自律エージェントの行動獲得における状態と行為の抽象化に関する研究

矢入健久

1998 年度 学位請求論文

自律エージェントの行動獲得における状態と行為の抽象化に関する研究

矢入 健久

東京大学 大学院 工学系研究科 航空宇宙工学専攻

本論文の要旨

近年，知能ロボットなどの知的自律システムの実現を目指す分野では，「行動に基づく知能」あるいは「環境との密な相互作用」などと呼ばれる基本原理に基づいたアプローチが盛んに研究されるようになってきた．この方法論の本質は，自律システムの「知的さ」をそれらの行動主体（エージェントと呼ばれる）と外部環境との間の密接な相互作用に求めるという考えである．

この分野における今日の主要テーマとして，エージェントが環境との相互関係からいかにして適切な行動則を獲得するかという「行動学習問題」が挙げられ，これまでに様々な機械学習理論に基づく具体的な手法やシステムが提案されている．そして，そのほとんどは，人間の手によってあらかじめ離散化された状態集合と行為集合との間の適切なマッピングルール（もし状態がSならば行為Aを実行すべきだ，というような行動則）を，何らかの評価基準と行動経験に基づいて獲得するという問題に帰着して扱っている．

しかし，ロボットのような実環境エージェントの場合，実際の環境とのインタラクションは，プリミティブで連続的，かつ多次元のセンサー入力およびモーター出力を介して行われるので，そのような抽象的な状態空間，行為空間を人間があらかじめ定義することは困難である．しかも，そのような人間のヒューリスティクスによって天下り的に抽象化された状態集合，行為集合を用いてエージェントが実環境における行動決定や学習を行った場合，記号的世界モデルに基づく古典的人工知能と同様，シンボルグラウンディング問題やフレーム問題という難問に正面からぶつかることを免れ得ない．

このような背景から本論文では，エージェント自身がその身体性やタスク，環境の性格にとって適切な状態空間や行為空間をプリミティブなセンサー情報，モーター出力から抽象化（離散化）するにはどうすべきかという問題，すなわち**状態・行為の自律的抽象化問題**を扱う．この状態・行為の抽象化問題は，ここ数年その重要性がにわかに注目されるようになり，実際いくつかの具体的アプローチが先行研究によって提案され，一定の成果が報告されている．しかし，これらの従来研究は，それぞれごく限定された問題に対してアド・ホックな具体的手法を提案し適用しているにすぎず，状態・行為の抽象化問題に対する一般的かつ体系的な解決法が完成されるには，まだ多くの本質的難問が残されている．

本研究では，このような自律エージェントの状態・行為抽象化問題における重要な未解決問題のうち，以下に挙げる2つのテーマに焦点を当て，その一般的な問題枠組の定義と解決法を提案する．

- 異種冗長なセンサー入力の一般化と状態クラスの表現法に関する問題
- 状態および行為の抽象化基準に関する問題

まず前者は，エージェントが環境から得るプリミティブなセンサー入力信号を，実際にどのような方法によって一般化し，状態クラスとして表現するかという問題である．従来の諸研究ではこの問題に関して，決定木，線形判別関数，マハラノビス超楕円体，最近傍法などによる状

態の一般化・表現法が提案されているが，それらでは専ら，「領域をいかに精密に分割するか」，「いかに少ない分割数ですむか」という２つの項目に主眼が置かれている．しかし，実環境で柔軟かつ知的に振舞う自律エージェントを実現しようとする立場からは，センサー情報が必然的に含む不確定性要素に対して，それらの状態一般化，表現法がどれほど頑強であり得るかということの方が，より重要である．また，そのように頑強な状態一般化・表現を行うためには，多種多様で冗長なセンサー情報をいかに効果的に統合するか，ということが鍵になるが，従来手法ではいずれもこれらの点に関する十分な考慮が行われていない．

そこで本研究では，この問題へのアプローチとして，単純ベイズ分類器に基づく状態一般化・表現法を提案する．この手法では，エージェントの行動経験データから統計的に推定された各センサー入力ごとの対数尤度の分布関数によって各状態の一般化と表現が行われ，新しいセンサー入力ベクトルの状態クラスへの分類は，各センサー入力ごとに分散的に計算された対数尤度の和によって決定される．この方法は，従来の手法に比べて以下のような長所を有している．

- 各種センサーから得られる異種冗長な情報を，その入力信号の連続／離散性や確率分布の型などの性格に関係なく，柔軟かつ効率的に統合しつつ，状態の一般化と表現を行うことが可能である．
- またその結果として，従来の状態一般化・表現手法と比較して，センサーノイズや故障（フォールト）など，実環境にとって不可避な不確定性要因に対する頑強性が大きく改善される．

一方，後者のテーマは，「エージェントの状態や行為を，何に基づいて，すなわち，どのようなポリシーで抽象化するべきか」という問題である．これはより詳細には，「エージェントが観測するセンサー入力，および実行するモーター出力について，どのような基準に従ってその近い・遠いの尺度を決め，同じ状態あるいは同じ行為として分類，一般化を行うか」という問題として定義することができる．この問題に対して，従来研究では一般に，行為の集合をあらかじめ人間が定義した上で，「同じ行為によって同じ結果が得られるセンサー入力を同じ状態とみなす」か，センサー入力空間における距離尺度を事前に仮定した上で，「近いセンサー入力において同じ結果をもたらすモーター出力ベクトルを同じ行為とみなす」というヒューリスティクスを適用することによって，どちらか一方のみを抽象化していた．しかも，これらのヒューリスティクスにおいて，何を「同じ結果」とみなすかはそれぞれの研究によってまちまちであるうえ，その違いが本質的にどのような意味を持つのかは全く議論されていない．

これに対し本研究では，ゴール状態やサブゴールへの到達，報酬の獲得，センサー入力の変化など，エージェントの行為結果を表す複数の属性に関する「ばらつき」（各結果属性に関する情報エントロピーの重み和によって定義される）を考え，これを最小化するような状態集合，行為集合を探索する過程として抽象化問題を定式化し，この枠組に基づいた状態・行為抽象化法を提案する．この新たな手法的枠組は，従来研究において思い付き的に用いられていた様々な抽象化基準に統一化された視点を与えるとともに，以下に挙げる４つの効果をもたらす．

- ゴール・サブゴール状態への到達，報酬の獲得，センサー入力変化など，複数の異なる行動結果属性の類似性に基づいた状態および行為の抽象化が可能である．
- 従来，別々に扱われていた状態の抽象化と行為の抽象化が，「行為結果のばらつき最小化」という一つの枠組の上で統合される．
- 提案した状態・行為抽象化法と，従来の行動政策獲得法とを交互に繰り返し適用することによって，エージェントの状態・行為空間の性格がデータ駆動型から，リスク回避型，ゴール指向型へと順次変化していき，エージェントの行動性能が改善される．
- エージェントの状態集合や行為集合を完全に0から構成する手段としてだけでなく，人間が初期値として与えた状態集合や行為集合を漸次的に再構成していくことによって，学習コスト面の上で大きな改善を図る方法を提供する．

本論文の後半では，状態・行為の抽象化問題におけるこれら2つのテーマに対する本研究の提案手法の有効性を検証するために，自律移動ロボットの目標物追従タスク，車庫入れタスクを想定したシミュレーション実験の結果を二部にわけて示した．

まず第一の実験では，接触センサー，ソナー，画像センサーなど異種冗長なセンサー入力が得られるという状況下で，提案した「単純ベイズ分類器に基づく状態一般化・表現法」が，外部環境の不確定性，すなわちセンサーノイズや故障などに対してどのような性格を持つかを調べた．その結果，この提案手法が，異種冗長センサー入力を効率的に統合し，これらの不確定性に対して従来の手法に比べて高い頑強性を持つことが示された．

一方，第二の実験では，「行為結果のばらつき最小化に基づく状態・行為抽象化」の検証を行った。この結果，

- 複数の行為結果属性の類似性を抽象化基準に含め，強化学習による従来の行動政策獲得と組み合わせることによって，前述のように抽象化の性格が理想的に変化していくこと．
- 状態空間と行為空間の両方を自律的に抽象化することによって，どちらか一方を単独で行う場合と比較してエージェントの行動パフォーマンスが大きく改善されること．
- 人間が暫定的に与えた状態・行為空間を初期値とし，これを再構成していくことによって学習に要するコストが大きく軽減されること．

などが示された．

また，これらの結果を踏まえ，本論文の最後では，状態・行為の抽象化と行動学習の完全なオンライン化，記号的プランニングシステムとの統合など，今後この研究が進むべき方向について議論している．

目次

図目次

表目次

第1章　序論

1.1　知的自律システム実現への道のり

近年,「行動に基づく知能」(behavior-based intelligence) あるいは「環境との密な相互作用」(intensive interaction with environment) などといった言葉が知的自律システムの研究分野において盛んに聞かれるようになった．これらの言葉の根底に共通する概念は，人間や他の生物，および（いまだ実現されていないが）それと同等に自律的で合理的な人工システムにおける「知能」の本質を，それらの行動主体（「エージェント」という言葉によって表される）と外部環境との間の絶えまない，物理的，情報的な相互作用の結果として現れる現象に帰着する考え方として説明される．このような方法論が広く認知されるようになった背景には，記号によって表現された世界モデル上でのプランニングという過程が知能の核心であるという考えのもとに始められた従来の知的自律システムの研究が，記号的世界モデルの利用という前提によってもたらされる深刻な問題 ― フレーム問題やシンボル・グラウンディング問題などの前に明らかに行き詰まっており，この新しい考え方がこの分野に大きなブレークスルーを提供するものとして期待されるようになったということが挙げられる．

この考え方は知的自律システムを人工的に実現する上で重要なヒントを示していることは間違いないが，それ自体が実際のシステムを構築するための必要十分な方法を提供しているわけではない．実際，初期においては人間によって埋め込まれた反射的な行動則によって環境との密なインタラクションを行う自律システムが提案されていたが，そのような行動則の獲得自体も知的システムの重要な一条件であるはずであるという根本的な問題と，人間が逐一そのような行動則系を設計することは非常に困難であるという現実的な理由から，今日では「環境との相互関係からエージェントがいかにして適切な行動則を獲得するか」という**自律的行動獲得問題**がこの分野における中心的テーマになりつつある．

1.2　抽象化 ― 行動のための内部表現獲得

この行動獲得問題へのアプローチとして，今日までに様々な機械学習手法を利用した方法が提案されている．しかし，その大部分においては，設計者があらかじめエージェントのプリミティブなセンサー空間をいわば天下り的に事前に離散化してしまっており，エージェント自身による行動学習は，その抽象化された有限個の要素から成る状態集合とアクション集合との間の適切な組合せを学ぶというレベルにとどまっている．

このように人間によって事前に離散化・抽象化された状態空間および行為空間を用いた行動学習には2つの大きな問題点が存在する．第一点は，人間による状態・行為の抽象化が必ずしも容易ではなく，また仮にできたとしてもその状態・行為空間がエージェントにとって最適であるという保証がどこにもないという点である．なぜならば，状態・行為の抽象化は本来，そのエージェント自身のタスク，センサー，アクチュエータなどの身体性，および環境の性格に深く依存しており，人間が直観で与えるものとは一致しないと考えられるからである．第二に，何らかの理由で環境やセンサーやアクチュエータの性格の変化が生じ，抽象化された状態と実際のセンサー入力との間，あるいは行為と実際のモーター出力との間の対応関係にずれが生じた場合，あらかじめ人間によって与えられた状態空間，行為空間を固定的に用いたのでは著しい機能低下が生じると考えられることである．

これらの問題に対して，そもそもエージェントは抽象的な内部表象を一切持つべきではなく，プリミティブなセンサー空間−モーター空間レベルで全ての行動獲得が行われるべきだという考え [15][14] がある一方で，知的エージェント自身が自ら経験に基づいて適切な状態の抽象化 [2][5][45] [62][77][80] および行為の抽象化 [61][44] を行うことによって解決を試みるアプローチが提案されるようになって来ている．

この自律的状態・行為空間構成というアプローチを正当づける根拠としては，以下の3点が挙げられる．

- 連続なセンサー空間中の全てのセンサー点とモーター空間中の全ての点の間の適したマッピングを逐一学習して記憶し，またマッチングを行うことは計算コスト，メモリコストなどの面で非現実的である．それに対して，もし適切に一般化抽象化されたサイズの小さい状態集合と行為集合が得られれば，これらのコストを低減することができる．
- 類似の情報を担ったセンサー入力を一般化して扱うことによって，環境の特定の状況に強く依存した外乱や不確定性に対するロバスト性が増すと考えられる．また，環境やセンサー自体の変化に対して，センサー入力と抽象状態間の対応関係の修正によって容易に対応できることが期待される．
- 抽象化された状態と行為の集合は，強化学習などの反射的な行動政策の獲得に使われるだけでなく，従来の記号操作に基づくプランニングシステムにおける状態・オペレータの記述に用ることができると考えられる．従って反射的エージェントと古典的なプランニングシステムとの統合を可能にするものとして期待される．

1.3 本研究の目的と意義

本研究では，上で述べた知的自律エージェントにおける状態・行為の抽象化というテーマにおいて，依然未解決となっている2つの主要問題への接近を目的とする．この2つの問題とは，

- 状態・行為の抽象化基準と両者の同時抽象化に関する問題

- センサー入力の一般化と状態クラスの表現法に関する問題

である．

前者は「エージェントが観測し得るセンサー入力，および実行し得るモーター出力について，どのような基準に基づいてその近い・違いを測り，同じ状態あるいは同じ行為として判断し分類するか」という問題である．従来研究では一般に，行為の集合をあらかじめ定義した上で，「同じ行為によって同じ結果が得られるようなセンサー入力を同じ状態とみなす」か，センサー入力空間における距離尺度を事前に定義した上で，「近いセンサー入力において同じ結果をもたらすモーター出力ベクトルを同じ行為とみなす」というヒューリスティクスを適用することによって，状態と行為のどちらか一方のみを抽象化していた．しかも，これらのヒューリスティクスにおいて何をもって「同じ結果」とみなすかという基準はそれぞれの研究によってまちまちであるうえ，その違いが本質的にどのような意味を持つのかは全く議論されていなかった．そこで本研究では，ゴール状態やサブゴールへの到達，報酬の獲得，センサー入力の変化など，エージェントの行為結果を表す複数の属性に関するばらつき，すなわち情報エントロピーを考え，これを最小化するような状態集合，行為集合を探す過程としてこの問題を定式化することを提案する．この状態・行為抽象化の新しい定式化は，従来研究においてまちまちに用いられていたヒューリスティックな抽象化基準に統一化された視点を与えるとともに，これまで困難であった状態と行為の同時抽象化をも可能にする．

一方後者は，同じ状態クラスあるいは行為クラスとして分類されたセンサー入力集合，モーター出力集合を，実際にどのような方法によって一般化し，表現するかという問題である．従来の諸研究ではこれに関して，決定木，線形判別関数，マハラノビス超楕円体，最近傍法などが用いられているが，「領域をいかに精密に分割するか」，「いかに少ない分割数ですむか」という点に関しては議論されてるものがあるものの，一般にはその一般化・表現法を用いる根拠について十分な議論がなされていない．自律ロボットなどの実環境エージェントへの適用を考えた場合にはむしろ，その状態一般化・表現法が，「多種多様で冗長なセンサー情報を効果的に統合することができるか」，そしてその結果「実環境おいて不可避な様々な不確定性に対して頑強であるか」ということの方がより重要である．しかし，上に挙げた従来手法はいずれも，この点に関して満足のいく解決法になっていない．そこで本研究では，この問題へのアプローチとして，単純ベイズ分類器に基づく状態一般化・表現法を提案する．この手法は異種冗長なセンサー情報源を用いた状態一般化および表現を可能にすることによって，センサーノイズやフォールトなどの不確定性要因に対して頑強な状態認識を実現する．

1.4 論文の構成

本論文は本章（第1章）を含めて9つの章から構成される．

第2章では，本研究が対象とする知的自律エージェント，特に環境とのインタラクションに基づく反射的エージェントの方法論の基本的枠組と，この方法論における行動学習問題につい

て整理を行う．

第3章では，反射的自律エージェントの行動獲得において，状態と行為の抽象化問題がどのように定義され，またどのような重要性を持つのかを明確にする．そして従来研究のアプローチを体系的に整理した後に，この問題における未解決の重要課題を指摘する．

第4章では，状態・行為抽象化問題における第1の課題（前節で述べた後者の課題）－ **異種冗長なセンサー情報の統合と実環境の不確定性に対する頑強化**について，**単純ベイズ分類器に基づく状態一般化・表現法**を提案する．

第5章では，第2の課題（前節で述べた前者の課題）－ **抽象化基準の統合と状態・行為の同時抽象化**に対して，**複数行為結果のばらつき（エントロピー）最小化に基づく状態・行為抽象化法**を提案する．

第6章，第7章ではそれぞれ，第4，5章で提案した手法を自律移動ロボットのゴール到達タスク，および車庫入れタスクに適用したシミュレーション結果を示し，提案手法の有効性を検証する．

第8章では，第6，7章の実験結果を踏まえつつ提案した2つの手法の評価を行い，現状の成果と限界，そしてその改善法について考察を行う．

第9章，すなわち最終章では，本研究で提案した新しい状態・行為抽象化法が自律エージェントの行動獲得問題にもたらす貢献をまとめるとともに，今後の課題と展望を述べる．

第2章　反射的自律エージェントと行動学習

本章では，**自律エージェント**（autonomous agent），－センサーを介して得られた環境に関する情報に基づいて行動を決定し，アクチュエーターを介して環境への作用を行うことによって，合目的的な行動を実現するシステム － に関して行われてきた従来の研究の概略を述べるとともに，それらが抱える問題点を明らかにする．

ここではまず，様々な研究分野でしばしば異なる意味で用いられる**エージェント**という概念の曖昧さをなくすため，本論文におけるその意味を，合理性や自律性などの付随する概念とともに明らかにする．

その後，記号的世界モデルおよびプランニングに基づく方法論について，その基本的な考え方と，その中で仮定されている様々な前提条件について考察する．そして，これらの前提条件が実環境においては必ずしも成り立たないということによって生じる様々な問題点を説明する．

次に，これらの問題を解決するための新しいアプローチとして近年盛んに研究されるようになった，**反射的自律エージェント**という方法論について，環境との密な相互作用，状況依存性，適応性，行動規範，などといった基本コンセプトを説明し，この方法論に基づいて提案されたいくつかの代表的な自律システムアーキテクチュアや，古典的なプランニングシステムとの融合を試みたシステムの例を概観する．

そして最後に，この反射的自律エージェントの方法論において最も中心的な研究対象になっている，**行動決定則の獲得**，すなわち行動学習の問題を取り上げ，これまでどのようなアプローチが試みられて来たか，また，依然どのような問題が残っているかということについて考察する．

2.1　自律エージェント

近年，「エージェント」という言葉は人工知能研究に限らず，情報科学やソフトウェア工学など様々な分野においても盛んに用いられているが，多くの研究者が多様な意味で用いているため，この言葉が指し示す全ての概念を網羅的に表す完全な定義は存在しないということが一般的に言われている．[88][96][115]

その中で，最も一般的な定義として，「エージェントとは自律的なシステムの総称である」[88]というものが挙げられるが，これは単に「エージェントとは何か」という元の問題を，「自律性とは何か」という新たな問題にすり替えているに過ぎないとも言える．そこで，ここでは，より具体的で一般性を失わないエージェントの定義として，「センサを通して環境を知覚し，エ

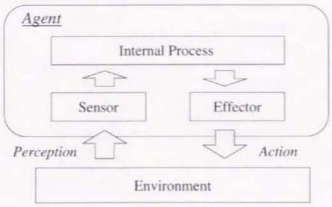


図 2.1: 一般的なエージェントの概念

フェクターを通して環境に対して行動するとみなすことができる何らかのもの」[115]，あるいは「環境を観測して得られた情報に基づき行動する処理系」[96],「認識し，行動する主体」[70] という表現を採用することにする（図 2.1）．

この極めて一般的な「エージェント」の定義において注目すべき第一点は，この定義がエージェントの必要条件として，「環境から情報を得，それに基づき環境に対して作用（働きかけ）を行う」ということしか求めておらず，得られた情報と，実行される作用との間の処理に関して何ら規定がなされていないという点である．したがって，この定義に従えば，エージェントが環境から得る情報と環境に及ぼす作用との間のメカニズムがどのようになっていても良く，また，その結果として実現される行動に何らかの意味があってもなくても，適切なものであってもなくても構わないということになる．つまり，人間や動物のように，生存や種の保存などといった目的に基づいて極めて臨機応変に行動するものから，スイッチのオンオフを感知してついたり消えたりするだけの電気スタンドまで，ほとんど全てのものが「エージェント」と呼べるということになる．また，この定義によれば，エージェントは人間や動物，あるいはロボットのように生物学的，物理的に独立した個体として存在する必要もない．従って，生物の脳細胞の 1 つ 1 つや，1 台のロボットを構成するカメラやモーターなどの個々のパーツなどもエージェントとしてみなすことができるということになる．

このように，本来「エージェント」という言葉は極めて一般的で広範な概念を表すものであり，「全てのものはエージェントとしてみなし得る」という極論も間違いではないと言える．この意味においては，あるものが「エージェントであるかどうか」ということよりも，それを我々が「エージェントとして扱うかどうか」ということが問題であると言える．

しかし，本研究ではこのような一般的な「エージェント」の定義が指し示す全てのものを対象とするのではなく，移動ロボットや惑星探査ローバーなどに代表されるような，より限られたクラスのエージェントを扱うことを目的としている．この限定されたエージェントの概念には，電気スタンドや（個々の）脳細胞は含まれない．そこでここでは，本研究で考慮の対象とす

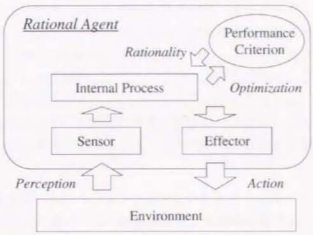


図 2.2: 性能尺度に基づく合理的エージェント

る，より限定された意味でのエージェントを「**知的自律エージェント**」(intelligent autonomous agent) として定義しなおす．知的自律エージェントは，上で述べた最も一般的なエージェントの定義を満たすことに加えて，**合理性** (rationality) と**自律性** (autonomy) という2つの重要な性格を持つものであるとする．また，本研究ではそのような知的自律エージェントのうち，ロボットやソフトウェアシステムのような**人工的**なエージェントをどのようにして構築するか，ということが主要なテーマであり，人間やその他の動物などの**自然**の知的自律エージェントがどのような仕組みになっているかを明らかにすることが目的ではないということも断っておく．以下では，この合理性と自律性がどういうものであるかについて考察する．

合理性

エージェントが「知的である」(ように我々が感じる) ための第一の条件として，その行動が何らかの評価基準の上で「意味を持つ」ということが挙げられる．例えば，地面をランダムに動きまわるようにプログラムされた移動ロボットの行動を「知的」であるとは言いがたいが，それはこの行動が何の役にも立たない，すなわち意味がないであると考えられる．この「役に立つかどうか」,「意味をもつかどうか」という基準を端的に言い表すのが「**合理性**」という概念である．文献 [70] では，エージェントの「合理的な行動」とは,「自己の信念をもとにゴール達成のために行動すること」であるとし,「**理想合理的エージェント**」を「可能な知覚列のすべてに対して，性能尺度を最大にする動作を選択するようなエージェント」として定義している (図2.2)．ここで**性能尺度**とは，エージェントの行動がどの程度成功したかを定性的，あるいは定量的に評価する何らかの評価関数である．例えば，蟻というエージェントの性能尺度は，どれだけ子孫 (遺伝子) を残せるかであると解釈できるし，人間というエージェントの性能尺度には，これに加えて，どれだけ個人が (いろいろな意味で) 満足するか，ということが含まれるであろう．

このような「合理性」の定義において問題となるのは，まず第一点として，あるエージェントが合理的であるかどうかは結局のところ，性能尺度を最大にする「理想的合理的エージェント」の場合を除いて，一般には相対的にしか定まらない，つまり，程度の問題であるということである．したがって，性能指標の値がこれ以上であれば合理的で，これ以下なら合理的でない，というような絶対的な線を引くことはできないということになる．また，もう一つ考慮に入れるべきことは，前述のように本研究では人工的なエージェントのみを考えるが，その場合，エージェントの性能尺度は実質的にエージェントの設計者，あるいは使用者によって決定されるということである．

したがって，本研究では（人工の）エージェントの行動が合理的であるということを，「**設計者あるいはユーザーによって定められた性能指標を，大局的に見て十分に良くすること**」として定義する．

自律性

上で定義した合理性はエージェントが知的であることの必要条件ではあるが，十分条件であるとは言えない．例えば，製パン工場でベルトコンベアーによって運ばれてくる菓子パンを袋詰めする機械は，その作業の正確性や故障率の低さなどによって定義された性能基準を十分に満足することができるという意味で合理的であることは間違いないが，その作業の様子に我々が**知的さ**を感じるとは思われない．

この袋詰め機械という合理的エージェントが「知的」であると感じられない理由は，その行動が至極特定の状況だけを想定した組み込みプログラムによって実現されていて，想定した状況以外では合理的に振舞うことが保証されないからであろう．すなわち，状況の多様性に対する柔軟性が欠如しているということである．逆に言えば，エージェントが知的であるためには，様々な状況に柔軟に対応し，合理的な行動を取れることが重要であるということになる．

この考察から，エージェントが知的であるためのもう一つの条件「**自律性**」を定義することができる．文献[70]では，エージェントの行動は，**あらかじめエージェントに備わっている組み込み知識**と**エージェント自身が直接知覚によって得る情報**によって決定されるとした上で，エージェントが組み込み知識だけに基づいて行動する場合が完全に非自律的であり，逆にエージェントが知覚によって得られた情報（およびその経験）のみに基づいて行動する場合が完全に自律的であると表現している．この考えに基づけば，先の袋詰め機械は，組み込み知識によってその大部分の行動が決定づけられており，その点で自律的であるとは言えないということになる．

このことからわかることは，知的エージェントにとっての**自律性**という基準もまた，程度の問題であり，完全な自律性というものは現実的にはあり得ないということである．なぜならば，人間や動物でさえも直接知覚経験だけで行動を決定しているわけでなく，進化という過程によって獲得された少なからぬ**組み込み知識**によって合理的な行動を実現しているからである．以上

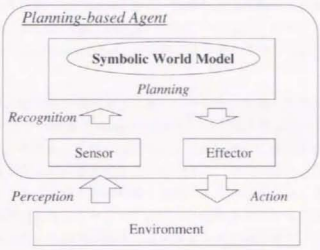


図 2.3: 記号的世界モデルとプランニングに基づくエージェント

のことから，本論文では**自律性**の現実的な定義として，「エージェントが，組み込み知識だけでなく，環境から知覚した経験に基づいて，様々な状況に応じた合理的な行動を取れること」という表現を採用することとする．

なお，この**自律性**の定義は，世間一般で用いられる伝統的な意味「人の直接的な制御下にないこと」には必ずしも一致しない．なぜなら先の袋詰め機械は人間が直接制御しなくても作動するという点ではこの伝統的な自律性の条件を満たしているからである．この点で，伝統的な意味の**自律性**は自動性，自立性が類義であるのに対し，エージェントにおける**自律性**では，適応性，柔軟性などの性格が重要であると言える．

2.2 記号的世界モデルに基づく自律エージェントとその問題点

2.2.1 記号的世界モデルに基づく自律エージェント

前節で述べたような**自律的な知的エージェント**を実現しようとする試みとしては，初期の人工知能研究においては記号的に記述された世界モデルと探索による問題解決，すなわち推論に基づく方法が主流であった．

この方法論では，エージェントがその内部に**世界モデル**（world model）を持つことが大きな特徴になっている（図 2.3）．

この世界モデルは，一階述語論理（first-order predicate）や意味ネットワーク（semantic networks）などに代表されるシンボリックな表現法により記述された，エージェントの状態（state）の集合と，状態間の遷移オペレータとして定義される行為（action）の集合から成り立っている．そしてエージェントは，「この状態においてこの行為を適用することによってこの

状態に遷移する」という**推論**（inference）を行いながら，現状態から目標状態に到達するための経路（行為オペレータ列）を**探索**（search）することによって適切な（合理的な）行動を決定する．この**問題解決**（problem solving）の過程は，一般に**プランニング**（planning）と呼ばれる．

このエージェントの行動全体の過程は，外界からセンサーを通して得られる情報を世界モデル中の状態にマッピングする**認識**（recognition）と，上で述べた探索に基づく**プランニング**，およびその結果得られたオペレータ列を実際のアクションとして適用する**実行**（execution）の3つのフェーズから成り立っていると見ることができる．そしてこの方法論では，記号的な世界モデルに基づいたプランニングの過程が最も重要であると考えられ，残りの2つのフェーズはプランニングの初期状態を与えたり，プランニング結果を実現するための過程に過ぎないと考えられていた．すなわち，この方法論は，エージェント内部に存在する世界モデルの上で推論と探索に基づいてプランニングを行う能力こそが知能（intelligence）であるという考えに基づいていると言える．

この記号的世界モデルとプランニングに基づくエージェントの考え方は，エージェント自身，外部世界，および両者の相互作用に関して，以下に挙げるいくつかの重大な前提を仮定している[1]．

- エージェント内部の記号的モデルが，世界（エージェントと外部環境の状態，および両者の相互作用の法則）を完全に表現する．つまり現実世界におけるエージェントにとって重要な事象が内部モデルによって過不足なく表現される．（**記号的世界モデルの完全性**）
- エージェントのアクション以外の要因による環境の変化はない．（**静的環境**）
- 現実世界，および内部モデルにおけるエージェントの状態遷移は決定論的である．すなわち不確定性を持たない．（**状態遷移の決定性**）
- エージェントはセンサーを介した知覚によって，自身と世界の状態を完全に同定することができる．（**状態認識の完全性**）
- エージェントはアクチュエーターを介した行為によって，期待した状態遷移を実環境において忠実に実現することができる．（**行為実現の完全性**）

これらの前提は全体として，エージェント内部の世界モデルの上で行われるプランニングと実世界における現実の状態遷移とが完全に一致することを仮定していると言える．すなわち，最初の3つの前提によって，内部モデルで行われるプランニング結果が現実世界においても正しい（矛盾がない）ことを保証し，残りの2つの前提によって，そのプランをそのまま実世界で実現できることを保証している．

