$ と $\zeta'(\lambda_\nu)$ の外挿近似式を考え, この近似式を最小にする $\lambda_{\nu+1}$ を以下の式で計算する.

$$\lambda_{\nu+1} \;=\; \lambda_\nu - \zeta'(\lambda_\nu)\frac{\lambda_\nu - \lambda_{\nu-1}}{\zeta'(\lambda_\nu) - \zeta'(\lambda_{\nu-1})}. \qquad (5.7)$$

しかし, もし $\zeta'(\lambda_\nu) \le \zeta'(\lambda_{\nu-1})$ ならば, $\lambda_{\nu+1}$ が存在しないので, $\lambda_{\nu+1}$ を $\lambda_{\nu-1} + \gamma_2(\lambda_\nu - \lambda_{\nu-1})$ とする. ただし, γ_2 は適当な定数である (e.g. 9). この方法では, λ_0 を 0 とし,

$$\lambda_1 \;=\; \min(1, -2\zeta'(0)^{-1}(f(\mathbf{\Phi}_{current}) - f(\mathbf{\Phi}_{previous})))$$

に設定する．なお，速いが不正確な方法では，外挿手段がないため，λ_1 の値が非常に小さいときに不都合が起こるので，この λ_1 の推定法を適用すべきではない．以下では，オリジナル BFGS 法に，妥当な精度の直線探索法を組み込んだ準ニュートン法のアルゴリズムを *BFGS2* と呼ぶ．

厳密だが遅い直線探索法は，上述の妥当な精度の探索法を厳密な終了条件が満たされるまで繰り返し適用することにより構築できる．なお，終了条件には $\|\zeta'(\lambda)\| < 10^{-8}$ を用いた．以下では，オリジナル BFGS 法に，厳密だが遅い直線探索法を組み込んだ準ニュートン法のアルゴリズムを BFGS3 と呼ぶ．

5.3.5 探索幅の新計算法

ここでは，**Step 4** で最適探索幅 λ を求める新計算法を提案する．

基本計算法

$\zeta(\lambda)$ の 2 次 Taylor 展開式は

$$\zeta(\lambda) \approx \zeta(0) + \zeta'(0)\lambda + \frac{1}{2}\zeta''(0)\lambda^2.$$

である．$\zeta'(0) < 0$ かつ $\zeta''(0) > 0$ のとき，この近似式の最小点は

$$\lambda = -\frac{\zeta'(0)}{\zeta''(0)} \left(= -\frac{(\nabla f(\Phi))^T \Delta\Phi}{(\Delta\Phi)^T \nabla^2 f(\Phi) \Delta\Phi} \right). \tag{5.8}$$

で与えられる．他のケースについては後述する．

(5.1) 式で定義した 3 層ニューラルネットでは，$\zeta'(0)$ と $\zeta''(0)$ が以下のように効率良く計算できる．$\zeta(\lambda)$ を微分して λ に 0 を代入すれば，

$$\zeta'(0) = -\sum_{t=1}^{m}(y_t - z_t)z'_t, \qquad \zeta''(0) = \sum_{t=1}^{m}((z'_t)^2 - (y_t - z_t)z''_t)$$

となる．ここで，$z_t = z(\mathbf{x}_t; \Phi)$ の微分は $\frac{d}{d\lambda} z(\mathbf{x}_t; \Phi + \lambda\Delta\Phi)_{|\lambda=0}$ で定義され，

$$z'_t = \Delta w_{00} + \sum_{i=1}^{h}(\Delta w_{0i}\sigma_{it} + w_{0i}\sigma'_{it}), \qquad z''_t = \sum_{i=1}^{h}(2\Delta w_{0i}\sigma'_{it} + w_{0i}\sigma''_{it})$$

である．ただし，$\sigma_{it} = \sigma(\mathbf{w}_i^T \mathbf{x}_t)$, $\sigma'_{it} = \sigma_{it}(1-\sigma_{it})(\Delta\mathbf{w}_i)^T\mathbf{x}_t$, $\sigma''_{it} = \sigma'_{it}(1-2\sigma_{it})(\Delta\mathbf{w}_i)^T\mathbf{x}_t$ であり，Δw_{ij} は **Step 2** で計算される w_{ij} の修正量を表す．

ここで，(5.8) 式を用いた最適探索幅計算の計算量について考察する．明らかに，中間ユニット i と入力ベクトル $\mathbf{x}_t$ の各ペアに対して，$(\Delta\mathbf{w}_i)^T\mathbf{x}_t$ を計算しなければならないので，中間ユニット数が h で入力事例数が m であることより，まず nhm 回の乗算が必要となる．残りの計算は $O(hm)$ 回の乗算で完了し，$N = nh + 2h + 1$ より，全体の計算量は $Nm + O(hm)$ となる．

この最適探索幅計算法は，cross-entropy 関数を目的関数とする分類問題にも同様に適用可能であり，実験により有効であることを示している [121]．さらに，微分可能な異なる活性化関数を有する多層ネットワークへ一般化することができ，完全相互結合するリカレントネットの学習やガウス混合分布の推定にも適用することができる [109, 110, 120]．以下，計算法の一般化について説明する．τ を出力層に直接結合する中間ユニットとし，その出力値は $v = a(\mathbf{w}^T\mathbf{u})$ で定義されるとする．ただし，$\mathbf{u}$ はユニット τ へ結合するユニットの出力値，$\mathbf{w}$ はその結合における結合重み，$a(\cdot)$ は適当な活性化関数を表す．このとき，λ に対する v の 1 次と 2 次の微分は

$$\begin{aligned} v' &= a'(\mathbf{w}^T\mathbf{u})\left(\Delta\mathbf{w}^T\mathbf{u} + \mathbf{w}^T\mathbf{u}'\right), \\ v'' &= a''(\mathbf{w}^T\mathbf{u})\left(\Delta\mathbf{w}^T\mathbf{u} + \mathbf{w}^T\mathbf{u}'\right)^2 + a'(\mathbf{w}^T\mathbf{u})\left(2\Delta\mathbf{w}^T\mathbf{u}' + \mathbf{w}^T\mathbf{u}''\right) \end{aligned}$$

で計算できる．よって，この式を順次後ろ向きに適用することにより，任意の多層ネットワークで最適探索幅を計算できる．なお，u_i が入力ユニットの出力値ならば $u'_i = u''_i = 0$ である．

望ましくないケースへの対処

上述のケースでは，$\zeta'(0) < 0$ を仮定した．$\zeta'(0) > 0$ のとき，その探索方向で目的関数の値を減少できないので，$\Delta\Phi^k = -\nabla f(\Phi^k)$ とし，これまでに蓄えた探索情報ベクトル $(\mathbf{p}, \mathbf{r})$ をすべて消去する．すなわち，これ以降，各反復での探索方向を勾配方向から再開する．このとき，$g'(0) = (\nabla f(\Phi^k))^T\Delta\Phi^k = -\|\nabla f(\Phi^k)\|^2 < 0$ より，$g'(0) < 0$ が保証される．

$\zeta'(0) < 0$ かつ $\zeta''(0) \leq 0$ のとき，(5.8) 式の値は負または無限大となるので，ガウス - ニュートン法を用いて，このケースに対処する．$z(\mathbf{x}_t; \Phi + \lambda\Delta\Phi)$ の 1 次近似は $z_t + z'_t\lambda$ となるので，$\zeta(\lambda)$ の近似は

$$\begin{aligned} \zeta(\lambda) &\approx \frac{1}{2}\sum_{t=1}^{m}(y_t - (z_t + z'_t\lambda))^2 \\ &= \zeta(0) + \zeta'(0)\lambda + \frac{1}{2}\sum_{t=1}^{m}(z'_t)^2\lambda^2 \end{aligned}$$

であり，この式の右辺の最小値は

$$\lambda = -\frac{g'(0)}{\sum_{t=1}^{m}(z'_t)^2}. \tag{5.9}$$

で与えられる．明らかに，$\zeta'(0) < 0$ のとき，(5.9) 式の値は正となる．

多くの場合，各反復で Φ の修正量の上限を設定することは有効である [29]．よって，$\|\lambda\Delta\Phi\| > 1.0$ ならば，λ を $1.0/\|\Delta\Phi\|$ とする．

λ は近似に基づき計算されるので，目的関数 $\zeta(\lambda)$ の値が常に減少するとは限らない．$\zeta(\lambda) \geq \zeta(0)$ のときは，(5.6) 式を用いて λ を縮小する．

探索幅計算法のまとめ

上述の探索幅計算法をまとめれば, 基本アルゴリズムの **Step 4** は以下となる.

Step 4-1: もし $\zeta'(0) > 0$ ならば, $\Delta\Phi_k = -\nabla f(\Phi_k)$, $k = 1$ とする.
Step 4-2: もし $\zeta''(0) > 0$ ならば, (5.8) 式, さもなければ, (5.9) 式で λ を計算する.
Step 4-3: もし $\|\lambda\Delta\Phi_k\| > 1.0$ ならば, $\lambda = \|\Delta\Phi_k\|^{-1}$ とする.
Step 4-4: もし $\zeta(\lambda) > \zeta(0)$ ならば, $\zeta(\lambda) < \zeta(0)$ となるまで (5.6) 式で λ を計算する.

以下では, 小記憶 BFGS 法に基づく準ニュートン法において, 提案した最適幅計算法を採用する 2 次学習アルゴリズムを *BPQ* (*BP based on Quasi-Newton*) と呼ぶ. なお, 記憶なし BFGS 法と提案した最適幅計算法を組み合わせて改良した OSS アルゴリズムを OSS2 と呼び, OSS2 は良いアルゴリズムとなる可能性があるので, 実験で評価する.

5.4 計算量の考察

BPQ と既存学習法に関して, 全ての訓練事例をそれぞれ 1 回利用する 1 反復の計算量を考察する. まず, オフライン BP での計算量 (乗算の回数) は, 目的関数の値を求めるのに $nhm + O(hm)$, 勾配ベクトルでは $nhm + O(hm)$ が必要となる. よって, $N = nh + h + 1$ より, オフライン BP の全体の計算量は $2Nm + O(hm)$ である.

上述に加え, BPQ では, 高々 s 反復の履歴を持つ小記憶 BFGS で探索方向を計算し, さらに, 最適探索幅を計算する. 既に述べたように, 前者の計算量は $O(Ns)$, 後者は $Nm + O(hm)$ である. しかるに, BPQ で目的関数の値を求める計算量は $Nm + O(hm)$ から $O(hm)$ に減少する. なぜなら, 各中間ユニットの出力値は $\sigma(\mathbf{w}_i^T\mathbf{x}_t + \lambda(\Delta\mathbf{w}_i)^T\mathbf{x}_t)$ であるが, 直前の反復での最適探索幅計算において, $(\Delta\mathbf{w}_i)^T\mathbf{x}_t$ は既に計算されているからである. よって, BPQ の全体の計算量は $2Nm + O(Ns) + O(hm)$ である. 未知の事例に対する汎化誤差を小さくするには, 一般に, m は N と比較して, ある程度大きくなければならない. PAC- 学習理論 [134] に基づけば, 汎化誤差の上限を ϵ 以下とするのに必要な事例数は, 概算で $\epsilon^{-1}N$ 以上である [7]. よって, s は N より小さいことより, 計算量 $O(Ns)$ は, $2Nm$ と比較して, かなり小さくなるので, BPQ の計算量のオーダは BP とほぼ等価になる.

$\nabla^2 f(\Phi)\Delta\Phi$ を求めた後, 内積により (5.8) 式の分母を計算する一般的な方法は既に提案されている [79, 70]. 数値結果は, 提案した最適探索幅計算法と同じになるが, 少なくとも 3 層ネットワークにおいては, 既存法は, 提案法と比較して, 多くの計算量が必要となる. 以下にその理由を示す.

$$\Re_{\Delta\Phi}\{f(\Phi)\} = \frac{\partial}{\partial\lambda} f(\Phi + \lambda\Delta\Phi)_{|\lambda=0}$$

で定義される Pearlmutter のオペレータ [79] を用いれば, $\Re_{\Delta\Phi}\{\nabla f(\Phi)\} = \nabla^2 f(\Phi)\Delta\Phi$ より, $\Re_{\Delta\Phi}\{\frac{\partial}{\partial w_{ij}} f(\Phi)\}$ は $\nabla^2 f(\Phi)\Delta\Phi$ の要素であることが分る. ここで, 出力ユニットと中間ユニット i 間の結合重み w_{0i} に関して計算すれば,

$$\begin{aligned}\Re_{\Delta\Phi}\left\{\frac{\partial}{\partial w_{0i}} f(\Phi)\right\} &= \Re_{\Delta\Phi}\left\{-\sum_{t=1}^{m}(y_t - z_t)\sigma_{it}\right\} \\ &= \sum_{t=1}^{m}(\Re_{\Delta\Phi}\{z_t\}\sigma_{it} - (y_t - z_t)\Re_{\Delta\Phi}\{\sigma_{it}\})\end{aligned}$$

となる. ただし,

$$\Re_{\Delta\Phi}\{z_t\} = \Delta w_{00} + \sum_{i=1}^{h}(\Delta w_{0i}\sigma_{it} + w_{0i}\Re_{\Delta\Phi}\{\sigma_{it}\}), \quad \Re_{\Delta\Phi}\{\sigma_{it}\} = \sigma_{it}(1-\sigma_{it})(\Delta\mathbf{w}_i)^T\mathbf{x}_t$$

である. 明らかに, 提案法と同様に, 中間ユニット i と入力ベクトル $\mathbf{x}_t$ の各ペアに対して, $(\Delta\mathbf{w}_i)^T\mathbf{x}_t$ を計算しなければならないので, 少なくとも nhm 回の乗算が必要となる. さらに, $\Re_{\Delta\Phi}\{\frac{\partial}{\partial w_{0i}} f(\Phi)\}$ を計算するのに, 少なくとも $O(m)$ の計算量が必要である. したがって, 同様な議論が他の結合重みに対しても適用され, 結合重みの総数は N であることより, 既存法の計算量は $nhm + O(Nm)$ となる. これに対して, 提案法の計算量は $nhm + O(hm)$ であり, $N = nh + 2h + 1$ なので, より効率的であることが分る.

SCG (Scaled Conjugate Gradient) アルゴリズム [71] では, 差分近似により探索幅を求める方法が提案されている. すなわち, (5.8) 式で定義される最適探索幅は

$$\nabla^2 f(\Phi)\Delta\Phi \approx \frac{\nabla f(\Phi + \delta\Delta\Phi) - \nabla f(\Phi)}{\delta},$$

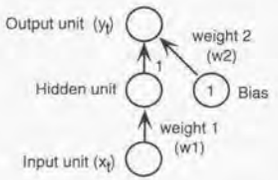
を用いて近似される. ただし, δ は $\delta = \delta_0\|\Delta\Phi\|^{-1}$ で定義され, δ_0 は小さな定数である. 明らかに, 勾配ベクトル $\nabla f(\Phi + \delta\Delta\Phi)$ を各反復で計算するので, SCG の計算量はおよそ BP の 2 倍となる. もし, パラメータ δ がほぼ最適探索幅 λ に等しければ, 本論文の提案法と比較して, SCG の方法がより正確な探索幅を与える. しかし, ここでの目的は最適探索幅 λ を求めることであり, そのような δ を予め知ることはできない.

5.5 実験による評価

ここでは, 3 種の問題を用いた実験により, BPQ の性能を他の代表的な学習アルゴリズムと比較して評価する.