[1] 記号的世界モデルとプランニングに基づく方法論の枠組の中で，これらの前提を緩和しようという研究の流れも存在する

2.2.2 実環境エージェントへの適用における問題点

記号的世界モデルとプランニングに基づくエージェントの方法論を，惑星探査ロボットなどの実環境自律エージェントに適用することの根本的な問題点は，上で挙げた仮定が現実の実環境ではほとんど成立しないということにある．

まず，この方法論では，記号的な世界モデルが実世界における（エージェントにとって）重要な事象を完全に表し得ることを仮定しているが，現実的にはこの仮定はまず成り立ち得ない．というのは，多数の物体の無数の状態によって成り立っている実環境において，何が重要で何が重要でないかをエージェントが完全に把握し，記号的モデルの中に表現することは非常に困難だからである．つまり，この考えに従えばエージェントのあるアクションによって環境中の何がどう変化し何が変化しないということを（たとえ我々一般的な人間にとっては自明に思われるようなことであっても，）全て網羅的に記述しなければならないが，そのようなモデルを用いてプランニングを行うには，無限のメモリーと推論処理能力を持たなければならない．この問題は**フレーム問題** (frame problem) [102][103] と呼ばれ，より一般的には「限定された情報処理能力しかないシステムがその能力をはるかに上回る複雑性を持つ情報をどのように扱うかという問題」[102] として表現される．フレーム問題の完全な解決は現在では（人間でさえも）不可能であるとされているが，記号的世界モデルに基づく方法論はこの問題の影響を真正面から受けてしまっている．

また，エージェントの内部モデルが実環境と完全に一致しない場合，記号的世界モデルとプランニングに基づく方法論では，このモデルと実世界の間のギャップを推論の過程で解消するようなメカニズムが備わっていないため，エージェントにとって致命的な結果（ループやデッドエンドなど）をもたらす可能性が常に存在するという問題がある．この問題は**シンボル・グラウンディング問題** (symbol grounding problem) と呼ばれ，フレーム問題とともに記号的アプローチを実環境システムに適用しようとする際に免れ得ない課題として指摘されている [99]．

次に，外部環境が静的であるという第二の仮定であるが，これも実環境のアプリケーションではまず成立しない．すなわち，通常の環境はエージェントによる行為以外の外的な要因によって変化し得るという点で動的（dynamic）である．したがって，この環境の動的性格を考慮した場合，世界モデルの記述はさらに複雑になり，上述のフレーム問題，シンボル・グラウンディング問題を一層困難なものにすると考えられる．

同様に，エージェントの状態遷移を決定論的に扱うという第三の仮定も多くの場合不適切であることは容易に想像される．すなわち，実環境における状態の遷移は本質的に不確定性を持つものとして扱わねばならず，それを無視して内部世界モデル上で行われたプランニングの結果は現実の環境に適用しても期待する結果が得られない可能性が高い．

また，センサーによる状態認識，アクチュエータによる行為の実行も決して完璧なものではない．すなわち，センサー入力から得られる情報は質的，量的に限られており，ノイズやフォールトの影響も免れないので，常にエージェントの状態を正確に同定できるという保証はない．また，アクチュエーター出力に関しても実在するハードウェアは不可逆的な誤差を免れ得ない

ため，プランニング結果と同じ状態遷移を実環境で完全に実現できる保証はない．

これらの理由から，現在ではこの方法論だけに基づいて自律的な知的エージェントを構成することは困難であるという考えが支配的である．

2.3 反射的エージェント

2.3.1 基本的枠組

記号的モデル／プランニングに基づくエージェントアプローチの問題点を端的に言えば，エージェントの知的さの実現を内部世界モデル上での推論に基づくプランニングという過程にだけ求め，実際のエージェントと環境との相互作用に注意を払わなかったということに集約される．

ここで述べる**反射的エージェント**というアプローチは，まさにこの記号的アプローチの問題点を踏まえて提案された方法論であると言える．すなわち，この方法論では，エージェントと外部環境との密接な**相互作用**（interaction）の過程の結果として合理的な行動が発現することを知能の本質として考える．したがって，環境から遊離したエージェント内部の記号的モデルの上でのプランニングによる行動決定よりも，環境から得られた情報に基づいて即応的に行動を決定・実行し，その結果を観測によって行動決定にフィードバックするという，エージェントと環境間の相互作用ループを重視する（図 2.4）．この方法論に基づくエージェントが反射的エージェントと呼ばれる理由は，このような密接な相互作用を実現するのがエージェント内部の条件反射的な行動決定だからである．ここで重要なことは，「反射」はエージェントと環境との間の密なインタラクションを実現するための**手段**であって，決してそれ自体が目的なわけではないということである．この点で**反射的エージェント**という名前はこの方法論の手段に着目した呼び方であると言える．

エージェントと環境との，センシングおよびアクチュエーションを介した相互作用は，広い意味でエージェントの**行動**そのものであり，その点でこの方法論は知能の本質が行動にあることを主張している．**行動規範型**（behavior-based）エージェントという呼び方はこのような思想に基づいた名前である．また，環境との密（intensive）な相互作用はエージェントに**状況に即した**（situated）行動をもたらすと考えられる．言い替えれば，反射的エージェントでは組み込み知識よりも，直接知覚される情報と実際の行為結果を重視することによって，状況に対して柔軟，すなわち**適応的**（adaptive）な行動が実現される．この理由から反射的エージェントは，**シチュエーテッド エージェント**（situated agent），**適応エージェント**（adaptive agent）とも呼ばれるが，これはエージェントの行動の性格に基づく呼び方であると言える．

このように，反射的エージェントは，**反射的な行動決定に基づく環境との密な相互作用によって適応的で合理的な行動を実現するエージェント**として一般的に定義することができる．しかし，この方法論に属する様々な具体的なアプローチを分類すると，より細かい部分に関していくつかの異なる立場が存在する．その中でも最も重要な事項は，エージェントがその内部に抽

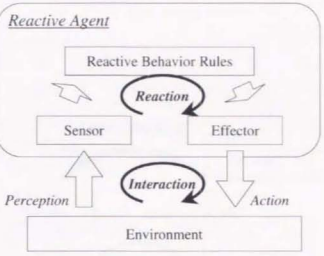


図 2.4: 反射に基づくエージェント

象的な**表象**（representation）を持つことを認めるかどうかということである．抽象的な表象とはこの場合，エージェントがセンサーによって直接知覚したり，アクチュエータによって直接環境に作用する出力よりも抽象度が高い情報を表す属性群のことであるが，記号的世界モデルとプランニングにもとづく方法論における世界モデルは，この抽象的な表象の一つの極限であるとみなせる．反射的エージェントアプローチに関する一連の研究において，先駆的役割を行った R.A.Brooks は，記号的アプローチが直面したフレーム問題やシンボル・グラウンディング問題を引き起こした根源はこの抽象化された表象にあり，自律的エージェントはそのような抽象的表象をどのようなものであれ持つべきではない，という立場を取っている [15]．実際，彼が提案した**サブサンプション アーキテクチュア**（subsumption architecture）[13] は，抽象的な表象を持たず，特定のセンサーとアクチュエーターを直接的に結びつける複数の専門の反射行動モジュールが並列的に働くことによってエージェントの行動を制御するようになっている（図 2.5）．

しかし，この Brooks の主張は反射的エージェント，あるいは行動規範型アーキテクチュアの分野において一般的に受け入れられているとは言いがたい．その大きな理由は，抽象的な表象を一切排除した場合，エージェントの可能なセンサー入力とそれに対応するモーター出力との関係を規定する**行動マッピング**を，組み込み知識として埋め込むにせよ，エージェント自身が経験から獲得するにせよ，表現しエージェントの内部に実現することが困難であるということである．そのため，その他のほとんどの反射的エージェントに基づくアプローチでは，エージェントが何らかの抽象的な表象（信念空間）を持っている（図 2.6）．本論文でも，この事項に関しては Brooks の主張とは異なる立場を取り，エージェントが抽象的な表象を持つこと自体が絶対的に悪いのではなく，そのような表象を人間が天下り的に与えてしまうことが，エージェントの自律性，柔軟性を損なわせる原因になっているという考えに基づく．そして，後述のようにエージェント自身が自ら抽象的な表象を獲得する手法を提案する．

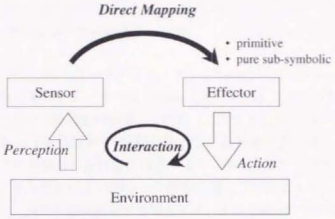


図 2.5: 抽象的内部表象を持たない反射に基づくエージェント

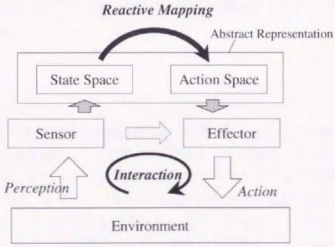


図 2.6: 抽象的内部表象を持つ反射に基づくエージェント

2.3.2 反射的エージェントの代表例

反射的エージェントの方法論の具体的なシステム構成法としては，これまでに多くの方法が提案されている [13][56][4] [48][9] が，ここではその主要な代表例して，Brooks，Maes，Arkin によって提案されたものを紹介する．

Brooks の サブサンプション・アーキテクチュア

Brooks のサブサンプション・アーキテクチュア（subsumption architecture）は，レベルの異なる複数の専門的な行動モジュール（層：layer と呼ばれる）によって構成される．ここで，下位レベルのモジュールは「障害物の回避」,「彷徨（うろつくこと）」などの行動を担当し，上位レベルのモジュールは「物体認識」「経路計画」などの行動を受け持つ．そしてこれらの行動モジュールはそれぞれが担当する行動に関連するセンサーおよびアクチュエータだけと直接的に結び付いており，それぞれが並列的かつ反射的に働くことによってエージェント全体の行動が決定される．また，複数の層が競合する場合には，上位の層が下位の層を抑制（subsume）し，下位の層は上位の層を意識しないようになっている．

サブサンプション・アーキテクチュアの大きな特徴としては，行動決定過程の並列性とともに，前述したように抽象化された表象を全く用いないということが挙げられる．このことは各行動モジュールがハードウェア的な結線によって実現（hard-wired）されなければならないことを意味するが，このようにして上位レベルの行動層を構築することは現実的に非常に困難であると指摘されている．

Maes の 行動ネットワーク

Maes は，エージェントの知覚（perception），行動（behavior），目標（ゴール）の３種のモジュールを活性化（activation），抑制（inhibition）の２種のリンクによってネットワーク状に構成された**行動ネットワーク**による反射的行動アーキテクチュアを提案している [56]．このアーキテクチュアでは，まずエージェントの現在のセンサー入力，および目標に対応して知覚モジュール，目標モジュールが活性化される．次にこれらのノードから２種のリンクを介して活性化エネルギー，あるいは抑制エネルギーが周辺の行動モジュールに伝播される．そして各行動モジュールの活性値があらかじめ決められた閾値を超えたとき，それに対応する行動が実行される．

この反射的エージェントアーキテクチュアの大きな特徴は，エージェントにとってのゴールを，目標モジュールという形で明示的に扱える点であり，そのおかげで合理的な行動の実現が容易になっている．また，Maes 自身が記している行動ネットワークの例によれば，知覚モジュールや行動モジュールはセンサー入力やモーター出力そのものではなくて,「物を荷台にのせているかどうか」,「ものを持ち上げる」などといった，ある程度一般化された状態や行為を表して

いる．すなわち，行動ネットワークは抽象化された表象とみなすことができ，この点で前述のサブサンプション・アーキテクチュアと明らかに異なっている．

Arkin の AuRA アーキテクチュア

Arkinは，人間や動物における知覚や行為のメカニズムを体系づけるものとして認知心理学の分野において盛んに研究されてきたスキーマ理論（schema theory）を応用した反射的エージェントアーキテクチュア，AuRA（Autonomous Robot Architecture）を提案している[4]．

スキーマ（schema）とは，一般的には「構造化された知識の単位」であるが，ここではエージェントの知覚／運動活動の構成単位，あるいは一種のパターンであると考えて良い．AuRAでは，まず現在のエージェントのタスクとゴールから，それを実現する行為パターン，すなわち**モータースキーマ**（motor schema）が決定される．そしてそのモータースキーマを実行するのに必要な情報を収集するための**知覚スキーマ**（perceptual schema）が計算された後に，それらの結果に基づいた実際の行動が実行されるようになっている．

AuRAアーキテクチュアにおける行動決定過程の最大の特徴は，エージェントの行動に関する内部知識がモータースキーマ，知覚スキーマという単位で表現されるという点である．スキーマは抽象化された知識であり，エージェントの内部表象とみなすことができるので，Brooksらの立場とは一線を画している．また，このアーキテクチュアにおけるスキーマは，記号ではなくて力学系的な表現がなされており，エージェントの行動は記号による推論のような手続きではなくて，モータースキーマが形成するポテンシャル場の重ね合わせによって反射的に決定される．

2.3.3 プランニングシステムとのハイブリッドシステム

知能の本質は記号による推論ではなくて，環境との密接な相互作用の結果として合理的な行動が創発されることであるとする反射的エージェントの思想に基づいて，既に多くの実システム，例えば集餌行動を行うロボットなどが作られている．しかし，それらは極めて単純な合理的行動を行うだけで，我々人間にとって本当に嬉しい実環境エージェントと呼ぶには程遠い．この理由は，我々にとって役に立つタスクアプリケーションは，どれも多かれ少なかれ何らかの推論能力を必要とするものだからだと考えられる．実際，知的自律エージェントの究極的目標である人間にそのような推論能力が多かれ少なかれ備わっていることは事実である．純粋な反射的エージェントアプローチの信念に従う立場では，そのような記号に基づく推論能力さえも反射的な相互作用の結果として実現されると主張されるが，実際にはその具体的な方法どころか，それが可能であることを裏付ける証拠すら明かになっていない．

この理由から，従来の記号的世界モデルとプランニングに基づく方法論を完全に否定するのではなく，これと反射的エージェントに基づく方法論とを統合することによって，高度なタス

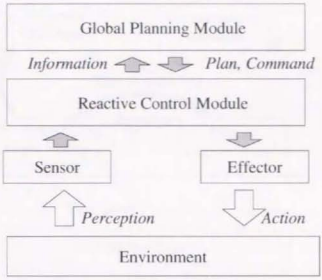


図 2.7: 反射的行動決定とプランニングシステムとのハイブリッドエージェント

クを行う実環境自律システムを作ろうとする様々な試みが行われている．これらのハイブリッドアプローチの基本的な形態はほぼ共通しており，記号的世界モデルに基づくプランニングシステムがエージェントの大局的な行動プランを生成し，それに基づいて反射モジュールが局所的な行動をコントロールするようになっている（図 2.7）．つまり，状況に強く依存しかつ即応性が求められるレベルでの行動が反射モジュールによって制御される一方，より一般的で目的達成のために熟考が要されるようなレベルでの行動はプランニングモジュールで決定される．また，多くのシステムでは，プラン通りの行動実行が不可能な場合には反射モジュールからプランニングモジュールにそれが知らされ，再プランニングが行われるようなメカニズムが考えられている．

このハイブリッドアプローチの代表例としては，プランニングモジュールによる大局的なプランニング結果を疑似センサー（psuedo sensor）の値として反射的行動モジュールに参照させることによって両者を統合する Segre の SEPIA アーキテクチュア [73]，反射的な行動を担当するレイヤーと熟考的な行動を担当するレイヤーから成り，それぞれの黒板システム（black board）間で環境に関する情報とシステムのゴールに関する情報を交換し合う Spector の Supervenient Architecture[74]，反射的行動モジュールと記号的的プランニングモジュールとの間にそれらを仲裁する独立したモジュールを置く Gat の ATLANTIS[38] などが挙げられる．

2.4 反射的行動則の獲得

反射的エージェントに関する研究における最大の問題は，与えられたタスクを適切に遂行するエージェントをどのように設計するのかということである．というのは，一般に、あるタス

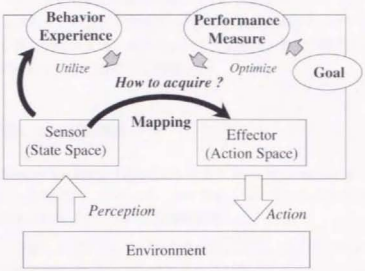


図 2.8: 行動学習を行う反射的エージェント

クを実現するようなシステムを設計することは一種の逆問題ととらえることができるが，反射的エージェントでは，エージェントと環境との複雑な相互作用のループの結果として全体としての行動が実現されるので，この逆問題を人間が直接解くことは極めて困難であると考えられるからである．そのため，与えられたタスクに対してエージェント自身が適切な反射的行動系を獲得するためのメカニズム，すなわち学習系に関する研究が盛んにおこなわれている．

2.4.1 学習エージェントの基本的枠組

学習とは一般的に，「同一の試行もしくは同一の母集団からの試行を繰り返すとき，次の回の試行は前の回のものより良い性能を示すようなシステムの変化」として定義される [104]．反射的自律エージェントの場合，そのシステムとしての挙動は，センサー入力とモーター出力との間のマッピング，あるいはそれらを抽象化した内部表現である状態空間と行為空間との間のマッピングによって決定されるので，その行動学習過程は，「センサー入力（状態）とモーター出力（行為）のマッピング（以後，これを**行動マッピング**と呼ぶ）を経験に基づいて獲得・変更していくことによって，自身の行動性能を向上させる過程」として限定的に定義することができる（図 2.8）．

このような反射的自律エージェントの行動学習機能は一般に，3つの要素から構成される [70]．すなわち，行動マッピングを変更したり新規に定義する**学習エレメント**（learning element），エージェントの知覚経験を監視しその行動の善し悪しを評価する**評価エレメント**（critic），そしてエージェントの学習にとって有用な経験をもたらすような問題（タスク）を示唆する**問題生成エレメント**（problem generator）である．

実際の学習エージェントにおける行動学習部のインプリメンテーションは，行動決定部の形態，すなわち行動マッピングに関する知識をどのように表現するか，また，幅広い機械学習理論の中のどのような具体的手法を用いるかによって，非常に大きな多様性を持っている．以下ではその代表的な研究例を手法に基づいて概観する．

2.4.2 帰納的学習による行動獲得

帰納的学習（inductive learning）は**例からの学習**（learning from experiences），**概念学習**（concept learning）とも呼ばれ，一般には「ある事象について与えられた入出力の事例から，その関数関係の記述を生成すること」として表現できる．

この学習手法は，自律エージェントの行動学習においては，反射的な状態－アクション規則（state-action rule）の獲得に用いられる．すなわち，「状態 s においてアクション a を実行して結果 r を得る」という，エージェントの過去の行動経験の集合を一般化することによって，「ある望ましい結果を得るためには状態 s においてアクション a を実行すれば良い」という有用な一般的行動則を得るというものである．この手法では，同じ結果部を持つ行動経験が一般化されるため，コンパクトな反射行動マッピングが得られる反面，一般化の過程は基本的にはバッチ的な処理によって行われるため，大量の経験例を記憶しておく必要がある．

帰納的学習を行動獲得に用いた例としては，決定木によって自律エージェントの行動を分類／一般化した[62]や，**背景知識**を用いることによって少ない訓練例からの一般化を可能にする**説明に基づく学習**（explanation-based learning : EBL）を適用した[58][73][10]などが挙げられる．

2.4.3 強化学習による行動獲得

強化学習の基本的枠組は，学習者のある行動に対して与えられた**報酬**（reward）に基づき，その行動が選択される強度を修正することによって，学習者の振舞いを望ましいものに近付けていくというものである．その代表的な具体的手法としては，Q 値（Q-value）と呼ばれる各状態における各アクションの効用（utility），すなわち**行為－価値関数**を，直接的な獲得報酬と，状態遷移にともなって与えられる間接報酬に基づいて更新していくことによって，適切な行動政策（離散マルコフ決定過程においては無限期間の割引報酬の合計の期待値を最大化するという意味で最適な政策）を獲得する **Q 学習**（Q-learning）[82]や，エージェントの一連の行動に対して与えられた報酬を強化関数に基づいて分配し，その行動において適用されたルールを一括して強化する**利益共有**（profit sharing）法[116]などが挙げられる．これらの手法は，エージェントの行動決定過程と行動政策学習の並列化，すなわちオンライン学習が可能であり，また帰納的学習のように大量の行動経験を記憶しておく必要がないなどの長所を有するが，タスクが複雑であったり状態空間が大きい場合には膨大な試行を要するという問題もある．

強化学習を用いて自律エージェントの反射的行動を獲得する研究例は非常に多く存在するが，実環境エージェントへの適用を議論したものとしては，[47] などがある．また，先に述べた Maes の**行動ネットワーク**でも適切なネットワークを構築する方法として取り入れられている [55]．

2.4.4 進化的プログラミングによる行動獲得

進化的プログラミング（evolutionary programming）の基本的な枠組は，一個体以上のエージェント集合から始めて，各エージェントの環境に対する適応度を評価する**適応度関数**（fitness function）に基づく「選択」と「再生産」の操作を適用していくことによって，この集合を「進化」させていくというものである．この方法論を自律エージェントの行動獲得に利用した例としては，遺伝的アルゴリズム（genetic algorithm : GA）を分類子システム（classifier system）とともに用いた [25] や，LISP で書かれたエージェントの行動プログラムを直接進化させていく遺伝的プログラミング（genetic programming : GP）を用いた [51] などがある．また，Brooks もサブサンプション・アーキテクチュアの自動設計に対して GP 利用の有効性を示唆している [12]．

2.4.5 自律エージェントの行動学習における問題点- 状態・行為の事前定義

本節で述べた，機械学習理論の様々な手法を反射的自律エージェントの行動獲得問題に適用しようとする試みは，いずれも理想的で単純な環境を仮定した問題から始まり，徐々に実環境特有の様々な性格を考慮するように研究の拡張が進められている．例えば強化学習に基づくアプローチにおいては，単純な離散マルコフ決定過程に従う理想的な環境下ではなく，エージェントの状態に関して不確定で不完全な情報しか得られないという**部分観測マルコフ決定過程**（partially observable Markov decision process : POMDP）における従来の学習アルゴリズムの挙動の分析や拡張に関する研究が盛んに行われている [91]．また，進化的プログラミングに基づくアプローチでも，従来困難とされてきたオンラインの行動進化を扱った研究 [75] などが存在する．

しかし，この分野において依然未解決である重要な課題の一つに，エージェントの行動学習が行われる次元（粒度）をどのように決定するか，という問題がある．従来の研究の大部分では，エージェントの行動獲得という過程が，あらかじめ人間によってヒューリスティックに（経験的に）定義された有限個の状態の集合と行為（アクション）の集合との間のマッピングを学ぶ問題として定式化されているが，実際にそのマッピング学習がうまくいくかどうかは状態集合・行為集合をどのように定義するかということに大きく依存する．したがって本来，自律エージェントの行動獲得は，マッピング学習が行われる状態空間，行為空間自体の定義という過程まで含めて考えられるべきである．また，この過程はエージェント自身による**内部表現獲得**として捉えることができ，人工システムの知能がどのように構築されるべきかという根本問題に深く関わっていると言える．

次章では，この問題をエージェントの行動獲得における状態および行為の抽象化問題として定義し，より詳しい考察を行う．

第3章　自律エージェントにおける状態と行為の抽象化問題

本章では前章で述べた自律エージェントの行動決定過程，および行動学習過程において用いられる状態集合と行為集合をどのようにして自ら獲得するかという問題，すなわち“状態と行為の自律的抽象化問題”について，その意義，従来研究のアプローチ，未解決事項などを整理して述べる．そして本研究における主眼点を明らかにする．

3.1　状態と行為の自律的抽象化

3.1.1　問題の概要

前章で述べたように，反射的自律エージェントにおける重要な問題は，“いかにして適切な反射的行動ルールを獲得するか”ということであり，この問題を解決するために多くの学習手法が提案されて来た．だが，その行動学習に関する研究のほとんどは，あらかじめ人間によって定義されたエージェントの有限個の内部状態と，有限個の行為選択肢との間の適切な[1]関係，すなわち行動ルールあるいは行動政策の獲得に関するものである（図3.1）．それらの中には，“If（状態 S）ならば then（アクション A）を取れ”というプロダクションルール的な規則を学ぶものや，“状態 S においてアクション A を実行することによって最終的に見込まれる報酬”というような有用度（utility）を学ぶタイプのものが含まれる．

しかし，極めて単純で人工的なゲーム世界のような環境におけるエージェントの場合はともかくとして，実環境で活動するエージェントにとって，そのような有限個の状態とアクションをあらかじめ人間が定義することは容易ではない．というのは，そのように定義されたエージェント内部の状態空間・行為空間が，真のエージェントの状況や行動を反映したものであるという保証はなく，また，仮に反映していたとしてもエージェントが実際に環境から得ることができるのは低レベルなセンサー入力であり，環境にもたらすことができるのはやはり低レベルなモーター出力であるが，それらの抽象的な状態および行為と，低レベルなセンサー入力およびモーター出力との間の関係が必ずしも自明ではないからである．

つまり，従来の反射的エージェントの行動決定・行動学習に関する研究は，その大部分が，エージェントの行動決定過程を示した図3.2において，あらかじめ人間が決めた状態空間（B）

[1]ゴール状態に近付くとか，報酬を獲得するとかという意味で

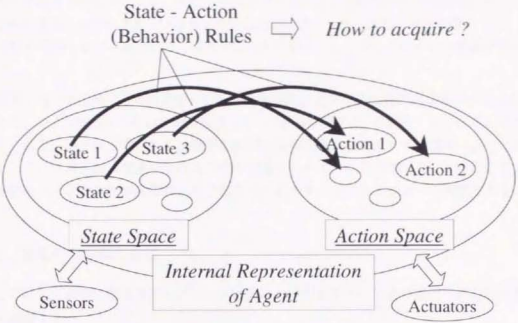


図 3.1: 状態－行為マッピングの獲得による行動学習

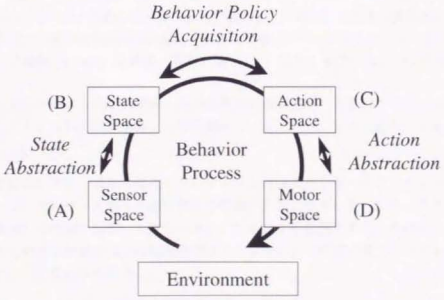


図 3.2: 自律エージェントの行動決定／学習過程

と行為（アクション）空間（C）との間の適切なマッピングを獲得するという問題にのみ論点を置いており，それらの行動学習において用いられる状態空間と行為空間そのものが本当に適切なものであるのかということ，また，それらのエージェント内部の表現空間と，エージェントが実際に環境とやりとりしているセンサー入力およびモーター出力との間の関係（A－B間，C－D間の関係）をどのようにして獲得するのかということについては十分な議論が行われていない．

本研究で扱う“状態・行為の抽象化”という問題は，まさにこの部分を議論の中心とした問題である．つまり図3.2において状態空間（B）－行為空間（C）の間のマッピングを学習する以前に，あるいはそれと並行して，適切な状態空間の定義とセンサー入力空間（A）との間のマッピングの学習，および適切な行為空間の定義とモーター出力空間（D）との間のマッピングの学習をエージェントが自律的に行うためにはどうすべきであるかという問題である．

3.1.2 抽象化の意義と目的

ここではより詳細に実環境エージェントにおける状態および行為の自律的な抽象化の意義と目的を議論する．

まず，自律エージェントの行動獲得問題における従来の多くの研究のように，エージェントが行動決定や行動政策学習に用いる有限個の状態集合と行為集合をあらかじめ人間が定義し，それらの抽象的な状態・行為と，低レベルなセンサー入力・モーター出力との間の関係，すなわち“センサー入力がこういう値の範囲（の組み合わせ）を取ったときを状態S_iとする”とか，“アクションA_jはモーター出力をこういう値（の組合せ）に設定したときに実現されるアクションとする”といった関係をも固定的に定めてしまう問題点について考える．このような“ヒューリスティックで静的（static）な状態・行為定義”には，大きく分けて次の2つの問題がある．

- 人間によるそのような状態空間・行為空間の定義が必ずしも容易でなく，また，そのようにして与えられた状態空間・行為空間がエージェントにとって必ずしも最適だとは限らないという点．
- 実世界においては，環境の変化やエージェント自身のセンサーやアクチュエータの変化によって，センサー入力と内部状態との関係，あるいはモーター出力とアクション（の内部表現）との間の関係にも変化（ずれ）が生じる可能性がある．そのような場合，固定された静的な状態空間，行為空間ではエージェントの行動に著しいパフォーマンス低下が生じると考えられる点．

このうちまず第一点は，人間にせよ，他の自律エージェントにせよ，それにとって“適した”状態空間・行為空間とは，その個体の身体性（すなわちセンサーやアクチュエータなどの性格），取り巻く環境の性質，目的とするタスクによって大きく異なるということに関連する．例えば，同じ人類でも身体の特徴や生活する環境，および生きることの目的（価値観）を異にする民族

同士では，概念の形成の仕方，すなわち“世界の切り取り方（分節の仕方）”は異なってくる[2]．これはロボットなどの人工的なエージェントにとっても全く同じであり，持っているセンサーやアクチュエータ，与えられたタスク，そのタスクが行われる環境などが異なればそのエージェントにとって最適な状態と行為の分節の仕方は異なるはずである．したがって，持っているセンサーやアクチュエータが少なく，タスクや環境も単純なエージェントの場合はともかく，多数の（しかも我々の感覚系・運動系とは根本的に異なる）センサーやアクチュエータを持つような複雑なエージェントのための状態空間・行為空間を我々人間が最適に定義することは困難である．

第二点目も上の第一点と同様，エージェントにとって適した内部表現というものがそのエージェントの身体性と環境の性格に強く依存することに関連する．例えば，車体のまわりにいくつかの超音波センサーを持つ移動ロボットエージェントが，障害物にぶつかることなく所定の位置に移動するようなタスクを考える．設計者が定義した状態空間はおそらく，この複数の超音波センサーの信号レベルを適当な範囲に区切ったものの組合せであり，行為（アクション）集合はロボットの車輪の特定の回転数（比）によって規定された前進，後退，左回転，右回転などの我々にとって常識的なものであるだろう．この天下り的な状態空間，行為集合は設計者が想定したような環境においてはうまく機能する．すなわち，エージェント（ロボット）はこの状態集合とアクション集合を使って適切な行動政策（両者の適切なマッピング）を獲得することができると考えられる．しかし，ひとたびセンサーの特性が変化（例えばある超音波センサーの取付向きが変わってしまう，供給電圧の低下によりセンサーが返す信号レベルが一定値ずれてしまう，など）したり，アクチュエータの入出力関係が変化（例えば，片方の車輪のモーター出力が変化してしまって，今までの“前進”コマンドでは右側にずれてしまうようになる，など）したり，あるいは環境自体が変化（例えば雨が降って地面の摩擦係数が変化するとか，障害物表面の超音波の反射特性が変化するなど）してしまった場合，センサー入力空間と状態空間との間の対応関係（図 3.2 の A-B 間），あるいは，モーター出力空間と行為空間との間の対応関係（図 3.2 の C-D 間）が変わってしまい，それまでに獲得した行動政策の一部は無効になってしまう．そのような場合，行動政策の再獲得，すなわち状態空間と行為空間とのマッピング（B-C 間）の再構築によってある程度は機能の回復が見込まれるであろう．しかし，より根本的な解決は，変化してしまったセンサー，アクチュエータ，環境の特質に基づいて状態空間，行為集合を再構築しなければ期待できない．この観点からもエージェントが自律的にその状態空間，行為空間を構成することが重要であることがわかる．

さて，上で挙げた 2 つの問題はいずれもエージェントが抽象化された状態空間や行為空間という，ある種の内部表現を持つことに由来するのだと考える立場がある．というのは，これらの問題はいずれも結局のところ，「エージェントがある行為を行うにあたって，環境の中でその行為と関連していることが何であるかを効率的に見分けるにはどうしたら良いか」という**フレーム問題**の一部であるとみなせるが，エージェントが何らかの方法によって抽象化された有限な内部表象である状態空間や行為空間に基づいて行動決定を行う限り，それが人間が与えたものであれ，エージェント自身が獲得したものであれ，所詮，記号論的な世界モデルの操作に基づ

[2]その違いは日常使われる“ことば”の違いとして現れる

く古典的人工知能と同様，このフレーム問題を免れ得ないだろうと思われるからである．

ギブソンによって創始された，認知科学における新しいパラダイムであるアフォーダンス理論[40][103]では，人間などの動物は，環境から得られる知覚入力を内部的に（脳で）処理して抽象的な意味を得るのではなく，環境に存在しているあらゆるものがはじめから（動物によって知覚されるべき）価値，情報を持っており，人間や動物はその混沌とした情報のプールの中を探索し，必要な情報を取り出しているのだと考えることによって，フレーム問題に陥らないようになっているとしている．また，前述のサブサンプション アーキテクチュア[13]は，これと似た考えを実際に人工的なエージェント，ロボットに適用したものとして解釈できると言われている．すなわちサブサンプションアーキテクチュアの考えでは，エージェントは従来の古典 AI のような内部表象を一切持たず，知覚（センサー）と行為 (アクチュエータ) を直接的に結びつけたいくつものレイヤー間の競合によって行動が "発現" するようになっている．

このようにアフォーダンス理論やサブサンプションアーキテクチュアの考え方は確かにフレーム問題を解決する糸口になる可能性を持っている．しかし，アフォーダンス理論はその環境の認知方法を具現化するシステムの構築法までは現在までのところ提供していない．例えば，アフォーダンス理論では，情報は環境自体に存在しているとされているが，エージェントがその情報をどうやってピックアップするのか，ということに関しては明確な答が与えられていない．また，そのピックアップされた情報からどうやって行動を決定したり，学習するのか，という問題についても同様である．サブサンプションアーキテクチュアにおいても，センサー入力とアクチュエータ出力を直接的に結ぶ行動レイヤを一般的にどうやって作るのか，あるいはエージェントがどのようにして獲得するのか，ということは解決されていない．実際，サブサンプションアーキテクチュアに基づく，あるいは触発された学習エージェントも多く提案されているが，そのほとんどにおいてはエージェントが利用出来る情報に関してなんらかの抽象化があらかじめ施されてしまっており，結局のところ前述の反射的エージェントの枠組（状態空間−行為空間のマッピングの獲得）に収まってしまっているように見受けられている．また，人間や動物などの実在する知的エージェントが，内部表象を全く使わないで行動を行っているということも真実ではないと考えられる．というのは，人間や一部の高等生物は明らかに自分をとりまく状況を汎化する，すなわち抽象化して扱う能力を持っているし，人間もフレーム問題に陥る状況があるということも言われているからである[102]．

このように，実環境エージェントが抽象化された状態空間や行為集合を用いることの是非については，今なお様々な議論が展開されている状況にある．しかし本研究は，"どんな形であれ抽象化された内部表象を使う以上，フレーム問題を根本的に解決することはできない" と認めた上で，自律ロボットなど比較的明確なタスクとゴールを持つような実環境エージェントを実現する場合には，現段階では何らかの内部表象を持つことが，実環境中の膨大な情報を効率的に扱う上で現実的な方法であるという考えに基づく．そしてその上で最も重要なことは，そのような内部表象が従来のように設計者によって天下り的・静的に与えられるのではなく，エージェント自身によってその身体性，環境，タスクの特質に適したものが自律的に構成されるということ，そしてセンサーやアクチュエータ，環境の変化に対してはそれらの柔軟な再構成が

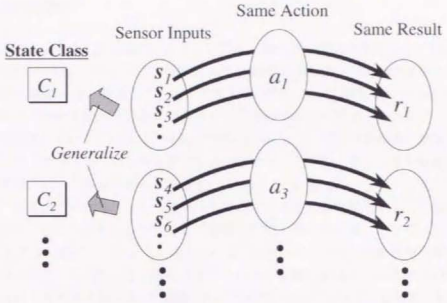


図 3.3: 行為と結果の同一性（類似性）に基づく状態抽象化

行われることであるという立場を取る．

また，ローレベルなセンサー入力およびモーター出力からの状態および行為の抽象化という問題は，－ 本研究で扱う範囲を越えるが － 反射的エージェントと古典的 AI において研究されてきたプランニングに基づくシステムとの間の溝を埋める（実環境に即した）シンボル生成という問題，さらにはエージェント同士のコミュニケーションに用いられる言語の自律的獲得という問題に対しても重要な関連を持つものであると考えられる．

3.2 従来研究における抽象化の方法

ここでは反射的エージェントによる自律的な状態と行為の抽象化に関してこれまで行われた従来研究のアプローチを整理するとともに，この問題において何が未解決な課題であるのかを考察する．

本論文ではここまで「状態と行為の抽象化」という表現を用いてきたが，これに関連する従来研究はこれまでのところ，状態の抽象化と行為の抽象化のどちらか一方のみを扱ったものがほとんどである．そこでこの節でもこの 2 つのテーマについてそれぞれどのような試みが行われて来たのかを述べる．

3.2.1 状態の抽象化

反射的エージェントの状態抽象化に関する研究は，既に多くの研究者によって行われている．それらの大部分は Q-Learning などの報酬に基づく強化学習において状態空間をどう適切に構成するかという問題と深く関連している．これらの研究において提案されている手法は一見，状態の一般化の仕方や表現の方法などの点において互いに非常に異なっているが，その基本的なアイデアは共通している．それは，“**同じアクションによって同じ（類似の）結果をもたらすような状態（センサー入力ベクトル）同士を近いものとして考え，それらを一般化して状態クラスを定義する**”というものである（図 3.3）．例えば，“前進（というアクション）を行うことによって，目的地への到達（という結果）を得るような前状態の集合を 1 つの状態クラスとして一般化する”ということである．ここで重要なのは，異なる状態（センサー入力）同士の“近さ”を定義するために，アクション同士（行為）の同一性，および行動結果の類似性が用いられている点である．したがってこれらのアクションと行動結果はあらかじめ定義されていなければならない．先の例で言えば，“前進”というアクション，および“目的地に到達する”という結果がどういうものであるかということがあらかじめ決められている．

以下に挙げる関連研究は，いずれもこの基本的枠組に基づきつつも，何を“行動結果”として考えるかという点と，“近い”とみなされた状態同士をどのような表現法によって一般化しているかという点において異なっていると考えることができる．例えば，同じ**報酬**を獲得することを同じ結果とみなすものもあれば，センサー入力の変化の仕方によって行動結果を定義するものもある．また，各センサー入力を属性とみなした決定木によって一般化された状態を表現するものもあれば，センサー入力のベクトル空間を超楕円体領域によって分割することによって表現するものもある．

［浅田の方法］

浅田ら [5] は同じアクション（要素の繰り返し）によるゴール状態への到達，あるいは既に定義されたサブゴールへの到達に基づく状態空間の構成法を提案している．

この手法ではまず，あらかじめ定義されたゴール状態，もしくは既に定義されたサブゴール状態に同じアクションによって到達するようなセンサー入力がエージェントの経験から収集され，分類される．そして各集合ごとにその平均ベクトル（中心点）と分散共分散行列が計算され，その中心点からのマハラノビス距離がある値以下となる領域（超楕円体になる）によって新しいサブゴールが定義される．新しいサブゴールの生成は最終的なゴール状態に隣接する領域から次第に遠い領域へと広がって行き，エージェントのセンサー入力空間がサブゴールによって覆い尽くされるまで続けられる．なお，各サブゴールが定義される際に，その領域が表す状態においてとるべきアクションも同時に学習されるので，別途行動政策（状態−アクションマッピング）の学習を行う必要がない．しかしその反面，状態抽象化の初期過程においてはゴール状態に到達したときのみ学習が行われることになるので，それ以外の多くの行動経験は無駄に

なるという点で効率は悪い．また，マハラノビス距離を用いていることからもわかるように，センサー入力は全て連続であると仮定している．

［上野の方法］

上野らのASRR（Adaptive Situation Recognition based on Rewards）は上の浅田らの方法をさらに発展させた状態の抽象化法だと言える．つまり，ゴール・サブゴール状態への到達だけでなく報酬の獲得と他状態への遷移に基づいてセンサー入力が分類・一般化される．また，一般化された状態の表現に関しても，マハラノビス楕円体による大局的な領域分割に加えて，特に境界周辺では最近傍法（nearest neighbourhood method）を併用することによって精度の高い状態表現を可能にしている．

上野らはまた，ASRRによって抽象化された状態空間を用いて部分的プランニングと実行を交互に行いつつ（インターリーブプランニング），行動規則の学習を報酬に基づいて行う，IPRL（interleave planning based reinforcement learning：インターリーブプランニングに基づく強化学習）を提案している．

［石黒の方法］

石黒らは，エージェントのプリミティブなセンサー入力から作り出された抽象属性であるEOP（Empirically Obtained Perceiver）によって状態空間を構成する方法を提案している．EOPはセンサー入力空間中で定義された線形判別関数（linear discriminant function）であり，センサー入力空間全体はこの多数のEOPにより階層的に分割される．状態の分割にあたっては，アクションと報酬の類似性に基づいた政策が採用されている．すなわち，エージェントの行動経験はまず取られたアクションによって分類された後，獲得報酬について頻度分布に基づきクラスタリングを行い，それらのクラスターをうまく分割するようにEOPが定義される．

問題点としては，直接得られる報酬についてのみ基づいているので，報酬を直接得ることのない状態領域では分割が行われないということ，また，線形判別関数に基づいたEOPもまた，連続的なセンサー入力空間のみを想定しているという点である．

［高橋の方法］

高橋ら[77]は同じアクションによるセンサー入力変化の類似性に基づく状態分割ストラテジーと，最近傍法（nearest neighbourhood method）によるノンパラメトリックな状態表現法を提案している．この状態分割ストラテジーが他の方法と著しく異なる点は，ゴール状態への到達や報酬の獲得という情報は一切使わず，センサー入力自体の変化の類似性のみに基づいて状態の分割を行っている点である．すなわち，まず上の3つの方法と同様に行動経験データが

アクションに基づいて分類され，次にセンサー入力の微小変化ベクトル δ をもとのセンサー入力ベクトル s によって線形モデルによって回帰（regression）を行う．このときのデータのモデルからのばらつきがある閾値を超えた場合には，データ集合を重み付きユークリッドノルムを距離尺度として用いたクラスタリングによって2つの集合に分ける，ということをデータのモデルからのばらつきが閾値内に収まるまで繰り返す．そして最終的に得られた各行動経験集合のセンサー入力部に最近傍法を適用することによってセンサー入力空間全体を分割する．

この方法の特徴は状態分割がゴール到達や報酬獲得といったタスクに強く依存する情報によらないため，得られた状態集合が同じエージェントの異なるタスクにも再利用しやすいという点である．このことは同時に，行動政策（状態−アクションルール）の獲得が，状態の抽象化とは別途に行われる必要があることを意味している．ここで問題なのは，センサー入力変化の類似性に基づいた状態分割が必ずしも報酬の分布を忠実に反映するとは限らないということである．そのためこの方法によって得られた状態空間を用いて行動政策学習を行っても低いパフォーマンス（獲得報酬の期待値など）しか得られないという可能性がある．また，最近傍法を用いた状態クラス表現は，どんな複雑な形をした領域でも精密に表現できるが，新しいセンサー入力を分類するときに各状態に属するインスタンス（の代表点）との距離を計算するので，その分の記憶コスト，計算コストは高くなると考えられる．

［Chapmapn の方法］

Chapman ら [16] は Q-Learning におけるセンサー入力一般化問題（sensor input generalization problem）へのアプローチとして，ビット列によってエンコーディングされた状態空間を報酬値に基づいて分割していく G アルゴリズムを提案している．G-アルゴリズムでは行動政策獲得のための Q 学習に並行して，状態表現のビット列中の各ビットがエージェントの直接獲得報酬および将来見込まれる割引報酬に強く関わっているかどうかのテストを行う．そして関連していると判断された場合にはそのビット（の 0 − 1）に関して分割を行うことで最初は 1 つであった状態空間を 2 分木（binary tree）によって次第に細分化していく．

ただしこの方法で扱われている問題は，プリミティブなセンサー入力空間から一般的状態空間への抽象化というものではなく，あらかじめビット列によって離散的に表現されている状態空間をどう適切に分割していくかというものである．したがって，連続的な値を取るセンサー群によって構成されたセンサー空間の抽象化においても同様に有効かどうかは不明である．というのは，G-アルゴリズムでは 1 ビットごとに分割を行うので，実数値を高次のビット列にエンコーディングしているような場合，明らかに非効率的な分割が行われると考えられるからである．

[Albusの方法]

Albus[2]らはローレベルなセンサー空間，アクチュエーション空間で記述されたエージェントの“良い”行動経験をクラスタリングして一般化することによってゴール状態へ到達するための一般的な因果則（“状態Sにおいてゴール Gに到達するためにはアクション Aを行えば良い”というもの）を獲得する方法を提案している．

このアルゴリズムではゴール状態かサブゴール状態へ到達するような行動経験が“良い”経験として記録され，アクションが適用されたセンサー状態と適用されたアクションを一緒にしたベクトル空間の中でクラスタリングされる．そして各クラスターについて前条件部とアクションがそれぞれ一般化され，“この状態においてこのアクションを行えば，良い結果が得られる”という一般的な因果則を得る．その意味では浅田らの方法と同様に，ゴール・サブゴールに基づく抽象化方法の1つとみなすことができる．この手法の注目すべき点は，枠組自体は状態の抽象化とアクションの抽象化の両方を包含しているということである．しかしながら実際にはセンサー－アクチュエータベクトルをどのようにクラスタリングするのかということが明らかでなく，例題の中でも既に抽象化されたアクション（前進，回転など）が用いられているので，実質的には状態の一般化のみが自律的に行われていると考えられる．また，センサー情報（例えば「ゴールへの向き」）とアクチュエータコマンド（例えば「前進」）とを単純な足し算／引き算することによって新たな属性を定義し，状態の表現に用いるなど，不明確な点も多い．

[Mooreの方法]

Moore[59]はあらかじめ粗く分割された状態空間を，状態遷移の不確からしさに基づいてさらに細分化していくParti-gameアルゴリズムを提案している．このアルゴリズムは，本来本質的に異なる状態であるのに不適切な状態空間構成によって同じ状態として見なされてしまっているような状態を，特定のアクションによる状態遷移の違いに基づいて徐々に細分化するというものである．すなわちこのアルゴリズムはChapmanらの方法と同様，現在の状態空間を行動経験に関するある統計量に基づいて変化（細分化）させていくというものであり，経験データをバッチ的に一般化して状態クラス集合を得るような他の方法に比べて，オンライン化するのに適している．

しかし，極めて単純な状態遷移モデルを用いていることや，エージェントの実際のセンサー入力空間を用いるのではなく真の状態が直接的に観測できることを前提としている点，アクションをはじめから「ある状態からある状態への遷移」として定義している点，状態の分割が単純に各状態変数の軸に対して垂直に2等分するだけなので不必要な細分化が行われ易い点など，実環境エージェントの状態抽象化問題に適用する上では多くの問題が指摘されている．

［中須賀の方法］

中須賀 [62] らはエージェントをランダムに行動させて集めた行動経験のデータ集合から帰納的な学習法により一般的な因果則（causal relationship rules）を獲得する方法を提案している．この方法ではゴールセンサーと呼ばれるエージェントのタスクゴールへの距離を示す属性がある特定の値を取るためには，その前の状態で各センサー値がどのような値を取っているべきかということが，過去の行動経験に基づいて帰納的に学ばれ，その前状態がサブゴールとして一般化されていく．ここで特筆すべき点は，他の手法では「ある決まった（certain）アクション」によって起こされた状態遷移の結果の類似性に基づいて状態が一般化されるのに対して，この手法では「何らかの（any）アクション」によって既に定義されているサブゴール状態（S_a とする）に到達するような前状態が新たなサブゴール（S_b とする）として定義されるという点である．そのため，実際の S_b から S_a への状態遷移はある特定のアクションによって常に実現されるわけではなく，場合によって異なるアクションが取られることになる．例えば「移動ロボットエージェントが何か障害物に接している」という状態 S_b から，「何にも接していない」という状態 S_a に遷移するためには，ロボットの前の部分が触れている場合には後退，後ろ部分が触れている場合には前進，という正反対のアクションを取ることになるが，この手法ではどちらの場合も包含する形で前状態の一般化，状態遷移の一般化が行われる．（実際にどのアクションを取るべきかということは決定木によって表現される）

このようにして獲得された一般的な状態遷移因果則は，各ルールの前条件部と後条件部（post-condition）とのマッチングにより，サブサンプションアーキテクチュアライクな階層的な行動ネットワークへとコンパイルされるようになっている．

しかしこの手法は，センサー入力が本当の意味でのプリミティブなセンサー情報ではなく，ある特定の値を取ることが望ましくなるように定義された恣意的な属性として定義されている．言い替えれば，それぞれのセンサーがエージェントにとってのゴール／サブゴールや報酬への近さ／遠さを反映するように定義されている必要がある．さらにそのセンサー値が最初ある程度離散化されているということもあり，完全な意味での自律的な状態抽象化を実現しているとは言いがたい．

［椹木の方法］

椹木ら [72] もまた，過去の行動経験集合をもとに，「同じアクションによって得られる行動結果のばらつきが小さくなるように」状態集合を一般化する方法を提案している．この手法では，ある状態分割の仕方の善し悪しを評価する基準として，概念クラスタリング手法である COBWEB[31] において用いられた CU（categorization utility）を変形したものを用いている．この基準は同じアクションによって同じ行動結果が得られるように状態分割を行ったときほど，大きな値になるようになっており，これを（局所）最大化するように行動経験の階層的なクラスタリングが行われ，状態が一般化されていく．椹木らはさらに，この手法によって一般化さ

れた状態空間を用いて強化学習（Q-Learning）を行う方法を示している．

しかし，この手法もまた，プリミティブなセンサー入力からの状態抽象化を実現したものではなく，既に人間の手によってかなり抽象化されてしまっている属性（例えば，温度が高い／低い，（観測する物体の）大きさが大きい／小さい，など）によって表現される比較的小さな状態空間をさらに一般化するという観点で提案されており，よりローレベルなセンサー入力からの抽象化にも適用できるかどうかは明らかでない．また，行動結果として想定しているものも恣意的であり（「状態が変化しない」，「（自分が）壊れる」など），実環境エージェントへの適用は難しいと考えられる．

［矢入の方法］

矢入ら [85] もまた，ゴール／サブゴール状態への到達に基づく状態抽象化法を提案している．この手法では [5] や [16] などと同様に，始めは最終的なゴール状態領域のみが定義されているが，特定のアクションによってこのゴール状態，または既に定義されているサブゴール状態へ到達するような前状態におけるセンサー入力が一般化されることによって，新しいサブゴールが定義される．したがって [5] と同様，状態の一般化とともにその状態における適切なアクションも同時に学習されるが，状態抽象化に要する行動経験データは一般に多い．

この手法の第一の特徴は，状態クラス（サブゴール）が超矩形領域（hyper-rectangle）によって表される点である．すなわち，各サブゴール領域は，センサー入力軸に垂直な超平面によって仕切られている．この状態表現方法は経験データからの一般化におけるコストは比較的小さく済むが，領域を正確に表現するという観点からは（最近傍法などを用いた方法と比較して）不利である．つまり，各状態クラスの領域が正確に表現されないために，状態抽象化後のエージェントの状態遷移が意図したものと異なった結果になる可能性が高くなる．しかし，実際にはそのような状態表現の不正確さによる誤った状態遷移が生じても，遷移先の状態で適切とされるアクションを再び選択することによって最終的にゴール状態に到達できる場合が多い．さらにこの問題は，この手法のもう一つの特徴であるサブゴール同士のオーバーラップによって緩和される．すなわち，他の状態抽象化法では大抵，状態クラスは排他的，すなわち互いに重なる部分がないように定義される．それに対してこの手法ではサブゴール状態同士がある程度重なり合うことを許している．つまり，一つのセンサー入力が2つ以上の状態クラスに属することが起こり得る．このときゴール到達までに見込まれるコストや過去の行動履歴などを考慮してどのサブゴールとしてみなすかを決定することにより，デッドロックなどに陥りにくいロバストな行動決定を実現している．

3.2.2 行為の抽象化

ローレベルなモーター出力空間を自律的に抽象化し，一般的なアクション（行為）クラスを定義する“行為抽象化”（アクション抽象化）に関する研究は，上で述べた状態抽象化に関する

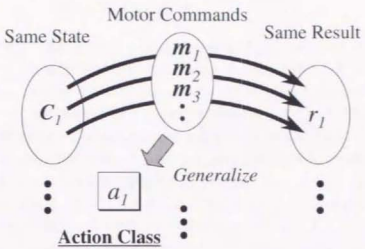


図 3.4: 状態変化と結果の同一性（類似性）に基づく行為抽象化

研究に比べると極端に少ない．その理由の１つは，人間にとってエージェントのアクション集合を定義することが，状態を定義することに比べて一般的に容易であるということが考えられる．というのは，行動学習に関する諸研究で取り上げられている問題を見てみてもわかるように，一般にあるエージェントの（利用可能な）アクション数やモーター出力の次元は，(区別すべき）状態の数やセンサー入力の次元に比べて少ないということ，エージェントのあるモーターコマンドがどのような状態変化をもたらすかということが人間に取って想像しやすいということ，そして状態の抽象化の方はセンサー入力空間をきれいに分割仕尽くさなければならないのに対し，アクションの方はセンサー出力空間の中で代表的な点だけを定義するだけで済む場合がほとんどであるからである．

しかし，エージェントのアクションが多数のモーター出力の和として実現されるような場合には，ローレベルなモーターベクトル空間中の点で表されるモーターコマンドのうち，エージェントの望ましい状態遷移をもたらすようなものをアクションとして一般化することは意味がある．また，環境やモーターの性格が変化してそれまでのアクションでは望みの状態遷移が得られなくなってしまったような場合にアクション集合を再構成するということも考えられる．

アクション抽象化のアプローチとしてまず考えられるのは，先に述べた状態抽象化の方針とアナロジカルなものである．つまり，“同じ状態（クラス），あるいは近いセンサー入力において適用した場合に同じ行動結果をもたらすようなモーターコマンドを同じアクション（行為）として一般化する”というものである（図 3.4）．以下ではそのような行為抽象化に関する代表的な研究を２つ挙げる．

［中村の方法］

中村ら [61] が提案した運動スケッチ（motion sketch）は，エージェントの様々なアクションがもたらすセンサー入力変化の主成分を求めることによって，代表的なアクションを抽出するというものである．この方法ではまず，エージェントはアクションをランダムに選択実行し，そのときのセンサー入力の変化を記録する．そしてそのデータについて特異値分解による主成分解析を行い，主成分ベクトルを求める．この主成分ベクトルはエージェントの代表的なアクションによるセンサー入力変化を表しており，それらの主成分ベクトルの線形和によって他の全てのアクションによるセンサー変化が表される．つまり，エージェントのアクション空間はこのセンサー変化の主成分ベクトルをもたらすアクションによって抽象化されたことになる．具体的にこの研究では，移動ロボットの２５の移動コマンドに対するカメラ画像のオプティカルフローを記録し，そこから２つの主成分（それぞれ「右回転」,「後進」を実行したときのオプティカルフローに対応する）が求められている．

この方法が最初に述べたアクション抽象化の基本的枠組と異なっている点は，アクションをモーター空間で一般化するのではなく，アクションがもたらすセンサー入力変化のベクトル空間の中で一般化を行っている点である．したがって，観測されたセンサー入力変化からエージェントが現在どのようなアクションを行っているかを推測するなどの目的で利用できるが，プリミティブなモーター空間から一般的なアクションを切り出すという目的には直接使えない．また，オプティカルフロー以外のセンサー入力，例えば離散的な値を取る接触センサーからの入力などでも一般的に利用できるかが明らかになっていない．

［Iba の方法］

Iba らは、ロボットアーム (実際には平面リンク機構) の様々な動き（例えば「投げる（throw）」）に関するデータを収集し，これをインクリメンタルにクラスタリングすることによってモータースキーマ（motor schema）として一般化するシステム OXBOW を提案している [44]．OXBOW では、エージェントのアクションは状態ベクトル（この場合は各リンクの位置・速度などの変数から構成されている）のシーケンスによって表現されており，一般化されたモータースキーマは、各変数についての条件が確率分布によって表される．つまり，ここではセンサー入力とモーター出力との区別が厳密になされていない．クラスタリングのアルゴリズム自体は [72] と同様，COBWEB[30] を応用したものであり，CU をクラスタリング結果の評価基準として，データの入力に対してインクリメンタルにアクションの一般化が行われていく．

ただし，OXBOW でも先の中村らの方法と同様，状態遷移の類似性のみに基づいてアクションが一般化されており，報酬やゴールなどの行為結果などに関する情報は使われていない．そのためこの方法によって一般化されたアクション集合が必ずしも報酬獲得やゴール到達を目的とする行動政策学習に適している保証はない．

3.2.3 従来の状態・行為抽象化法の問題点

本節で述べた反射的エージェントにおける状態抽象化・および行為抽象化に関する従来研究のアプローチは，いずれも基本的には共通のアイデアに基づいていると言って良い．すなわちそれは，“過去の行動経験において，同じアクションによって同じ行為結果が得られるようなセンサー入力の集合を同じ状態として一般化し，同様に同じ状態あるいは類似のセンサー入力において同じ行為結果をもたらすようなモーター出力の集合を同じアクションとして一般化する”という方針である．本論文ではこのような考えに基づいた抽象化法を，**“行為結果の類似性に基づく経験的状態・行為抽象化”**と呼ぶ．

ここでまず興味深い問題は，この抽象化基準の妥当性に関するものである．本章の冒頭で述べたように，状態抽象化および行為抽象化の本来の目的は，エージェントにとって“最適な”状態空間および行為集合を自律的に獲得することであり，“最適な”状態空間および行為集合とはこの場合，それらを用いて行うプランニングや行動政策学習が，効率（コスト）および最終的な行動パフォーマンスの面において優れているような状態空間および行為集合ということになる．しかし，これらの基準はエージェントのプランニングや行動政策学習の結果に依存するため，状態・行為の抽象化の評価基準として直接的に用いることは難しい．そのため，先に述べた従来研究においては，「“同じ状態において同じアクションを実行して得られた行動結果が同じものになる”ように構成された状態空間・アクション空間ほど，エージェントの行動パフォーマンスは良くなる」というヒューリスティックを暗黙のうちに用いているということができる．この仮定が直観的には概ね正しいものであろうということは，“同じアクションを適用してもその結果がどうなるかまるで想像がつかない”ような気まぐれな状態集合・行為集合を用いて報酬に基づく行動政策学習や，探索に基づくプランニングを行った場合と，“同じアクションを適用すれば常にほぼ同じ結果が得られるような”決定論的な状態遷移が実現される状態集合・行為集合を用いてそれらを行った場合とで，どちらの方が学習効率や最終的なパフォーマンスの面で良くなるか，ということを考えても容易に想像がつく．無論，この仮定が厳密な意味で正しいかどうかについては，より注意深く調べる必要がある．しかし，本研究では，従来の諸研究と同様，このヒューリスティックが妥当なものであると仮定した上で，今後の議論を進める．

さて，上で挙げた従来の手法は，いずれもこの行為結果の類似性に基づく経験的状態・行為抽象化の枠組の中で実現された具体的手法であると言えるが，その中でそれぞれの特徴を作り出しているのは次の2点に関する違いだと考えられる．

- エージェントの行為結果として何を考えることによって，状態・アクションの類似性を定義するか．
- 同じ状態，あるいは同じアクションとして分類されたセンサー入力集合，モーター出力集合を実際にどのような表現形態によって一般化するか．

まず前者の問題に関しては，行為結果の類似性に基づく経験的状態・行為抽象化では，エージェントに“同じ”あるいは“近い”行動結果（outcome）をもたらすようなセンサー入力同士，

およびモーター出力同士が同じ状態，同じアクションとして見なされ，一般化される．ここで“同じ（近い）行動結果”という言葉は曖昧な表現であり，様々なものが考えられる．実際，従来研究では大きく分けて，“同じゴール状態あるいはサブゴール状態へ到達すること”，“同じ報酬値を環境から獲得すること”，“センサー入力の変化が近いこと”などが“同じ（近い）行動結果”として用いられ，それぞれの基準に基づいて状態の一般化，行為の一般化が行われているが，これらの違いはエージェントの状態・行為抽象化の結果の違いに反映されると考えられる．つまり，行為結果の類似性に基づく抽象化では，行為結果として何を考えるかということの違いが**抽象化ポリシーの違い**となる．しかしながら，従来研究ではこの行為結果の違い，抽象化ポリシーの違いが本質的に何を意味し，また，どのように使い分けられるべきかなどの問題に関して全く明らかになっていないばかりか，議論さえ満足に行われていない．