5.5.1 人工問題

学習効率を視覚的に評価するため, 2 つの結合重みだけからなるネットワークの学習問題を設計した. 図 5.1 に問題を示す. すなわち, その 3 層ネットワークでは, 各層が 1 つのユニットで構成され,



Input value (x_t)	Target value (y_t)
$x_1 = 1$	$y_1 = 0.73$
$x_2 = 2$	$y_2 = 0.88$
$x_3 = 3$	$y_3 = 0.95$
$x_4 = 4$	$y_4 = 0.98$
$x_5 = 5$	$y_5 = 0.99$

図 5.1: 2- 結合重み問題

中間 - 出力ユニット間の結合重みが 1 に固定される．この問題では，入力 - 中間ユニット間の結合重みを w_1 と表記し，出力ユニットのバイアス重みを w_2 と表記する．図 5.1 に示した目標出力値は，$(w_1, w_2) = (1, 0)$ と設定して入力値から計算したものである．よって，この結合重みを用いれば，学習の目的関数は最小となる．

この実験では，BPQ を 8 種の学習アルゴリズム (オンライン慣性項付き BP, オフライン BP, 慣性項付き BP, 適応 BP [128], SCG [71], BFGS1, BFGS2, BFGS3) と比較する．なお，$N = 2$ のときには，小記憶 BFGS はオリジナル BFGS と等価になる．各アルゴリズムのパラメータは，提案者の推奨する値 [128, 71] または試行錯誤に基づき決定した．すなわち，オンライン慣性項付き BP では，学習定数と慣性項の係数を，それぞれ，$\eta = 1.0$ と $\alpha = 0.9$ に設定し，オフライン BP では，学習定数 η を 0.25 に設定した．慣性項付き BP では，学習定数と慣性項の係数を，それぞれ $\eta = 0.025$ と $\alpha = 0.9$ に設定した．適応 BP では，増加係数 (increase factor) と減少係数 (decrease factor) を $u = 1.1$ と $d = u^{-1}$ に設定し，初期更新値 η_0 を 0.01 に設定した．SCG では，定数 δ_0 を 10^{-4} に設定し，初期スケーリング値 (scaling value) λ_1 を 10^{-6} に設定した．

w_1 と w_2 に対する誤差曲面上で，$(w_1, w_2) = (-0.5, 0.0)$ を初期値とし，最大で 100 反復させた各学習アルゴリズムの学習軌跡を図 5.2 と 5.3 に示す．ただし，MSE は以下で定義される平均自乗誤差を表す．

$$\mathrm{MSE} = \frac{1}{5}\sum_{t=1}^{5}(y_t - (\sigma(w_1 x_t) + w_2))^2$$

図 5.2 に示すように，1 次学習アルゴリズムの探索は，誤差曲面の谷底で非常に効率悪くなるり，100 反復まででは，どのアルゴリズムでも最小点に到達することができなかった．これは，谷底近くで，連続する 2 つの勾配ベクトルの方向がほぼ逆向きになるためであり，1 次学習アルゴリズムの本質的な問題点である．また，連続する 2 つの勾配ベクトルの符合に基づく学習定数の適合化では，学習効率を改善できなかったことが分る．なお，オンライン BP の 1 反復は，全ての事例を用いた m 回の結合重みの更新であり，また，学習定数 η を大きくすれば，BP の学習軌跡は大きく振動する．

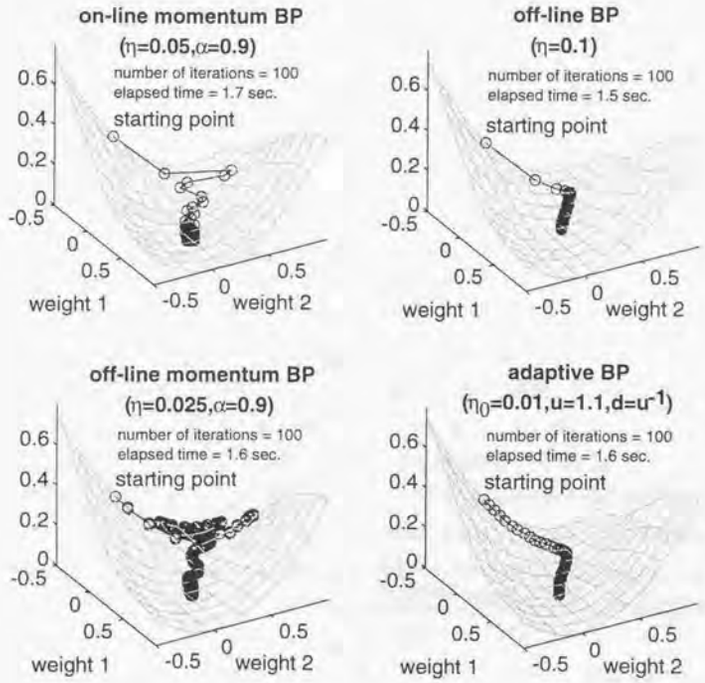


図 5.2: 1 次学習アルゴリズムの学習軌跡

これに対し, 図 5.3 に示すように, SCG, BFGS1, BFGS2, BFGS3, および, BPQ は 100 反復以内で最小点に到達した. BPQ と比較すれば, 最小点に到達するまでに, SCG は多くの反復回数を必要とした. BFGS1 では, 最も多くの反復回数が必要となり, この問題を解くのに探索幅計算が重要であることを示している. BFGS2 は BPQ と同様に効率良く最小点到達した. BFGS3 の反復回数は BPQ より僅かに少なかったが, BFGS3 は厳密に探索幅を計算するので, 全体の計算量は多くなった. これらの実験結果より, 曲がった谷底を形成する誤差曲面においても, 2 次の学習アルゴリズムを採用すれば, 効率良く誤差を減少できることが期待される.

5.5.2 パリティ問題

学習アルゴリズム評価のベンチマークとしてパリティ問題は広く採用されているので, 8- ビットのパリティ問題を用いて BPQ を 7 種のアルゴリズム (オンライン慣性項付き BP, 適応 BP, SCG, OSS2, BFGS1, BFGS2, BFGS3) と比較した. なお, この問題規模は比較的小さいので, BPQ ではオリジナル BFGS 法を採用した. 実験には, 中間ユニット数 8 のネットワークを用い ($h = 8$, $N = 82$), 可能な全ての入力パタンを訓練事例として用いた ($m = 256$). なお, 目標出力値は 1 または 0 に設定した. 各アルゴリズムのパラメータについては, オフライン慣性項付き BP の学習定数 η を 0.1 に設定したことを除き, 前回の実験と全て同じ値に設定した. また, 全てのアルゴリズムの結合重みの初期値を, $[-1, 1]$ の範囲の一様分布に基づいて生成した. 実験では, 最大 CPU 処理時間を 100 秒に設定した. ただし, $\|\nabla f(\mathbf{w})\| < 10^{-8}$ のときには, 反復を終了させた.

図 5.4(a) では, 100 試行の平均 RMSE (平均自乗誤差の平方根: $\sqrt{2f(\mathbf{w})/m}$) を用いて, BPQ と 1 次学習アルゴリズムの収束性を評価する. これらのアルゴリズムの中で, BPQ が最も速く収束し, 反復を初期の段階で停止できることが分る. 図 5.4(b) では, BPQ と 2 次学習アルゴリズムの収束性を評価する. BPQ と比較して, 他のアルゴリズムの収束は遅かった. これは, 準ニュートン法において, 最適探索幅計算が重要な役割を果たしていることを示している.

5.5.3 音声合成問題

大規模問題での BPQ の有効性を評価するため, parrot-like speaking データ [76] を用いた音声合成問題での実験を行った. この問題は, 与えられた状況に依存して, 現在までの音声波形より, それに続く音声波形を適切に求める自己回帰問題であり, 詳細には, 入力情報は 8 ユニットの状況ベクトルと 10 ユニットの音性波形からなり, 入力した音性波形の次の要素の値を出力ユニットの目標出力値とする. すなわち, 入力ユニット数の合計は 18 ($n = 18$) である. 実験では, 中間ユニット数を 36 ($h = 36$), 訓練事例数を 12,800 ($m = 12{,}800$) に設定した. すなわち, 結合重みの総数は 721 ($N = 721$) である. 実験では, BPQ を 8 種のアルゴリズム (オンライン BP, オンライン慣性項付き

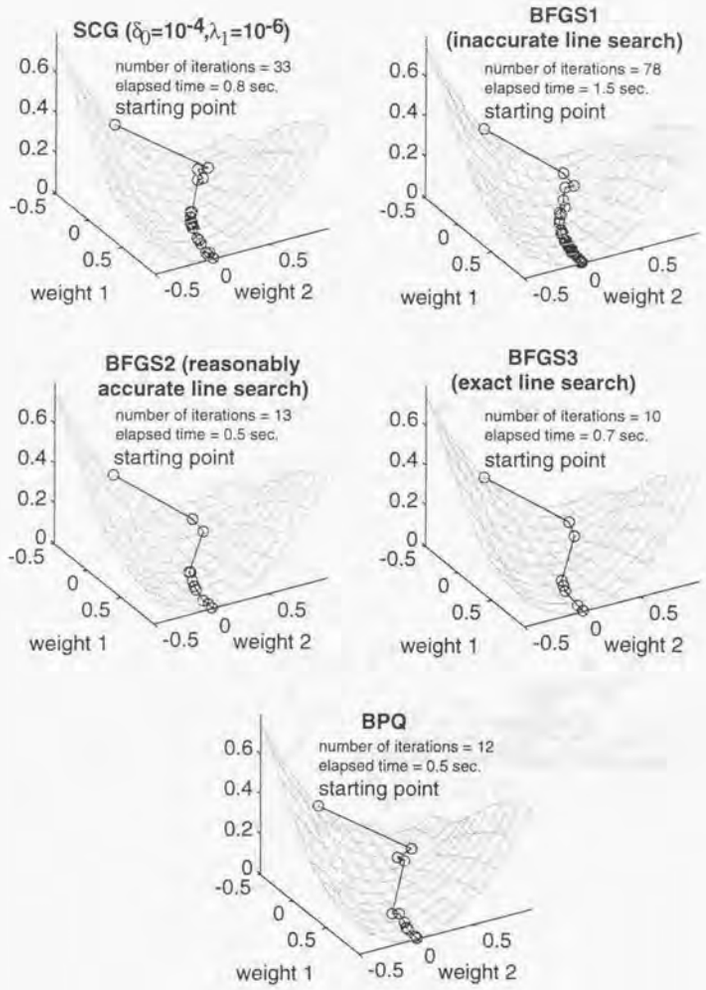
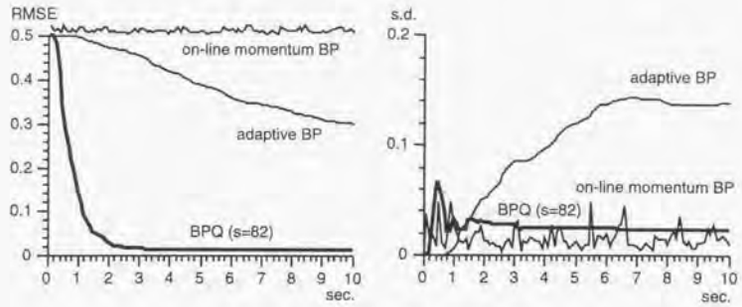
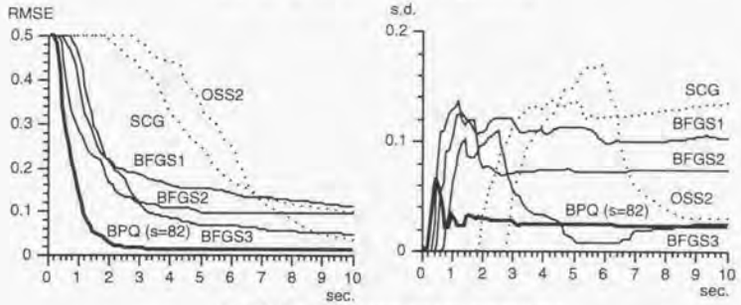


図 5.3: 2 次学習アルゴリズムの学習軌跡



(a) BPQ vs. first-order algorithms



(b) BPQ vs. second-order algorithms

図 5.4: パリティ問題での収束性

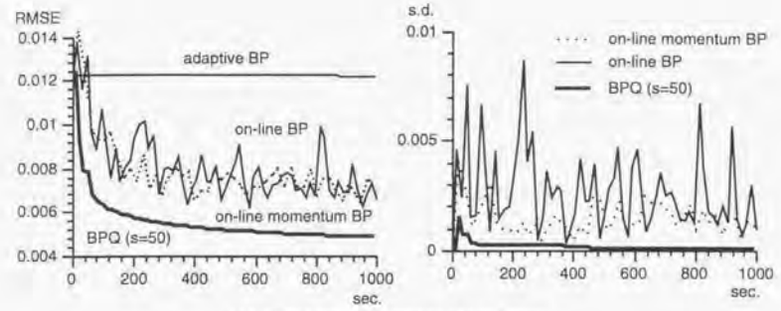
BP, 適応 BP, SCG, OSS2, BFGS1, BFGS2, BFGS3) と比較した. 各アルゴリズムの初期結合重みの設定法については, まず, 入力ユニットと中間ユニット間の結合重みを $[-1, 1]$ の範囲のランダムな値に設定し, 一方, 中間ユニットと出力ユニット間の結合重みを全て 0 に設定した. ただし, 出力ユニットのバイアス項の値は, 全訓練事例の目標出力値の平均値に設定した. アルゴリズムの終了判定については, 最大反復回数に上限を設け, その数を 100 に設定した.

図 5.5(a) では, BPQ と 1 次学習アルゴリズムを, 各アルゴリズムの 10 回の試行の平均 RMSE と標準偏差を用いて比較した. なお, 適応 BP の標準偏差は, 常にほぼ 0 だったので, 図示していない. オンライン BP では, 学習定数 η を 1.0 に設定したとき, 実験では最も誤差を減少できたが, 学習曲線は大きく振動した. 慣性項の係数 α を 0.9 に設定したオンライン慣性項付き BP では, 学習定数 η を 1.0 に設定したとき, 実験では最も誤差を減少でき, 学習曲線は大きく振動しなかったが, オンライン BP とほぼ同等の収束性であった. 適応 BP では, 学習定数 η を 0.1 または 1.0 に設定したが, いずれの値でもほとんど誤差を減少させることができなかった. BPQ では, パラメータ s を 50 に設定したところ, これらのアルゴリズムの中で最も速く収束し, 反復を初期の段階で停止できることが分る.

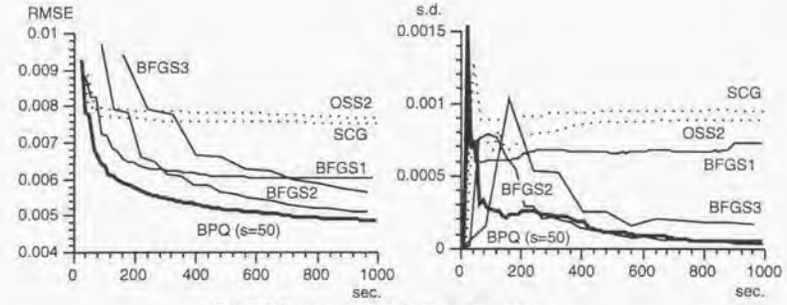
図 5.5(b) では, 2 次学習アルゴリズムの収束性を 10 回の試行の平均値で比較する. BPQ との比較では, BFGS1 はあまり誤差を減少できず, BFGS3 では, BPQ の最良レベルまで RMSE を減少させるのに, 多くの計算時間を必要とした. 一方, BFGS2 の収束性は, BFGS1 や BFGS3 に優り, BPQ に近いものであった. これらの結果より, 準ニュートン法において収束性を向上させるのに, 探索幅計算が重要であることが分る.

図 5.5(c) では, 全てのアルゴリズムの 1 反復の平均処理時間を比較する. BPQ と OSS2 の 1 反復時間は適応 BP とほぼ等価であった. オンライン BP は, 各事例に対して結合重みを修正するので約 1.3 倍遅くなり, オンライン慣性項付き BP では, さらに慣性項係数の乗算が必要となるため, 約 2 倍遅くなった. BFGS1 では, 探索幅を縮めるために数回の目的関数の評価が必要な場合があり, 約 1.5 倍遅くなった. BFGS2 では, 終了条件を満たすための付加的な直線探索計算が必要なため, 約 4 倍となり, BFGS3 では, 厳密直線探索のため, 約 7 倍もの計算時間を必要とした. SCG では, 1 反復で 2 度勾配ベクトルを計算するので, 約 2 倍の計算時間を必要とした.

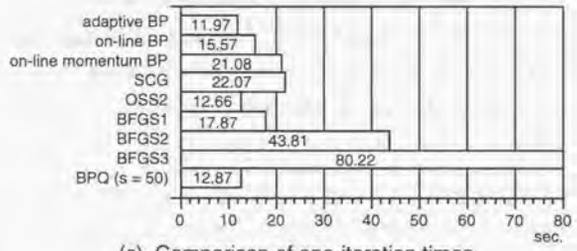
図 5.6 では, 4 種の直線探索法について, それぞれの収束性に対する局部性パラメータ s の影響を比較する. ただし, 図の曲線は 10 回の試行の平均 RMSE である. 一般に, 直線探索法の違いに対して, 局部性パラメータの値の影響は小さかった. BPQ では, $s = 5$ と $s = 50$ でほぼ同等の収束性を示したが, $s = 2$ では, $s \geq 5$ のときよりも収束性が悪かった. 結論として, 探索幅計算法は準ニュートン法の収束性に重要な役割果たすが, 一方, 小記憶 BFGS 法は, 適切な s を用いれば, 収束性を変えずに記憶容量を大幅に減少できることが示された.