また，この抽象化基準に関する問題はより一般的には，エージェントの状態および行為の同時（並行）抽象化という問題とも深く関わっている．前述の通り，従来研究のほとんどが状態の抽象化か，行為（アクション）の抽象化のどちらか一方のみを扱っている．これは，2つの抽象化基準がそれぞれ，“同じアクションによって同じ行為結果を得る”，“同じ（似た）状態において同じ行為結果を得る”というように，互いに依存し合っているということによる．すなわち，異なる状態同士の類似性を論じるためには行為空間内の距離尺度が定義されてなければならず，逆に異なるアクション同士の類似性を論じるためには先に状態（センサー入力）空間内の距離尺度が定義されてなければならないという，“鶏と卵の関係”[113]になっているためである．したがって，プリミティブなセンサ入力空間とモーター出力空間から一般的な状態空間と行為空間への抽象化を並列的に行うためには，このように互いに依存しあう抽象化基準を見直す必要がある．

前者の問題が，“（状態や行為の）抽象化はどのような原理・基準に従って行われるべきか”という極めて根本的な問題であったのに対して，後者の問題はこれとは対照的に，より実際的な問題である．前者の問題が関わるところのセンサー入力同士，モーター出力同士の近さ／遠さ，すなわち距離尺度がひとたび決まれば，その基準に従ってエージェントの過去の行動経験におけるセンサー入力集合とモーター出力集合が分類される．そしてそれぞれが一般化されて状態空間あるいはアクション空間が定義されるわけであるが，実際にどのような表現手法によってセンサー入力集合が状態として一般化され，モーター出力集合がアクションとして一般化されるか，ということについては，やはり従来研究によって様々である．特に状態の表現法については，決定木（decision tree）による表現，マハラノビス距離によって定義された楕円体領域による表現，最近傍法（nearest neighbourhood method）による表現など，様々な方法が提案されている．それにも関わらず，それらの異なる状態表現・一般化手法同士の比較というものはあまり議論されておらず，稀になされていてももっぱら，“どれだけ少ないコストで正確（精密）に状態（の集合）を表現できるか”ということに関してのみであり，実環境の様々な不確定性要因によってセンサー値に誤りが生じたり，故障が発生した場合にその状態表現系がロバストであるかどうかということは全く考慮されていない．また，実環境エージェントでは，ゲーム世界（toy world）などの理想的なエージェントと異なり，自分の“真の状態”に関する必要十分な情報を得ることはできず，ある事象に関しては不十分な，そして別のある事象に関して

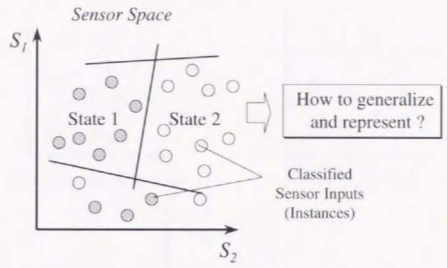


図 3.5: 状態の一般化と表現の問題

は異なる性格を持った複数のセンサー源から冗長な情報を得ることになるが，それらの膨大で不均一のセンサー情報をいかに効率的に統合することができるかということも，この状態一般化・表現法の問題に強く依存している．

次節以降ではまず，後者の問題すなわち“状態と行為の一般化と表現の問題”を先に詳しく述べ，その後，前者の問題“抽象化基準に関する問題”について詳しく述べる．

3.3 状態と行為の一般化と表現の問題

3.3.1 概念の獲得と表現

本節では前節で述べた状態と行為の経験的抽象化に関する２つの重要課題のうち，“状態と行為の一般化と表現の問題”，すなわち，ある類似性基準に基づいて分類されたセンサー入力の集合，モーター出力の集合をどのような表現形式によって一般化し，利用するか（図 3.5）という問題について，その意義や論点を述べる．

同じ状態に属するものとして分類されたセンサー入力の集合，あるいは同じアクションに属するものとして分類されたモーター出力の集合を一般化することは，見方を変えればセンサー入力のベクトル空間，あるいはモーター出力のベクトル空間を，どのように分割するかという問題である．前節で紹介した従来研究において用いられている決定木，マハラノビスの超楕円体，最近傍法などの手法によって状態・アクションの一般化を行った場合に，センサー入力空間あるいはモーター出力空間がどのように分割されるかを表したのが図 3.6 である．これらの異なる表現方法の善し悪しを決める基準としては，従来の帰納学習における議論と同様，“い

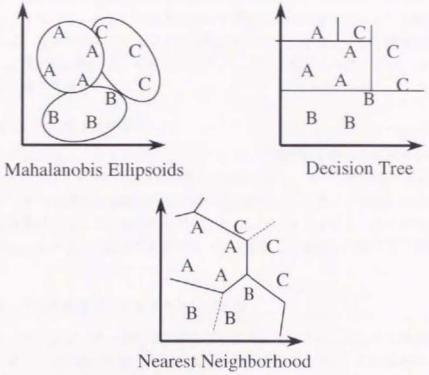


図 3.6: 従来手法によるセンサー空間・モーター空間の分割

かに単純な記述で，いかに正確にインスタンス（この場合はセンサー入力ベクトルやモーター出力ベクトル）をクラス（一般化された状態またはアクション）に分類するか" というものがまず考えられる．言い変えれば，"状態分類のパフォーマンスを維持しつつ，いかに状態クラスの数を減らすか" という基準である．

しかし，実環境エージェントの状態／行為の一般化という問題をより細かく考えると，一般的な帰納的機械学習の分野で扱われている一般化学習，あるいは概念学習とは明らかに異なる点がいくつか存在する．そしてその違いの多くは，一般化された状態およびアクションというものが，実環境における行動決定に用いられることを目的としていることに起因している．

3.3.2 実環境エージェントの状態表現系に求められる性格

センサー入力分類の正確性

まず，通常の概念学習ではインスタンスを正確にそれが属するクラスに分類することを目的にするが，エージェントの状態認識では観測されたセンサー入力を一般化された状態に正確に分類すること自体が目的なのではなく，その後に適切な行為が選択され，望ましい状態遷移がもたらされることが重要である．したがって，たとえ状態の表現が大雑把であるために，観測さ

れたセンサー入力が本来分類されるべき状態の近隣の状態として分類されたとしても，その状態で選択されたアクションが正しい状態において選択されるべきアクションと同じか，近いものであるならば，結果的にエージェントの行動は同じものになる．そして通常の環境とタスクでは，この“近隣する状態において取るべきアクションは似たものになる”という仮定は一般に妥当なものであると考えられる．

ノイズやフォールトに対する頑強性

ロボットなどの実環境エージェントがセンサーを介して環境から得る情報が有する重要な特徴として，ノイズやフォールトの存在ということが挙げられる．ここで重要なことは，ハードウェアとしてのセンサーの精度や信頼性が技術的に向上すれば，当然ノイズのレベルやフォールトの発生確率は減るが，ここでの本質的な問題はそういうことではなく，センサー入力レベルでのノイズやフォールトをいかに状態表現のレベルで吸収し，頑強な行動決定に貢献するかいうことである．

大量のセンサー入力を利用するときのコストや効率

エージェントが用いるセンサー情報は，そのエージェントに求められるタスクの複雑性に応じて増大する．また，上で述べた個々のセンサーのノイズやフォールトの影響を小さくするという観点からも実環境エージェントでは多くのセンサーが用いられる．しかし，それらのエージェントでは同時に実時間での行動決定が要求される．したがって，センサー入力が増えた場合の状態一般化・表現に要する計算コストや記憶コストがどのようなオーダーで増えるかということは重要な問題である．

異種冗長なセンサー入力の統合

また，高度な実環境エージェントが環境から得るセンサー情報は，我々人間が様々な感覚器官を持ち合わしているように，非常に多種多様なものである．例えば，簡単な移動ロボットエージェントでさえも，画像センサー（ビジョン），赤外線センサー，接触センサーなど，連続性（センサーが返す値が連続か離散か），分布の性格，範囲などの面で全く異なったセンサーを持つ．そしてこれらの異種（heterogeneous）のセンサーは，それぞれ環境についての異なった情報を返すだけでなく，ときに非常に似た情報も返す．例えば移動ロボットの前に岩石などの障害物が有った場合，画像センサーも赤外線センサーもソナーもそれぞれ異なった形式でその障害物の存在や大きさ，距離などに関する情報を提供する．このような異種冗長性は，先に述べたようなノイズやフォールトに対するロバスト性を実現するという観点からは非常に重要な役割を担っているが，そのような頑強性を実際に実現するためには，一部のセンサーに誤りが生じた場合でも，残りのセンサーから得られる類似の情報を利用して状態の分類を行えるように状態表現系が作られている必要がある．

3.3.3 従来研究における状態の表現法

このような観点に基づき，状態抽象化に関する従来研究で用いられてきた状態の一般化，表現法について分類を行い，それらの特徴と問題点について考察を行う．

[決定木を用いた状態一般化・表現法]

まず，この問題に対する代表的なアプローチとしては，決定木（decision tree）を用いた方法が挙げられる [62], [16]．決定木は帰納的機械学習の分野では最も標準的な手法になっており，ID3[68] や C4.5[69] など様々なインプリメンテーションが存在し，盛んに研究が行なわれている．

決定木による学習の基本的なメカニズムは，クラスの ID によってラベル付けされたインスタンス集合を，集合中のクラス分布に関するエントロピー（ばらつき具合）が最も減るような分割（分割すべき属性軸と分割値）を繰り返し求めていくことによって，任意のインスタンスをいずれかのクラスに分類する木を構成するというものである．決定木によって分割された属性空間（センサー入力空間）は，図 3.6 のように属性軸に対して垂直な平面によって切り分けられた形になっている．そのため，例えばあるクラスのインスタンスが 2 つの属性軸に対して斜めになるように分布している場合，そのクラスの領域を正確に表現しようとすると，分割が細かくなり過ぎるという問題が生じる．そのような場合の解決策としては，その 2 つの属性を適当に組み合わせることによって新たな属性を作り，その属性について分割を行なうということが考えられる．[2] では，属性同士の単純な足し算・引き算によって作られた新しい属性による分類が行なわれており，不完全ながらもこの考えを実現している例といえる．

決定木を用いてセンサー入力ベクトル空間を分割した場合の長所としては，各センサーが離散値を取るものであっても，連続的な値を取るものであっても，またそれらが混ざっていても，容易に適用することができるという点である．すなわち，エージェントのセンサー系が異種のセンサー群によって構成されている場合にも用いることができる．ただし，この場合上で述べたような新属性の生成，特に順序関係のない（nominal）離散値属性と連続値属性から新しい属性を作ることは困難になる．

一方，決定木を用いる場合の問題点は，センサー入力群の中に互いに近い情報を提供するものが含まれている場合に，その冗長な情報を効果的に統合し，ノイズやフォールトに対してロバストな状態分類を行なうことが難しいという点にある．つまり，例えば 2 つの属性が非常に似た情報を表すという状況では，基本的により多くの情報量ゲインが得られる方について枝が展開され，もう一方の属性に関する情報は利用されないことになる．そのため，生成された決定木の中で頻繁に用いられている属性（センサー）について，ノイズやフォールトによって誤りや欠損が生じた場合に，冗長なセンサー情報が生かされず，大きなパフォーマンス低下に陥る可能性がある．

[マハラノビス楕円体を用いた状態一般化・表現法]

次に典型的な状態の一般化・表現手法としては，センサー入力空間中において代表点とその点からのマハラノビス距離がある値以下となる楕円領域によって各状態クラスを定義するという

ものである [5][80].

今，ある状態クラス C_i に属するセンサー入力のインスタンス集合がセンサー入力空間中に分布しているとする．このとき，この手法ではまず，C_i に属するインスタンス集合の中心点（平均ベクトル）$\boldsymbol{\mu}_i$ と分散共分散行列（variance covariance matrix）$\boldsymbol{\Sigma}_i$ を求める．そして不等式

$$(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) \leq m+2 \tag{3.1}$$

によって表される超楕円体領域の内部をその状態クラスとして定義する．すなわち，あるセンサー入力 $\boldsymbol{x}$ は，上式を満たす状態クラス C_i として分類される．

このマハラノビスの超楕円体を用いた状態表現の特徴として，先の決定木による方法とは異なり，センサー入力ベクトルに含まれる全情報が状態認識に用いられるという点が挙げられる．つまり，上式による状態クラス C_i とあるセンサー入力ベクトル $\boldsymbol{x}$ とのマハラノビス距離の計算式には $\boldsymbol{x}$ の全要素が（然るべき重みを付加されて）含まれている．そのためセンサー系が冗長に構成されている場合には，一部のセンサー入力にノイズやフォールトによる誤りが含まれていても，他の類似するセンサー情報によって緩和され，状態分類のパフォーマンス低下が比較的緩やかであることが期待される．つまり，冗長なセンサー情報の統合による状態分類のノイズやフォールトに対するロバスト性が見込まれる．

しかし，この手法ではインスタンス集合についての平均ベクトル，分散共分散行列を計算することからもわかるように，全センサー入力が連続値を取ることが前提となっており，離散値，とくに順序関係のないような離散値を取るセンサー入力を扱うことができない．例えば物体までの距離を返すソナーからの入力は扱えるが，障害物に触っている（On）か触っていない（Off）かの値を返す接触センサーなどからの入力をこの枠組の中で扱うことはできない．また，この手法では各状態クラスに属するセンサー入力のインスタンスがその平均ベクトルを中心とする多次元正規分布に従うことを仮定しているが，この仮定は実環境エージェントのセンサー入力および状態空間の性格から考えてあまり現実的ではない．さらに，センサー入力ベクトルの中に互いに極めて近い入力が含まれている場合，分散共分散行列の逆行列を求めることが数値計算上不安定になる．また，全てのセンサー入力間の相関係数を求めることから，センサー入力の数の約2乗に比例して計算量も大きくなる．

これらの理由から，この手法も始めに述べた異種冗長なセンサー入力からの状態一般化には適していないと言える．

[線形判別関数を用いた状態一般化・表現法]

センサー入力空間を線形判別関数（linear discriminant function）で表される超平面によって分割することによって状態の一般化と表現を行う方法 [45] は，インスタンスの分布が同じ分散共分散行列に従う2つのクラスの境界面が線形判別関数になることからも，その性格はマハラノビス楕円体による方法とほぼ同じであると考えられる．したがって，連続／非連続のセンサーを柔軟に扱えるという点，インスタンスの分布に関して強い制約を課さない点などにおいて，提案手法の方が異種センサー情報源を用いて状態を一般化し表現する上で適していると考えられる．

ただし，線形判別関数はあくまでも2つの状態クラスの境界を指定するものであるから，複数の判別関数を適切に組み合わせることによって，マハラノビス楕円体による方法よりも，複雑な領域を持った状態クラスを正確に表現することができる．

[最小近傍法を用いた状態一般化・表現法]

各状態クラスのセンサー入力空間内での領域を複数の代表点によって表現し，任意のセンサー入力をその代表点までの重み付きユークリッド距離（weightedeuclid distance）が最も小さい状態クラスに分類する方法[77]は，各センサー入力を独立に扱っており，一部のセンサーのフォールトが全体の状態認識に及ぼす影響は小さい．また，各状態クラスの代表点を増やすことによって，どのような形の領域も自由に表現することができ，この点はここで述べた他の状態表現法よりも優れていると考えられる．

しかし，マハラノビス楕円体による方法や，線形判別関数による方法と同様に，離散値を取るセンサーが含まれている状態一般化問題には使用できない．また，これはこの手法に限らず，マハラノビス楕円体を用いた手法や線形判別関数を用いた方法についても言えることであるが，これらの手法はいずれも重み付きユークリッド距離，あるいはマハラノビス距離という距離尺度がセンサー入力空間中の状態の近さを適切に反映していることを大前提としている．すなわち，「センサー入力同士，あるいはセンサー入力と状態クラスの間の（マハラノビスあるいは重み付きユークリッド）距離が小さければ小さいほど，両者は（状態として）近い」ということを仮定している．よって，その仮定が成り立たないような悪構造のセンサー入力空間では利用できない．このことを端的に示す簡単な例としては，角度（angle）に関する情報を返すようなセンサー入力，例えばロボットアームの台座に対する回転角θなどを考えてみると良い．$\theta=-\pi$と$\theta=\pi$は値の距離（差）としては大きいが，状態としては同一のものである．

以上の考察から，いずれの状態表現・一般化手法も，多種多様で冗長なセンサー入力を統合し抽象化した上で，実環境の様々な不確定性要因に対してロバストな状態分類を実現する一般的枠組としては不十分であると言える．

3.4 抽象化基準に関する問題

本節では状態および行為の抽象化における抽象化基準の問題，すなわち，過去の行動経験から集められたセンサー入力の集合，あるいはモーター出力の集合を，どのような基準に基づいて互いの類似性を定義し，状態クラス，アクションクラスに分類するのかという問題，あるいは，過去の行動経験のデータ$\mathcal{D}$が与えられたときに，ある状態クラスの集合$\mathcal{C}=\{C_1,C_2,\cdots\}$，またはあるアクションクラスの集合$\mathcal{A}=\{A_1,A_2,\cdots\}$の善し悪しを評価するのか，という問題について考察する．

3.4.1 従来研究における状態・行為の抽象化基準

前述のように，あるエージェントの抽象化された状態空間・および行為空間の善し悪しを評価する最終的な基準は，1）それらを用いて行動政策学習を行ったり，プランニングに基づく行動決定を行ったときのパフォーマンス（例えば獲得報酬和の期待値や，ゴール状態に如何に早く到達するかなど）が良く，2）かつ，それらの状態集合・行為集合の記述に要するコストが少ない（状態クラス・行為クラスの数が少なく，個々の状態・行為が単純な形をしているということ）であると言える．しかし，エージェントの行動パフォーマンスは行動政策学習やプランニングの結果に基づいて実際にエージェントが行動して初めて評価できるので，特に1つめの基準を直接最適化するように状態空間あるいは行為空間を決定することは現実的には困難である．

そのため，実際の状態抽象化・行為抽象化に関する従来研究では，これらの基準に代わるものとして，「同じアクションによってどの程度同じ行為結果を得ることができるか」という基準に従ってセンサー入力を一般化し，同様に「同じ状態においてどの程度同じ行為結果を得ることができるか」という基準に従ってモーター出力の一般化を行っているが，ここで「同じ行為結果」というものを何によって定義するかという新たな問題が存在する．そこで以下では，従来の状態・行為の抽象化研究において，「何」に基づいてセンサー入力ベクトルやモーター出力ベクトル同士の近さ・遠さが定義され，状態あるいはアクションとして一般化されているか，ということについて考察する．

[同一ゴール／サブゴール状態への到達に基づく抽象化]

異なるセンサー入力同士の近い・遠いを定める基準としてはまず，同じアクションによって，同じゴール状態，もしくはそれに準ずる状態に達する場合には両者は「近い」とし，そうでない場合には「遠い」とみなすというものが考えられる．

この考えを採用したものとしては，[5][2][85]が挙げられる．これらの手法ではまず，ある1つのゴール状態 C_0 を定義し，そのゴール状態に到達することをエージェントのタスクとして定める（図3.7(a)）．そしてエージェントにランダムなアクション選択および実行を行わせ，このゴール状態に到達したときの，**実行したアクション** $a \in \mathcal{A}$ と，アクションを**実行する前のセンサー入力** $\boldsymbol{s}$ を記録する．そのようにして集められた実行する前のセンサー入力 $\boldsymbol{s}$ の集合を，実行したアクション a の違いによって分類し，それぞれの集合を一般化して新しいサブゴールとする．例えば，$a = MF$（前進）によって C_0 に到達したセンサー入力集合を一般化してサブゴール状態 C_1 を，$a = MB$（後退）によって到達したセンサー入力を一般化してサブゴール C_2 を定義したとする（図3.7(b)）．そして同様に今度はサブゴール状態 C_1，C_2 に到達するような行動経験を集めてアクションについて分類し，そのセンサー入力を一般化して新たなサブゴール状態 C_3，C_4，$\cdots$ を定義する．このようにして次々とサブゴール状態を生成して行き，センサー入力空間全体がサブゴール群によって覆われた時点で終了する（図3.7(c)）．

この状態一般化ストラテジー，およびこれに従って一般化された状態クラス群は，次のような性格を有している．まず，上に示した過程からもわかるように，センサー入力空間における

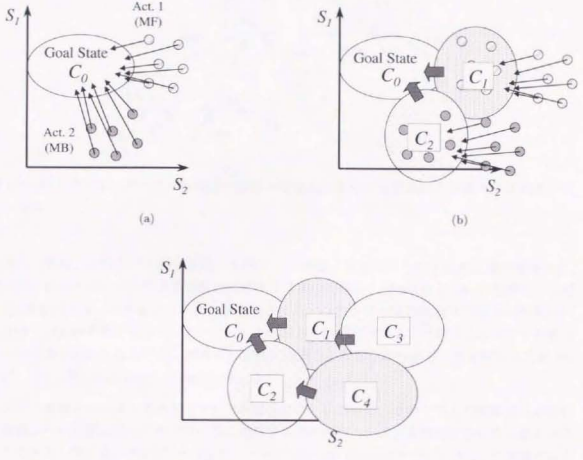


図 3.7: 同一ゴール／サブゴール状態への到達に基づく状態抽象化における状態クラス生成の概要

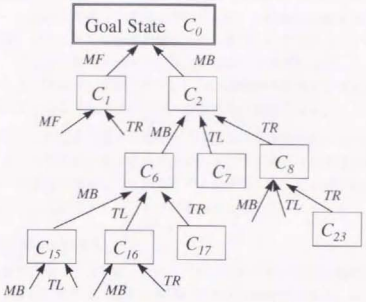


図 3.8: 同一ゴール・サブゴール到達に基づく抽象化によって生成された状態クラスの単一ツリー構造

状態の一般化，状態クラスの生成は，最終ゴール状態に近いところから徐々に遠い領域へと，遠心的に行われる．そして生成された状態クラス間の関係は，図3.8のように，最終ゴール状態 C_0 を根（root）とするツリー状に書き表される．このツリーは各状態クラスから最終ゴール状態への経路も表しており，エージェントは現在の状態（クラス）から，このツリーを根に向かって遡っていくことによってゴール状態に到達できることがわかる．各状態クラスが“サブゴール”と呼ばれるのは，そのためである．

ここで重要なことは，各サブゴール（状態クラス）において，その1つ上の階層の（すなわち最終ゴール状態に近い）サブゴールに遷移するためのアクションも既に分かっているということである．例えば，サブゴール C_k が，アクション MF によってサブゴール C_j に到達するようなセンサー入力を一般化することによって定義されたものだとすると，エージェントは状態 C_k においてはアクション MF を取れば，最終ゴール状態に近付くことができることがわかる．このことは，この**同一ゴール／サブゴール状態への到達に基づく抽象化**ストラテジーに従って状態の抽象化を行った場合，同時に行動政策学習も行われているということを意味している．すなわち，一旦状態の抽象化が終了してしまえば，エージェントは改めて行動政策（状態－アクションのマッピング）を学ぶ必要がない．その意味でこのポリシーに基づいて抽象化された状態空間は設定されたタスクに対して全く“無駄がない”ということが言える．

しかし，この状態抽象化ポリシーは同時に以下に述べるように，おもに2つの問題点を持っている．まず，この状態抽象化ストラテジーでは，エージェントが最終ゴール状態か既に生成されているサブゴール状態のどちらかに到達するような行動経験を得ない限り，新しい状態ク

ラスの生成も行動政策学習も全く行われないということである．特に最初の学習段階では，“たまたま”最終ゴール状態に到達しなければ次のサブゴール状態が定義されないので，ほとんどの行動経験が無駄になり，非常に効率が悪い．この問題へのアプローチとして，始めはエージェントに易しい，すなわち最終ゴール状態に近い領域から学習を始め，徐々に難しい（すなわちゴール状態から遠い領域で）問題へ移行していく**段階的学習** [85] が考えられるが，全てのタスクや状況においてそのようなストラテジーが取れるとは限らない．

もう一つの問題は，この抽象化基準に基づいて構成された状態空間はそのタスクとゴール状態に完全に特化したものであるため，タスクやゴール状態が変わった場合の再利用が難しいという点である．これは最初に想定したタスクとゴール状態に対しては理想的な無駄のない状態空間が得られるということとのトレードオフであるとも言える．

[類似する報酬獲得に基づく抽象化]

ゴール状態への到達というイベントは，タスクに設定された唯一の正の報酬を獲得することとして捉えることができる．そこで，より一般的に正負の直接報酬の獲得，および状態遷移に伴う間接的な報酬の獲得に基づく状態・行為の一般化というものを考えることができる．すなわち，同じアクションによって獲得される直接報酬または将来に見込まれる割引報酬が同じであるようなセンサー入力を同じ状態クラスとして一般化し，同様に同じ状態において同じ（直接・間接割引）報酬を獲得するようなモーター出力を同じアクションとして一般化するというものである．

この考えに基づいた状態抽象化法の例としては，[80][45][16] が挙げられる．ただし，[45] では直接獲得される報酬にのみ基づいた状態一般化が行われているのに対して，[80] と [16] では，状態遷移によって間接的に獲得される割引報酬に基づいた状態の一般化も考慮されており，より一般的な枠組になっていると言える．

類似報酬獲得に基づく状態抽象化ストラテジーの過程は，ゴール／サブゴール到達に基づく抽象化ストラテジーのそれに似ている．まず，正負の報酬 R_{+1}，R_{+2}，$\cdots$，R_{-1}，R_{-2}，$\cdots$，が定義され，将来に渡ってより多くの（正の）報酬を獲得することがエージェントのタスクとして定義される（図 3.9(a)）．エージェントはランダムに行動を行いながら，それらの報酬を獲得した行動経験をアクションの違いによって分類し，それぞれのセンサー入力ベクトルを一般化して新しい状態クラスを定義する（図 3.9(b)）．そしてそれ以降，状態クラスに到達した行動経験から同様に新しい状態が一般化され，センサー入力空間全体が状態クラスによって覆い尽くされるまで続けられる（図 3.9(c)）．

この状態抽象化ストラテジーの第一の特徴は，始めに定義された正負の直接報酬から遠心的に状態の一般化が行われていくことである．それらの状態クラスの関係は，ゴール／サブゴール到達に基づく状態抽象化の場合と同様，各直接報酬（を獲得する状態）を根とする複数のツリーによって表される（図 3.10）．そして，各状態クラスでは，それが属するツリーの根にあたる報酬を獲得するためのアクションが同時に学ばれる．つまり，あるアクション a によって報酬 R を獲得する状態として一般化された状態クラス S では，アクション a によってその報酬を

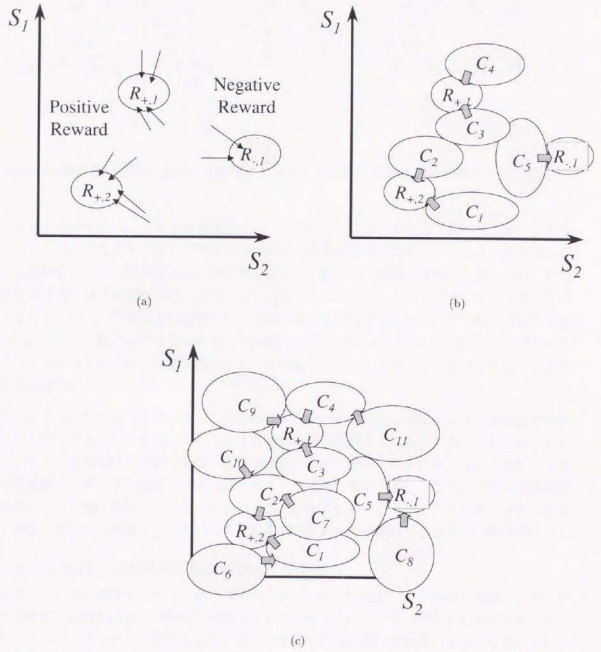


図 3.9: 類似獲得報酬に基づく状態抽象化における状態クラス生成の概要

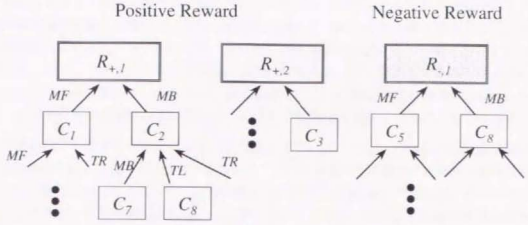


図 3.10: 類似報酬獲得に基づく抽象化によって生成された状態クラスの複数ツリー構造

獲得することができる．ここで重要なこと － ゴール／サブゴール到達に基づく抽象化と異なる点は，この抽象化ストラテジーではローカルな報酬を獲得するためのアクションは学ばれるが，大局的なゴールに到達するためのアクションは状態抽象化過程では得られないため，別途行動政策学習を行う必要があるということである．例えば，$a = MB$（後退）というアクションによって，$R_{-,j}$（障害物との衝突）という負の報酬を獲得する状態として一般化された状態 C_k では，“この状態ではアクション MB を選択してはいけない” ということはわかるが，“ではどのアクションを実行すれば大局的なゴールに近付けるか” ということに関しては，別途学習する必要がある．

このように，類似報酬獲得に基づく抽象化はゴール／サブゴール到達に基づく抽象化を一般化したものであると言えるが，このことは同時に同様の問題点を抱えていることを意味している．つまり，直接報酬か間接を獲得した行動経験のみが状態の一般化に用いられるため，学習の初期段階では多くの行動経験が無駄になるという点で，効率が悪い．また，正負の直接報酬は最終ゴール状態と同様，エージェントのタスクに強く依存するため，ある報酬セットに基づいて抽象化された状態集合，アクション集合を他のタスクで再利用することは一般に難しい．

[センサー入力変化・状態遷移の類似性に基づく抽象化]

上で述べた2つの抽象化ポリシーがいずれも，“ゴールへの到達” や，“報酬の獲得” というタスクに特化した行為結果の類似性を基準として状態やアクションを一般化するものであったのに対し，タスクやゴールに依存しない，より一般的な行為結果の類似性に基づいてそれらを一般化するという方法が考えられる．すなわち，センサー入力の変化や状態遷移が互いに類似する行動経験におけるセンサー入力，モーター出力を一般化して状態クラスや行為クラスを定義するというものである．

この考えに基づいた状態抽象化法の例として [77][59] が挙げられる．[77] では，同じアクションによってもたらされるセンサー入力ベクトルの微小変化を1つの線形回帰モデルによってう

まく表現できるようなセンサー入力集合を同じ状態としてみなし，モデルと実際のデータとの誤差がある閾値を超える場合に，新たな状態を定義するということを繰り返すことによって状態空間構成している．一方，[59]では，あらかじめ有限個の状態クラスに粗く分割された状態空間を用いて，エージェントの状態遷移が不確かであるような状態クラスを徐々に細分化していくという方法を提案している．また，先に挙げた2つの行為抽象化の研究例[61][44]も，センサー入力変化・状態遷移の類似性に基づく行為の抽象化を実現したものであると言える．

この抽象化ストラテジーにおける重要な特徴はまず，エージェントの全ての行動経験が状態抽象化に利用されると言う点である．すなわち，ゴール状態へも到達せず，報酬も獲得しなかったような行動経験からも，センサー入力の変化，あるいは状態の変化の類似性に基づいて，状態あるいはアクションが一般化される．したがってゴール状態への到達や報酬の獲得を経験しない限り抽象化が行われない他の2つのストラテジーに比べてデータの有効使用率という観点からは非常に効率が良い．状態生成の過程も，センサー入力空間の全領域において同時多発的に行われることになる．また，このようにして抽象化された状態集合や行為集合は，報酬やゴール状態の定義に依存しないので，行動政策学習を別途行わなければならないが，このことは同時に，異なるゴール状態や報酬が定義されている他のタスクへの再利用が容易であることを意味している．

しかし，ゴールや報酬に依存しない状態空間・行為空間が得られるというこの抽象化ストラテジーの長所は，ある面では同時に短所でもある．まず，このストラテジーによって得られた状態集合と，実際のタスクにおけるゴール状態や報酬の分布が完全に一致しているとは限らないので，その状態空間を用いて行動政策学習を行っても最適な結果得られないという可能性があるということである．例えば図3.11のように，この抽象化法によって分割された状態集合が実際の報酬の分布を反映していない場合には，この状態空間上でQ-Learningなどの強化学習を行ってもあまり良い結果が得られないであろうということは容易に想像がつく．また，一般化された状態や行為がゴールや報酬と直接関係がないということは，エージェントにとっては全く興味がないような状態やアクションも多く生成されてしまう可能性があるということを意味している．

これらの考察より，センサー入力変化・状態遷移に基づく状態・行為抽象化ポリシーは，状態や行為の抽象化自体の効率は非常に良いものの，それを用いた行動政策学習の結果が必ずしも望ましいものにならないという問題をもつことがわかる．

3.4.2 複数の抽象化基準を用いた状態と行為の抽象化

このように，同じ“行為結果の類似性に基づく状態・行為抽象化”であっても，何を行為結果として考えるかによって，抽象化の基準が変わり，抽象化の結果が大きく異なってくることが示された．表3.1は上で述べた3つの抽象化ポリシー（ゴール／サブゴール到達に基づく抽象化，報酬獲得に基づく抽象化，状態変化に基づく抽象化）の違いを端的にまとめたものである．

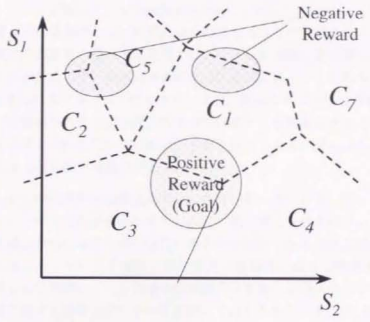


図 3.11: 状態クラス定義と報酬の分布との不一致

表 3.1: 抽象化ポリシーの比較

	ゴール到達	報酬獲得	センサー変化
全体的性格	目的指向	報酬指向	データドリブン
クラス生成のされ方	ゴールから徐々に	報酬から徐々に	全領域同時並行
行動政策	得られる	ある程度得られる	得られない
タスク依存性	大	大	小
データ使用率	低い	中程度	高い

これからわかるように，ゴール／サブゴール到達に基づく抽象化，報酬獲得に基づく抽象化では，ゴールへの到達や報酬を獲得しない限り状態や行為の一般化が行われないので，抽象化における学習効率は良くなく，異なるタスクへの再利用も難しい．しかし，結果として得られる状態空間，アクション空間はゴールや報酬を直接的に反映して定義されるので，ゴール到達，報酬獲得という観点からは無駄のない理想的なものになることが保証される．一方，センサー入力変化／状態変化の類似性に基づく抽象化では，ゴールや報酬に直接関わらないような行動経験からも学習が行われるので，抽象化自体の効率は非常に良い．すなわち，他の2者に比べて状態・行為の抽象化に要するコストは小さい．また，異なるタスクへの再利用も容易である．しかし，抽象化の結果得られた状態空間，行為空間は必ずしも特定のタスクにとって理想的なものであるとは限らず，それを用いて行動政策学習を行ってもエージェントの行動パフォーマンスが期待通りに良くならない可能性がある．

つまり，これら3つの代表的な抽象化基準はどれも一長一短であり，唯一正しいあるいは最適な抽象化ポリシーというものは存在しない．ここで次に考えられるのが，これらの異なる抽象化基準の長所を組み合わせることができないかということである．例えば，学習の初期段階においては効率の良いセンサー入力変化／状態変化の類似性に基づく抽象化ポリシーを用い，行動政策の学習によって次第にゴール到達や報酬獲得が頻繁に経験されるようにつれて，ゴール到達，報酬獲得に基づく抽象化ポリシーに切替えていくことができれば，(抽象化に要する行動経験が少ないという意味で) 効率が良く，かつ，エージェントの最終的なパフォーマンスの良さを保証するような状態抽象化・行為抽象化が実現できると考えられる．

3.4.3 抽象化尺度の具体的表現

一般に，ある属性ベクトルによって記述されたインスタンスの集合をクラスタリングし一般化するには，大きく分けて次の2つの方針が考えられる．

- インスタンス間の距離尺度を何らかの方法で定義し，その距離尺度に基づいて近いインスタンス同士をクラスタリングし，一般化する．
- インスタンス集合のある分割の仕方 P の評価関数を定め，これを最適化するようにインスタンス集合を分割し，一般化する．

前者はインスタンス間の**ローカル**な距離尺度に基づいて分割・一般化するのに対して，後者は分割の仕方に対する**グローバル**な評価基準に基づいてインスタンスを分割・一般化するというものである．

この分類は本研究で扱う "行為結果の類似性に基づく状態，行為の抽象化問題" についても当てはまり，それぞれ，

- "同じアクションによって同じ（類似の）行為結果をもたらすようなセンサー入力を同じ状態クラスとして一般化する" または，"同じ状態において同じ（類似の）行為結果をもた

らすようなモーター出力を同じアクション（行為）クラスとして一般化する" というローカルな方針に基づいて状態クラスあるいはアクションクラスを一つ一つ定義していく．

- ある状態クラスの集合 $\mathcal{C} = \{C_1, C_2, \cdots\}$ と，あるアクションクラス集合 $\mathcal{A} = \{A_1, A_2, \cdots\}$ が与えられたときに，それらによって行動経験のデータ集合の分類を行った結果行為結果に関する曖昧さがどのように減少するか，言い替えれば，それらの状態集合とアクション集合の定義を用いればエージェントの行為結果がどの程度正確に予測できるか，というグローバルな評価関数を定義し，この評価関数を最適化するような状態集合 $\mathcal{C}$ または行為集合 $\mathcal{A}$ を求める．

のように記述することができる．本論文では前者の方針を "ローカルな距離尺度に基づく状態・行為の抽象化"，後者を "グローバルな評価基準に基づく状態・行為の抽象化" と呼ぶことにする．

この観点から 3.2 節で述べた状態抽象化および行為抽象化に関する関連研究を比較してみると，その多くは "ローカルな距離尺度に基づく状態・行為の抽象化" に従っていることがわかる．すなわち状態抽象化の場合であれば，同じアクションによって同じ報酬を獲得したり同じ状態に遷移することができるかどうかということによってセンサー入力間のローカルな距離尺度を定義し，その結果「近い」と判断されたセンサー入力を集めて 3.3 節で分類したような然るべき方法で一般化している．

これに対して [72] と [44] は "グローバルな評価基準に基づく状態・行為の抽象化" の数少ない例である．[72] は状態の抽象化，[44] は行為の抽象化をそれぞれ扱っているが，どちらも**概念クラスタリング** [30] の方法に基づいており，CU（categorization utility）と呼ばれるクラスタリング結果のグローバルな評価関数をインクリメンタルに最大化することによって状態または行為の一般化を行っている．ここで，あるインスタンス集合を K 個のクラスタに分けるような分割 $\mathcal{C} = \{C_1, C_2, \cdots, C_K\}$ の CU は次のように計算される．

$$CU(\mathcal{C}) = \frac{\sum_{k=1}^{K} P(C_k)[\sum_j \sum_l P(X_j = V_{jl}|C_k)^2 - \sum_j \sum_l P(X_j = V_{jl})^2]}{K} \tag{3.2}$$

これは，「分割 $\mathcal{C}$ を与えたときの，正しく予測できる属性の数の期待値が，この分割を考えない場合に比べてどれだけ増加するか」を表したものとして解釈できるが，理論的には特に根拠のないヒューリスティックな指標である．

"グローバルな評価基準に基づく状態・行為の抽象化" は，従来多く採用されてきた "ローカルな距離尺度に基づく状態・行為の抽象化" に比べると，一つの評価関数を最大化（あるいは最小化）するような状態クラス集合ないしはアクションクラス集合を見つける過程として実現されるので，その結果得られる状態空間や行為空間は全体として一貫性の取れたものになると考えられる．しかし，抽象化結果の善し悪しを適切に反映する評価基準を定義することは必ずしも自明でない上，最適化の対象となる解空間が一般に膨大であるため，真の最適解を求めることは極めて困難である．実際，CU を抽象化の評価関数として利用する [72] [44] においても，その最適解を直接的に探索するのではなく，Fisher の COBWEB[30] で提案されたヒューリスティックな探索法を用いて状態空間，あるいは行為空間の構成を行っている．この方法ではイ

表 3.2: 従来の状態／行為の抽象化に関する研究の分類

	状態（State）の抽象化	行為（Action）の抽象化
ゴール・サブゴール 状態到達	Asada[5] Albus[2]	-
類似報酬獲得	Chapman[16] Ishiguro[45] Ueno[80]	-
センサー変化・ 状態遷移の類似性	Moore[59] Takahashi[77]	Iba[44] Nakamura[61]

ンスタンスを一つずつ加えるごとに，既成クラスへの帰属，そのインスタンスからのみ成るクラスの生成，複数クラスの統合，クラスの分割，という4つのオペレータによってCUがどのように変化するかを計算し，最も効果がある（すなわちCUを大きくする）ものを実際に適用していくことによって，逐次的にクラスタリングを行っている．

3.4.4 状態・行為の同時抽象化

3.2節で見たように，“行為結果の類似性に基づく状態・行為の抽象化”に関する従来研究では，状態の抽象化か行為の抽象化のどちらか一方のみを扱ったものばかりであり，両者を同時に抽象化する方法を明らかにした例はない．[3]この背景には，両者の抽象化の過程が互いに依存しているという問題が挙げられる．すなわち，“同じ（似た）アクションにより同じ行為結果を得るようなセンサー入力の集合”という定義によって状態クラスを定めるにはその前に“同じアクション”あるいは“似たアクション”が何なのか定義されていなければならず，同様に，“同じ（似た）状態において同じ行為結果をもたらすようなモーター出力の集合”という定義によってアクションクラスを定めるには事前に“同じ状態”あるいは“似た状態”が何であるかが定義されていなければならない．この関係は[113]が指摘しているように，「にわとりと卵の関係」になっていると言える．

表3.2は，3.2節で概観した従来の関連研究を，（1）状態と行為のうちどちらの抽象化を扱ったものかという軸と，（2）前述した3種類の抽象化基準のうちのどれに基づいているかという軸とで分類して表したものである．

行為結果の類似性に基づいて状態空間と行為空間の両方を自律的に抽象化するには，大きく分けて次の2つのアプローチが考えられる．

[3][5]は状態とアクションの一般化を同時に行ったと主張しているが，実際には，あらかじめ決まったモーターコマンド（要素アクション）を異なる回数繰り返し適用したものを1つのアクションとして扱うことを“アクションの一般化”と称しており，プリミティブなモーター空間全体を抽象化しているわけではない．

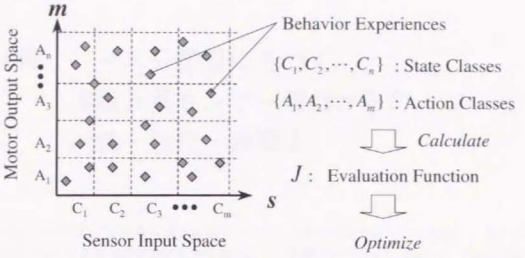


図 3.12: グローバルな評価基準に基づく状態・行為空間構成

まず一つ目は，状態空間とアクション空間のどちらか一方の抽象化を（もう片方を固定した上で）交互に繰り返すことによって漸次的に両者の抽象化を行うことである．すなわち，まず始めに暫定的なアクション空間を仮定しておいて状態空間の抽象化を行う．そして次にその状態空間を固定してモーター出力空間からアクション空間への抽象化を行う．この2つの抽象化過程を収束するまで繰り返すというものである．この方法は従来提案されて来た“（アクション空間を固定した）状態抽象化法”と，“（状態空間を固定した）アクション抽象化法”をそのまま用いることが出来る点が長所である．しかし，2つの抽象化過程の繰り返しが発散せずに収束するか，また収束するとしてもそれが実時間内に終了するかということに関してはいずれも明らかでない．

もう一つのアプローチは，前述した“ある状態空間とアクション空間のグローバルな評価基準”を利用して，その評価基準を最適化するような状態空間とアクション空間を同時に求めるというものである．つまり，図3.12のようにセンサー入力空間 S に関する状態分割 $\mathcal{C} = \{C_1, C_2, \cdots\}$ とモーター出力空間 $\mathcal{M}$ に関するアクション分割 $\mathcal{A} = \{A_1, A_2, \cdots\}$ によって過去の行為経験の集合を分類したときに行為結果に関する曖昧さがどれだけ減るか，という評価基準を設け，これを最大化するように $\mathcal{C}$ と $\mathcal{A}$ を決めるということである．しかし，既に述べたようにこの評価基準に基づく最適化問題は状態空間，アクション空間のいずれか一方だけを対象にしても大変な難問であるので，両者を対象とした最適化ではより適切なヒューリスティックスを用いる必要があると考えられる．5章では，このグローバルな評価基準に基づく状態および行為の抽象化法を提案する．

第4章　ベイズ分類器に基づく異種冗長センサー情報からの状態一般化・表現法

本章では，多種多様で冗長な情報を含むセンサー入力をどのようにして一般化し，様々な不確定性要素に対して頑強な状態表現を得るかという，「異種冗長センサー情報からの状態抽象化問題」を扱う．この問題は3章で述べた自律エージェントの行動抽象化問題における2つのテーマのうち，主に**抽象化された状態と行為の表現と獲得法**に関わるものであり，本研究ではそのアプローチとして，単純ベイズ分類器（Naive Bayesian Classifier）を用いた状態一般化・表現手法を提案する．

4.1　研究の目的

3.3節で述べたように，状態抽象化における重要な問題の1つに，どのような表現手法によってローレベルで連続的なセンサー信号空間から，より一般的な離散化された状態空間に変換するかということがある．この状態一般化・表現法の問題は，例えば自律移動ロボットのような実環境エージェントにとって，特に次の2つの理由から重要である．

まず，これらのエージェントが活動する実環境は非常に不確定性が強い世界であり，センサーやアクチュエータの誤差，故障などの不確定性要素を免れ得ないということである．そのための1つの現実的なアプローチとして，センサー系を冗長にすることによって観測の信頼性や耐故障性を高めるということが考えられるが，実際にこれを実現するにはエージェントの内部状態表現系がそのような冗長なセンサー入力情報を効率的に統合し利用できるようになっている必要がある．つまり，冗長なセンサー情報が与えられたときに，そのうちの（特に有用な）情報だけを選択するというのではなく，利用可能な情報全てを用いて状態認識を行えるような状態表現法が望ましい．

もう1つの理由は，これらの実環境エージェントは通常，ハードウェアとしてのセンサーを複数種類持っており，それらからのセンサー入力が連続性・非連続性，分布の形，取りうる値の範囲などの性質において，多様になっているということである．例えば自律移動ロボットの場合，接触センサーはものに触れているかいないかを1 (On)，0 (Off) の離散値で返し，超音波センサーは近くのものへの距離を返し，画像センサー（カメラ）は物の画面上での位置や大き

さ，色などを返す．したがってエージェントの内部表現系は，このような多種多様のセンサー情報を扱えなければならず，全センサー入力が連続性や分布形などの面において同様な性格を持つことを前提としているような手法は不適切である．

これら2つの理由から，本章では異種（heterogeneous）冗長（redundant）なセンサー入力をいかに効率的に一般化し，実環境の様々な不確定性要因に対してロバストな内部状態表現系を獲得するかという問題に焦点を当てる．

4.2 想定する環境とエージェントの性格

ここではこの章を通じて想定するエージェントと環境それぞれの性格，および両者の関わり合い方に関する性格を明らかにする．

4.2.1 エージェントと環境とのインタラクションに関する仮定

本章ではエージェントの行動過程，あるいは環境との相互作用の過程を，「センシング − 行動（意志）決定 − アクチュエーション − 状態遷移」の1サイクルを単位時間とした（時間的）離散的過程として扱う（行動過程の時間的な離散性）．また，エージェントの状態遷移は決定論的なものではなく不確実性を伴ったものとし，その状態遷移確率は現在の状態のみに依存し，過去の履歴には依存しないものとする．（**状態遷移のマルコフ性**）

4.2.2 エージェントの真の状態とセンシングに関する仮定

ここで言うエージェントの「真の状態 (real state)」とは，エージェントの「内部状態 (internal state)」，あるいは「信念 (belief)」とは異なり，エージェントのタスクや目的，さらには内部表現系などには依存しないエージェントの絶対的な状態を表すものであるとする．また，一般に真の状態空間は連続的で無限数の個々の状態から成り立っている．

エージェントはこの「真の状態」自体を直接観測することはできず，実際に得られるのはその状態においてある確率分布で生起されるセンサー入力 $\boldsymbol{s}$ である．したがってエージェントは真の状態の一部分に関する不確定な情報を得ているに過ぎず，あるセンサー入力状態から真の状態を唯一に特定することはできない（**センサー入力の不完全性，状態の部分観測性**）．

また，各センサー入力は，必ずしも分布範囲，分布型，連続・非連続性などの性格の面で同種とは限らない．むしろ，異種なものが多数含まれているという状況を想定する（**センサー入力の異種性**）．さらに，異なるセンサー入力同士が真の状態に関して同種の冗長な情報をエージェントに提供することがあるものとする（**センサー入力の冗長性**）．

本章におけるテーマは次節で述べるように，このような異種冗長なセンサー入力によって構

成されるセンサー空間をどのように抽象化し，離散化された領域（状態クラス）を表現するか，ということである．

4.2.3 エージェントのアクチュエーションに関する仮定

一般にはエージェントの行為空間もまた，状態空間やセンサー入力空間と同様に連続的で，無限個の要素から成り立っているが，本章ではエージェントが選択可能な行為の基本単位は，あらかじめ定義された有限個のモーターコマンドを単位時間適用したアクション要素（action element）とする．そして，[5] などのアプローチに習い，継続的に適用される同じアクション要素の系列がエージェントに他の状態クラスへの遷移をもたらすか，（正または負の）報酬を獲得するまでを1つアクションとして定義する．

つまり，本章ではアクションに関しては，あらかじめ抽象化がなされているという立場を取る．

4.3 状態抽象化過程の概要

本章では，アクション空間に関してはあらかじめ離散化されているとしているので，状態抽象化は，「同じアクション $a \in \mathcal{A}$ によって，同じ行動結果 $\boldsymbol{r} \in \mathcal{R}$ を獲得するという基準に従ってセンサー入力 $\boldsymbol{s} \in \mathcal{S}$ を分類し，その集合をそれぞれ一般化することによって，状態クラス集合 $C = \{C_1, C_2, \cdots\}$ を定義する過程」として表現することができる．

ここで行動結果 $\boldsymbol{r}$ として何を用いるかということに関しては，3.4 節で述べたように主に，1）ゴール・サブゴールへの到達，2）報酬の獲得，3）センサー入力の変化の3つが考えられ，そのうちのどれを採用するかによって上記の抽象化の過程と結果は異なってくる（抽象化ポリシーの違い）．この状態抽象化ポリシーの問題と，状態の一般化・表現手法の問題は互いに独立した問題であり，この章で提案するベイズ分類器を用いた状態一般化・表現法は，基本的にはいずれの抽象化ポリシーを用いた場合にも適用可能である．

しかし，状態クラスがセンサー空間においてどのような順番で形成されていくのか，また抽象化の過程で行動政策に関する知識も得られるかどうかなどという点は抽象化ポリシーに依存するため，状態抽象化と行動政策学習を包含する全体の学習過程，特に両者を同時並列的に行うためのアルゴリズムは大きく異なってくる．そのため本章ではひとまず，扱う対象をゴール・サブゴールへの到達，および報酬の獲得に基づく状態抽象化過程に限定して提案手法の説明を行う．したがって，状態の抽象化はゴール状態や報酬に近い領域から遠心状に行われ，その生成された状態クラスについて有用度あるいはアクションの価値関数が順次学ばれていくことになる．また，行為結果ベクトル $\boldsymbol{r}$ は，ゴール・サブゴールへの到達と，その他の有限個の正負の報酬から構成される．すなわち $\boldsymbol{r} = (r_g, r_o)$（$r_g$ は到達したゴール状態あるいは既存状態クラスの識別子，r_o はその行為によって環境から得た報酬の値）となる．

以上のような条件を仮定したとき，ゴール・サブゴール到達，および報酬獲得に基づく状態

抽象化と，行動政策の獲得を含むエージェントの全体の学習過程は以下のようになる．

[ゴール・サブゴール到達，および報酬獲得の類似性に基づく状態抽象化]

1. （**ゴール状態／報酬の設定**）そのタスクにおけるエージェントのゴール状態，およびいつどのような（正負の）報酬を獲得するかを人間（設計者）が決める．この段階では最終ゴール状態以外の状態クラスは存在しない．
2. （**訓練タスクの提示**）ランダムに生成した訓練タスクをエージェントに与える．
3. （**行動決定と経験の蓄積**）エージェントは与えられたタスクに対して，ゴール状態に到達するか，あるいは一定時間以上経過するまで，以下の規則に従って行動を実行し，その結果を記録する．
   (a) 現在のセンサー入力 $\boldsymbol{s}$ が既存の状態クラスのいずれかに合致すると判断したときには，その状態クラスの有用度に基づきアクションを選択し実行する．その結果（他状態への到達，報酬の獲得）に関する情報は後に状態クラスの有用度を更新するのに用いられる．
   (b) センサー入力が現在定義されている状態クラスのいずれにも合致しない場合，エージェントはランダムにアクション a を選択して実行する．そしてその結果，既存の状態クラスに達するか，報酬を獲得した場合にはその行動経験（アクションを実行する前のセンサー入力，実行されたアクション，行為結果）$beh =< \boldsymbol{s}, \mathrm{a}, \boldsymbol{r} >$ を，行動経験データベース $\mathcal{D}_{beh}$ に追加する．ただし，同じアクションをあらかじめ設定された時間以上実行しても，ゴール状態／既存状態クラスに到達せず，何の報酬も得られなかった場合はデッドロックに陥っているものとみなしてそのアクションは打ち切り，あらたな別のアクション要素を選択する．
4. （**既存状態クラスの有用度の更新**）各既存状態クラス c_j の有用度関数あるいは行為－価値関数を 3-(a) で記録された経験に基づき，Q-Learning や Profit Sharing 法などの強化学習法を用いて更新する．[1]
5. （**新状態クラスの生成**）3-(b) のステップで集められた行動経験の集合 $\mathcal{D}_{beh}$ に含まれるインスタンスをアクション $\boldsymbol{a}$ および報酬あるいは遷移先の状態クラスによって表される行動結果 $\boldsymbol{r} = (r_g, r_o)$ における r_g または r_o の値が等しいもの同士を同じ行為経験集合として分類する（例えば，「前進によってゴール状態に到達した行動経験の集合」，「左回転によって負の報酬 -5 を獲得した行動経験の集合」という具合に）．ただし，既存の状態クラスに達し，かつ報酬を獲得した経験インスタンスはどちらの集合にも属するものとして扱う．そして含まれる行為インスタンス数が一定数以上の集合について，**アクションが実行される前のセンサー入力ベクトル $\boldsymbol{s}$ をベイズ分類器によって一般化し，新しい状態クラス C_{new} として定義する．**
6. （**終了判定**）前ステップによって定義された状態クラス集合によって，エージェントが取り得る全状態がカバーされ，かつステップ (4) の行動政策に関する学習が収束したなら

ば終了し，そうでなければ2に戻る．

図4.1はこの学習アルゴリズムをフロー図として表したものである．

この全体アルゴリズムの大きな特徴は，行動政策学習（ステップ4）と状態抽象化（ステップ5）とが部分的に並列化されているという点である．前述の通り，「ゴール状態・状態クラスへの到達に基づく状態抽象化ポリシー」か，「獲得報酬の類似性に基づく状態抽象化ポリシー」を採用した場合，ある状態の一般化はその状態において取るべき(あるいは取るべきでない)アクションの学習とともに行われる．特に「ゴール状態・状態クラスへの到達に基づく状態抽象化ポリシー」の場合は状態クラス（＝サブゴール）の生成とともに最終ゴールへの最短経路も学習されるので，ステップ4自体が不要になる．それに加えてセンサー入力空間中における状態の抽象化（すなわち新しい状態クラスの生成）はゴール状態や報酬源に近い領域から遠い領域に向かって順々に行われる（図3.7，図3.9）ため[1]，ゴール状態以外にも正負の報酬が多数存在するようなタスク環境においても，まだ抽象化された状態クラスが存在しない領域では状態抽象化を行いつつ，すでに抽象化が完了した領域ではその状態クラスの有用度関数やQ値を学ぶことが可能になっている．

これに対して，例えば[77]のような「センサー入力変化の類似性に基づく状態抽象化ポリシー」を採用した場合，状態の抽象化はゴールへの到達や報酬獲得とは直接関係なく行われるので，各状態（クラス）においてどのアクションを取るべきかという知識は得られず，行動政策に関する学習は状態抽象化とは完全に独立して行われる必要がある．加えてセンサー入力空間中での状態クラスの生成は全領域で同時多発的にに行われるので，上のようなストラテジーによって状態の抽象化と行動政策の獲得を並列化することはできない．本章で扱う状態抽象化問題のクラスを「ゴール状態・状態クラスへの到達に基づく状態抽象化」と「獲得報酬の類似性に基づく状態抽象化」に限定した理由はこの点によるところが大きい．

より完全な状態の抽象化と行動政策獲得の同時並列化，すなわち抽象化されている最中の状態クラスについてもその有用度関数やQ値が学ばれるような学習過程を実現するためには，状態抽象化と行動政策獲得との複雑な干渉による安定性や収束性などの問題を考える必要がある．この問題については本章の最後（4.7節）でより詳しく考察するとともに，次章でその解決への1つのアプローチを示す．

次節では，上記アルゴリズムのステップ（5）において行われる単純ベイズ分類器による状態の一般化について詳しく述べる．

[1]実際には，Q-Learningの場合は1回の行動が行われるごとにQ値の更新が行われるのに対し，Profit Sharing法では1つのエピソード（報酬が獲得されるまでの行動シーケンス）が終るごとにそのエピソードで用いられたルールを一括して強化するなどの違いがある．

[2]ここでの「近い」「違い」という言葉の意味は，単なるセンサー空間内での物理的な距離というよりは，選択可能なアクションによってどれだけ容易に到達できるかという「コスト的な」距離のことである．

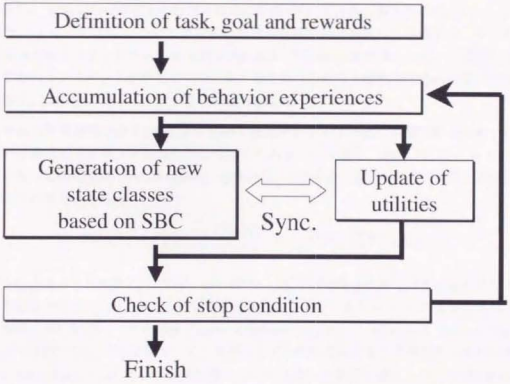


図 4.1: 単純ベイズ分類器（SBC）を用いたゴール・サブゴール到達および報酬獲得に基づく状態抽象化

4.4 単純ベイズ分類器による状態の一般化

4.4.1 単純ベイズ分類器

ベイズ分類器（Bayesian classifier）はパターン認識の分野において昔から用いられて来た確率的分類手法である [27][106] が，近年になって，帰納的機械学習における高い性能がにわかに注目されるようになっている [53].