(a) BPQ vs. first-order algorithms



(b) BPQ vs. second-order algorithms



(c) Comparison of one-iteration times

図 5.5: 音声合成問題での収束性

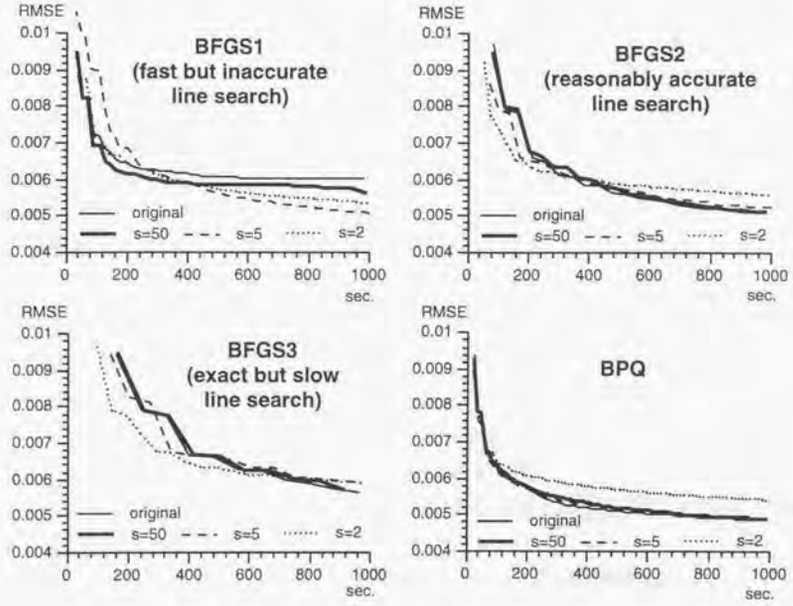


図 5.6: 局部性パラメータの影響

5.6 結言

本章では, 3 層ネットワークに対する 2 次の学習アルゴリズム BPQ を提案した. BPQ は準ニュートン法をベースとし, 探索方向を小記憶 BFGS で計算し, 最適探索幅を 2 次近似の最小点として効率良く計算することを特徴とする. 人工問題, パリティ問題, および, 音声合成問題を用いた実験では, 他の代表的な学習アルゴリズムより BPQ が効率良く働くことを示した. さらに, 探索幅計算法は準ニュートン法の収束性に重要な役割果たし, 一方, 小記憶 BFGS 法は収束性を変えずに記憶容量を大幅に減少できることを示した.

なお, 収束性能だけでなく, 汎化性能はニューラルネット学習における重要な評価尺度である. 汎化性能を向上させるアプローチの一つには, ニューラルネットの学習目的関数に, 訓練事例に関する誤差項だけでなく, 結合重みの値が大きくなることを抑制する正則化 (ペナルティ) 項を付加する方法がある. 正則化項にはさまざまな提案 [35, 81, 8, 34, 39, 55, 139] があるものの, 自乗値正則化項は BPQ と組合せることにより, 適切な正則化率において, 高い汎化性能を示すとともに, 他の組合せと比較して最も高速に収束することを示した [117, 118, 102, 119]. しかし, そこでは適切な正則化率を予め設定することが問題となる. これに対して, MDL 原理 [90] に基づく新正則化法を考案し, 実験により考案法の有効性を示している [101, 123].

第 6 章

ニューラルネットを用いた法則発見法: RF5

6.1 序言

科学的発見を支援するシステムにおいて, データからの数法則 (numeric law) の発見は中心的な課題である. このようなシステムでは, 例えば, 金属の電気伝導度 σ, 入射光の周波数 ν, および, 金属の光の反射率 R から, ハーゲン - ルーベンスの法則 $R = 1 - 2(\sigma/\nu)^{1/2}$ を発見できる.

BACON システム [48, 49] での先駆的な研究の後, いくつかの方法 [50, 23, 51, 126, 132] が提案されている. ただし, これらの基本的な探索戦略はほとんど同じである. すなわち, 乗算, 除算, または, 予め定義した関数を用いて, 2 つの既存変数を組み合わせ, 新たな変数を再帰的に生成する. BACON と FAHRENHEIT [50] は trend detector と呼ばれる尺度で変数を組み合わせ, ヒューリスティックスを加えた深さ優先探索を行なう. ABACUS [23] は proportional graph を生成し, ビーム探索を行なう. IDS [51] では, 相関分析が適用され, ビーム探索を行なう. E* アルゴリズム [126] は 2 変数の法則のみを対象とする. また, Sutton-Matheus アルゴリズム [132] は回帰を行ない, 自乗誤差と変数値の自乗についての相関 [125] を用いて変数を組み合わせる.

これらの既存法に対して, 以下の課題が指摘できる. 第 1 に, 2 つの変数を順番に組み合わせて新たな変数を作るので, 多くの変数からなるデータにおいて複雑な法則を探索すれば, 容易に組合せ爆発が起き, また, 探索パラメータが適切でなければ, 望ましい法則を発見できないことが予想される. 第 2 に, 法則に現れる指数の値が整数ではないとき, 適当な関数 (e.g. $\sigma^{1/2}$) を予め定義しなければ, 法則の発見は困難になる. しかし, 多くの場合, 事前知識はない. 第 3 に, 現実の観測データは確実にノイズを含むが, 既存法は比較的ノイズに弱いことが指摘されている [51, 126].

ニューラルネットを用いるアプローチは上記の課題解決に有望であると考える. 指数の値が整数に制限されない一般化した多項式の各項を直接学習するには, 入力値の重み付け和の代りに, 入力値を結合重みで累乗した値の積を計算する product unit と呼ばれる計算ユニットが提案されている [19]. しかし, 標準的な BP アルゴリズム [93] では, このタイプのユニットを含むネットワークの学習は極め

て困難であることが報告されている [53]. これに対して, いくつかの学習アルゴリズムを組み合わせる方法 [53] などが提案されているが, BP と比較して, それらの有効性はあまり顕著ではない. また, これらの初期の研究では, 2 値データのみを扱い, 数法則の発見問題を対象としていない.

本章では, 数法則をニューラルネットを用いて発見する方法 RF5 [103, 116, 115, 100] を提案する. まず, 数法則の発見問題をニューラルネットの学習問題として定式化する. 次に, BPQ を学習アルゴリズムとし, 複数の学習結果から適切なものを選択する評価尺度を備えた RF5 アルゴリズムについて述べる. 最後に, 人工問題と現実問題による実験を行ない, RF5 の有効性を評価する.

6.2 ニューラルネットを用いた法則の発見

数法則の発見をニューラルネット [19] を用いて定式化する.

$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ を事例集合, $\mathbf{x}_t$ を n 次元入力ベクトル, y_t を $\mathbf{x}_t$ に対する目標出力値とする. ここでは,

$$y_t = c_0 + \sum_{i=1}^{h} c_i x_{t1}^{w_{i1}} \cdots x_{tn}^{w_{in}} \tag{6.1}$$

で表される数法則のクラスについて考える. ここで, 各パラメータ c_i, w_{ij} は未知の実数, h は未知の整数である. なお, 対象とする法則が周期関数や不連続関数からなる場合, (6.1) 式では厳密には対処できない. しかし, このような関数でも, 入力ベクトル $\mathbf{x}$ のレンジが限られている場合には, 有限項数の多項式を用いて, ある程度の精度で近似可能であり, さらに, 実数指数の多項式ならば, 各項の表現能力が向上するので, 少ない項数での近似が期待できる. 一方, (6.1) 式で c_0 を除いた定数項なしモデルも考えられるが, ここでの議論を簡潔にするため (6.1) 式のみを対象にする. 以下では, $(c_0, \cdots, c_h)^T$, $(w_{i1}, \cdots, w_{in})^T$ をそれぞれ $\mathbf{c}$, $\mathbf{w}_i$ と表記する. ただし, $\mathbf{a}^T$ は $\mathbf{a}$ の転置を意味する. また, 全てのパラメータからなる 1 つのベクトル $(\mathbf{c}^T, \mathbf{w}_1^T, \cdots, \mathbf{w}_h^T)^T$ を $\mathbf{\Phi}$ で表し, $N(= nh + h + 1)$ を $\mathbf{\Phi}$ の次元 (パラメータ数) とする.

必要ならば適当な値を各入力ベクトルの要素に加えることにより, $x_{ti} > 0$ を仮定できる. よって, (6.1) 式 は

$$y_t = c_0 + \sum_{i=1}^{h} c_i \exp\left(\sum_{j=1}^{n} w_{ij} \ln(x_{tj})\right) \tag{6.2}$$

と等価である. (6.2) 式 は各中間ユニットの活性化関数が $\exp(s) = e^s$ である 3 層ニューラルネットとみなすことができる. つまり, h, $\mathbf{w}_i$, および, $\mathbf{c}$ はそれぞれ中間ユニット数, 全入力ユニットと中間ユニット i との結合重み, および, 全中間ユニットと出力ユニットとの結合重みである. 以下では, 中間ユニット i の出力値を $v_{it} = v_i(\mathbf{x}_t; \mathbf{w}_i) = \exp\left(\sum_{j=1}^{n} w_{ij} \ln(x_{tj})\right)$, 出力ユニットの出力値を

$z_t = z(\mathbf{x}_t; \mathbf{\Phi}) = c_0 + \sum_{i=1}^{h} c_i v_i(\mathbf{x}_t; \mathbf{w}_i)$ で表す. なお, このタイプの中間ユニットは *product unit* [19] と呼ばれる. よって, (6.1) 式を対象とする数法則の発見問題は

$$f(\mathbf{\Phi}) = \frac{1}{2} \sum_{t=1}^{m} (y_t - z(\mathbf{x}_t; \mathbf{\Phi}))^2 \tag{6.3}$$

を最小化する $\mathbf{\Phi}$ を求めるニューラルネットの学習問題として定式化できる.

6.3 RF5 アルゴリズム

RF5 は, 中間ユニット数の異なるネットワークを学習して法則候補を作り出し, その中から適切なものを法則として特定する. 以下では, ネットワーク学習法と法則特定法の詳細について述べる.

6.3.1 ネットワーク学習法

初期の研究 [53] で報告されたように, 我々の実験でも, 標準的な BP による (6.3) 式の最小化は極めて困難であった. そこで, 常に効率良く望ましい結果を得るため, (6.3) 式の最小化には, 前章で提案した 2 次の学習アルゴリズム BPQ を採用する. すなわち, BPQ は探索方向を小記憶 BFGS 法で計算し, 最適ステップ幅を 2 次近似の最小点として計算する.

前章と同様に, $f(\mathbf{\Phi} + \lambda \Delta \mathbf{\Phi})$ を $\zeta(\lambda)$ で表せば, $\zeta(\lambda)$ の 2 次近似式の最小値は

$$\lambda = -\frac{\zeta'(0)}{\zeta''(0)} \tag{6.4}$$

で与えられる. 以下では, (6.3) 式で定義したニューラルネットでも, $\zeta'(0)$ と $\zeta''(0)$ を効率良く計算できることを示す. $\zeta(\lambda)$ を微分し, λ に 0 を代入すれば,

$$\zeta'(0) = -\sum_{t=1}^{m} (y_t - z_t) z'_t, \qquad \zeta''(0) = \sum_{t=1}^{m} ((z'_t)^2 - (y_t - z_t) z''_t)$$

となる. ここで, $z_t = z(\mathbf{x}_t; \mathbf{\Phi})$ の微分は $\left(\frac{d}{d\lambda} z(\mathbf{x}_t; \mathbf{\Phi} + \lambda \Delta \mathbf{\Phi})\right)_{(\lambda=0)}$ で定義され,

$$z'_t = \Delta c_0 + \sum_{i=1}^{h} (\Delta c_i v_{ti} + c_i v'_{ti}), \qquad z''_t = \sum_{i=1}^{h} (2 \Delta c_i v'_{ti} + c_i v''_{ti})$$

となる. ここで, $v'_{ti} = v_{ti} \times \sum_{j=1}^{n} \Delta w_{ij} \ln(x_{tj})$, $v''_{ti} = v'_{ti} \times \sum_{j=1}^{n} \Delta w_{ij} \ln(x_{tj})$ であり, Δc_i, Δw_{ij} は小記憶 BFGS 法で計算される c_i, w_{ij} の変化量である.

6.3.2 法則特定法

一般に, 与えられたデータ集合に対して, 最適な中間ユニット数を予め知ることはできない. また, データは普通ノイズを含むので, (6.3) 式を最小にする法則候補がベストとは限らない. よって, 中間

ユニット数を変えて発見した法則候補を適切に評価するための尺度が必要である．ここでは，ノイズを想定して，目標出力値とニューラルネットの出力値の差 $y-z$ が正規分布に従うと仮定する．すなわち，その対数尤度を

$$\log p(\mathbf{x}, y; \mathbf{\Phi}, \sigma) \;=\; -\log\sigma - \frac{1}{2\sigma^2}(y - z(\mathbf{x}; \mathbf{\Phi}))^2 + r \tag{6.5}$$

で定義する．ただし，σ は標準偏差を表し，正規化定数 r は $-(1/2)\log(2\pi)$ である．よって，最適な中間ユニット数を求めることは，最尤推定におけるモデル選択問題となるので，その評価尺度として MDL 基準 [90] を採用する．

訓練事例に対する負の対数尤度は

$$-\sum_{t=1}^{m}\log p(\mathbf{x}, y; \mathbf{\Phi}, \sigma) \;=\; m\log\sigma + \frac{1}{2\sigma^2}\sum_{t=1}^{m}(y - z(\mathbf{x}; \mathbf{\Phi}))^2 - mr \tag{6.6}$$

である．ただし，m は事例数である．(6.6) 式の最小化を考えれば，$\mathbf{\Phi}$ については，(6.3) 式の最小化と等価であり，ニューラルネットの学習で最尤推定量 $\hat{\mathbf{\Phi}}$ を得ることができる．一方，σ については，(6.6) 式を σ で微分して 0 とおけば，

$$\frac{m}{\sigma} - \frac{1}{\sigma^3}\sum_{t=1}^{m}(y - z(\mathbf{x}; \mathbf{\Phi}))^2 \;=\; 0$$

であり，よって，分散の最尤推定量 $\hat{\sigma}^2$ は

$$\sigma^2 \;=\; \frac{1}{m}\sum_{t=1}^{m}(y - z(\mathbf{x}; \mathbf{\Phi}))^2 \tag{6.7}$$

となる．なお，この $\hat{\sigma}^2$ の値は平均自乗誤差 (MSE) に他ならない．(6.7) 式を (6.6) 式に代入すれば，(6.6) 式の第 2 項はパラメータ数とは独立の値 $(m/2)$ となり，第 2 項と第 3 項は各モデルで共通なので，ここでは，次の MDL 値を評価尺度として採用する．

$$\mathrm{MDL} \;=\; 0.5m\log(\mathrm{MSE}) + 0.5N\log(m) \tag{6.8}$$

ただし，N は $\mathbf{\Phi}$ の総パラメータ数である．

以下では，(6.3) 式を目的関数とし，BPQ アルゴリズムと MDL 尺度を用いた数法則発見法を *RF5* (*Rule extraction from Facts version 5*) と呼ぶ．

6.4 実験による評価

人工問題と現実問題を用いて，RF5 の性能を評価する．

6.4.1 人工データ

Sutton-Matheus の問題 [132] とその修正問題を用いて，数法則発見アルゴリズム RF5 を評価した．オリジナル問題は

$$y \;=\; 2 + 3x_1x_2 + 4x_3x_4x_5 \tag{6.9}$$

を復元する問題である．事例については，$x_1, \cdots, x_5$ の各変数に $[0, 1]$ の範囲でランダムな値を与え，対応する y の値を (6.9) 式より計算する．ただし，変数の総数は 9 $(n = 9)$ であり，不要変数 $x_6, \cdots, x_9$ にも $[0, 1]$ の範囲でランダムな値を与え，合計 200 $(m = 200)$ 事例を生成した．実験では，各結合重みの初期値を，平均 0，分散 0.1 の正規分布に基づいて独立に生成し，MSE 値が十分小さいとき，

$$\frac{1}{m}\sum_{t=1}^{m}(y_t - z(\mathbf{x}_t; \mathbf{\Phi}))^2 \;<\; 10^{-8},$$

勾配ベクトルの大きさが十分小さいとき，

$$\frac{1}{N}\|\nabla f(\mathbf{\Phi})\|^2 \;<\; 10^{-8},$$

または，CPU 処理時間が 100 秒を越えたときに，アルゴリズムの反復を終了させた．

オリジナル問題

実験では，中間ユニット数を 1 から 3 まで変化させ $(h = 1, 2, 3)$，それぞれ 100 回の試行を行なった．実験結果の MSE 値，MDL 値，反復回数，および，処理時間 (秒) に関する基本統計量を表 6.1 に示す．表より，$h = 2$ で MDL 値が最小になり，正しい中間ユニット数を発見できたことが分かる．また，このとき 100 回全ての試行が最適解に収束した．発見した法則は

$$y \;=\; 2.000 + 3.000x_1^{1.000}x_2^{1.000} + 4.000x_3^{1.000}x_4^{1.000}x_5^{1.000}$$