ベイズ分類器の基本的考えは次のようなものである．まず任意のインスタンスがあるクラス C_i に属する事前確率（prior probability）$P(C_i)$ と，クラス C_i に属するという条件のもとでの各属性（変数）X_j が特定の値を取る**条件付き確率**の分布 $P(X_j|C_i)$ を，各クラス C_i $(i = 1, 2, \cdots)$，各属性 X_j $(j = 1, 2, \cdots)$ について訓練例の集合から統計的に推定し，保存する．そしてクラス名が未知の新しいインスタンス $\boldsymbol{x}$ が提示されると，Bayes の定理 $P(C_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|C_i)}{P(\boldsymbol{x})} \cdot P(C_i)$ を用いてそのインスタンスがあるクラスに属するための条件つき確率 $P(C_i|\boldsymbol{x})$ を各クラスについて計算し，その値が最も大きなクラスに割り当てる．

ここで条件付き確率分布 $P(\boldsymbol{X}|C_i)$ を正確に推定することは一般に困難であるため，ベイズ分類器では属性同士がクラス変数に関して独立であるという条件，$P(X_j|C_i, X_k) = P(X_j|C_i)$ を仮定する（**単純化条件 ： naive assumption**）．その結果，条件付確率（尤度）$P(C_i|\boldsymbol{x})$ は以下のように書き直される．

$$P(C_i|\boldsymbol{x}) = P(C_i) \prod_j \frac{P(x_j|C_i)}{P(x_j)} = \alpha \cdot P(C_i) \prod_j P(x_j|C_i) \tag{4.1}$$

ここで $P(x_j)$ はクラス変数 C_i に依存しないので，しばしば省略される（つまり $\boldsymbol{x}$ がどのクラスに属するかは $P(C_i) \prod_j P(x_j|C_i)$ の値の大小比較のみによって定まる）．この仮定に基づいたベイズ分類器は特に**単純ベイズ分類器**（naive Bayesian classifier ： NBC [54]，simple Bayesian classifier ： SBC [24]）と呼ばれている．単純ベイズ分類器におけるこの属性間の条件つき独立条件（naive assumption）は多くの実問題においては決して厳密に成立しない非現実的な仮定である．そのため，単純ベイズ分類器は長年の間，大した性能を持たないと思われ，さしたる注意を向けられることがなかった．しかし近年になって，属性同士の条件付き独立仮定が全く成り立たないような問題ドメインにおいても，単純ベイズ分類器が C4.5[69] など洗練された分類器と同等もしくはそれ以上の性能を持ち得るということが実験的に，また一部理論的に明らかにされている [24][33]．さらに，ベイズ分類器は適用の容易さ，計算コストの低さ，ノイズに対する頑強性などの望ましい性格を有している [52][65].

現在のこの研究分野では，単純ベイズ分類器のパフォーマンスを改善しようとするいくつかの試みが行われている．その一つは，4.1 式における条件つき確率分布 $P(x_j|C_i)$ を，インスタンスデータからいかに正確に推定するかに関する研究である．実世界における多くの概念学習問題は，連続的な属性値（continuous attribute）を扱うものであるが，単純ベイズ分類器によるこれらの問題への伝統的なアプローチでは，(1) 各属性の定義域をあらかじめ分割しておき離散値属性として扱うか，(2) 平均値と分散だけを計算し，正規分布（ガウス分布）によって

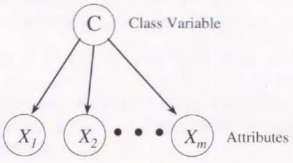


図 4.2: 単純ベイズ分類器のネットワーク表現

推定するものがほとんどである．そこで，より正確な推定を行うために，複数のガウス分布の重ね合わせによる分布モデルを仮定しそのパラメータを推定する方法 [39] や，情報量基準を用いて連続値属性の離散化を最適に行う方法 [26] などが提案されている．

もう一つの試みは，単純ベイズ分類器における“属性間の条件付き独立仮定”（naive assumption）を緩和しようとするものである．単純ベイズ分類器は図 4.2 に示すように，クラス変数 C を根（root）とし，説明属性 X_i を互いに独立な葉（leaf）とする，ベイジアン ネットワーク（Bayesian network）[66][43] を特殊化（単純化）したものとして表現されるが，これらのアプローチではこの分類器をより精密なネットワーク構造に拡張したり [33]，相関（相互依存性）の強い葉同士を統合したり [65] することによって性能を向上させることに成功している．

また，他のアプローチとしては，ブースティング（boosting）[32] の単純ベイズ分類器への応用が上げられる [29][7]．この方法は，あるインスタンス集合を用いてベイズ分類器を構成した後，そのエラー率と各インスタンスの正誤に基づいてインスタンス集合に重み付けをおこなって再び分類器を作る，ということを繰り返し，得られた複数の分類器による選挙（voting）によってインスタンス分類を行うという技術である．

4.4.2 センサー入力から状態クラスへの一般化

ここでは，この単純ベイズ分類器を前節で示した状態抽象化−行動学習アルゴリズムのステップ（5）におけるアクションと行為結果に基づくセンサー入力の一般化，すなわち状態抽象化に適用する方法について述べる．

前掲アルゴリズムのステップ（5）では，過去の一定期間に記録された行動経験の集合 $\mathcal{D}_{bch}$ をアクション a と行動結果 $\boldsymbol{r} = (r_g, r_o)$ の違いによって部分集合 $\{\mathcal{D}_{N+1}, \mathcal{D}_{N+2}, \cdots\}$ に分類し，それぞれの集合に含まれる行動経験インスタンスのセンサー入力 $\boldsymbol{s}$ を一般化することによって新しい状態クラス C_i を定義している．提案手法ではこのセンサー入力ベクトル $\boldsymbol{s}$ から状態クラス C_i への一般化にベイズ分類器を用いる．

今，あるセンサー入力 $\boldsymbol{s}$ が観測されたときにそれが状態クラス C_i に属する確率 $P(C_i|\boldsymbol{s})$ すなわち尤度を単純ベイズ分類器の基本式 4.1 に基づいて推定すると，

$$P(C_i|\boldsymbol{s}) = \alpha \cdot P(C_i) \prod_j P(s_j|C_i)$$

となるが，本手法ではこの式中の「ある状態クラス C_i に属するときにセンサー S_j が値 s_j を取る」条件付き確率分布 $P(s_j|C_i)$ をセンサー S_j が値 s_j を取る一般的な確率 $P(s_j)$ によって正規化し，さらに式全体の対数を取る．

$$F_{C_i}(\boldsymbol{s}) = \log P(C_i|\boldsymbol{s}) = \sum_{\substack{all \\ sensors}} \log \frac{P(s_j|C_i)}{P(s_j)} - \log \frac{1}{P(C_i)} + \log \alpha' \tag{4.2}$$

センサー入力 $\boldsymbol{s}$ はこの対数尤度 $F_{C_i}(\boldsymbol{s})$ が最も大きな状態クラスに分類され，センサー入力空間はこの関数の値によって離散的な状態に分割される．したがって状態クラスの生成・定義は，この式中の確率比（の対数）$\log \frac{P(s_j|C_i)}{P(s_j)}$ と，任意の状態がある状態クラス C_i に属する事前確率 $P(C_i)$ を過去のデータから推定するということに帰着される．

4.4.3 異種センサーの扱いと確率分布の推定

上の 4.2 式は各センサーが離散的な値を取る場合のものであるが，連続な値を取るセンサーが含まれる場合には，それらのセンサーについては確率 $P(s_j|C_i)$，$P(s_j)$ ではなく，確率密度 $p(s_j|C_i)$，$p(s_j)$ を用いれば良い．すなわち 4.2 式はより一般的に，

$$\begin{aligned} F_{C_i}(\boldsymbol{s}) &= \log P(C_i|\boldsymbol{s}) \\ &= \sum_{\substack{continuous \\ sensors}} \log \frac{p(s_j|C_i)}{p(s_j)} + \sum_{\substack{discrete \\ sensors}} \log \frac{P(s_j|C_i)}{P(s_j)} - \log \frac{1}{P(C_i)} + \log \alpha' \end{aligned} \tag{4.3}$$

となる．ここで，ある状態クラスに属するという条件のもとであるセンサーが特定の値を取る条件付き確率（密度）$P(s_j|C_i)$ または $p(s_j|C_i)$ は一般的な（すなわち属する状態クラスが未知のときの）確率（密度）$P(s_j)$ または $p(s_j)$ によって正規化されているので離散値センサー・連続値センサーを問わず，$\log \frac{P(s_j|C_i)}{P(s_j)}$ または $\log \frac{p(s_j|C_i)}{p(s_j)}$ の値を「センサー S_j に関する現センサー入力の状態クラス C_i へのメンバーシップ（属する確からしさ）」を測る指標として同等に扱うことができる．

次に実際の確率分布 $P(s_j|C_i)$，$P(s_j)$，確率密度関数 $p(s_j|C_i)$，$p(s_j)$，および状態クラス C_i に属する事前確率 $P(C_i)$ をどのようにして行動経験データ $\mathcal{D}_{beh}$ から求めるかということについて説明する．まず，離散値センサーの場合の確率分布 $P(S_j|C_i)$ は，C_i に属するインスタンスの各センサー値 $S_j = s_{j1}, s_{j2}, \cdots$ を取る頻度分布から直接的に求められる．

一方，S_j が連続値センサーである場合，本手法ではノンパラメトリック推定法の一種である k_n-最近傍推定法 [27] を用いる．すなわち，センサー S_j 上のある値 s_j における確率密度 $p(S_j = s_j|C_i)$ は $\mathcal{D}_{beh}$ 中で $S_j = s_j$ の近傍に分布する C_i のインスタンスから間接的に求められ

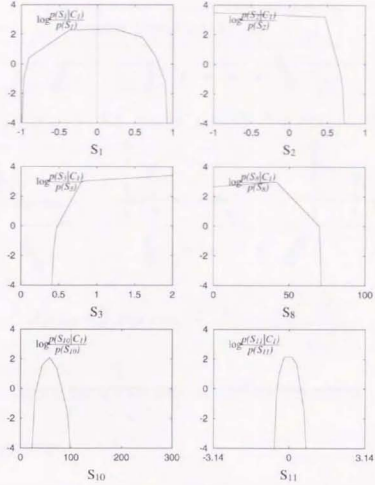


図 4.3: 推定された条件付き確率分布（対数比）の例

る．図 4.3 はこのようにして推定された $\log\frac{p(s_j|C_1)}{p(s_j)}$ を示したものである．連続値を取る変数についての確率（密度）関数を扱う方法としては，他に値の取り得る範囲を分割して離散値センサーとして扱う方法や，何らかの標準的な分布形に当てはめるパラメトリックな推定方法を用いることが考えられる．まず連続値領域を離散化するという方法は，適切な分割を求めること自体が自明でなく不適である．一方，分布モデルを仮定したパラメトリックな方法はデータ数が少ない場合においても比較的安定に分布を求めることができるが，例えばガウス（正規）分布などのような単純な分布モデルでは不十分であり，混合分布モデルと EM 法などを用いた方法などが考えられる．

また，各センサーの一般的分布関数 $P(S_j)$，$p(S_j)$ については，$P(S_j) = \sum_i P(S_j|C_i) \cdot P(C_i)$，あるいは $p(S_j) = \sum_i p(S_j|C_i) \cdot P(C_i)$ として求めることができる．しかし，既に述べたようにこれらの一般的分布は正規化のために用いられているだけであり，単純ベイズ分類器の基本式 4.1 からもわかるように，必ずしも正確に求める必要はない．

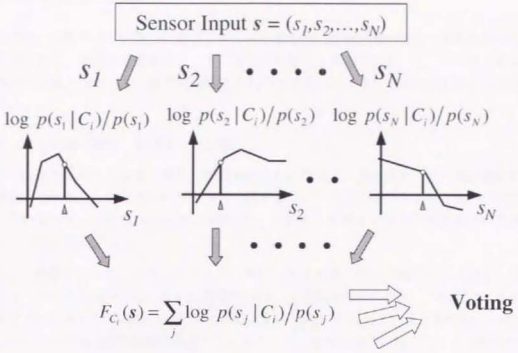


図 4.4: 単純ベイズ分類器における分散処理的状態認識

4.4.4 冗長情報の利用

式 4.3 は，現在のセンサー入力全体によって表される状態がどの状態クラスに属するかを，1 つ 1 つのセンサー入力値ごとに並列的に評価しその結果を足し合わせることで総合的な判断を行うある種の**分散処理**として捉えることができる（図 4.4）．そのため，十分に冗長な構成になっている場合には，一部のセンサーに擾乱やフォールトが生じてもそれが全体に影響しづらく，またセンサー入力の一部が欠けていても残りの入力で暫定的な $F_{C_i}(\boldsymbol{s})$ の値を同様に計算することによって緩やかな機能低下（graceful degradation）が実現されると考えられる．言い替えれば，冗長なセンサーシステムにおいてこの状態一般化・表現法を使うことによって，不確定性による観測のばらつきが平滑化されロバスト性が増すと考えられる．

ただし，このような性格はいずれも条件つき独立を仮定しているセンサー群が "バランス良く" 配置されていることが前提となっている．つまり，エージェントの状態認識に必要なセンサー情報が "偏りなく" 冗長になっているということが想定されている．どのようにしてそのような "バランスの良い" センサー系を構成するかということについては，後で述べる各センサー情報の有用性基準，および任意のセンサー間の類似性基準などを用いることが考えられる．

4.5 他手法との比較

ここでは，単純ベイズ分類器に基づく本手法と他の代表的な状態一般化・表現手法との比較を定性的に行う．比較する他手法は，1）決定木を用いた方法 [16][2]，2）マハラノビス楕円体を用いた方法 [5][80]，3）線形判別関数を用いた方法 [45]，4）最近傍法を用いた方法 [77] である．

決定木による状態一般化・表現法との比較

まず，決定木を用いた状態一般化・表現法と比較した場合，両者はセンサー値が連続的であるか離散的であるか，またある状態クラスに属するという条件下でのセンサー値の分布形がどうなっているかという性格に基本的に依存せず，それゆえ異種センサー情報の統合が柔軟に行えるという点は共通する．

しかし，複数のセンサーがエージェントの状態のある側面に関して類似の（冗長な）情報を提供するという本研究で想定している環境下では，決定木を用いた分類・一般化はより多くの情報量ゲインをもたらす属性（センサー情報）のみを用いてセンサー入力を分類・一般化し，少ないながらも同様の有効な情報を含んでいるセンサー情報は捨てられてしまう可能性が高い．すなわち，全てのセンサー情報を（ある意味で公平に）用いる単純ベイズ分類器を用いた状態分類・一般化法に比べて効率は良いが，擾乱やフォールトなどに対して脆くなりがちだと考えられる．この問題の改善法としては複数の属性を組み合わせることによって新しい属性軸を作るという提案 [2] もあるが，その場合は次元やタイプ (連続性など) が違うセンサー情報から単純な四則演算によって新たな属性を作るという現状の方法には無理がある．

一方，決定木は分割を増やすことによって，かなり精密にインスタンスを分類することが可能である．例えばセンサー入力空間中の XOR の領域を単純ベイズ分類器で表現することは難しいが，決定木では容易である．これらの性質に関しては近年，帰納的機械学習（教師付き学習）やデータマイニングなどの分野で両者の実験的な比較が数多く行われている [52]．これらの結果からは，多くのドメインにおけるベンチマークテストにおいて，ベイズ分類器やその拡張版は，C4.5 などの決定木アルゴリズムと同等かそれ以上のパフォーマンスを持つことが示されている．

マハラノビス楕円体表現を用いた状態一般化・表現法との比較

次に中心点からのマハラノビス距離がある値以下になるような超楕円体領域によって状態クラスの一般化と表現を行う方法と提案手法を比較する．両者は冗長，すなわち互いに相関のあるセンサー入力を柔軟に扱え，例えば一部のセンサー入力が欠けている場合に残りの入力によって状態分類を比較的小さなパフォーマンス低下で行うことができるという点は共通である．しかし，マハラノビス楕円体を用いた方法では全センサーが連続値を取り，また各状態クラスに含まれるインスタンスが多次元正規分布に従うことを前提としており，性格の異なる異種センサー情報を統合するという目的には適さない．また，マハラノビス距離を計算するときに用いる分散共分散行列の非対角成分は各センサー情報間の相関を明示的に扱うことを意味するが，こ

れは全センサーが想定したノイズレベルの範囲内で正常な値を返す場合においては，センサー値間の相関の補正を行わない単純ベイズ分類器を用いた提案手法よりも正確な分類を行えると考えられる．しかし，一部のセンサー入力に大きな誤りが生じた場合にはそれが全体の状態認識に対して大きな影響を与えてしまう．個々のセンサーレベルではそのようなフォールトが常に生じ得るような実環境エージェントではこの性格は望ましくない．また，エージェントにとって入手可能なセンサー情報が増えることによってセンサー入力の次元が高くなった場合の基本的な計算量は，単純ベイズ分類器ではその次元に比例するオーダーであるのに対して，マハラノビス楕円体による方法では任意のセンサー入力間の相関係数を計算するので，次元の2乗のオーダーで増加することになる．そのうえ，2つのセンサー入力が非常に近い性格を持つ場合，分散共分散行列が非正則に近付くため逆行列の数値計算が不安定になり，正確なマハラノビス距離を計算できないという問題もある．

線形判別関数を用いた状態一般化・表現法との比較

線形判別関数が同じ分散共分散行列を持った2つのクラスの境界面になることからも推測されるように，線形判別関数を用いた状態一般化・表現法の特徴は，マハラノビス楕円体を用いた手法に近い．したがって，本手法と比較した場合，離散値を取るセンサー入力を扱えない点，連続値であっても状態クラスの分布が多次元正規分布と著しく異なる場合には適用できない点，などが指摘される．

最小近傍法を用いた状態一般化・表現法との比較

最後に最近傍法（nearest neighbourhood method）を用いた状態一般化・表現法と比較する．

あるセンサー入力がどの状態クラスに属するかを，各状態クラスの代表点までの重み付きユークリッド距離の小ささによって決めるこの方法は，各センサー入力を独立に扱うという点では，マハラノビス楕円体による方法よりも，単純ベイズ分類器を用いた本提案手法の考え方に近く，一部のセンサーのフォールトが全体に及ぼす影響はやはり小さいので，同様のロバスト性を有していると考えられる．また，各状態クラスの代表点を増やすことによって，どのような形の領域も自由に表現することができるという点は本提案手法よりも優れている．しかし，この方法はマハラノビス楕円体による方法や，線形判別関数による方法と同様に，離散値を取るセンサーが含まれている状態一般化問題には使用できない．また，この手法は重み付きユークリッド距離という距離尺度がセンサー入力空間中の状態の近さを適切に反映していることを大前提としているので，その仮定が成り立たないような悪構造のセンサー入力空間では利用することができない．

これに対して，単純ベイズ分類器を用いた提案手法ではこのような**物理的な距離と本質的な状態の類似性との間が線形関係にある**という仮定を排することによって，より柔軟な状態一般化・表現を可能にしていると言える．つまり提案手法で，は式4.3における条件付き確率分布の関数 $\log \frac{P(s_j|C_i)}{P(s_j)}$，$\log \frac{p(s_j|C_i)}{p(s_j)}$ をノンパラメトリックな方法で求めることによって**物理的な距離と本質的な状態の類似性との間の関数関係**自体を自律的に獲得していると言える[2]．

[2] $p(s_j|C_i)$ の分布を，正規分布のような単純な分布形に従うと仮定し，パラメトリックな方法により推定した場

4.6 センサー情報の有用性・類似性の基準

従来の状態抽象化問題に関する研究の大部分では，エージェントにあらかじめ与えられたセンサー群からの情報をいかに正確に，効率的に一般化するかという点に重点が置かれており，そのエージェントがタスクを遂行する上でどのようなセンサーが必要であるかということや，異なる個々のセンサー入力同士がエージェントの状態に関する共通の情報をどの程度含むか，という問題についてはほとんど議論が行われていない．

しかし，これまで再三述べてきたように，簡単な内部表現（モデル）によって記述することが不可能で，ノイズや故障など様々な不確定性要素に支配されるような実環境において頑強な行動決定を行うエージェントを実現するためには，異種冗長なセンサー系に基づいた状態の自律的抽象化が必要であり，そのセンサー系に含まれる個々のセンサーの定量的な有用度（重要度）や相互の類似度（冗長度）を測る何らかの指標が必要である．

ベイズ分類器による帰納学習を研究する分野では，インスタンスの分類に用いられる各属性の個々の重要性や，属性同士の依存性（dependency）を定量的に評価するのに，情報理論に基づいた指標を定義し用いていることが多い [24][33]．本研究でもこれに従い，状態一般化および状態分類における個々のセンサー入力の有用性とセンサー間の非独立性（類似度）を定義する．

4.6.1 センサー情報の有用性基準

まず，状態クラス集合 $\mathcal{C}$ の分類における 1 つのセンサー S_j の重要度 $Imp_{\mathcal{C}}(S_j)$ を次式のように定める．

$$
\begin{aligned}
Imp_{\mathcal{C}}(S_j) &= I(S_j;\mathcal{C}) = H(\mathcal{C}) - H(\mathcal{C}|S_j) = H(S_j) - H(S_j|\mathcal{C}) \\
&= \begin{cases} \sum_i \sum_l P(C_i) P(s_{j,l}|C_i) \log \frac{P(s_{j,l}|C_i)}{P(s_{j,l})} & (S_j \text{が離散値の場合}) \\ \sum_i P(C_i) \int p(s_{j,l}|C_i) \log \frac{p(s_{j,l}|C_i)}{p(s_{j,l})} ds_{j,l} & (S_j \text{が連続値の場合}) \end{cases}
\end{aligned} \tag{4.4}
$$

$I(S_j;\mathcal{C})$ はセンサー入力と状態クラス集合との間の**平均相互情報量**，$H(\mathcal{C})$ はクラス集合 $\mathcal{C}$ に関する**情報エントロピー**（＝あいまいさ），$H(\mathcal{C}|S_j)$ はセンサー S_j の値が得られたとしたときのクラス集合 $\mathcal{C}$ に関する**平均条件付きエントロピー**である．つまり，$I(S_j;\mathcal{C})$ はエージェントがセンサー入力 S_j の値を知ることによって，現状態がどの状態クラスに属するかということに関してどの程度の情報が得られるかという情報量ゲインの期待値であり，その値の大きさは直観的にセンサー S_j が状態を認識する上でどの程度有用であるかを示している．

平均相互情報量 $I(S_j;\mathcal{C})$ は S_j と $\mathcal{C}$ に関して対称なので，実際には求めるのが困難な $H(\mathcal{C}|S_j)$ の代わりに，経験データから $H(S_j|\mathcal{C})$ と $H(S_j)$ を計算することによって $I(S_j;\mathcal{C})$ を求めることができる．

合にはこの主張は成り立たず，距離と本質的な状態の類似性との間の線形関係を仮定したことと同じになる

4.6.2 センサー同士の類似性基準

同様に2つのセンサー S_j, S_k がどの程度似た情報を提供するかという類似性を表す指標 $Sim_{\mathcal{C}}(S_j, S_k)$ として，センサー S_j, S_k と $\mathcal{C}$ との3者間共通の相互情報量 $I(S_j; S_k; \mathcal{C})$ を用いる．

$$Sim_{\mathcal{C}}(S_j, S_k) = I(S_j; S_k; \mathcal{C}) = I(S_j; \mathcal{C}) + I(S_k; \mathcal{C}) - I(S_j S_k; \mathcal{C}) = I(S_j; S_k) - I(S_j; S_k|\mathcal{C})$$

$$= \begin{cases} \sum_l \sum_m P(s_{j,l}, s_{k,m}) \log \frac{P(s_{j,l}, s_{k,m})}{P(s_{j,l})P(s_{k,m})} \\ -\sum_i P(C_i) \sum_{l,m} P(s_{j,l}, s_{k,m}|C_i) \cdot \log \frac{P(s_{j,l}, s_{k,m}|C_i)}{P(s_{j,l}|C_i)P(s_{k,m}|C_i)} \quad \text{（離散値の場合）} \\ \int_{s_j} \int_{s_k} p(s_j, s_k) \log \frac{p(s_j, s_k)}{p(s_j)p(s_k)} ds_j ds_k \\ -\sum_i P(C_i) \int_j \int_k p(s_j, s_k|C_i) \cdot \log \frac{p(s_j, s_k|C_i)}{p(s_j|C_i)p(s_k|C_i)} ds_j ds_k \quad \text{（連続値の場合）} \end{cases} \tag{4.5}$$

ここで，$I(S_j S_k; \mathcal{C})$ は S_j, S_k を一つのセンサーとして扱った場合の $\mathcal{C}$ との平均相互情報量であり，$I(S_j; S_k|\mathcal{C})$ は現在の状態クラスが分かったとしたときの2センサー S_j, S_k 間の**条件付き平均相互情報量**である．すなわち $I(S_j; \mathcal{C}) + I(S_k; \mathcal{C}) - I(S_j S_k; \mathcal{C})$ は2つのセンサー S_j と S_k の値を別々に知ったとしたときに見込まれる $\mathcal{C}$ に関する情報量ゲインから，両者を同時に知ったとしたときに得られる実際の情報量ゲインを引いたものであり，S_j と S_k が $\mathcal{C}$ のに関してどの程度互いに似た情報を与えるかという尺度として見ることができる．つまりこの値が大きい程，2つのセンサーは同種の情報を担っているということを意味する．

6章で述べるシミュレーションの結果では，これら2つの指標がそれぞれ，各センサーの重要度と，任意の2センサー間の機能的類似性を経験的に良く表すことが示される．

4.7 提案手法の限界と拡張

4.7.1 状態抽象過程のオンライン化

本章で提案した単純ベイズ分類器に基づく状態一般化法は，異種冗長なセンサー情報を効率的に統合することによって，実環境エージェントにとって不可避なセンサーノイズや故障などの不確定性に対して頑強な状態一般化・認識を可能にする．

しかし，本提案手法では完全には解決されていない重要課題として，抽象化過程のより完全なオンライン化という問題がある．ここで言う“オンライン化”とは，エージェントの状態抽象化，行動政策の獲得，実際のタスク遂行の3つが同時並行的，あるいは繰り返し的に行なわれることである．その意味で提案手法のオンライン化を実現するためには，大きく2つの課題が考えられる．1つは現状のようにまとまった行動経験データからバッチ的に状態クラスの表現を得るのではなく，新しい行動経験データが得られる度に状態クラスの定義を少しずつインク

リメンタルに更新していくにはどうしたら良いかという問題である．これはエージェントを取り巻く環境が時間的に変化していくような場合にその変化に追従していくという観点からも重要である．もう1つはエージェントの学習のもう一段上のレベルのループで起こる問題で，状態抽象化と，それと並行して行われる行動政策学習との間の干渉をどのように扱うかという問題である．すなわち，行動政策の学習は抽象化された状態クラスの集合がどうなっているかに強く依存するが，状態の抽象化もまた行動政策を学習したエージェントの行動経験データに強く影響を受ける．

前者については，本稿で採用しているバッチ的な状態クラス生成の単純な拡張として，新しい経験インスタンスが得られる度に最も古いインスタンスを除きながら逐次抽象化を行うことによって，環境の変化に対応していくということが，まず考えられる（疑似オンライン化）．この方法は状態クラスの定義に必要な経験データを全て記憶しておく必要があり，また大きな計算量を要するという点であまり本質的な解決策とは言えない．一方，単純ベイズ分類器における各連続値センサーに関する確率密度分布をパラメトリックな方法により推定する場合には，その分布モデルに付随するパラメータを**ベイズ学習** [27] などにより順次更新していく方法などが考えられる．

後者の問題については，本章のはじめに述べたように，扱う問題クラスをゴール・サブゴールへの到達，および報酬の獲得に基づく状態抽象化に限定し，政策学習すなわち状態−アクションの有用度に関する学習は既に生成された状態クラスについてのみ行うことで，両者が相互に干渉してしまう状況を考慮の対象から外している．しかしながら，抽象化がまだ完了していない状態クラスに関してもその有用度や各アクションの価値を学ばなければならないような状況，すなわち状態抽象化と行動学習が完全に同時並列に行われる場合には，両者の干渉問題は非常に重要である．まず，ベイズ分類器においては，各クラスに属するインスタンスの頻度分布から確率密度分布 $p(S_j|C_i)$ を求めるが，オンラインでこれを行う場合，行動学習の影響を受けて，得られる経験インスタンスに偏りが生じると予想される．この場合直観的には，新しい経験インスタンスが得られる度に，現在の状態クラシファイヤーを用いて分類し，それが正しければそのセンサー値における $p(S_j|C_i)$ を大きくし，誤っていれば逆に小さくするという強化則によって更新することが一つの方法として考えられる．また，各状態クラスの領域が徐々に変化していく場合における有用度や価値関数の更新については，例えば [77] で提案されているように，領域が変化した後のある状態の価値関数を，変化前の各状態における価値関数の重み和によって見積もるという方法が考えられる．

4.7.2 状態抽象化におけるロバスト化

本研究では主に，エージェントが実環境から，様々なノイズやフォールトなどの不確定性要因を含んだセンサー情報を得て行動決定を行う場合に，いかにしてそれらの不確定性に対して頑強な状態認識を実現するか，という観点から，異種冗長なセンサー情報源を効率的に統合して状態の一般化および表現を行う方法を議論した．すなわち，ここで扱った「頑強性」は，精

密な呼び方をすれば「センサー入力に含まれる不確かさやフォールトに対する状態分類メカニズムの頑強性」ということになる．しかし，状態に関する一般的な概念の獲得という観点からこの問題を見れば，もう一つの本研究では扱わなかった「状態空間（というモデル）の状態のばらつきに対する頑強性」の存在というものが指摘される．

すなわち，前者の頑強性は情報に含まれる不確定性に対してのロバストさを指し示すものであったのに対し，後者はある状態クラスに含まれる状態インスタンス自体のばらつきに対するクラスモデルのロバストさを意味している．そしてより具体的には，有限個の（状態の）インスタンスを一般化することによって状態クラスの表現を得る際に，いかにそのインスタンスに対する過一般化（over-generalization）や過特化（over-specification）を防ぐかという問題と深く関連する．この点に関して本研究では，各状態クラスを単純ベイズ分類器で表現する際に，k_n-最近傍法というノンパラメトリックな推定法によって条件付き確率分布を計算しているため，精密な状態表現ができる反面，インスタンスが少数である場合に過特化が起こりやすいという問題がある．つまり，後者の文脈での「頑強性」は満たされていないということになる．

しかし，この概念モデルの一般性と精密性との間のトレードオフという問題に関しては，あらゆる知識表現法（例えば決定木や人工ニューラルネットワークなども含まれる）に共通する問題であり，本研究で用いたベイズ分類器に限定された話ではない．したがって，解決策もまたこれら他の表現系において研究されてきた理論，例えば**赤池の情報量**（AIC）や**最小記述長**（MDL）理論などを用いれると考えられる．実際，ベイズ分類器の一般系であるベイジアンネットワークを事例データベースから自動構築することを目的とする研究分野では，MDL を用いた評価基準が提案されている [76]．