である．ただし，各値を小数点第 4 位で四捨五入した．図より RF5 は元の法則を完全に復元できたことが分かる．ここで，各試行の平均反復回数は 93.7 回，平均処理時間は 0.878 秒であり，この実験に要した全処理時間は 4.4 分であった．

修正問題

指数の値が実数の場合での RF5 の有効性を評価するため，(6.9) 式の代わりに，

$$y \;=\; 2 + 3x_1^{-1}x_2^{3} + 4x_3x_4^{1/2}x_5^{-1/3} \tag{6.10}$$

表 6.1: オリジナル問題

ユニット数	MSE 値			MDL 値			反復回数		処理時間	
	best	avg.	s.d.	best	avg.	s.d.	avg.	s.d.	avg.	s.d.
1	0.126	0.13	0.0	-177.9	-178	0.0	71	1.0	0.32	0.01
2	0.000	0.0	0.0	-1786.4	-1786	0.0	81	8.4	0.68	0.07
3	0.000	0.0	0.0	-1759.9	-1717	64	130	29	1.65	0.36

表 6.2: 修正問題

ユニット数	MSE 値			MDL 値			反復回数		処理時間	
	best	avg.	s.d.	best	avg.	s.d.	avg.	s.d.	avg.	s.d.
1	1.317	1.32	0.00	56.7	57	0.0	70	5	0.32	0.02
2	0.000	0.03	0.17	-1786.4	-1731	314	116	27	0.97	0.23
3	0.000	0.00	0.00	-1760.0	-1727	47	240	104	3.02	1.31

を用いて実験を行なった．ただし，実験の条件はオリジナル問題のときと全て同じとした．結果を表 6.2 に示す．この実験でも正しい中間ユニット数を発見できた．しかし，$h = 2$ のとき，数回の試行は望ましくない局所最適解に収束した．発見した法則は

$$y = 2.000 + 3.000x_1^{-1.000}x_2^{3.000} + 4.000x_3^{1.000}x_4^{0.500}x_5^{-0.333}$$

である．これは (6.10) 式と等価である．既存法では，適切な関数を用意しなければ，このような法則を発見できないので，既存法と比べて，RF5 には重要な長所があることが示された．

ノイズ許容性

RF5 のノイズ許容性を評価するため，(6.9) 式，または，(6.10) 式で計算する各 y の値に，平均 0，分散 0.1 の正規分布に基づく独立なノイズを与えて実験を行なった．ただし，これ以外の実験の条件は，以前のものと全て同じとした．結果を表 6.3 と 6.4 に示す．最良の MSE 値は $h = 3$ のときであるが，最良の MDL 値は $h = 2$ のときであり，いずれの問題でも，正しい中間ユニット数を発見できた．オリジナルと修正問題で RF5 が発見した法則は

$$\begin{aligned} y = {} & 1.968 + 3.028x_1^{1.000}x_2^{0.969}x_4^{-0.007}x_5^{-0.007}x_6^{0.004}x_7^{0.008}x_8^{-0.007}x_9^{0.001} \\ & +3.880x_1^{-0.027}x_2^{-0.014}x_3^{1.025}x_4^{0.995}x_5^{1.048}x_6^{-0.008}x_7^{-0.020}x_8^{0.010}x_9^{-0.014} \end{aligned}$$

表 6.3: ノイズありオリジナル問題

ユニット数	MSE 値			MDL 値			反復回数		処理時間	
	best	avg.	s.d.	best	avg.	s.d.	avg.	s.d.	avg.	s.d.
1	0.160	0.16	0.0	-154.0	-154	0.0	68	4	0.31	0.02
2	0.009	0.01	0.0	-416.0	-416	0.0	93	9	0.77	0.07
3	0.008	0.01	0.0	-405.5	-405	0.8	784	81	9.80	1.02

表 6.4: ノイズあり修正問題

ユニット数	MSE 値			MDL 値			反復回数		処理時間	
	best	avg.	s.d.	best	avg.	s.d.	avg.	s.d.	avg.	s.d.
1	2.326	2.33	0.0	113.6	114	0.0	90	12	0.41	0.05
2	0.010	0.03	0.21	-403.9	-399	53	228	134	1.90	1.11
3	0.009	0.01	0.0	-388.9	-385	1.2	753	127	9.41	1.58

$$\begin{aligned} y = {} & 2.012 + 3.004x_1^{-1.000}x_2^{3.001}x_6^{-0.001}x_7^{0.001} \\ & +3.983x_1^{0.002}x_2^{-0.003}x_3^{1.022}x_4^{0.500}x_5^{-0.333}x_6^{-0.005}x_7^{-0.002}x_8^{0.003}x_9^{-0.007} \end{aligned}$$

である．各値を小数点第 2 位で四捨五入した結果は

$$y = 2.0 + 3.0x_1^{1.0}x_2^{1.0} + 3.9x_3^{1.0}x_4^{1.0}x_5^{1.0}$$

$$y = 2.0 + 3.0x_1^{-1.0}x_2^{3.0} + 4.0x_3^{1.0}x_4^{0.5}x_5^{-0.3}$$

となる．いくつかの値はわずかに異なるが，元の法則とほぼ等価な法則を発見できた．このことは，RF5 は頑強で，ある程度のノイズを許容できることを示している．

6.4.2 学習効率の評価

ここでは，法則発見問題における学習アルゴリズム BPQ の効率を評価する．

BP との比較

学習効率をグラフィカルに評価するため，2 変数からなる人工問題を作成した．法則の一般形を

$$y = x^{w_1} + w_2 \tag{6.11}$$

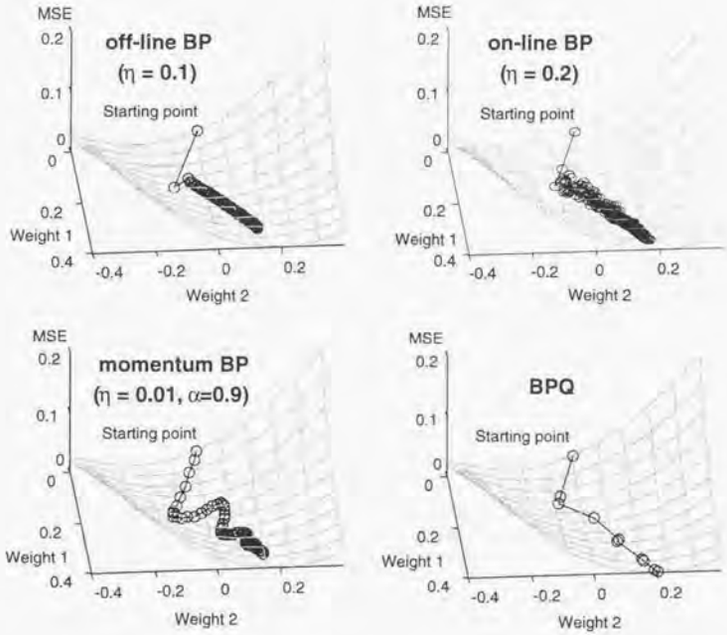


図 6.1: 学習軌跡

とし, $(w_1, w_2) = (0.4, 0.2)$ で真の法則を与えるとする. 入力事例 x_t は $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ の各要素とし, 目標出力値 y_t は (6.11) 式に真のパラメータを代入して各 x_t から計算した. すなわち, 最小値は真のパラメータの値で与えられる. 実験では, オフライン BP, オンライン BP, 慣性項付き BP, および, BPQ を比較した. 誤差曲面上において, $(w_1, w_2) = (0.0, 0.0)$ を初期値とし, 最大で 100 反復させた学習軌跡を図 6.1 に示す. ただし, 図の学習定数や慣性項の係数は試行錯誤で決定した. 図より, オフライン BP, オンライン BP, および, 慣性項付き BP では, 初期値を比較的最小値の近くに設定したにもかかわらず, 100 反復では最小値に到達できなかった. この理由は, 前章でも述べたように, 谷底近くでは, 連続する 2 つの勾配ベクトルの方向がほぼ逆向きになるためであり, これは 1 次の学習法の本質的な問題点である. 一方, BPQ では, 十数反復で効率良く最小値に到達できた. さらに, BPQ には試行錯誤で決定するパラメータがないので, 一般の問題への適用が容易となる.

適応 BP との比較

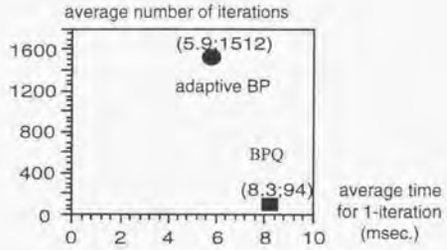
ノイズありの Sutton-Matheus のオリジナルと修正問題を用いて, BPQ の効率を評価した. この実験では, 標準的な BP では, すべての試行が収束しなかったので, Silva-Almeida の学習定数適応規則 [128] を用いたアルゴリズム (適応 BP) と比較した. なお, 適応 BP では, 各結合重み $\mathbf{\Phi}_i$ に対する学習定数 η_i は

$$\eta_i^k = \begin{cases} \eta_i^{k-1} \times u, & \text{if } \frac{\partial f(\mathbf{\Phi}^k)}{\partial \mathbf{\Phi}_i} \times \frac{\partial f(\mathbf{\Phi}^{k-1})}{\partial \mathbf{\Phi}_i} \geq 0, \\ \eta_i^{k-1} \times u^{-1}, & \text{otherwise} \end{cases}$$

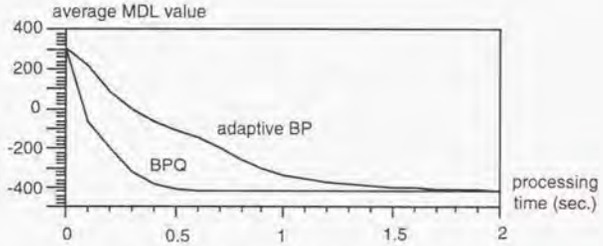
で調整される. ここで, k は反復回数を表し, パラメータ u は提案者が推奨するように 1.1 に設定した [128]. ただし, 目的関数の値が減少しないときには, 全ての学習定数の値はその半分の値に設定される.

実験結果を図 6.2 に示す. ただし, 図の値は 100 回の試行の平均である. 図 6.2(a) では, ノイズありオリジナル問題において, 1 反復の処理時間と収束までに要した反復回数の関係を示す. 図より, 適応 BP の 1 反復の処理時間は BPQ より僅かに少ないが, 適応 BP の反復回数は BPQ の 16.1 であり, 全体では, BPQ は適応 BP より 11.4 倍速いことが分かる. 図 6.2(b) では, ノイズありオリジナル問題において, MDL 値による収束性を比較する. 明らかに, BPQ の収束性は適応 BP より優れている. 図 6.2(c) では, ノイズあり修正問題において, MDL 値による収束性を比較する. この問題では, 適応 BP の全ての試行が収束しなかった.

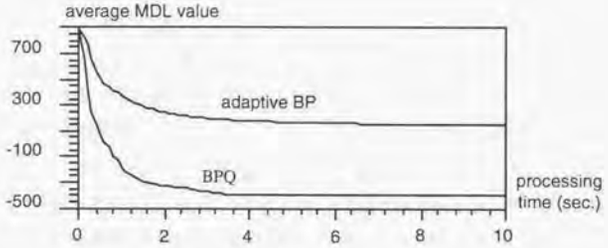
適応 BP がノイズあり修正問題をうまく学習できなかった理由については, 目標値 y_t に関する基本統計量がヒントを与える. 今回の実験では, ノイズありオリジナル問題での平均と標準偏差は 3.33 と 0.90 (値のレンジは $[1.8, 6.3]$) であったが, ノイズあり修正問題では, 19.40 と 60.95 (値のレンジは $[2.1, 555.9]$) であった. これは, 結合重みベクトルが変化すると, オリジナル問題と比較して, 修正問題



(a) 計算効率 (ノイズありオリジナル問題)



(b) 収束性 (ノイズありオリジナル問題)



(c) 収束性 (ノイズあり修正問題)

図 6.2: 学習結果

表 6.5: 現実データによる評価

法則名	参照法則	発見した法則	データ数
Hagen-Rubens の法則	$R = 1 - 2\left(\frac{\nu}{\sigma}\right)^{\frac{1}{2}}$	$R = 1.00 - 2.08\nu^{0.57}\sigma^{-0.57}$	9
Kepler の法則	$T = 0.41r^{\frac{3}{2}}$	$T = 0.41r^{1.50} + 0.19$	5
Boyle の法則	$V = 29.30/p$	$V = 29.05p^{-1.08} - 0.61$	19

での勾配ベクトルがより急激に変化することを意味する. これが適応 BP では学習できなかった理由であると考える.

6.4.3 現実データ

現実データとして, Hagen-Rubens の法則, Kepler の第 3 法則, および, Boyle の法則に従うデータを用いた実験を行なった. ここで, Hagen-Rubens の法則は金属の電気伝導度 σ, 入射光の周波数 ν, および金属の光の反射率 R の関係, Kepler の法則は太陽との距離 r と惑星の公転周期 T の関係, Boyle の法則は気体の圧力 p と体積 V の関係である. この実験では, 各データの事例数は少ないので, 中間ユニット数は 1 に固定した. なお, この場合でも, 定数項 c_0 を考慮するので, この学習は単純な回帰問題に帰着されない.

表 6.5 に, 参照法則, 発見した法則, および発見に用いたデータ数を示す. ただし, 結果は 10 回の試行で MDL 値を最小にしたものであり, 各値は小数点第 3 位で四捨五入した. Hagen-Rubens の法則では, 参照法則からかなり外れたデータも含まれるが, 参照法則と類似した法則を発見できた. 一方, Kepler の第 3 法則と Boyle の法則では, 望ましくない定数項が現れたが, 参照法則と類似した法則を発見できた. また, これらの結果からは, 定数項の値が比較的小さいので, それを 0 に固定したモデルでの試行が示唆される.

6.5 結言

数値データから未知の法則を発見するため, コネクショニストアプローチに基づく方法 RF5 を提案した. RF5 では, 数法則の発見問題がニューラルネットの学習問題として定式化され, 2 次学習アルゴリズム BPQ でネットワークを学習して法則候補を作り出し, その数法則候補から適切なものを MDL 基準を用いて選択する. 実験では, ある程度のノイズを含むデータからでも, RF5 は指数の値が整数に制限されない法則を効率良く発見することができた.

第 7 章

HME の構成的学習法

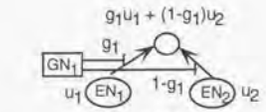
7.1 序言

Hierarchical Mixtures of Experts (HME) は複数のネットの調整を自己組織的に学習するモデルであり, その有効性は分類問題や関数近似問題にて示されている [41, 43, 135]. しかし, 既存の学習法では, 学習に先立ち予め適切な構造を与える必要があり, 結果の性能はその構造に依存する. また, 適切な構造を与えても, 期待した通りに学習が進む保証がない.

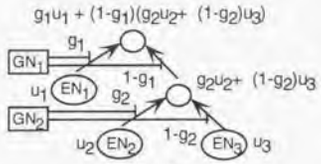
この課題の解決には, 枝刈り (pruning) アルゴリズム [88] と構成的 (constructive) アルゴリズム [46] による 2 つのアプローチが考えられる. 枝刈りアルゴリズムは, 特に, 不要な入力変数を多く含む問題において, ネットワークの汎化能力を向上させるのに重要な役割を果たすと期待できる. しかし, 初期ネットワークは最適なものを含まねばならず, その学習には一般に多くの計算量が必要となり, また, どのような初期ネットワークを設定すべきかについての事前知識も一般に得られない. 枝刈りアルゴリズムと比較して, 構成的アルゴリズムは上記の課題解決に有望であると考えられる.

多層ネットワーク (feed-forward networks) や分類木 (classification trees) においては, 多くの構成的アルゴリズム (e.g.[22, 11]) が提案されている. 一方, HME に関しては, 最近研究が始められた段階であり, 2 つのアルゴリズムが提案されている [98, 136]. すなわち, 1 つの構造の HME において, そのネットワークが収束してから拡張する著者らの方法 [98] と学習の過程で拡張する Waterhouse らの方法 [136] である. 両者を比較すれば, 後者の問題点は, ネットワークの拡張時点を定めるヒューリスティックスが必要になること, および, 単純に拡張するだけでは望ましい解に収束しないことが挙げられる. 実際, 後者の方法では, 枝刈りアルゴリズムとの併用が必要となる.

本章では, HME の構成的学習アルゴリズムを提案する [111, 114, 98, 122]. まず, 結合重みの初期化法, ネットワークの学習法, および, ネットワークの拡張法からなる提案アルゴリズムについて述る. 次に, パリティ問題と関数近似問題を用いた実験により, 提案法の有効性を評価する.