第5章　行為結果のばらつき最小化に基づく状態と行為の抽象化

本章では反射的エージェントにおける状態・行為抽象化問題の新たな枠組として，行動経験データにおける**行為結果のばらつき最小化に基づいた状態空間・行為空間の構成法**を提案する．これはエージェントの状態空間および行為空間の大局的な評価基準を，“それらの状態集合，行為集合によって過去の行動経験を分類したときに，行為結果（action outcome）に関するばらつき（情報エントロピー）がどれだけ減少するか”ということによって定義し，それを最適化するセンサー入力空間とモーター出力空間の分割（partition）を見つける過程として，状態・行為抽象化問題を定式化するものである．

この枠組はエージェントの状態遷移や獲得報酬，ゴール到達など，従来の“行為結果の類似性に基づく状態・行為抽象化法”で用いられて来たいくつかの行為結果を評価基準に併せ含めることによって，強化学習などの行動政策学習との並列化による行動経験データの性格の変化を利用して効率的な状態・行為空間構成を行うことができる．また同時に，人間が暫定的に与えた状態・行為空間を初期値として状態，行為を再抽象化したり，環境やエージェント自身の変化に対して柔軟に適応する手段も提供する．

本章ではこの一般的な状態・行為抽象化の枠組に基づき，ヒューリスティックな探索を用いた具体的な状態空間，行為空間の構成法も紹介する．

5.1　情報量基準による行動結果のばらつき表現

3章で述べたように，行為結果の類似性に基づく状態・行為抽象化は，「同じアクションによって同じ行為結果をもたらすようなセンサー入力を同じ状態とし，同じ状態において同じ行為結果をもたらすようなモーター出力を同じアクションとして一般化する」過程として説明されるが，これを言い替えれば，「**ある状態においてあるアクションを行ったときの行為結果が常に同じになるような状態空間，アクション空間ほど望ましい**」という基準が得られる（図5.1）．つまり，エージェントのある状態空間およびアクション空間は，エージェントの反射的行動の因果則（「状態Cにおいてアクション Aを実行すると，結果Rが得られる」という形の）に対する大きな意味での仮説モデルであると考えられ，その善し悪しは実際のエージェントの反射行動の結果をどれだけ正確に予測できるか，あるいは，過去の行動経験データをどれだけ正しく表現しているか，ということによって評価することができる．

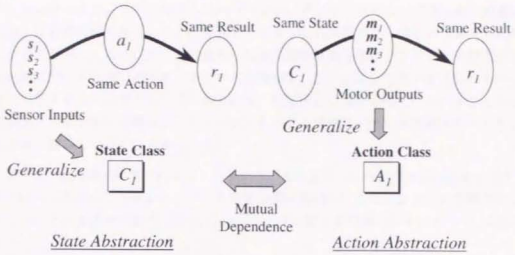


図 5.1: 行為結果の同一性に基づく状態・行為の抽象化

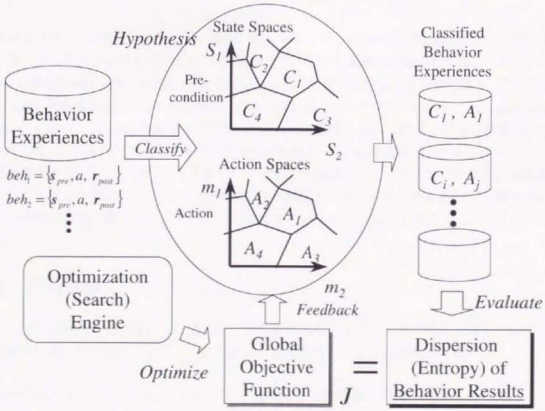


図 5.2: 行為結果のばらつき最小化に基づく・行為の抽象化 — 概要

ここで，過去の行動経験データをどれだけ“正しく”表現しているかということは，(アクション前の) センサー入力，(実行された) モーター出力，そしてそれによって得られた結果によって記述される行動経験の集合を，この仮説モデルすなわち状態クラス集合と行為クラス集合によってクロス分類し，それによって分割された各行動経験集合中のインスタンス同士がどれだけ行為結果に関して共通であるか，すなわち行為結果に関するばらつきが小さいかということであると解釈することができる．言い替えれば，行為結果に関するばらつきが小さくなるような状態空間・アクション空間ほど，エージェントのその環境における状態遷移モデルをより決定論的に表し得るということである．

このように，ある集合に含まれるインスタンスのある変数（この場合は行為結果を表す変数）に関するばらつき具合（不確からしさ）を測る定量的指標としては，**シャノンの情報エントロピー**がある．この集合中の変数 $X(=x_1, x_2, \cdots, x_n)$ に関する情報エントロピー H は次式で定義される．

$$H = -\sum_i P(x_i) \log_2 P(x_i) \tag{5.1}$$

ここで $p(x_i)$ はこの集合中において変数 X が値 x_i を取る確率（頻度）である．エントロピー H は $p(x_i) = \frac{1}{n}$ $(i = 1, 2, \cdots, n)$，すなわち，X に関して最も不確定であるとき最大値 $\log_2 n$ を取り，$p(x_j) = 1$, $p(x_i) = 0$ $(i \neq j)$，すなわち X に関する不確からしさが全くない場合に最小値 0 を取る．

本研究では，この情報エントロピーを用いてエージェントのある行動経験集合中の行動結果に関するばらつき具合を表す．つまり，次節で示すように，行為結果の類似性に基づく状態・アクション抽象化問題を，“行為結果に関するエントロピーを最小にする状態クラス集合，アクションクラス集合の探索問題”として再定義して扱う（図 5.2）．このアプローチは CU (categorization utility) を用いた [72][44] と同様，“グローバルな評価基準に基づく状態・行為の抽象化”であると言える．すなわち，“行為結果に関するエントロピー”が抽象化結果に対するグローバルな評価基準となる．ただし後に述べるように，一般に行為結果は 1 つの変数ではなく，獲得報酬やセンサー入力の変化など，いくつかの変数から構成されるベクトルであるので，実際にはそれらの各結果変数に関するエントロピーの重み和 (weighted sum) を考える．

5.2 問題定義

ここでは前節の考えに従い，行為結果に関する情報エントロピー最小化に基づいた状態・行為抽象化の定式化を行う．まず，ここで用いる記号を以下のように定義しておく．

$beh_i = < \boldsymbol{s}_i, \boldsymbol{m}_i, \boldsymbol{r}_i >$ ： エージェントの (i 番目の) 行動経験．3 つの要素は，

- $\boldsymbol{s}_i$ ： (アクションを行う前の) センサー入力ベクトル
- $\boldsymbol{m}_i$ ： (適用された) モーター出力ベクトル
- $\boldsymbol{r}_i$ ： 行動結果ベクトル

$\mathcal{B} = \{beh_1, beh_2, \cdots, beh_n\}$ ： 行動経験集合

$\mathcal{C} = \{C_1, C_2, \cdots C_{n_c}\}$ ： 状態クラス集合

n_c ： 状態クラスの数

$\mathcal{A} = \{A_1, A_2, \cdots A_{n_a}\}$ ： 行為（アクション）クラス集合

n_a ： 行為クラスの数

$w = [w_1, w_2, \cdots, w_{d_r}]$ ： 各結果要素に対する重み係数のベクトル

$\mathcal{B}_{j,k} = \{beh_i | \boldsymbol{m} \in C_j, \boldsymbol{s} \in A_k\}$ ： センサー入力が状態クラス C_j に属し，モーター出力が行為クラス A_k に属する $\mathcal{B}$ 中の行為経験の集合

$|\mathcal{B}_{j,k}|$ ： 経験集合 $\mathcal{B}_{j,k}$ に含まれる要素数

$H_{\mathcal{B}_{j,k}}(r_l)$ ：経験集合 $\mathcal{B}_{j,k}$ での l 番目の行動結果要素 r_l に関する情報エントロピー

$J_{\mathcal{B}_{j,k}} = \sum_l w_l \cdot H_{\mathcal{B}_{j,k}}(r_l)$ ： $H_{\mathcal{B}_{j,k}}(r_l)$ を行動結果ベクトル各要素について計算し，重み係数をかけて足したもの

Rep ：状態クラスとアクションクラスの表現形式（ex. 決定木，単純ベイズ分類器など）

このとき，状態と行為の抽象化問題を以下のように定式化する．

[**行為結果に関する情報エントロピー最小化に基づく状態・行為の抽象化**（状態・行為クラスの個数を指定する場合）]
行動経験集合 $\mathcal{B}$ と各結果要素の重み $\boldsymbol{w}$ および状態クラスとアクションクラスの表現形式 Rep と個数 n_c，n_a が与えれたときに，

$$J(\mathcal{C}, \mathcal{A}) = \frac{1}{|\mathcal{B}|} \sum_{j,k} |\mathcal{B}_{j,k}| \cdot J_{\mathcal{B}_{j,k}} \tag{5.2}$$

を最小化する状態クラス集合 $\mathcal{C}$ と行為クラス集合 $\mathcal{A}$ を求める．

式5.2の J は，状態クラス集合 $\mathcal{C}$ とアクションクラス集合 $\mathcal{A}$ によって全体の行動経験インスタンス集合 $\mathcal{B}$ を部分集合の集合 $\{\mathcal{B}_{j,k}\}$ に分割したときの**行動結果ベクトル $\boldsymbol{r}$ 各要素についての平均エントロピーの重み和**である．言い替えれば，$\mathcal{C}$ と $\mathcal{A}$ を用いてエージェントの過去の行動経験を記述したときの，行動結果に関するばらつきあるいは**曖昧さ**の程度を表しており，これを最小にするような $\mathcal{C}$ と $\mathcal{A}$ を求めることが最適な状態と行為の抽象化であると考えることができる．

ただし，ここで注意すべき点は，J の値自体には絶対的な意味がなく，2つの異なる状況における J の値の差に意味があるという点である．すなわち，ある状態クラス集合 $\mathcal{C}$，行為クラス集合 $\mathcal{A}$ に対する評価関数値 $J(\mathcal{C}, \mathcal{A})$ の大小自体には意味がなく，他の状態・行為クラス集合

$\mathcal{C}'$, $\mathcal{A}'$ における評価関数値との差 $J(\mathcal{C},\mathcal{A}) - J(\mathcal{C}',\mathcal{A}')$ の値の大小には意味があるということである.

5.3 異なる行動結果の考慮

3.4 節で述べたように, "行為結果の類似性に基づく状態・行為抽象化" に関する従来の研究では, "行為結果" として "同じゴール状態, あるいはサブゴール状態に到達するかどうか", "同じ報酬値を獲得するかどうか", "同じ状態クラスへの遷移, あるいは近いセンサー入力変化をもたらすかどうか" という主に3つの判断基準のうちのどれか1つだけを用いて, 異なるセンサー入力同士の近さ, あるいは異なるモーター出力同士の近さを定義し, 状態あるいはアクションへの一般化を行っている.

この "何をエージェントの (重要な) 行為結果と考えるか" という違いは, 当然, 状態やアクションの抽象化結果にも違いを与えるので,「抽象化ポリシー」の違いを表しているとも言える. 例えば, 目的地へ到達することをタスクとする移動ロボットについて, **報酬の類似性に基づく状態とアクションの抽象化**を行えば,「目標地点に到達する」とか「障害物に衝突する」というような正負様々な報酬に直接的/間接的に結び付くような状態あるいはアクションという観点からセンサー入力, モーター出力の一般化が行われ, **センサー入力変化の類似性に基づく状態とアクションの抽象化**を行えば,「接触センサーが On から Off になる」とか「画像センサー中で目標物の位置が右に移動する」というような報酬やゴールとは独立した一般的な情報に基づいて状態やアクションの切り分けが行われることになる.

ここで重要なことは, これらの異なる抽象化基準はどれも完璧なものではなく, 一長一短であるということである (表 3.1). センサー入力変化のような一般的な行為結果の類似性という基準を利用すれば, エージェントの少ない行動経験から効率的に状態・アクションの抽象化を行うことができ, しかも同じエージェントの (報酬の設定などが異なる) 他のタスクに再利用することが可能である. しかし, その反面, このようにして得られた状態空間とアクション空間を用いた場合, 同じ状態で同じアクションを行っても報酬の獲得やゴールへの到達に関しては毎回同じ結果が得られるとは限らず, 行動政策学習を行ってもあまり良い結果が得られない可能性があり, また (報酬やゴールとは全く関係ないという意味で) 無価値な状態やアクションが多く定義されてしまう可能性が大きい. 一方, 直接/間接報酬獲得の類似性に基づく抽象化を行った場合, 始めのうちは「たまたま」報酬を獲得しない限り学習が行われないので非常に効率が悪いが, "正の報酬を獲得するにはどうしたらよいか, また, 負の報酬を受けないにはどうしたらよいか" という観点から状態およびアクションが定義されていくので, 最終的な状態空間・アクション空間は理想的なものになると考えられる.

このことは, 別の捉え方をすれば, 状態・行為の抽象化はこれらの異なる抽象化基準を複数組み合わせる, あるいは状況に応じて使い分けることが重要であることを意味していると言える. 例えば, 学習の初期段階においては, エージェントが報酬獲得やゴール到達をあまり経験

表 5.1: 抽象化基準の遷移

	学習段階初期	学習段階中期	学習段階後期
基準になる行為結果	センサー入力変化	負の報酬	正の報酬
学習の性格	データ駆動型	リスク回避型	ゴール駆動型

表 5.2: 本研究で考慮する 3 種類の行為結果属性

行為後のセンサー入力	行為による直接獲得報酬	遷移状態クラス
s_{post}	rwd	C_{post}

しなくても効率的に状態とアクションの抽象化が行える "センサー入力変化の類似性に基づく抽象化" を行い，状態・アクションの抽象化学習と並行して行われる行動政策学習の成果によって次第に報酬獲得やゴール到達を含む行動経験回数が増えるにつれて，徐々に "獲得報酬の類似性に基づく抽象化" へと移行していくことができれば，学習の効率と最終的なパフォーマンスを両立するという意味で，理想的であると考えられる．このような抽象化基準の変化は，"**データ駆動型（data-driven）の学習**" から，"**ゴール駆動型（goal-driven）の学習**" への移行とも捉えることができる（表 5.1）．

このような考えに基づき，本提案手法では，アクションを行った**後のセンサー入力ベクトル**，その結果**獲得した報酬**，**到達した状態クラス**という 3 つの異なる複数の行動結果要素を考慮して状態と行為の抽象化基準を定める（表 5.2）．すなわち，エージェントの行為結果ベクトル $\boldsymbol{r}$ は，

$$\boldsymbol{r} = [s_{post,1}, s_{post,2}, \cdots, s_{post,d_s}, rwd, c_{post}] \tag{5.3}$$

と表される．そして，この各行為結果要素に関する情報エントロピーに重み係数をかけて足し合わせたものを，行為結果に関するトータルのばらつき尺度として扱う．つまり，ある行動経験集合 $\mathcal{B}_{j,k}$ の行為結果に関するトータルのばらつきは，

$$J_{\mathcal{B}_{j,k}} = \sum_{l} w_l \cdot H_{\mathcal{B}_{j,k}}(r_l) \tag{5.4}$$

によって計算される．ここで w_l が各行動結果要素 r_l に対する重み係数である．

したがって，各行動結果要素に対する重みを変更することによって前述したようなエージェントの状態・行為抽象化ポリシーの移行を実現することができるが，実際の抽象化結果は重み値だけでなく，抽象化に用いる経験データの傾向にも大きく影響を受ける．すなわち，行動政策学習が進むにつれて正の報酬を獲得したりゴール状態に到達する行動経験の比率が相対的に多くなるため，それらに対する重み係数を大きくするのと同様な効果が得られることになる．

5.4 最適化問題としての性格

5.2節で定義した行為結果に関する情報エントロピー最小化に基づく状態・行為の抽象化は，行動経験集合 $\mathcal{B}$，各行為結果要素に対する重みのベクトル $\boldsymbol{w}$，定義する状態クラスとアクションクラスの表現形式 Rep と各個数 n_c，n_a を入力とし，状態クラス集合 $\mathcal{C}$ とアクションクラス集合 $\mathcal{A}$ を出力とする最適化問題の体裁を取っているが，極めて単純なケースを除いて真の最適解を求めるのは困難である．

その根拠は，まず全体の評価関数が局所的な評価関数が多数組み合わさったものであるため，解空間中のいたるところに局所最適解が存在するような問題になっていると考えられるということである．局所的な評価関数とはこの場合，状態クラスとアクションクラスによって分類された行動経験データの部分集合 $\mathcal{B}_{j,k}$ における，各行動結果要素に関するエントロピー $H_{\mathcal{B}_{j,k}}(r_l)$ であるが，これが $n_c \cdot n_a \cdot n_r$ 個だけ存在することになる．(n_r は行為結果として考える変数の数)

また，評価関数が複雑なだけでなく，解空間あるいは探索空間の大きさも問題になる．つまり，実際の解のパラメータ空間の次元は，1つの状態クラスおよびアクションクラスを表現するのに要するパラメータをそれぞれ dim_c，dim_a とすると，$n_c \cdot dim_c + n_a \cdot dim_a$ になる．dim_c，dim_a は状態クラス，アクションクラスをどのような形式で表現するかによって異なるが，大まかにはそれぞれエージェントのセンサー入力，モーター出力の次元数に比例して増えると考えられる．

上のような評価関数と探索空間の性格を考慮すると，全探索はもちろんのこと，局所的な傾きを用いた山登り法的な探索も現実的なアプローチでないことは容易に想像がつく．そこで本手法では探索空間を制限するために1つのヒューリスティクスを用いる．それは，始めから多数の状態クラス，アクションクラスの存在を仮定して最適化を行うのではなく，最初はセンサー入力空間もモーター出力空間も少数の分割だけを考えて最適化を行い，次に最も行為結果の重みつきエントロピー和 $H_{\mathcal{B}_{j,k}}(r_l)$ が大きかった行動経験集合 $\mathcal{B}_{j,k}$ を分割するように状態クラス，アクションクラスを徐々に増やしていくというものである（図5.3）．このようなヒューリスティック探索は，最適解への収束を保証されないが，後のシミュレーションでは概ね好結果をもたらすことが示される．また，今回は扱わないが，本問題と同様の複雑で悪構造な最適化問題の実際的な解法としてよく用いられている遺伝的アルゴリズム（genetic algorithm : GA）や，焼きなまし法（simulated annealing method）などを適用することも考えられる．

5.5 各行為結果要素と情報エントロピーの計算法

本提案手法では，エージェントの行為結果ベクトル $\boldsymbol{r}$ の要素として，アクション実行後の1）各センサー入力，2）（直接）獲得報酬，3）到達した状態クラス の3つを考えるが，これらの行動結果要素のそれぞれの意味とその情報エントロピーの求め方は次のようになっている．

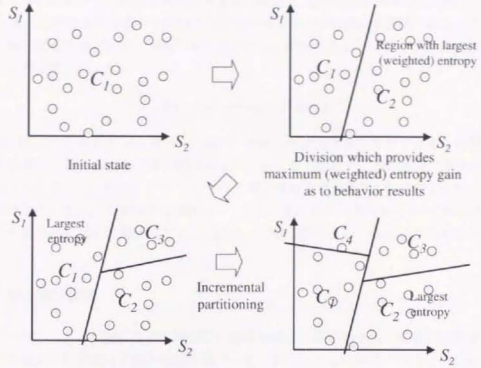


図 5.3: 行為結果のばらつきの漸次的最小化による状態分割

5.5.1 アクション後のセンサー入力

エージェントのセンサー入力の変化は，そのタスクにおけるゴール状態や報酬に対して直接依存しない極めて一般的な行為結果属性である．これを状態と行為の抽象化基準に利用することは，“エージェントの状態が近ければそのときのセンサー入力の値も近く”，また，“近いセンサー入力値（を示す状態）をもたらすアクション同士も近い”という考えに基づいている．

ある行動経験サブ集合 $\mathcal{B}_{j,k}$ のアクション後のあるセンサー入力 S_l に関する情報エントロピーは，S_l が例えば接触センサーのように，**離散値**を取るセンサー入力である場合は，$P(s_{l,m}) = P(S_l = s_{l,m})$ をその行動経験集合中でアクション後に観測された S_l が値 $s_{l,m}$ $(m = 1, 2, \cdots)$ を取る頻度確率（frequency）であるとして，

$$H_{\mathcal{B}_{j,k}}(S_l) = -\sum_m P(s_{l,m}) \log_2 P(s_{l,m}) \tag{5.5}$$

の定義式に従って容易に計算することができる．

一方，S_l が例えば（障害物との）距離センサーのように**連続値**を取るセンサー入力である場合には，連続情報におけるエントロピー定義

$$H_{\mathcal{B}_{j,k}}(S_l) = -\int_{s_l} p(s_l) \log_2 p(s_l) ds_l \tag{5.6}$$

を用いることになるが，これを忠実に計算するためには S_l に関する確率密度分布 $p(s_l)$ を推定し，かつ S_l の定義域全体で積分を実行する必要があるので一般的には困難である．そこで本手法では，確率密度分布 $p(s_l)$ を正規分布であるものと仮定することによってこの値を近似的に求める．この場合，式 5.6 は，

$$H_{\mathcal{B}_{j,k}}(S_l) = \log_2(\sqrt{2\pi e}\sigma_{s_l}) \tag{5.7}$$

のように，$\mathcal{B}_{j,k}$ 中の S_l の分布に関する分散さえ分かれば計算することができる．実際の分布が正規分布と著しく異なる場合にはこの近似は成り立たなくなるが，その場合でも上式が "集合 $\mathcal{B}_{j,k}$ におけるアクション後のセンサー入力 S_l に関するばらつき具合" を反映するものとして扱うことはできる．また，連続信号の情報エントロピーを近似的に求めるもう一つの方法として，その信号の値空間を適当に離散化することによって離散信号として扱うことも考えられる[1]．

5.5.2 直接獲得報酬

エージェントのある行動において獲得した直接報酬は，特定のタスクに密接に関係する行動結果属性である．したがって獲得報酬に基づいて一般化された状態やアクションはそのタスクに強く依存するものとして定義される．例えば，天井にぶらさがっているバナナを手に入れることをゴールとするサルエージェントの場合は，バナナという報酬に直接結び付く "天井に向かって手を伸ばす" というアクションや，"椅子の上の登っている" という状態が一般化されるのに対し，檻からの脱走をゴールとするサルエージェントであれば，"鍵を鍵穴に入れる" というアクションや，"鍵が手の届くところにある" という状態が直接報酬に基づいて定義されることになる．

提案手法では，正負の獲得報酬に関する情報は 1 次元の離散値を取るスカラー量として扱っている．すなわち，$Rwd \in \{r_{+,1}, r_{+,2}, \cdots, r_{-,1}, r_{-,2}, \cdots, \}$ とし，行動経験サブ集合 $\mathcal{B}_{j,k}$ における直接獲得報酬に関するエントロピーは，

$$H_{\mathcal{B}_{j,k}}(Rwd) = -\sum_{r \in \{r_{+,1}, \cdots r_{-,1}, \cdots\}} P(r)\log_2 P(r) \tag{5.8}$$

によって計算されるが，より一般的に，報酬値が連続値を取ったり多次元ベクトルで表される場合への拡張も可能である．

5.5.3 到達した状態クラス

行為結果ベクトルに含まれるのが直接獲得報酬だけだと，直接的に報酬に到達するアクションやその直前の状態だけが一般化され，それ以外の状態やアクションは一般化されない．そこで，アクションの実行によりどの状態クラスに遷移するかという情報 c_{post} もエージェントの

[1] この場合，連続スカラー量をどのように離散化すべきかという新たな問題が生じる

行為結果ベクトルに含めることによって，直接は報酬を獲得しないがその1つ前の状態に遷移できる状態，すなわちサブゴール状態やそれをもたらすアクションも一般化されるようにする．先のモンキーバナナの問題を例とすれば，"バナナがぶら下がっている下に椅子を運ぶ" とか，"椅子のそばにいる" というそれ自身が直接はバナナの獲得に即時結び付かないが，それに近付くようなアクションや状態が定義されることになる．

この C_{post} は離散的な値を取る行為結果属性なので，

$$H_{\mathcal{B}_{j,k}}(C_{post}) = -\sum_{c \in \mathcal{C}} P(c) \log_2 P(c) \tag{5.9}$$

によって，行動結果サブ集合 $\mathcal{B}_{j,k}$ における情報エントロピーを計算する．

5.6 状態・行為クラスの表現法

本章では反射的エージェントの状態・行為抽象化問題における2つの問題のうち，"抽象化基準の問題" に専ら焦点を当てており，"表現・一般化の問題" についての議論は4章に譲っている．実際，5.2節で述べた問題の枠組自体は状態クラスやアクションクラスをどのように一般化し表現するかという問題には依存しておらず，問題の性質に合わせてどのような表現法を用いても良い．

本章では，決定木を用いて一般化された状態・行為クラスを表現する方法と，単純ベイズ分類器を用いた具体的な方法を後に示す．4章で示したように冗長なセンサー入力が得られる場合のノイズやフォールトに対するロバスト性という観点からは単純ベイズ分類器を用いた表現法の方が望ましいと考えられるが，5.2節で定義した問題は各インスタンスにどの状態・アクションクラスに属すべきかというラベルがついているようないわゆる**教師つき学習**の問題ではなく，全体の重みつきエントロピー和を最小にするようなセンサー入力空間，モーター出力空間の分割を求めるという問題であるため，アルゴリズム的には決定木を用いた方法の方がシンプルになる．つまり，ID3やC4.5などの決定木を用いた教師つき学習アルゴリズムは，もともとインスタンス集合におけるクラス（ラベル）に関する情報エントロピーを最小化することを目的としているので，5.2節で述べた問題への拡張も容易である．すなわち，クラスに関するエントロピーではなくて行為結果属性に関するエントロピーの重み和を最小化するようなアルゴリズムに変更すれば良い．一方，単純ベイズ分類器による概念学習は教師つき学習が基本であるため，そのままではこの問題に適用することは難しい．そこで，5.9.2で提案する手法では教師なし（unsupervised）のクラスタリング手法である **k-means 法**を併用している．

5.7 抽象化された状態・行為の複雑さの指標

5.2節の問題定義では，最終的な状態クラス，行為クラスの個数をあらかじめ指定することを前提としているが，それらを指定せずに**抽象化された状態・行為の複雑さの指標**あるいは表

現コストを表す項 $c(\mathcal{S},\mathcal{A})$ を評価関数に含め，全体の行為結果に関する不確からしさを表す関数 J とのトレードオフによって最適な状態・行為クラス数も自律的に決定する方が望ましい場合が考えられる．このとき，5.2 節の問題定義は次のように修正される．

[**行為結果に関する情報エントロピー最小化に基づく状態・行為の抽象化**（状態・行為クラスの個数を指定しない場合）]
行動経験集合 $\mathcal{B}$ と各結果要素の重み $\boldsymbol{w}$ および状態クラスとアクションクラスの表現形式 Rep と記述コストを表す関数 $c(\mathcal{S},\mathcal{A})$ が与えられたときに，

$$J'(\mathcal{C},\mathcal{A}) = \frac{1}{|\mathcal{B}|}\sum_{j,k}|\mathcal{B}_{j,k}|\cdot J_{\mathcal{B}_{j,k}} + c(\mathcal{S},\mathcal{A}) \tag{5.10}$$

を最小化する状態クラス集合 $\mathcal{S}$ と行為クラス集合 $\mathcal{A}$ を求める．

$c(\mathcal{S},\mathcal{A})$ は状態クラス集合 $\mathcal{S}$ と行為クラス集合 $\mathcal{A}$ 全体の「複雑さ」を表す指標であるが，これは一般に（個々のクラスの記述に要するパラメータの数）×（クラスの数）に比例するものと考えられる．すなわち，$J'(\mathcal{C},\mathcal{A})$ を最小化することは，“できるだけ単純かつ少数の状態クラス集合および行為クラス集合で，できるだけ行為結果に関するばらつきが少なくなるよう”な $\mathcal{C}$ と $\mathcal{A}$ を求めるということになる．

この問題は**帰納的概念学習**や**統計的確率モデル推定**などの分野においてしばしば見られる，“あるデータセットを説明する**仮説モデルの正確性と簡潔さとの間のトレードオフ**”に関する問題とほぼ同じものである．すなわち，ある仮説モデルの善し悪しは，それの元になった観測データといかに無矛盾であるかということだけでなく，そのモデルがいかに簡潔であるかということも含めて評価されるべきであるということであり，**オッカムの剃刀**（**Ockham's laser**）の原理とも呼ばれている．このトレードオフを理論的に扱った，仮説モデルの評価関数としては，**赤池の情報量基準**（**Akaike's Information Criterion : AIC**）や，**最小記述長**（**Minimum Description Length : MDL**）などが考えられている．特に MDL は，ある仮説モデルの評価基準を，“そのモデルを用いてデータ集合（の例外）を記述するのに要する記述長（すなわちモデルの不正確さ）とモデル自体の記述に要するビット数（すなわちモデルの複雑さ）”として定義しており，上に挙げた分野などで盛んに応用されている．

この MDL 理論をほぼそのまま用いて上の抽象化評価関数 $J'(\mathcal{C},\mathcal{A})$ を定義することは，状態・行為クラス集合を決定木を用いて表現した場合でも，単純ベイズ分類木を用いて表現した場合でもさほど難しくない．ある行為結果属性 r_l に関する平均のエントロピー $H(r_l) = \frac{1}{|\mathcal{B}|}\sum_{j,k}|\mathcal{B}_{j,k}|\cdot H_{\mathcal{B}_{j,k}}(r_l)$ は，$\mathcal{B}$ 中の行動経験要素を $\mathcal{C}$ と $\mathcal{A}$ を用いて分類したときの r_l に関する例外を記述するのに要する平均のビット数とみなすことができるからである．したがってこの行動結果属性だけを考えるならば，MDL は，

$$MDL(\mathcal{C},\mathcal{A}|\mathcal{D}_{r_l}) = |\mathcal{B}|\cdot H(r_l) + \frac{1}{2}n_p\log|\mathcal{B}| \tag{5.11}$$

となる [100][76]. ただし n_p は $\mathcal{C}$ と $\mathcal{A}$ を記述するのに要する自由パラメータの数である.