(a) 最も単純な HME



(b) 拡張した HME

図 7.1: HME の例

7.2 フレームワーク

$\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ を事例集合とする．ただし，$\mathbf{x}_t$ は n 次元入力ベクトル，y_t は $\mathbf{x}_t$ に対する目標出力値である．HME は複数の Expert Networks (ENs) と Gating Networks (GNs) から構成され，ENs を端点とし，GNs をその他のノードとする木として定義されるが，2 分木で多分木と同じ動作をする HME を構成できるので，本稿では 2 分木のみを考える．以下では，EN_i の結合重みベクトルを $\mathbf{w}_i = (w_{i0}, \cdots, w_{in})^T$ で表し，学習過程では，その出力値を

$$u_{ti} \;=\; u_i(\mathbf{x}_t, y_t; \mathbf{w}_i) \;=\; \exp\Big(-\frac{1}{2}(y_t - \mathbf{w}_i^T \mathbf{x}_t)^2\Big)$$

で定義する．ただし，$\mathbf{a}^T$ は $\mathbf{a}$ の転置を表す．GN_i の結合重みベクトルを $\mathbf{v}_i = (v_{i0}, \cdots, v_{in})^T$ で表し，多分木では soft-max 関数であるが，2 分木では GN がシグモイド関数となるので，その出力値を

$$g_{ti} \;=\; g_i(\mathbf{x}_t; \mathbf{v}_i) \;=\; \Big(1 + \exp(-\mathbf{v}_i^T \mathbf{x}_t)\Big)^{-1}$$

で定義する．但し，w_{i0}, v_{i0} はバイアス項であり，$x_{t0} = 1$ とする．いま，EN_1, EN_2 および GN_1 からなる最も単純な HME を (GN_1, EN_1, EN_2) で表し，その出力値を $g_1u_1 + (1-g_1)u_2$ とする．また，任意の HME は，$(GN_1, EN_1, (GN_2, EN_2, EN_3))$ のようなリスト構造で表現でき，その出力値を $g_1u_1 + (1-g_1)(g_2u_2 + (1-g_2)u_3)$ のように再帰的に定義することができる (図 7.1)．

7.3 構成的学習アルゴリズム

7.3.1 関係行列と目的関数

以下の説明のため，EN_i と GN_j の関係を表す行列 $\mathbf{R}$ を導入する．関係行列 $\mathbf{R}$ の要素 r_{ij} は $\{1, -1, 0\}$ の 3 値をとり，$r_{ij} = 1$ ならば $g_j u_i$ を，$r_{ij} = -1$ ならば $(1-g_j)u_i$ を，$r_{ij} = 0$ ならば EN_i と GN_j は互いに関係のないことを表す．例えば，(GN_1, EN_1, EN_2) の関係行列 $\mathbf{R}^{(2)}$ と $(GN_1, EN_1, (GN_2, EN_2, EN_3))$ の関係行列 $\mathbf{R}^{(3)}$ は以下となる．

$$\mathbf{R}^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \qquad \mathbf{R}^{(3)} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}$$

以下では，EN の個数が c である HME$^{(c)}$ において，全ての結合重みからなるベクトルを $\mathbf{\Phi}^{(c)} = (\mathbf{v}_1^T, \cdots, \mathbf{v}_{c-1}^T, \mathbf{w}_1^T, \cdots, \mathbf{w}_c^T)^T$ で表すと，$\mathbf{\Phi}^{(c)}$ の総パラメタ数は $N^{(c)} = (2c-1)(n+1)$ となる．したがって，関係行列 $\mathbf{R}^{(c)}$ を用いれば，EN_i に対する積項は

$$h_{ti} \;=\; h_i(\mathbf{x}_t; \mathbf{\Phi}^{(c)}) \;=\; \prod_{\{j | r_{ij}^{(c)} = 1\}} g_{tj} \times \prod_{\{j | r_{ij}^{(c)} = -1\}} (1 - g_{tj})$$

となるので，HME$^{(c)}$ の学習は

$$L(\mathbf{\Phi}^{(c)}) \;=\; \sum_{t=1}^{m} \log\left(\sum_{i=1}^{c} h_{ti} u_{ti}\right) \tag{7.1}$$

を最大化する $\mathbf{\Phi}^{(c)}$ を求める最適化問題として定義できる．ここで，結合重み $(\mathbf{v}_i,\ \mathbf{w}_i)$ に対する (7.1) 式の微分は，対数微分 $(a'(x) = a(x)(\log(a(x)))')$ を用いて，以下のように計算できる．

$$\nabla_{\mathbf{v}_i} L(\mathbf{\Phi}) \;=\; \sum_{t=1}^{m} \left\{ \frac{\sum_{\{j | r_{ji}=1\}} h_{tj} u_{tj} (1 - g_{ti})}{\sum_{k=1}^{c} h_{tk} u_{tk}} - \frac{\sum_{\{j | r_{ji}=-1\}} h_{tj} u_{tj} g_{ti}}{\sum_{k=1}^{c} h_{tk} u_{tk}} \right\} \mathbf{x}_t$$

$$\nabla_{\mathbf{w}_i} L(\mathbf{\Phi}) \;=\; \sum_{t=1}^{m} \frac{h_i u_i (y - \mathbf{w}_i^T \mathbf{x}_t)}{\sum_{k=1}^{c} h_{tk} u_{tk}} \mathbf{x}_t$$

7.3.2 アルゴリズムの概要

提案アルゴリズムは，HME$^{(c)}$ の学習を行ない，学習後の EN_i $(i = 1, \cdots, c)$ の中で誤差が最も大きい EN_b を選択し，EN_b を (GN_c, EN_b, EN_{c+1}) に置き換え，HME$^{(c+1)}$ を構成する処理を繰り返す．この基本的なアイデアは CART [11] などの既存法で採用されている "divide-and-conquer" アプローチに基づくものである．以下に，提案する構成的アルゴリズムの要点を示す．

Step 1: $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{v}_1$ を初期化し，$\mathbf{R}^{(2)} = (1, -1)^T$, c=2 とする．

Step 2: HME$^{(c)}$ の学習を実行する．

Step 3: 終了条件を満たせば反復を停止する.

Step 4: 拡張箇所となる EN_b を選択する.

Step 5: $\mathbf{w}_{c+1}$, $\mathbf{v}_c$ を初期化し, $\mathbf{R}^{(c+1)}$ を計算する.

Step 6: $c = c+1$ とし, **Step 2** へ戻る.

7.3.3 初期化法 (Step 1)

パリティ問題を用いた予備実験では, $\Phi^{(2)}$ の初期値を 0 近くの小さな値としてランダムに設定したとき, 多くの試行は $\hat{\Phi}^{(2)}$ $(\mathbf{w}_1 = \mathbf{w}_2 = \hat{\mathbf{w}},\ \mathbf{v}_1 = \mathbf{0})$ へ収束した. ただし, $\hat{\mathbf{w}}$ は全事例に対する最小自乗解, すなわち,

$$\hat{\mathbf{w}} \;=\; \arg\min\left\{\sum_{t=1}^{m}(y_t - \mathbf{w}^T\mathbf{x}_t)^2\right\}$$

である. さらに, $\hat{\Phi}^{(2)}$ に対して, 各値が小さいランダムなベクトル $\Delta\Phi$ を加えて学習を再開すれば,

$$L(\hat{\Phi} + \Delta\Phi) \;<\; L(\hat{\Phi})$$

が多くの試行で成り立ち, それらの多くで $\Phi^{(2)}$ は元の点 $\hat{\Phi}^{(2)}$ に収束した.

以下では, ヘス行列 (Hessian matrix) $\nabla^2 L(\Phi_0^{(2)})$ を調べることにより, こようなことが起こる理由を解析する. 一般に, $\sum_{t=1}^{m}\mathbf{x}_t\mathbf{x}_t^T$ は正定値 (positive definite) であると仮定できる. また,

$$\sum_{t=1}^{m}(y_t - \hat{\mathbf{w}}^T\mathbf{x}_t)\mathbf{x}_t \;=\; \mathbf{0}$$

であることを確認する.

まず, $\hat{\Phi}^{(2)}$ が $\mathbf{w}_1 = \mathbf{w}_2 = \hat{\mathbf{w}}$ かつ $\mathbf{v}_1 = \mathbf{0}$ のとき,

$$\nabla L(\hat{\Phi}^{(2)}) \;=\; \mathbf{0}$$

となる. なぜなら, $\hat{\Phi}^{(2)}$ において, $u_{t1} = u_{t2}$ かつ $h_{t1} = h_{t2} = g_{t1} = 0.5$ より,

$$\begin{aligned}
\nabla_{\mathbf{w}_i} L(\hat{\Phi}^{(2)}) \;&=\; \sum_{t=1}^{m}\frac{h_{ti}u_{ti}(y_t - \mathbf{w}_i^T\mathbf{x}_t)}{\sum_{k=1}^{2}h_{tk}u_{tk}}\mathbf{x}_t \\
&=\; \frac{1}{2}\sum_{t=1}^{m}(y_t - \hat{\mathbf{w}}^T\mathbf{x}_t)\mathbf{x}_t \\
&=\; \mathbf{0} \\
\nabla_{\mathbf{v}_1} L(\hat{\Phi}^{(2)}) \;&=\; \sum_{t=1}^{m}\frac{h_{t1}u_{t1}(1-g_{t1}) - h_{t2}u_{t2}g_{t1}}{\sum_{k=1}^{2}h_{tk}u_{tk}}\mathbf{x}_t \\
&=\; \mathbf{0}
\end{aligned}$$

となるからである.

一方, 2 次微分を計算すれば, 目標出力値が $[0,1]$ で正規化されているとき, ヘス行列 $\nabla^2 L(\Phi^{(2)})$ は一般に半負定値 (semi-negative definite) となることが分る. 実際に 2 次微分を計算すれば,

$$\begin{aligned}
\nabla_{\mathbf{v}_i}\nabla_{\mathbf{v}_i} L(\hat{\Phi}^{(2)}) \;&=\; \nabla_{\mathbf{v}_1}\nabla_{\mathbf{w}_i} L(\hat{\Phi}^{(2)}) \;=\; \mathbf{0} \\
\nabla_{\mathbf{w}_i}\nabla_{\mathbf{w}_i} L(\hat{\Phi}^{(2)}) \;&=\; \frac{1}{4}\sum_{t=1}^{m}(y_t - \hat{\mathbf{w}}^T\mathbf{x}_t)^2\mathbf{x}_t\mathbf{x}_t^T - \frac{1}{2}\sum_{t=1}^{m}\mathbf{x}_t\mathbf{x}_t^T \\
\nabla_{\mathbf{w}_1}\nabla_{\mathbf{w}_2} L(\hat{\Phi}^{(2)}) \;&=\; -\frac{1}{4}\sum_{t=1}^{m}(y_t - \hat{\mathbf{w}}^T\mathbf{x}_t)^2\mathbf{x}_t\mathbf{x}_t^T
\end{aligned}$$

となる. ここで, $(n+1)\times(n+1)$ 行列 $\mathbf{A}, \mathbf{B}$ を

$$\begin{aligned}
\mathbf{A} \;&=\; \frac{1}{4}\sum_{t=1}^{m}\mathbf{x}_t\mathbf{x}_t^T \\
\mathbf{B} \;&=\; \frac{1}{4}\sum_{t=1}^{m}\left(1 - (y_t - \hat{\mathbf{w}}^T\mathbf{x}_t)^2\right)\mathbf{x}_t\mathbf{x}_t^T
\end{aligned}$$

とおけば, 直行行列 $\mathbf{P}$ を用いて, $\nabla^2 L(\hat{\Phi}^{(2)})$ は

$$\nabla^2 L(\hat{\Phi}^{(2)}) \;=\; \mathbf{P}^T\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\mathbf{A} & 0 \\ 0 & 0 & -\mathbf{B} \end{pmatrix}\mathbf{P}$$

$$\text{ただし,}\qquad \mathbf{P} = \begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & \mathbf{I} \\ 0 & \mathbf{I} & -\mathbf{I} \end{pmatrix}$$

と表せる. ここで, $\mathbf{I}$ は $(n+1)\times(n+1)$ 単位行列である. 明らかに, $\mathbf{A}$ は正定値であり, 目標出力値が $[0,1]$ で正規化されているときに, $\mathbf{B}$ は一般に正定値となる. よって, $\hat{\Phi}^{(2)}$ は一般に局所的最適解 (weak local maximum) であることが分る.

しかるに, $\Delta\Phi$ として, $\Delta\mathbf{w}_1 = \Delta\mathbf{w}_2 = 0$, $\Delta\mathbf{v}_1$ をランダムな値とすれば, この局所的最適解から容易に脱出することができる. ここで, g_{t1} の値を 0.5 の近くでランダムに設定するため, $\{v_{11}, \cdots, v_{1n}\}$ の値を $[-1,1]$ の範囲でランダムに設定し, v_{10} の値を

$$v_{10} \;=\; -\sum_{i=1}^{n}v_{1i}\left(\sum_{t=1}^{m}x_{ti}\right)$$

で設定する. すなわち, $\mathbf{v}_1$ は入力ベクトルの重心を含むランダムな超平面となり, この超平面により, 訓練事例はほぼ同数の 2 つのグループに分割される. つまり, この $\Delta\Phi$ を用いれば,

$$\begin{aligned}
L(\hat{\Phi}) \;&=\; L(\hat{\Phi} + \Delta\Phi) \\
\nabla_{\mathbf{w}_1} L(\hat{\Phi} + \Delta\Phi) \;&=\; \sum_{t=1}^{m}h_{t1}(y - \hat{\mathbf{w}}^T\mathbf{x}_t)\mathbf{x}_t \;\neq\; 0
\end{aligned}$$

となるので, 次の反復で目的関数の値を確実に増加させることができる.

7.3.4 訓練法 (Step 2)

HME における既存の学習アルゴリズム [43, 135] は EM (Expectation Maximization) アルゴリズム [16] に基づくものである．しかし，HME への適用では，M-ステップにおいて ENs の結合重みを更新するのに IRLS (Iterative Reweighted Least Squares) が必要となるが，この処理は数値的に不安定になる傾向がある [135]．我々の予備実験でもこの点が問題となったので，HME の学習には，準ニュートン法 [29] をベースとする 2 次学習アルゴリズム [108, 99] を採用する．

このアルゴリズムは収束するまで次の処理を繰り返す．まず，勾配ベクトルを求めた後，探索方向 $(\Delta\mathbf{\Phi})$ を小記憶 BFGS (Broydon-Fletcher-Goldfarb-Shanno) 法に基づき計算する．次に，$L(\mathbf{\Phi}+\lambda\Delta\mathbf{\Phi})$ を最大化する最適探索幅 λ を 2 次近似の最大点として計算する．以下では，(7.1) 式で定義した $\mathrm{HME}^{(c)}$ ネットワークにおいて，最適探索幅を効率良く計算できることを示す．

λ は $L(\cdot)$ の唯一の変数なので，$L(\mathbf{\Phi}+\lambda\Delta\mathbf{\Phi})$ を単に $\zeta(\lambda)$ で表す．2 次近似の最大点は

$$\lambda = -\frac{\zeta'(0)}{\zeta''(0)} \tag{7.2}$$

で与えられる．$\zeta(\lambda)$ を微分して，$\lambda=0$ とおけば，

$$\begin{aligned}
\zeta'(0) &= \sum_{t=1}^{m}\frac{\sum_{i=1}^{c}(h'_{ti}u_{ti}+h_{ti}u'_{ti})}{\sum_{i=1}^{c}h_{ti}u_{ti}} \\
\zeta''(0) &= \sum_{t=1}^{m}\left\{-\left(\frac{\sum_{i=1}^{c}(h'_{ti}u_{ti}+h_{ti}u'_{ti})}{\sum_{i=1}^{c}h_{ti}u_{ti}}\right)^2+\frac{\sum_{i=1}^{c}(h''_{ti}u_{ti}+2h'_{ti}u'_{ti}+h_{ti}u''_{ti})}{\sum_{i=1}^{c}h_{ti}u_{ti}}\right\},
\end{aligned}$$

となり，h_{ti} と u_{ti} に対する微分は

$$\begin{aligned}
h'_{ti} &= h_{ti}\sum_{\{j|r_{ij}=1\}}\Delta\mathbf{v}_j^T\mathbf{x}_t - h_{ti}\sum_{\{j|r_{ij}\in\{1,-1\}\}}g_{tj}\Delta\mathbf{v}_j^T\mathbf{x}_t \\
h''_{ti} &= h'_{ti}\sum_{\{j|r_{ij}=1\}}\Delta\mathbf{v}_j^T\mathbf{x}_t - h'_{ti}\sum_{\{j|r_{ij}\in\{1,-1\}\}}g_{tj}\Delta\mathbf{v}_j^T\mathbf{x}_t - h_{ti}\sum_{\{j|r_{ij}\in\{1,-1\}\}}g_{tj}(1-g_{tj})(\Delta\mathbf{v}_j^T\mathbf{x}_t)^2 \\
u'_{ti} &= u_{ti}(y-\mathbf{w}_i^T\mathbf{x}_t)\Delta\mathbf{w}_i^T\mathbf{x}_t \\
u''_{ti} &= u'_{ti}(y-\mathbf{w}_i^T\mathbf{x}_t)\Delta\mathbf{w}_i^T\mathbf{x} - u_{ti}(\Delta\mathbf{w}_i^T\mathbf{x}_t)^2
\end{aligned}$$

となる．ここで，$\Delta\mathbf{w}_i$ と $\Delta\mathbf{v}_i$ は，それぞれ $\mathbf{w}_i$ と $\mathbf{v}_i$ に対する探索方向を表す．

ここで，(7.2) 式を用いた最適探索幅計算の計算量について考察する．明らかに，EN_i と GN_i のそれぞれと各入力ベクトル $\mathbf{x}_t$ において内積を計算しなければならない．つまり，$(\Delta\mathbf{w}_i)^T\mathbf{x}_t$ と $(\Delta\mathbf{v}_i)^T\mathbf{x}_t$ を計算しなければならないので，EN 数が c，GN 数が $c-1$，事例数が m であることより，まず $(2c-1)nm$ 回の乗算が必要となる．一方，残りの計算は $O(cm)$ 回の乗算で完了するので，$N=(2c-1)(n+1)$ より，全体の計算量は $Nm+O(cm)$ となる．つまり，この計算量は $L(\mathbf{\Phi})$ の値を求める計算量とほぼ等価である．しかるに，次の反復での ENs と GNs の各値は

$$u_{ti} = \exp\left(-\frac{1}{2}(y_t-\mathbf{w}_i^T\mathbf{x}_t-\lambda(\Delta\mathbf{w}_i)^T\mathbf{x}_t)^2\right)$$

$$g_{ti} = \left(1+\exp(-\mathbf{v}_i^T\mathbf{x}_t-\lambda(\Delta\mathbf{v}_i)^T\mathbf{x}_t)\right)^{-1}$$

で求められ，これらの式に現れる内積は既に計算されているので，次の反復で $L(\mathbf{\Phi})$ の値を求める計算量は $O(cm)$ となる．したがって，最適探索幅計算による処理負荷は殆んどないことが分る．

7.3.5 結合重みの縮小

一般に，HME の構成が進むにつれて，GNs の結合重みの値は非常に大きくなる．すると，GNs の出力値は 0 または 1 に近づき，そこでの微係数はほとんど 0 となる．この状況では，GNs の結合重みはほとんど修正されなくなるので，それらの結合重みの値を縮小すべきである．なお，結合重みの縮小は sigmoid 関数の非線型性を減少させることと等価である．

我々の予備実験において，全ての GNs の結合重みを同時に縮小したときには，僅かな縮小ではあまり効果がなく，一方，大きく縮小すれば，これまでに HME が学習したものを破壊してしまう傾向があった．予め適切な縮小係数を知ることは困難なので，ここでは，各 GN 毎に結合重みを縮小する方法を採用する．すなわち，$\mathrm{HME}^{(c)}$ の学習後，

$$\mathbf{v}_k^{(new)} = \gamma\mathbf{v}_k^{(old)}$$

と設定し，$\mathrm{HME}^{(c)}$ を再学習させた．ただし，この処理は $k=1$ から $k=c-1$ まで順番に繰り返す．我々の実験では，γ を 0.1 に設定した．

7.3.6 終了判定条件 (Step 3)

終了条件には，AIC [1] や MDL [90] を採用することができる．評価尺度 $\mathrm{Cr}^{(c)}$ は

$$\mathrm{Cr}^{(c)} = -L(\mathbf{\Phi}^{(c)})+0.5N^{(c)}K,$$

で与えられ，K については，AIC では $K=2$，MDL では $K=\log(m)$ となる．すなわち，

$$\mathrm{Cr}^{(c)} \geq \mathrm{Cr}^{(c-1)}$$

で反復を終了させる．

7.3.7 拡張法 (Step 4)

$\mathrm{HME}^{(c)}$ を拡張するには，次の $\mathrm{Err}(EN_i)$ を最大にする EN_i を選択する．

$$\mathrm{Err}(EN_i) = \sum_{t=1}^{m}h_{ti}(y_t-\mathbf{w}_i^T\mathbf{x}_t)^2$$

表 7.1: パリティ問題

ビット数	4	5	6	7	8
収束回数	100	100	100	100	99
平均 EN 数	3.07	3.85	4.01	4.20	5.23
EN 数の分散	0.26	0.43	0.10	0.47	0.75
最小 EN 数	3	3	4	4	5

この式で, h_{ti} は各事例が EN_i に任される確率であり, 自乗項はその事例の誤差である. よって, 全体の和は EN_i に対する期待誤差となる. 以下では, EN_b が選択されたとする.

$\mathbf{R}^{(c+1)}$ の値は次のように計算する. まず, EN_{c+1} は EN_b の下に置かれるので, $1 \leq j \leq c-1$ では $r_{c+1,j} = r_{sj}$ のようにコピーする. また, GN_c は EN_b と EN_{c+1} にだけ関係するので, $r_{sc} = 1$, $r_{c+1,c} = -1$, そして $i \neq s$ では $r_{ic} = 0$ とすれば, $\mathbf{R}^{(c+1)}$ の全要素が確定する.

$\mathbf{w}_{c+1}$ と $\mathbf{v}_c$ の初期値は, Step 1 の初期化法と同様な方法で設定される. つまり, $\mathbf{w}_{c+1} = \mathbf{w}_b$, $\{v_{c1}, \cdots, v_{cn}\}$ の値を $[-1, 1]$ の範囲でランダムに設定し, v_{c0} の値を

$$v_{c0} = -\sum_{i=1}^{n} v_{ci} \left(\sum_{t=1}^{m} h_{ti} x_{ti} \right)$$

で設定する. なお, $\mathbf{v}_1$ は入力ベクトルの EN_b に対する重み付き重心を含むランダムな超平面となる.

7.4 実験による評価

7.4.1 パリティ問題

4 から 8 ビットのパリティ問題を用いて, 提案法を評価した. 実験では, 目標出力値を 0 と 1 に設定し, 全ての入出力パターンを事例として学習させた. 構成的学習法の EN 数の上限は 8 に設定し, 各段階では, 100 反復以上して

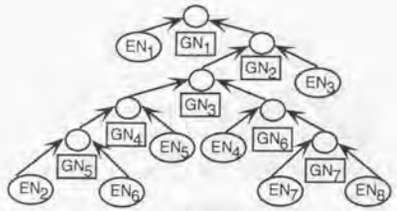
$$\frac{\|\nabla L(\mathbf{\Phi}^{(c)})\|}{N^{(c)}} < 10^{-8}$$

ならば収束したとみなした. また, 各事例の重み付き誤差が

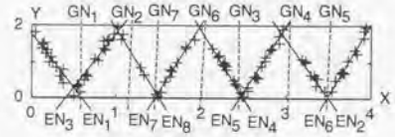
$$\sum_{i=1}^{c} h_{ti} (y_t - \mathbf{w}_i^T \mathbf{x}_t)^2 < 0.1$$

となれば, 望ましい解に収束したとして, アルゴリズムを終了させた.

各ビット数での 100 回の試行結果を表 7.1 に示す. ここで, n ビットパリティ問題に対して, 最小 EN 数は $\lfloor n/2 \rfloor + 1$ で与えられる. 表より, 提案法を用いれば, 最小に近い EN 数でほぼ確実に学習できたことが分かる. 一方, 既存法 [135] では, はるかに多くの EN がなければ, これらのパリティ問題を解けなかった. 例えば, 8- ビットパリティ問題のケースでは, 67% の事例を正しく学習するのに, 64 個もの EN (6 階層均等 2 分木) を必要とした.



(a) 提案法が構成した HME の例



(b) 提案法の学習結果の例

図 7.2: 関数近似問題 (提案法)

7.4.2 関数近似問題

x を入力値, y を目標出力値とし, $0 \leq y \leq 2$ の範囲で, $(x, y) = (0, 2)$ から $(4, 2)$ まで, 傾きが -4 と 4 の直線を交互に繋いだ区分線形関数の学習 (近似) 問題での評価を行なった. 実験では, x の値を $[0, 4]$ の範囲でランダムに設定し, 対応する y の値を求め, 各 y には, 平均 0, 分散 0.1 の正規分布に基づく独立なノイズを与え, 合計で 100 事例を生成した. パリティ問題と同じ終了条件で 10 回の試行を行なったところ, 構成的学習法では, 全試行において, 最小の EN 数でほぼ正確に学習できた. 学習結果の一例は

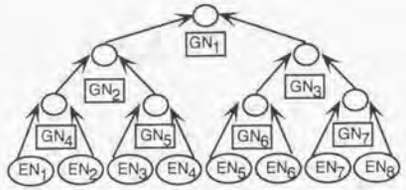
$$(GN_1, EN_1, (GN_2, (GN_3, (GN_4, (GN_5, EN_2, EN_6), EN_5), (GN_6, EN_4, (GN_7, EN_7, EN_8)), EN_3)))$$

である (図 7.2).

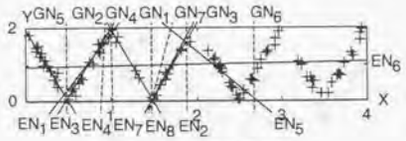
一方, 3 階層の均等 2 分木

$$(GN_1, (GN_2, (GN_4, EN_1, EN_2), (GN_5, EN_3, EN_4)), (GN_3, (GN_6, EN_5, EN_6), (GN_7, EN_7, EN_8)))$$

を予め設定した場合には (図 7.3(a)), どの試行でも適切に学習を行なえず, 例えば, 図 7.3(b) に示すような結果となった. 3 階層の均等 2 分木で正確に学習するには, まず, GN_1 は $x = 2$ で境界を形成しなければならないが, 実際に図 7.2(b) では, $x \approx 1.5$ で GN_1 の境界が形成されたため, $x < 1.5$ で



(a) 3 階層均等 2 分木



(b) 従来法の学習結果の例

図 7.3: 関数近似問題 (従来法)

は冗長な EN が存在し, 逆に, $x > 1.5$ では EN が不足した. すなわち, 予め構造を固定すれば, いくつかの GN の学習すべき境界などが予め規定されるので, 学習が困難になったと考える. 構成的学習アルゴリズムの学習過程を図 7.4 に示す. また, それぞれの段階での HME の構造を図 7.5 に示す. この試行では, 最初に GN_1 は $x \approx 0.5$ となったが, 新たな GNs により適切な境界が順次作られたので, 最終的に望ましい結果を得ることができた.

7.5 結言

本章では, 結合重みの初期化法, ネットワークの学習法, および, ネットワークの拡張法から成る HME の構成的学習アルゴリズムを提案した. パリティ問題と関数近似問題を用いた実験において, 従来法では困難であったが, 提案法を用いれば, 最小規模の HME でも望ましい結果が得られることを確認した.

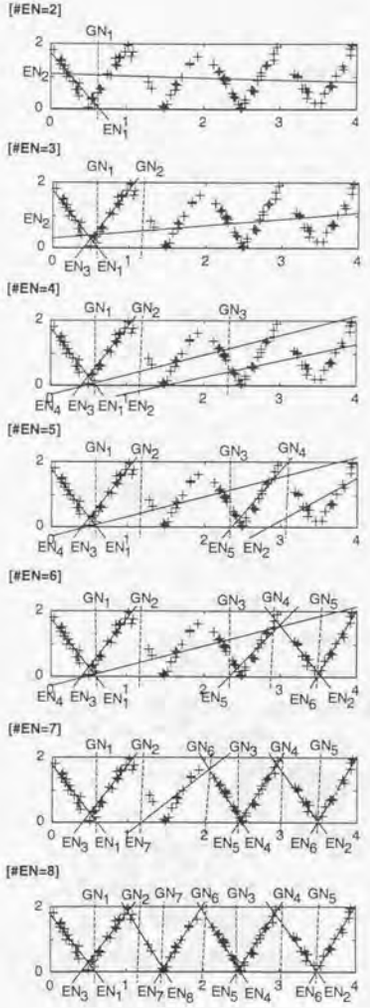


図 7.4: 提案法の学習過程

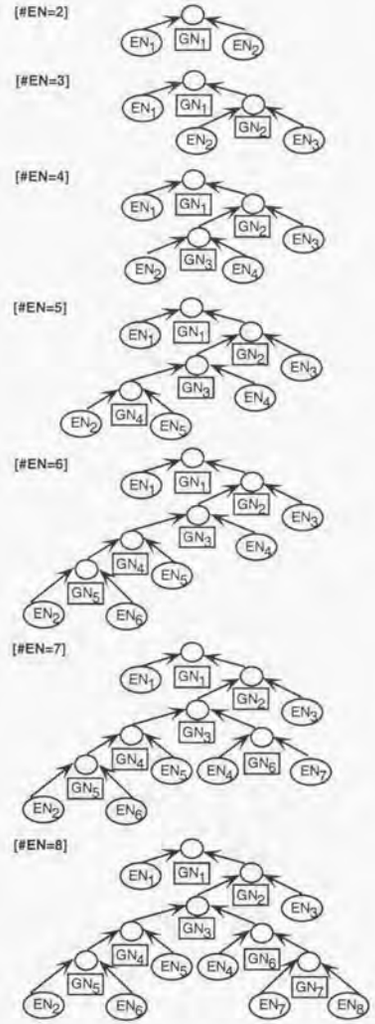


図 7.5: 各段階で構成した HME

第 8 章

結論

本論文は，記号ベースと数値ベースの概念学習アルゴリズムに関して，著者がこれまでに行なってきた研究成果をまとめたものである．以下にその成果を要約し，最後に今後の課題を述べる．

第 1 章では，事例に基づく教師あり概念学習アルゴリズムに関する研究の背景について述べるとともにそれらの問題点を明らかにした．次いで，それらの問題点を解決するための基本方針を示し，本論文の構成について述べた．

第 2 章では，エキスパートシステム構築における知識獲得ボトルネック解決のための試みとして，少ない事例からでも簡潔で十分に汎化した分類ルールを抽出する RF2 を提案した．RF2 はルール候補の生成と精錬の 2 フェーズからなり，正の事例の記述を逐次一般化することによりルール候補を生成し，IDA* 法を用いてそれらを最適ルール集合に精練する．RF2 を人工問題へ適用した結果，PAC- 学習で必要とされる 4 分の 1 の数の事例から，十分に正確なルールを抽出できた．また，最新のルール抽出法である GREEDY3 でも抽出できなかったルールを少ない事例からでも抽出できた．RF2 法を医療診断問題へ適用した結果，未知の事例に対する正答率が医者の知識を用いて作成したエキスパートシステムのものと匹敵した．

第 3 章では，第 2 章の成果を基に，ノイズを含む事例からでも，高い信頼性で良い正答率を保証する簡潔な分類ルールの抽出を目標とする RF3 を提案した．RF3 の特長は，ルール集合選択のための新評価尺度 MEF の導入にある．MEF 尺度では，任意の汎化誤り率の許容限界に対し，アルゴリズムの失敗確率の期待値が近似的に最小化される．また，MEF 尺度は，抽出したルール集合の複雑さと例外事例の個数の和を最小化する尺度としても解釈できた．実験の範囲では，RF3 を用いて，元のルール集合とほぼ同等なルール集合を抽出できた．また，訓練誤り許容限界を変化させ，MEF 尺度と汎化誤り率を比較したところ，両者の傾向が酷似することが分かった．

第 4 章では，過去に解いた問題を用いて，自ら適応して探索効率を改善する概念学習法 RF4 を提案した．すなわち，複数の枝刈り尺度と概念の複合化ルールを利用した深さ優先探索において，状況ベクトルから，原子式が識別概念の構成要素となる確率を計算することにより，適応して探索を高速化す

る．KRK 問題では，RF4 の探索効率が改善されるだけでなく，未知の事例に対する正答率も向上することを確認した．ボンガルド問題では，100 のボンガルド問題に対して，RF4 は 41 問を正答できた．また，RF4 が 41 問を解答した後では，その問題解決時間は平均して 1/3 に短縮され，問題解決を高速化する知識の一部も抽出できた．適応学習の基本となるタスク順序付け問題，および，この問題を一般化した論理式の真偽値判定問題では，計算機実験で，ベイズ推定を用いる方法が，ランダムなタスク列や最尤推定してタスク列を求める方法と比較して，任意の事例数において，期待コストが最小のタスク列を生成し，最小コストにも速く近付くことを確認した．

第 5 章では，3 層ネットワークにおける 2 次の高速学習アルゴリズム BPQ を提案した．BPQ は準ニュートン法をベースとし，探索方向を小記憶 BFGS 法で計算し，最適探索幅を 2 次近似の最小点として効率良く計算することを特徴とする．計算量の考察により，BPQ の 1 反復の計算時間は，一般的な状況で，標準的な BP とほぼ等しいことを示した．人工問題，パリティ問題，および音声合成問題を用いた実験では，他の代表的な学習アルゴリズムと比較して，BPQ は効率良く誤差を減少できることを確認した．さらに，この探索幅計算法は準ニュートン法の収束性に重要な役割果たし，一方，小記憶 BFGS 法は収束性を変えずに記憶容量を大幅に減少できたことを示した．

第 6 章では，数値データに内在する未知の法則を発見するアルゴリズムとして，ニューラルネットを用いた法則発見法 RF5 を提案した．RF5 では，法則の発見問題がニューラルネットの学習問題として定式化され，第 5 章で提案した BPQ アルゴリズムを利用して複数の法則候補を作り出し，その中から MDL 基準を利用して最良のものを法則として特定する．実験では，ある程度のノイズを含むデータからでも，RF5 は指数の値が整数に制限されない法則を効率良く発見できることを確認した．

第 7 章では，複数ニューラルネットの出力の組合せを自己組織的に学習する専門家の階層混合モデル HME の構成的学習アルゴリズムを提案した．このアルゴリズムの特徴は，鞍点に捕らわれることを防ぐための結合重み初期化法，第 5 章で提案した BPQ アルゴリズムを利用したネットワーク学習法，および，“divide-and-conquer” アプローチに基づくネットワーク拡張法を有することである．パリティ問題と関数近似問題を用いた実験では，従来法では困難であるが，提案法を用いれば，最小規模の HME でも望ましい結果が得られることを示した．

本論文では，記号ベースと数値ベースのそれぞれのパラダイムで，大規模な問題でも高品質な学習結果をもたらす基本学習アルゴリズム，ノイズを含む事例からでも汎化精度の高い学習結果を得るための評価尺度を導入した学習アルゴリズム，および，メタレベル学習機構の構築に向けた試みとして開発した学習アルゴリズムを提案した．これらを代表的既存法と比較すれば，アルゴリズムを精緻にすることにより，概念学習における探索性能を向上させることができた．つまり，記号ベース学習では，幅広い概念を妥当な計算量で探索できるようにし，数値ベース学習では，2 次微分情報を効率良く利用してアルゴリズムの高速化を実現した．しかるに，単純な構造の従来法と比較して，ある程度アルゴリズムが複雑になったことは，提案法の短所として指摘できる．また，本論文では従来法との比較実験等により提案法が優ることを示したが，さらに幅広い問題を用いて提案法を評価することも重要である．一方，メタレベル学習機構の構築に向けた研究については，まだ入り口の段階であり，さらなる検討を行わなければならない．本研究の次のステップをまとめれば，さらに幅広い問題を用いて提案したそれぞれのアルゴリズムの評価を行うこと，記号ベースや数値ベースに基づく複数モデルの出力や学習結果を適切に統合する学習機構を構築すること，多様な学習戦略を問題に応じて適切に使い分けるための学習機構を構築すること，さらに，現実環境との多様なインタラクションの可能な概念学習アルゴリズムを構築することなどが挙げられる．著者自身，今後これらの問題の解を見つけるべく，本研究をベースとしてさらなる検討を続けていきたい．

謝辞

本論文をまとめるにあたり東京大学工学部電子情報工学科 石塚満教授には終始懇切丁寧な御指導，御教示を賜りました．ここに慎んで深謝の意を表します．

また，内容について多くの御指導，御助言を賜りました東京大学工学部電気工学科 田中英彦教授，東京大学先端科学技術研究センター 岡部洋一教授に深く感謝致します．また，東京大学工学部電子工学科 近山隆教授，東京大学先端科学技術研究センター 廣瀬明助教授には多くの有益な御教示を賜りました．厚く御礼申し上げます．

本研究は主にコミュニケーション科学研究所にて多くの方々のご指導を得て行なわれました．研究の機会を与えて頂くと共に，御指導頂いた故 西川清史博士（元 NTT コミュニケーション科学研究所所長），同志社大学工学部知識工学科 河岡司教授（前 NTT コミュニケーション科学研究所所長），NTT コミュニケーション科学研究所 松田晃一所長に厚く御礼申し上げます．

また，本研究の期間中に直接の上司として御教授頂いた NTT コミュニケーション科学研究所 中野良平グループリーダには，本研究を進めるにあたり終始丁寧なる御指導を賜りました．また，本論文をまとめるにあたりましても御助言と激励の言葉を頂きました．ここに記して深く感謝致します．

適応概念学習の研究に関しまして University of Ottawa において Stan Matwin 教授をはじめとする機械学習グループの皆様には有益な御討論を頂きました．ここに感謝致します．また，多くの御討論を頂いた NTT コミュニケーション科学研究所 上田修功主幹研究員をはじめ，中野研究グループの皆様に感謝致します．

そして，本論文をまとめるにあたっては NTT 研究開発推進部 研究推進部門 岡田忠信部門長をはじめとする皆様のあたたかい御理解を頂きました．ここに記して深く感謝致します．

最後に，陰ながら著者を支えてくれている妻に感謝します．

参考文献

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Contor.*, Vol. AC-19, pp. 716–723, 1973.

[2] D. Angluin and P.D. Laird. Learning from noisy examples. *Machine Learning*, Vol. 2, No. 4, pp. 343–370, 1988.

[3] E. Balas and A. Ho. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. *Mathematical Programming*, Vol. 12, pp. 37–60, 1980.

[4] E. Barnard. Optimization for training neural nets. *IEEE Trans. Neural Networks*, Vol. 3, No. 2, pp. 232–240, 1992.

[5] R. Battiti. Accelerating back-propagation learning: two optimization methods. *Complex Systems*, Vol. 3, No. 4, pp. 331–342, 1989.

[6] R. Battiti. First- and second-order methods for learning between steepest descent and newton's method. *Neural Computation*, Vol. 4, No. 2, pp. 141–166, 1992.

[7] E.B. Baum and D. Haussler. What size net gives valid generalization. *Neural Computation*, Vol. 1, No. 1, pp. 151–160, 1989.

[8] C.M. Bishop. *Neural networks for pattern recognition*. Oxford Press, 1995.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Occam's razor. *Information Processing Letters*, Vol. 24, No. 6, pp. 377–380, 1987.

[10] N. Bongard. *Pattern recognition*. Spartan Books, 1970.

[11] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Montrey, CA, 1984.

[12] B.G. Buchanan and T.M. Mitchell. Model-directed learning of production rules. In D.A. Waterman and F. Hayes-Roth, editors, *Pattern-directed inference systems*. Academic Press, New York, 1978.

[13] W. Buntine. A critique of the valiant model. In *IJCAI-89*, pp. 837–842, Detroit, MI, 1989.

[14] F. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, Vol. 3, No. 4, pp. 261–283, 1989.

[15] G. DeJong and R. Mooney. Explanation-based learning: An alternative view. *Machine Learning*, Vol. 1, No. 1, pp. 145–176, 1986.

[16] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via EM algorithm. *J. Royal Statist. Soc. Ser. B (methodology)*, Vol. 39, pp. 1–38, 1977.

[17] L.P. Devroye. Automatic pattern recognition: a study of the probability of error. *IEEE trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 530–543, 1988.

[18] R.O. Duda and H.E. Hart. *Pattern classification and scene analysis*. John Wiley & Sons, New York, 1973.

[19] R. Durbin and D. Rumelhart. Product units: a computationally powerful and biologically plausible extension. *Neural Computation*, Vol. 1, No. 1, pp. 133–142, 1989.

[20] B. Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *J. American Statistical Association*, Vol. 78, No. 382, pp. 316–331, 1983.

[21] S.E. Fahlman. Faster-learning variations on back-propagation: an empirical study. In *Proceedings of the 1988 Connectionist Models Summer School*, pp. 38–51, San Mateo, CA, 1988.

[22] S.E. Fahlman and C. Libiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp. 524–532, Los Altos, CA, 1990. Morgan Kaufmann.

[23] B.C. Falakenhainer and R.S. Michalski. Integrating quantitative and qualitative discovery in the abacus system. In Y. Kodratoff and R.S. Michalski, editors, *Machine learning: an artificial intelligence approach, Volume* III, pp. 153–190. Morgan Kaufmann, San Mateo, CA, 1990.

[24] M.L. Fisher and P. Kedia. Optimal solution of set covering / partitioning problems using dual heuristics. *Management Science*, Vol. 36, No. 6, pp. 674–688, 1990.

[25] R. Fletcher. *Practical methods of optimization*. Vol. 1. John Wiley & Sons, 1980.

[26] S.I. Gallant. A connectionist learning algorithm with provable generalization and scaling bounds. *Neural Networks*, Vol. 3, No. 2, pp. 191–201, 1990.

[27] M. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, CA, 1979.

[28] M.R. Garey. Optimal task sequencing with precedence constraints. *Discrete Math.*, Vol. 4, pp. 37–56, 1973.

[29] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. Academic Press, London, 1981.

[30] I.J. Good. *The estimation of probabilities: An essay on modern Bayesian methods, Research monograph 30*. MIT Press, Cambridge, MA, 1965.

[31] S. Goonatilake and S. Khebbal. Intelligent hybrid systems: Issues, classes and future trends. In S. Goonatilake and S. Khebbal, editors, *Intelligent hybrid systems*. John Wiley & Sons, Baffins Lane, UK, 1995.

[32] R.D. Gordon. Values of mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. *Ann. Math. Statist.*, Vol. 12, pp. 364–366, 1941.

[33] R.P. Gorman and T.J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, Vol. 1, No. 1, pp. 75–89, 1988.

[34] S.J. Hanson and L.Y. Pratt. Comparing biases for minimal network construction with back-propagation. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, Vol. 1, pp. 177–185, San Mateo, CA, 1989. Morgan Kaufmann.

[35] G.E. Hinton. Learning translation invariant recognition in massively parallel networks. In J.W. de Bakker, A.J. Nijman, and P.C. Treleaven, editors, *Proc. PARLE Conference on Parallel Architectures and Languages Europe*, pp. 1–13, Berlin, 1987. Springer-Verlag.

[36] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Ass.*, Vol. 58, pp. 13–30, 1963.

[37] D.R. Hofstadter. *Gödel, Escher, Bach: An eternal golden braid.* Basic Books Inc, 1979.

[38] R.C. Holte, L.E. Acker, and B.W. Porter. Concept learning and accuracy of small disjuncts. In *IJCAI-89*, pp. 813–818, Detroit, MI, 1989.

[39] M. Ishikawa. A structural learning algorithm with forgetting of link weight. Technical report, Tech. Rep. TR-90-7, Electrotechnical Lab. Tsukuba-City, Japan, 1990.

[40] R. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, Vol. 1, No. 4, pp. 295–307, 1988.

[41] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, Vol. 3, No. 1, pp. 79–87, 1991.

[42] D.S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, Vol. 9, pp. 256–278, 1974.

[43] M.I. Jordan and R.A. Jacobs. Hierachical mixtures of experts and EM algorithm. *Neural Computation*, Vol. 6, No. 2, pp. 181–214, 1994.

[44] R.E. Korf. Search: A survey of recent results. In H.E. Shrobe, editor, *Exploring artificial intelligence*, pp. 197–237. Morgan Kaufmann, San Mateo, CA, 1988.

[45] G.M. Kuhn and P. Herzberg. Some variations on training recurrent neural networks. In *Proceedings of CAIP Neural Networks Workshop*, pp. 15–17, Rutgers University, 1990.

[46] T.Y. Kwok and D.Y. Teung. Constructive feedforward neural networks for regression problems: a survey. Technical report, HKUST-CS95-43, 1995.

[47] J.E. Laird, P.S. Rosenbloom, and A. Newell. Chanking in soar: The anatomy of a general learning mechanism. *Machine Learning*, Vol. 1, No. 1, pp. 11–46, 1986.

[48] P. Langley. Bacon.1: a general discovery system. In *Proceedings of the Second National Conferenee of the Canadian Soeiety for Computational Studies of Intelligenee*, pp. 173–180, 1978.

[49] P. Langley, H.A. Simon, G. Bradshaw, and J. Zytkow. *Scientific discovery: computational explorations of the creative process.* MIT Press, Cambridge, MA, 1987.

[50] P. Langley and J. Zytkow. Data-driven approaches to empirical discovery. *Artificial Intelligence*, Vol. 40, pp. 283–312, 1989.

[51] P. Langley and J. Zytkow. A robust approach to numeric discovery. In *Proc. seventh International Machine Learning Conference*, pp. 411–418, Austin, Texas, 1990.

[52] Y. LeCun, P.Y. Simard, and B. Pearlmutter. Automatic learning rate maximisation by on-line estimation of the hessian's eigenvector's. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Neural Information Processing Systems 5*, pp. 156–163. Morgan Kaufmann, San Mateo, CA, 1993.

[53] L.R. Leerink, C.L. Giles, B.G. Horne, and M.A. Jabri. Learning with product units. In G. Tesauro, D.S. Touretzky, and T.K. Lee, editors, *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA, 1995.

[54] D.G. Luenberger. *Linear and nonlinear programming.* Addison-Wesley, Reading, MA, 1984.

[55] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, Vol. 4, No. 3, pp. 415–447, 1992.

[56] 益沢ほか. 医療知識ベース利用による医療診断支援システム (doctors) の臨床評価. 第 4 回医療情報学連合大会, pp. 672–677, 1984.

[57] R.S. Michalski. Synthesis of optimal and quasi-optimal variable-valued logic formula. In *International Symposium on Multi-valued Logic*, pp. 76–87, 1975.

[58] R.S. Michalski. A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning: an artificial intelligence approach*, pp. 331–364. Morgan Kaufmann, San Mateo, CA, 1983.

[59] R.S. Michalski. Inferential theory of learning. In R.S. Michalski and G. Tecuci, editors, *Machine learning: A multistrategy approach, Volume* IV, pp. 3–61. Morgan Kaufmann, San Francisco, CA, 1994.

[60] R.S. Michalski and R.L. Chilauski. Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *Policy Analysis and Information Systems*, Vol. 4, pp. 125–160, 1980.

[61] R.S. Michalski and Y. Kodratoff. Research in machine learning: recent progress, classification of methods, and future directions. In Y. Kodratoff and R.S. Michalski, editors, *Machine learning: an artificial intelligence approach, Volume* III, pp. 3–30. Morgan Kaufmann, San Mateo, CA, 1990.

[62] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system aq15 and its testing application to three medical domains. In *AAAI-86*, pp. 1041–1045, 1986.

[63] R.S. Michalski and R.E. Stepp. Learning from observation: conceptual clustering. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning: an artificial intelligence approach*, pp. 331–364. Morgan Kaufmann, San Mateo, CA, 1983.

[64] M.L. Minsky and S. Papert. *Perceptrons: An introduction to computational geometry*. MIT Press, Cambridge, MA, 1969.

[65] S.N. Minton. Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence*, Vol. 42, pp. 363–392, 1990.

[66] S.N. Minton, J.G. Carbonell, C.A. Knoblock, D.R. Kuokka, O. Etzioni, and Y. Gil. Explanation-based learning: A problem-solving perspective. *Artificial Intelligence*, Vol. 40, pp. 63–118, 1989.

[67] T.M. Mitchell. Generalization as search. *Artificial Intelligence*, Vol. 18, pp. 203–226, 1982.

[68] T.M. Mitchell, R.M. Keller, and S.T. Kedar-Cabelli. Explanation-based genaralization: A unifying view. *Machine Learning*, Vol. 1, No. 1, pp. 47–80, 1986.

[69] T.M. Mitchell, P.E. Utgoff, and R.B. Banerji. Learning by experimentation: Acquiring and refining problem-solving heuristics. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning: an artificial intelligence approach*, pp. 331–364. Morgan Kaufmann, San Mateo, CA, 1983.

[70] M.F. Møller. Exact calculation of the product of the Hessian matrix of feedforward network error functions and a vector in O(N) time. Technical report, DAIMI PB-432, Computer Science Department, Aarthus University, 1993.

[71] M.F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, Vol. 6, No. 4, pp. 525–533, 1993.

[72] M.F. Møller. Supervised learning on large redundant training sets. *International Journal of Neural Systems*, Vol. 4, No. 1, pp. 15–25, 1993.

[73] R.J. Mooney, J.W. Shavlik, G.G. Towell, and A. Gove. An experimental comparison of symbolic and connectionist learning algorithms. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 775–780, Detroit, MI, 1989.

[74] S. Muggleton, M. Bain, J. Hayes-Michie, and D. Michie. An experimental comparison of human and machine learning formalisms. In *Proc. sixth International Machine Learning Workshop*, pp. 113–118, Ithaca, NY, 1989.

[75] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proc. First Conf. Algorithmic Learning Theory*, pp. 368–381, Tokyo, 1990.

[76] R. Nakano, N. Ueda, K. Saito, and T. Yamada. Parrot-like speaking using optimal vector quantization. In *Proc. IEEE International Conference on Neural Networks*, Parse, Australia, 1995.

[77] G. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, Vol. 5, No. 1, pp. 71–99, 1990.

[78] J. Pearl. *Heuristics*. Addison-Wesley, Reading, MA, 1984.

[79] B.A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, Vol. 6, No. 1, pp. 147–160, 1994.

[80] L. Pitt and L.G. Valiant. Computatinal limitations on learning from examples. *J. ACM*, Vol. 35, No. 4, pp. 965–984, 1988.

[81] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, Vol. 247, pp. 978–982, 1990.

[82] M.J.D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In *Nonlinear Programing, SIAM-AMS Proceedings, Vol 9*, Providence, R.I., 1976.

[83] J.R. Quinlan. Learning efficient classification procedures and their application to chess end game. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning: an artificial intelligence approach*, pp. 463–482. Morgan Kaufmann, San Mateo, CA, 1983.

[84] J.R. Quinlan. Induction of decision trees. *Machine Learning*, Vol. 1, No. 1, pp. 81–106, 1986.

[85] J.R. Quinlan. Simplifying decision trees. *Int. J. Man-Machine Studies*, Vol. 27, pp. 221–234, 1987.

[86] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, Vol. 5, No. 31, pp. 239–266, 1990.

[87] J.R. Quinlan and R.L. Rivest. Infering decision trees using the minimum description length. *Information and Computation*, Vol. 80, pp. 227–248, 1989.

[88] R. Reed. Pruning algorithms - a survey. *IEEE Trans. Neural Networks*, Vol. 4, No. 5, pp. 740–747, 1993.

[89] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *Proc. IEEE International Conference on Neural Networks*, San Francisco, CA, 1993.

[90] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. of statist.*, Vol. 11, No. 2, pp. 416–431, 1983.

[91] R.L. Rivest. Learning decision lists. *Machine Learning*, Vol. 2, No. 3, pp. 229–246, 1987.

[92] F. Rosenblatt. The perceptron: A probabilistic model for information strage and organization in the brain. *Psychological Review*, Vol. 65, No. 6, pp. 386–408, 1958.

[93] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, pp. 318–362. MIT Press, 1986.

[94] K. Saito and R. Nakano. Medical diagnostic expert system based on PDP model. In *Proceedings of IEEE International Conference on Neural Networks*, pp. 255–262, San Diego, CA, 1988.

[95] K. Saito and R. Nakano. Rule extraction from facts and neural networks. In *INNC-90-PARIS*, pp. 379–382, Paris, France, 1990.

[96] K. Saito and R. Nakano. Adaptive concept learning algorithm. In *IFIP 13th World Computer Congress*, pp. 294–299, Hamburg, Germany, 1994.

[97] K. Saito and R. Nakano. Concept learning algorithm with adaptive search. In K. Furukawa, D. Michie, and S. Muggleton, editors, *Machine Intelligence 14*, pp. 347–363. Oxford Press, 1995.

[98] K. Saito and R. Nakano. A constructive learning algorithm for HME. In *Proceedings of IEEE International Conference on Neural Networks*, pp. 1268–1273, Washington, D.C., 1996.

[99] K. Saito and R. Nakano. BFGS update and efficient step-length calculation for three-layer neural networks. *Neural Computation*, Vol. 9, No. 1, pp. 123–141, 1997.

[100] K. Saito and R. Nakano. Law discovery using neural networks. In *International Joint Conference on Artificial Intelligence*, pp. 1078–1083, Nagoya, Japan, 1997.

[101] K. Saito and R. Nakano. MDL regularizer: A new regularizer based on the MDL principle. In *Proceedings of IEEE International Conference on Neural Networks*, pp. 1833–1838, Houston, TX, 1997.

[102] K. Saito and R. Nakano. Second-order learning algorithm with squared penalty term. In *Advances in Neural Information Processing Systems 9*, pp. 627–633, Denver, CO, 1997.

[103] K. Saito and R. Nakano. A connectionist approach to numeric law descovery. In *Machine Intelligence 15*, (to appear).

[104] 斉藤, 中野. 事例とニューラルネットからの分類ルール抽出法. 情処研究報告, 知識工学と人工知能, Vol. 67-3, pp. 1–8, 1989.

[105] 斉藤, 中野. ノイズを含む事例からのルール抽出: RF3 アルゴリズム. 情報処理学会論文誌, Vol. 33, No. 5, pp. 636–644, 1992.

[106] 斉藤, 中野. 事例からのルール抽出: RF2 アルゴリズム. 情報処理学会論文誌, Vol. 33, No. 5, pp. 628–635, 1992.

[107] 斉藤, 中野. ボンガルド問題と概念学習アルゴリズム. 第 7 回人工知能学会全国大会, pp. 97–100, 1993.

[108] 斉藤, 中野. 3 層ニューラルネットにおける 2 階導関数を用いた学習アルゴリズムの高速化. 信学技報, Vol. NC94-7, pp. 49–56, 1994.

[109] 斉藤, 中野. 準ニュートン法に基づく Elman ネットワークの学習アルゴリズム. 信学技報, Vol. NC94-38, pp. 47–54, 1994.

[110] 斉藤, 中野. 準ニュートン法に基づくガウス混合分布の推定アルゴリズム. 神経回路学会第 5 回全国大会, pp. 256–257, 1994.

[111] 斉藤, 中野. HME の構成的学習アルゴリズム. 神経回路学会第 6 回全国大会, pp. 54–55, 1995.

[112] 斉藤, 中野. ベイズ推定に基づくタスク順序付け. 情報処理学会論文誌, Vol. 36, No. 3, pp. 572–578, 1995.

[113] 斉藤, 中野. 適応概念学習アルゴリズム: RF4. 情報処理学会論文誌, Vol. 36, No. 4, pp. 832–839, 1995.

[114] 斉藤, 中野. HME の構成的学習アルゴリズム. 信学技報, Vol. NC95-114, pp. 99–106, 1996.

[115] 斉藤, 中野. コネクショニストアプローチによる数法則の発見. 情報処理学会論文誌, Vol. 37, No. 9, pp. 832–839, 1996.

[116] 斉藤, 中野. ニューラルネットを用いた法則発見. 信学技報, Vol. NC95-165, pp. 85–92, 1996.

[117] 斉藤, 中野. 自乗値ペナルティ項を用いた 2 次学習アルゴリズム. 神経回路学会第 7 回全国大会, pp. 78–79, 1996.

[118] 斉藤, 中野. 自乗値ペナルティ項を用いた 2 次学習アルゴリズム. 信学技報, Vol. NC96-105, pp. 79–86, 1997.

[119] 斉藤, 中野. 自乗値ペナルティ項を用いた 2 次学習アルゴリズム. 情報処理学会論文誌, Vol. 38, No. 11, 1997 (to appear).

[120] 斉藤, 中野. 2 次学習アルゴリズム BPQ によるリカレントネットワーク学習とガウス混合分布推定. 電子情報通信学会論文誌, 1998 (to appear).

[121] 斉藤, 中野. 2 次学習アルゴリズム BPQ の分類問題への適用法とその評価. 電子情報通信学会論文誌, Vol. J81-D-II, No. 2, 1998 (to appear).

[122] 斉藤, 中野. HME の構成的学習アルゴリズム. 電子情報通信学会論文誌, Vol. J81-D-II, No. 2, 1998 (to appear).

[123] 斉藤, 中野. MDL 原理に基づく新正則化法. 人工知能学会誌, Vol. 13, No. 1, 1998 (to appear).

[124] A.L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, Vol. 3, pp. 211–229, 1959.

[125] T.D. Sanger. Basis-function trees as a generalization of local variable selection method for function approximation. In D.S. Touretzky, editor, *Neural Information Processing Systems 3*, pp. 707–713. Morgan Kaufmann, San Mateo, CA, 1991.

[126] C. Schaffer. Bivariate scientific function finding in a sampled, real-data testbed. *Machine Learning*, Vol. 12, No. 1/2/3, pp. 167–183, 1993.

[127] T.J. Sejnowski and C.R. Rosenberg. Parallel networks that learns to pronounce English text. *Complex Systems*, Vol. 1, pp. 145–168, 1987.

[128] F.M. Silva and L.B. Almeida. Speeding up backpropagation. In R. Eckmiller, editor, *Advanced Neural Computers*, pp. 151–160. North-Holland, Amsterdam, 1990.

[129] H.A. Simon and J.B. Kadane. Optimal problem-solving search: All-or-none solution. *Artificial Intelligence*, Vol. 6, No. 3, pp. 235–247, 1975.

[130] J.R. Slagle. An efficient algorithm for finding certain minimum-cost procedure for making binary decisions. *J. Assoc. Comput. Machinery*, Vol. 11, No. 3, pp. 253–264, 1964.

[131] P.A. Stefankis, J. Wnek, and J. Zhang. Bibliography of recent machine learning research (1985-1989). In Y. Kodratoff and R.S. Michalski, editors, *Machine learning: an artificial intelligence approach, volume* III, pp. 685–789. Morgan Kaufmann, San Mateo, CA, 1990.

[132] R.S. Sutton and C.J. Matheus. Learning polynomial functions by feature construction. In *Proceedings of the Eighth International Machine Learning Workshop*, pp. 208–212, Evanston, IL, 1991.

[133] J.A. Swets. Measuring the accuracy of diagnostic systems. *Science.*, Vol. 240, No. 3, pp. 1285–1293, 1988.

[134] L.G. Valiant. A theory of the learnable. *CACM.*, Vol. 27, No. 11, pp. 1134–1142, 1984.

[135] S.R. Waterhouse and A.J. Robinson. Classification using hierarchical mixtures of experts. In *Proc. of NNSP*, pp. 177–186, 1994.

[136] S.R. Waterhouse and A.J. Robinson. Constructive algorithms for hierarchical mixtures of experts. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pp. 524–532. MIT Press, Cambridge, MA, 1996.

[137] R.L. Watrous. Learning algorithms for connectionist networks: applied gradient methods of nonlinear optimization. In *Proc. IEEE International Conference on Neural Networks*, pp. II–619–627, San Diego, CA, 1987.

[138] S. Weiss and L. Kapouleas. An experimental comparison of pattern recognition, neural nets and machine learning classification methods. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 781–787, Detroit, MI, 1989.

[139] P.M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, Vol. 7, No. 1, pp. 117–143, 1995.

[140] P.H. Winston. Learning structural descriptions from examples. In P.H. Winston, editor, *The psychology of computer vision*, pp. 157–209. McGraw Hill, New York, 1975.

本論文に関する原著論文

学術論文

1. 斉藤 和巳, 中野 良平, "事例からのルール抽出: RF2 アルゴリズム", 情報処理学会論文誌, Vol. 33, No. 5, pp. 628-635, 1992.
2. 斉藤 和巳, 中野 良平, "ノイズを含む事例からのルール抽出: RF3 アルゴリズム", 情報処理学会論文誌, Vol. 33, No. 5, pp. 636-644, 1992.
3. 斉藤 和巳, 中野 良平, "ベイズ推定に基づくタスク順序付け", 情報処理学会論文誌, Vol. 36, No. 3, pp. 572-578, 1995.
4. 斉藤 和巳, 中野 良平, "適応概念学習アルゴリズム: RF4", 情報処理学会論文誌, Vol. 36, No. 4, pp. 832-839, 1995.
5. 斉藤 和巳, 中野 良平, "コネクショニストアプローチによる数法則の発見", 情報処理学会論文誌, Vol. 37, No. 9, pp. 1708-1716, 1996.
6. Kazumi Saito and Ryohei Nakano, "Partial BFGS Update and Calculating Optimal Steplength for Three-layer Neural Networks", *Neural Computation* Vol. 9, No. 1, pp. 123-141, 1997.
7. 斉藤 和巳, 中野 良平, "自乗値ペナルティ項を用いた 2 次学習アルゴリズム", 情報処理学会論文誌, Vol. 38, No. 11, 1997 (掲載予定).
8. 斉藤 和巳, 中野 良平, "MDL 原理に基づく新正則化法", 人工知能学会誌, Vol. 13, No. 1, 1998 (掲載予定).
9. 斉藤 和巳, 中野 良平, "2 次学習アルゴリズム BPQ の分類問題への適用法とその評価", 電子情報通信学会論文誌, Vol. J81-D-II, No. 2, 1998 (掲載予定).
10. 斉藤 和巳, 中野 良平, "HME の構成的学習アルゴリズム", 電子情報通信学会論文誌, Vol. J81-D-II, No. 2, 1998 (掲載予定).

11. 斉藤 和巳, 中野 良平, "2次学習アルゴリズム BPQ によるリカレントネットワーク学習とガウス混合分布推定", 電子情報通信学会論文誌 (採録決定).

国際会議

1. Kazumi Saito and Ryohei Nakano, "Medical Diagnostic Expert System based on PDP Model", *International Conference on Neural Networks*, pp. 255-262, 1988.

2. Kazumi Saito and Ryohei Nakano, "Rule Extraction from Facts and Neural Networks", *International Neural Network Conference*, pp. 379-382, 1990.

3. Kazumi Saito and Ryohei Nakano, "Adaptive Concept Learning Algorithm", *IFIP 13th World Computer Congress 94*, Vol. 1, pp. 294-299, 1994.

4. Kazumi Saito and Ryohei Nakano, "Concept Learning Algorithm with Adaptive Search", Furukawa, K., Michie, D. and Muggleton, S. eds., *Machine Intelligence 14*, pp. 347–363, Oxford Press, 1995.

5. Kazumi Saito and Ryohei Nakano, "A Connectionist Approach to Numeric Law Descovery", *International Workshop on Machine Intelligence*, 1995.

6. Kazumi Saito and Ryohei Nakano, "A Constructive Learning Algorithm for HME", *International Conference on Neural Networks*, pp. 1268-1273, 1996.

7. Kazumi Saito and Ryohei Nakano, "Law Discovery using Neural Networks", *NIPS*96 Post-conference Workshop: Rule-extraction from Trained Neural Networks*, pp. 62–69, 1996.

8. Kazumi Saito and Ryohei Nakano, "Second-order Learning Algorithm with Squared Penalty Term", In *Advances in Neural Information Processing Systems*, Vol. 9, pp. 627–633, 1997.

9. Kazumi Saito and Ryohei Nakano, "MDL Regularizer: A New Regularizer based on the MDL Principle", *International Conference on Neural Networks*, pp. 1833–1838, 1997.

10. Kazumi Saito and Ryohei Nakano, "Law Discovery using Neural Networks", *International Joint Conference on Artificial Intelligence*, pp. 1078–1983, 1997.

研究会, 大会

1. 斉藤 和巳, 中野 良平, "PDP モデルによる診断エキスパートシステム", 情報処理学会研究報告, 知識工学と人工知能 56-3, pp. 17-24, 1988.

2. 斉藤 和巳, 中野 良平, "事例とニューラルネットからの分類ルール抽出法", 情報処理学会研究報告, 知識工学と人工知能 67-3, pp. 1-8, 1989.

3. 斉藤 和巳, 中野 良平, "ボンガルド問題と概念学習アルゴリズム", 第7回人工知能学会全国大会, pp. 97-100, 1993.

4. 斉藤 和巳, 中野 良平, "3層ニューラルネットにおける2階導関数を用いた学習アルゴリズムの高速化", 電子情報通信学会技術報告, NC94-7, pp. 49-56, 1994.

5. 斉藤 和巳, 中野 良平, "準ニュートン法に基づく Elman ネットワークの学習アルゴリズム", 電子情報通信学会技術報告, NC94-38, pp. 47-54, 1994.

6. 斉藤 和巳, 中野 良平, "準ニュートン法に基づくガウス混合分布の推定アルゴリズム", 日本神経回路学会第5回全国大会, pp. 256-257, 1994.

7. 斉藤 和巳, 中野 良平, "HME の構成的学習アルゴリズム", 日本神経回路学会第6回全国大会, pp. 54-55, 1995.

8. 斉藤 和巳, 中野 良平, "HME の構成的学習アルゴリズム", 電子情報通信学会技術報告, NC95-114, pp. 99-106, 1996.

9. 斉藤 和巳, 中野 良平, "ニューラルネットを用いた法則発見", 電子情報通信学会技術報告, NC95-165, pp. 85-92, 1996.

10. 斉藤 和巳, 中野 良平, "自乗値ペナルティ項を用いた2次学習アルゴリズム", 日本神経回路学会第7回全国大会, pp. 78-79, 1996.

11. 斉藤 和巳, 中野 良平, "自乗値ペナルティ項を用いた2次学習アルゴリズム", 電子情報通信学会技術報告, NC96-105, pp. 79-86, 1997.