しかし, この MDL を本問題の $J'(\mathcal{C},\mathcal{A})$ として用いるには大きく 2 つの問題が考えられる. まず, 前述のように評価関数 $J(\mathcal{C},\mathcal{A})$ や $J'(\mathcal{C},\mathcal{A})$ の第 1 項は各行為結果属性に関するエントロピー $H(\tau_l)$ の重み和であり, すでに "記述長" としての意味を持っていない. したがって, 状態・行為クラス集合の記述長にあたる $\frac{1}{2}n_p \log|\mathcal{B}|$ に課すべき重みも絶対的には決まらない. もう一つの問題は, MDL に基づいた抽象化基準と, 抽象化の最終的な目的であるエージェントの行動パフォーマンスとの間に明確な関連性がないことである. つまり, $J'(\mathcal{C},\mathcal{A})$ を MDL によって定義し, それを最適化するような状態クラス集合 $\mathcal{C}$ とアクションクラス集合 $\mathcal{A}$ が求まったとしても, それがエージェントの最終的な行動パフォーマンスを最適にするかどうかは全くわからない. 以上の理由から, MDL や AIC などの情報理論的な基準は本問題の抽象化基準における $c(\mathcal{S},\mathcal{A})$ を定める上で重要なヒントは与えるものの, そのままの形では利用できず, 最終的には試行錯誤的に決めざるを得ないと言える.

5.8 状態と行為の同時抽象化

3 章で述べたように, 既存の状態・行為抽象化に関する研究では, あらかじめ定義されたアクションによって同じ行為結果をもたらすようなセンサー入力を同じ状態クラスとして一般化するという方針によって状態空間を自律的に構成するか, 逆に, あらかじめ離散化された状態空間中において同じ状態遷移をもたらしたり, あらかじめ距離尺度を仮定したセンサー入力空間において近いセンサー入力変化をもたらすモーター出力を同じ行為クラスとして一般化するという方針によって行為空間を自律的に構成するかのどちらか一方のみが行われ, 状態空間, 行為空間をともにエージェントの行動経験に基づいて自律的に構成する方法は提案されていない.

これに対し, 5.2 節で提案した**行為結果に関する情報エントロピー最小化に基づく状態・行為の抽象化**の枠組では, ある状態クラス集合 $\mathcal{C}$ とあるアクションクラス集合 $\mathcal{A}$ に対するグローバルな評価関数を定義し, これを最小化する $\mathcal{C}$ と $\mathcal{A}$ を見付ける問題として定式化されているので, 両方の問題がごく自然な形で統合されている.

ここで注目すべきことは, この最適化問題において, $\mathcal{C}$ を記述するパラメータ群と $\mathcal{A}$ を表すパラメータ群を合わせて 1 つの探索空間として考える場合には, 図 5.4 のようにセンサー入力空間とモーター出力空間を別々に分割し, 状態クラス集合とアクションクラス集合を定義するという従来の考え方とは異なり, センサー入力空間とモーター出力空間を結合した 1 つの空間を最適に分割する問題になるということである (図 5.5). この立場に基づいた場合, もはや "状態集合" と "アクション集合" が独立に定義されることはなく, "(反射) **行動スキーマクラス集合**" と言うべきものがセンサー入力空間とモーター出力空間を結合した空間を分割することになる. そして, 今まで "状態" と呼んでいたものはこのある行動クラスをセンサー入力空間に射影したものであり, 同様に "アクション" はある行動クラスをモーター出力空間に射影したものであるという解釈になる.

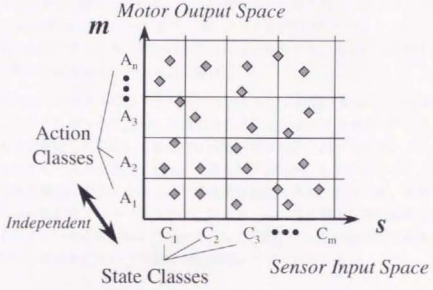


図 5.4: センサー入力空間とモーター空間の独立な分割による状態・行為空間定義

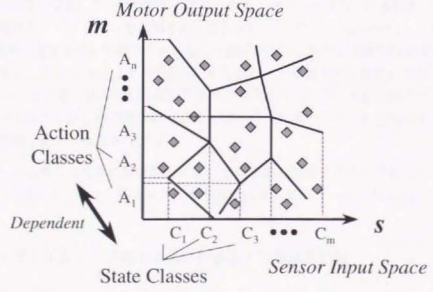


図 5.5: センサー入力空間とモーター空間の非独立な分割

この“状態とアクションは本来不可分であり，独立して定義できるものではない”という主張は，一般的には至極もっともなものであると考えられる．例えば，“ドアノブをまわす”というアクションは，エージェントの目の前にドアがあるという状態において初めて意味を持つものであり，食べ物を目の前にしているという状態において取り得るアクションの選択肢の集合 $\mathcal{A}$ に含めるのは直観的におかしい．また，同じ“ドアを開ける”というアクションでも，そのドアがノブ式のものであったり，押し引き型のものであったり，あるいは自動ドアである場合によって，それに対応するモーター出力は異なる．

しかし，このように状態空間とアクション空間を互いに非独立なものとして扱う立場は，“あるエージェントの状態 $c \in \mathcal{C}$ に対して適切なアクション $a \in \mathcal{A}$ を求める”というように，両者を独立なものと仮定して発展して来た行動政策学習や探索に基づくプランニングの従来研究とは相入れないものである．また，図5.4と図5.5の比較からも分かるように，最適化に要する探索コストも飛躍的に大きくなるという実際的な問題もある．そのため，次節で提案する2つの状態と行為の抽象化アルゴリズムでは，従来の方針通り状態と行為の抽象化を別々に，しかし並行して行うという立場に留め，センサー入力空間とモーター出力空間を統合した空間でのより一般的な行動抽象化という問題は将来課題に残すものとする．

5.9 行為結果のばらつき最小化による状態・行為抽象化のアルゴリズム例

本節では5.2節で定義した“行為結果のばらつき最小化による状態・行為抽象化”を具体的に実現した2種類のアルゴリズムを説明する．これらのアルゴリズムはいずれも，行動経験集合 B，各結果要素に対する重み係数ベクトル $\boldsymbol{w}$，状態・アクションクラス集合の表現形式 Rep，状態クラスの数 n_c，アクションクラスの数 n_a を入力とし，状態クラス集合 $\mathcal{C}$ とアクションクラス集合 $\mathcal{A}$ を出力とする．なお，どちらのアルゴリズムについても，状態空間とアクション空間のどちらか一方はあらかじめ定義されているものとし，もう片方だけを経験に基づいて抽象化するように変更することも容易である．

また，これらはあくまでも状態空間とアクション空間の抽象化だけを目的としており，行動政策学習 − 状態とアクションのマッピング学習は含まれていないということに注意されたい．

5.9.1 アルゴリズム1・分類木を用いた状態・行為抽象化法

このアルゴリズムでは，状態クラスと行為クラスがともに1つである状態から始めて，現在の目的関数の値を最も悪くしている，すなわち行為結果属性に関するエントロピーの重み和が最も大きい領域を分類木（classification tree）によって順々に分割していくことによって，探索範囲を効率的に狭めていく．

状態クラス集合 $\mathcal{C}$，およびアクションクラス集合 $\mathcal{A}$ は，それぞれ各センサー入力あるいは各

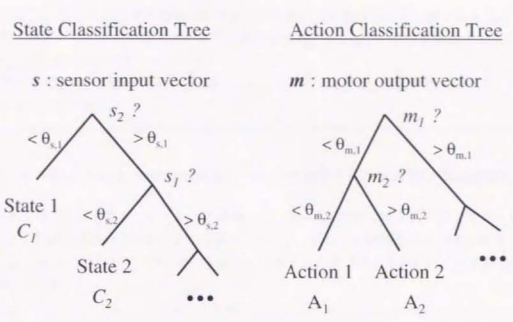


図 5.6: 分類木による状態と行為の表現

モーター出力の値について記述された分類木によって表される（図5.6）．これらをそれぞれ，状態分類木，アクション分類木と呼ぶ．

[分類木を用いた漸次的状態・アクション抽象化アルゴリズム

1. 状態クラス集合 $\mathcal{C}$，アクションクラス集合 $\mathcal{A}$ をそれぞれの要素数が 1 であるとして初期化する．
2. 現在の $\mathcal{C}$ と $\mathcal{A}$ によって $\mathcal{B}$ をクロス分類し，評価関数である行為結果に関する平均エントロピー和 $J(\mathcal{C},\mathcal{A})$ を計算する (その値を $J_{\mathcal{C},\mathcal{A}}$ とする)．
3. $|\mathcal{B}_{j,k}| \cdot J_{\mathcal{B}_{j,k}}$ の値が最も大きな集合 $\mathcal{B}_{j,k}$ を次に分割すべき行為要素集合として選ぶ
4. 選ばれた行動経験サブ集合 $\mathcal{B}_{j,k}$ について，状態とアクションそれぞれに関して次の分割を試みる．
   (a) **状態の分割** ：状態クラス C_j に対応する状態分類木中のリーフ（leaf）について，アクション実行前のセンサー入力ベクトル $\boldsymbol{s}$ のどの要素のどの値で分割したときに最も J が減少するかを調べる．そしてそのときの減少量を $\Delta J_{C,max}$，状態分類木の分割を $Div_{C,max}$ とする．ただし，現在の状態クラス数が n_c 以上の場合は $\Delta J_{C,max} = 0$ とする．
   (b) **行為の分割** ：アクションクラス A_k に対応するアクション分類木中のリーフについて，モーター出力ベクトル $\boldsymbol{m}$ のどの要素のどの値で分割したときに最も J が減少するかを調べる．そしてそのときの減少量を $\Delta J_{A,max}$，アクション分類木の分割を $Div_{A,max}$ とする．ただし，現在のアクションクラス数が n_a 以上の場合は $\Delta J_{A,max} = 0$ とする

$\Delta J_{C,max}$ と $\Delta J_{A,max}$ を比較し，前者が大きい場合は状態分類木を分割 $Div_{C,max}$ によって，後者が大きい場合はアクション分類木を分割 $Div_{A,max}$ によって展開する．そして $\mathcal{B}_{j,k}$ もそれに応じて2つに分割する．

5. 現在の状態クラス数，行為クラス数がともにそれぞれ n_c，n_a 以上であれば終了．そうでなければ2に戻る．

5.9.2 アルゴリズム2・k-means 法とベイズ分類器を用いた状態・行為抽象化法

このアルゴリズムもアルゴリズム1と同様，センサー入力空間とモーター出力空間を徐々に分割しながら評価関数の（局所）最小化を行っていく．また，分割の際には 行動経験インスタンスを k-means 法によってクラスタリングし，それぞれのクラスタを単純ベイズ分類器によって表現している．

[k-means 法とベイズ分類器を用いた漸次的状態・アクション抽象化アルゴリズム

1. 状態クラス集合 $\mathcal{C}$，アクションクラス集合 $\mathcal{A}$ をそれぞれ要素数が1であるとして初期化．
2. 現在の $\mathcal{C}$ と $\mathcal{A}$ によって $\mathcal{B}$ を分類し，評価関数 $J(\mathcal{C},\mathcal{A})$ を計算する．
3. $|\mathcal{B}_{j,k}| \cdot J_{\mathcal{B}_{j,k}}$ の値が最も大きな集合 $\mathcal{B}_{j,k}$ を次に分割すべき行為要素集合として選ぶ
4. 選ばれた $\mathcal{B}_{j,k}$ の要素を k-menas 法によって2つのクラスターに分割する．ただし，k-means 法で用いる（インスタンスとクラスターとの間の）距離尺度として，

$$d(\boldsymbol{r}_i, \boldsymbol{c}_m) = \sum_l \frac{w_l}{\sigma_l^2}(r_{il} - c_{il})^2$$

を用いる．このクラスタリングの結果，$\mathcal{B}_{j,k}$ が $\mathcal{B}_{j,k,1}$ と $\mathcal{B}_{j,k,2}$ に分かれたとする．

5. $\mathcal{B}_{j,k,1}$ と $\mathcal{B}_{j,k,2}$ それぞれの要素のセンサー入力ベクトル $\boldsymbol{s}$，モーター出力ベクトル $\boldsymbol{m}$ を Naive Bayesian Classifier によって一般化し，暫定的な新しい状態クラス $C_{j,1}$，$C_{j,2}$，アクションクラス $A_{k,1}$，$A_{k,2}$ を定める．これに伴い，暫定的な状態クラス集合 $\mathcal{C}' = \mathcal{C} - \{C_j\} + \{C_{j,1}, C_{j,2}\}$ とアクション集合 $\mathcal{A}' = \mathcal{A} - \{A_k\} + \{A_{k,1}, A_{k,2}\}$ を定める．
6. $\Delta J_C = J(\mathcal{C},\mathcal{A}) - J(\mathcal{C}',\mathcal{A})$，$\Delta J_A = J(\mathcal{C},\mathcal{A}) - J(\mathcal{C},\mathcal{A}')$ を求める．ただし，現在の状態クラス数が n_c 以上の場合は $\Delta J_C = 0$，アクションクラス数が n_a 以上の場合は $\Delta J_A = 0$ とする．
7. これらの値の大小によって分岐．
   (a) $\Delta J_C > \Delta J_A$ の場合 ：$\mathcal{C}' \to \mathcal{C}$ とする
   (b) $\Delta J_C < \Delta J_A$ の場合 ：$\mathcal{A}' \to \mathcal{A}$ とする
8. 現在の状態クラス数，行為クラス数がともにそれぞれ n_c，n_a 以上であれば終了．そうでなければ2に戻る．

5.10 行動政策学習との統合

前節で示したアルゴリズムは，行動経験のデータ集合$\mathcal{B}$を入力として，行為結果の重みつきエントロピー和を局所最小化する状態クラス集合$\mathcal{C}$とアクションクラス集合$\mathcal{A}$を求める過程のみに関するものであったが，本来の状態抽象化およびアクション抽象化の一つの目的は，報酬に基づく強化学習などの行動政策学習にとって最適な状態空間とアクション空間を提供することである．

この両方の学習過程を統合した最も単純な形態は，図5.7のように，まず最初にエージェントにランダムにアクションを実行させることによって行動経験集合$\mathcal{B}$を得，次に前節で提案した状態／行為抽象化法によって状態空間$\mathcal{C}$とアクション空間$\mathcal{A}$を構成し，最後にその$\mathcal{C}$と$\mathcal{A}$を用いて行動政策を行うというものである．しかし，この単純な学習戦略は2つの理由からあまり現実的でない．まず第一にランダムなアクション（だけ）によって十分な行動経験を集めることは非常に非効率的であり，自律ロボットなどの実際のアプリケーションでそれが可能であることは滅多にないと考えられることである．第二に環境やエージェント自身（のセンサーやアクチュエータ）に変化が生じた場合に，行動政策学習のやり直しだけでは不十分な場合があるということである．

したがって，より現実的なアプローチとしては図5.8のようにランダムアクションによる行動経験収集のフェーズを短くし，その代わりに状態／行為抽象化と行動政策学習とを繰り返し交互に行う方法が考えられる．これによって無駄なランダムアクションを減らすとともに，環境変化に適応するための再学習を状態／行為空間の再構成というレベルから行うことができる．

しかし，このような状態／行為抽象化フェーズと行動政策学習フェーズのループ化によって，両者の干渉という新たな問題が生じる．つまり，行動政策学習は状態空間，アクション空間を介して状態／行為抽象化に依存しているが，状態／行為抽象化もまた抽象化に必要な行動経験集合を介して行動政策学習に依存することになる．したがって，この学習ループの安定性，収束性ということが重要な問題になる．特に，行動政策学習が進むことによって収集される行動経験データには偏り（bias）が生じるが，これが状態／行為抽象化に悪影響を及ぼさないかという懸念がある．

実は，この行動経験データに生じるバイアスは，本章で提案した行為結果のばらつき最小化に基づく状態・行為抽象化法にとってはかえって好都合であると考えられる．というのは，本提案手法では，センサー入力の変化，獲得報酬，到達状態クラスという3種の異なる行為結果属性を考慮しているので，行動政策学習がまだあまり進んでおらずランダムなアクション選択に近い状況では主に**センサー入力変化の類似性と負の報酬獲得に基づく**状態／行為抽象化が行われ，行動政策学習が進んできて正の報酬獲得の頻度が増加してくると，次第に**ゴール／サブゴール到達に基づく**状態／行為抽象化へと移行していくことが予想されるからである．このことは状態・行為抽象化フェーズにおいて，各行為結果属性に対する重みベクトル$\boldsymbol{w}$を意識的に変更しなくても，行動政策学習との連係によって行動経験データ$\mathcal{B}$の性格が変わるために，結果としてデータ**指向型**の状態・行為空間から**ゴール指向型**の状態・行為空間へと徐々に移行し

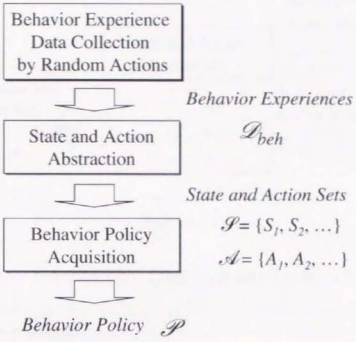


図 5.7: ランダム行動経験データを用いた状態・行為抽象化と行動政策獲得

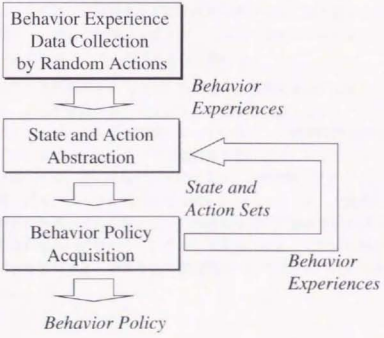


図 5.8: 状態・行為抽象化と行動政策学習の繰り返し学習

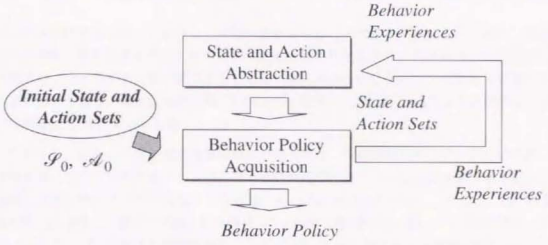


図 5.9: 初期状態・行為空間を利用した状態・行為抽象化と行動政策学習の繰り返し学習

て行くということを意味している.

5.11 初期状態・行為空間の利用による効率化

前節で述べたように状態・行為抽象化と行動政策学習を交互に繰り返すことによって，学習の高速化と環境変化への適応化が期待されるが，それでも最初のランダムなアクション選択による行動経験データの収集はコスト的に大きな問題になる.

そこで，状態・行為空間を全く 0（ゼロ）の状態から構成するのではなく，最初は人間が暫定的に与えてやり，これを経験に基づいて徐々に再構成していくというアプローチが考えられる．すなわち，まずエージェントには人間によって定義された初期状態空間と初期行為空間が与えられ，エージェントはこれらを用いて行動政策学習を行う．そしてこのときに集められた行動経験データに基づいて，先に提案した手法によって状態空間・アクション空間を再構成し，再び行動政策学習に戻る，というものである（図 5.9）．このように，大雑把な初期状態空間と初期行為空間を最初に与え，それを種として行動政策学習と行動抽象化を繰り返すことによって，全く何も定義されてない状態からランダムアクションによって行動経験を集めるよりもはるかに効率的な学習を実現でき，提案手法の実際問題へのアプリケーションを現実的なものにすると考えられる.

5.12 状態・行為空間の再構成時における行動政策学習結果の再利用

前々節および前節で述べたような，状態空間あるいは行為空間の再構成を行う場合，再構成前の状態空間と行為空間を用いて行った行動政策獲得学習の結果を，再構成後の状態空間と行為空間においてそのままの形で転用することは一般にできない．例えば，行動政策学習としてQ学習を用いた場合，異なる状態空間・行為空間の組同士 - $\{\mathcal{C},\mathcal{A}\}$ と $\{\mathcal{C}',\mathcal{A}'\}$ の間では，当然ながら最適なQ値のセットも異なる．

したがって，状態・行為空間が再構成された場合には，行動政策学習もその新しい状態・行為空間を用いて再び行う必要があるが，全く初めからやり直すのは非常に非効率的であり，行動経験の収集自体にコストが発生する実環境においては非現実的である．そこで，古い状態・行為空間 $\{\mathcal{C}',\mathcal{A}'\}$ と，新しい状態・行為空間 $\{\mathcal{C},\mathcal{A}\}$ との間の関係を何らかの方法で導き，それに基づいて $\{\mathcal{C}',\mathcal{A}'\}$ における行動政策を $\{\mathcal{C},\mathcal{A}\}$ において再利用することが考えられる．

このような行動政策学習結果の再利用の具体例としては，[77] で提案されている方法がある．この方法では，新しい状態集合（もとの状態空間を細分化することによって得られる）中の各状態が，もとの状態集合中のそれらにどの程度近いかを，行動経験集合中のセンサー入力ベクトルが両者においてどれだけ共有されているか，ということに基づいて計算する．そしてこの類似度の値を重みとして，もと状態集合の各状態についてのQ値を，新しい状態空間の各状態に再割当する．本研究ではこの手法を，状態空間だけでなく行為空間（行為集合）も再構成された場合にも利用できるように拡張して用いる．

あるエージェントの再構成前の状態空間および行為空間をそれぞれ，
$\mathcal{C}^{old}=\{C_1^{old},C_2^{old},\cdots,C_{n_{c,old}}^{old}\}$，$\mathcal{A}^{old}=\{A_1^{old},A_2^{old},\cdots,A_{n_{a,old}}^{old}\}$
とし，再構成後の状態空間および行為空間を同様にそれぞれ，
$\mathcal{C}^{new}=\{C_1^{new},C_2^{new},\cdots,C_{n_{c,new}}^{new}\}$，$\mathcal{A}^{new}=\{A_1^{new},A_2^{new},\cdots,A_{n_{a,new}}^{new}\}$
とする．そして，既にQ学習によって獲得されている $\mathcal{C}^{old}$，$\mathcal{A}^{old}$ の各要素 $C_{i'}^{old}$，$A_{j'}^{old}$ についてのQ値を $Q'(C_{i'}^{old},A_{j'}^{old})$ とし，推定すべき $\mathcal{C}^{new}$，$\mathcal{A}^{new}$ の各要素 C_i^{new}，A_j^{new} についての（初期）Q値を $Q(C_i^{new},A_j^{new})$ によって表す．

今，$\mathcal{C}^{new}$，$\mathcal{A}^{new}$ を自律再構成するのに用いられた行動経験集合 $\mathcal{B}=\{beh_1,beh_2,\cdots,beh_{n_b}\}$ を，$\mathcal{C}^{new}$，$\mathcal{A}^{new}$ および $\mathcal{C}^{old}$，$\mathcal{A}^{old}$ で分類することによって，次の条件付き頻度確率を計算する．

$$freq_{\mathcal{B}}(C_{i'}^{old},A_{j'}^{old}|C_i^{new},A_j^{new})=\frac{freq_{\mathcal{B}}(beh\in\{C_{i'}^{old},A_{j'}^{old}\}\wedge beh\in\{C_i^{new},A_j^{new}\})}{freq_{\mathcal{B}}(beh\in\{C_i^{new},A_j^{new}\})} \tag{5.12}$$

これは，新しい状態クラス C_i^{new} において新しい行為クラス A_j^{new} を実行することが，状態クラス $C_{i'}^{old}$ において行為クラス $A_{j'}^{old}$ を実行することと，どの程度類似しているかを表していると言える．そこで，この $freq_{\mathcal{B}}(C_{i'}^{old},A_{j'}^{old}|C_i^{new},A_j^{new})$ を用いて，次のように $Q(C_i^{new},A_j^{new})$ の初期値を計算する．

$$Q(C_i^{new},A_j^{new})=\sum_{i'}^{n_{c,old}}\sum_{j'}^{n_{a,old}}freq_{\mathcal{B}}(C_{i'}^{old},A_{j'}^{old}|C_i^{new},A_j^{new})\cdot Q'(C_{i'}^{old},A_{j'}^{old}) \tag{5.13}$$

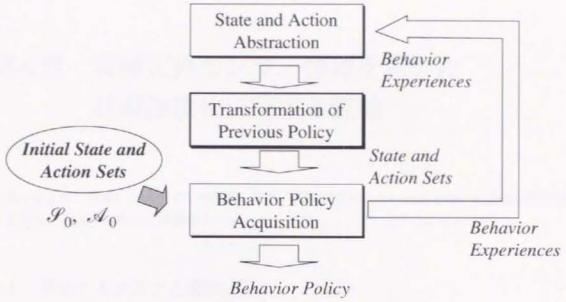


図 5.10: 再構成前の初期状態・行為空間において獲得した行動政策を再利用した場合の状態・行為抽象化と行動政策学習の繰り返し学習

このようにして得られた新しい状態・行為空間における行動政策（Q 値）を初期値として利用し，再び行動政策学習（Q 学習）を行うことによって，Q 値を全く初めから学習しなおす場合に比べて大きなコストの低減が期待される．

第6章　異種冗長センサー情報を用いた状態抽象化に関する実験

本章では4章で提案した「ベイズ分類器に基づく異種冗長センサー情報を用いた状態抽象化法」の有効性を検証するために計算機上で行ったシミュレーション実験の結果を示す．

6.1　想定するタスクと環境

ここでは移動ロボットによる**目標物追従タスク**のシミュレーションを行うことによって4章で提案したベイズ分類器に基づく異種冗長センサー情報を用いた状態抽象化法の有効性を検証する．想定するロボットは図6.1に示す4種類の物理センサーハードウェア，すなわち**機上カメラ**（OC），**接触センサー**（TS），ソナー（SN），**天井カメラ**（CC）を有し，それらから表6.1に示した11個のセンサー入力を得られるものとする．つまり，センサー入力ベクトル $\boldsymbol{s}$ は11次元になる．ここで重要なのは，このセンサー入力ベクトルが異種のセンサーから構成され，しかも互いに近い情報を含んでいるものがあるということ，すなわちセンサー系が**異種冗長**になっているということである．例えば，S_1 の“機上カメラ画像中の目標物の水平位置”と S_{11} の“目標物の方向角に対するロボットの向き”，S_3 の“機上カメラ画像中の目標物の高さ”と S_{10} の“（天井カメラ画像中の）ロボットから目標物への距離”は，それぞれ互いに強く関連した情報を含んでいる．実際，もし必ず正しいセンサー値が観測されるような環境であるならば，天井カメラから得られる S_{10}，S_{11} の2つのセンサー入力だけでもタスクを遂行するのに十分であると考えられる．しかし既に述べたように，この実験では，不確定性や故障の可能性を含んだ環境下においてこれらの異種冗長なセンサー情報が提案手法によってどのように統合され，エージェントの行動にどのような効果をもたらすかを調べるということが主要な目的の一つである．

また，この移動ロボットエージェントが使用できるアクションとしては，図6.2に描かれた8つの移動コマンドを考える．つまり，アクション変数 a は常に8つのコマンド $\{A_1, A_2, \cdots, A_8\}$ のいずれかの値を取る．この実験では状態の自律的抽象化のみを考えるので，これらのアクションはいずれも移動ロボットの左右2つのホイールの回転数比として，事前に定義されているものとする．

このタスクでは，移動ロボットであるエージェントが，目標物（ボール）に正面の向きから到達した場合に**唯一の正の報酬**が与えられるものとする．この唯一の報酬を獲得することはゴー

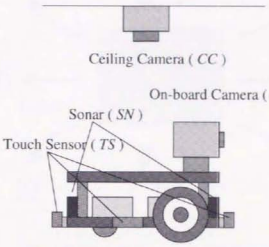


図 6.1: 移動ロボットエージェントが持つセンサー類

表 6.1: 移動ロボットエージェントが利用できるセンサー入力

センサー番号	情報源となるセンサー	センサー情報の内容	連続性
S_1	機上カメラ (OC)	カメラ画像内における目標物の水平位置	連続
S_2		カメラ画像内における目標物の垂直位置	連続
S_3		カメラ画像内における目標物の高さ	連続
S_4-S_7	接触センサ (TS)	物体との接触（左右前後）	離散
S_8-S_9	ソナー (SN)	前後に存在する物体までの距離	連続
S_{10}	天井カメラ (CC)	カメラ画像内におけるロボットと目標物との距離	連続
S_{11}		カメラ画像内におけるロボットの機軸と目標物への方向との相対角	連続

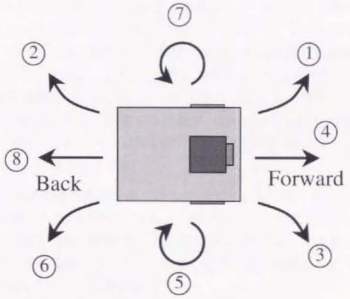


図 6.2: 移動ロボットエージェントが実行可能なアクション

ル状態への到達であると解釈できるので，3.4 節で述べたゴール／サブゴール到達に基づく状態抽象化問題として扱うことができる．よって，結果ベクトル $\boldsymbol{r}$ は**到達した状態**（ゴール状態又は既存状態クラス，およびそれ以外の未定義領域）の識別子を表す 1 次元の離散値変数になる．また，ゴール・サブゴール到達に基づく状態抽象化の場合，前述のように新状態クラス（サブゴール）の生成に伴ってその状態における適切なアクションも同時に学習されるので，報酬に基づく行動政策学習（状態−アクション価値関数の学習）は行う必要がない．したがって，エージェントはすでに抽象化が済んだ状態クラスではそのアクションを実行し，まだ抽象化が行われていない未定義の状態においては行動経験を集めるためにランダムにアクションを選択し実行する．

各センサーの事前確率（密度）分布 $P(S_j)$，$p(S_j)$ については，状態抽象化を始める前にランダムにサンプリングしたこのエージェントの 500 個の状況におけるセンサーデータから k_n-**近傍法**によって推定したものを用いている．また，4.3 節で述べた学習アルゴリズム中のステップ（5）（新状態クラスの生成）において，新しい状態クラスの定義はそのクラスに属するインスタンス数が **100 を超えたとき**に行われるものとする．

6.2 状態抽象化過程の様子

まず，提案する手法によって状態の抽象化が行われていく過程の様子を説明する（図 6.3(a)-(c)，図 6.4）．ここでは上で挙げた 11 のセンサー入力のうち天井カメラから得られる 2 つ（S_{10}，S_{11}）だけを用いるものとする．

図6.3において，エージェントのセンサー入力空間は，はじめは全く抽象化されていない，すなわち1つも状態クラスが定義されていない（**a**）．そこでエージェントはまず，ランダムな行動決定を繰り返して“たまたま”最終ゴール状態に到達するような行動経験を収集し，その経験集合からアクション A_4（前進）によって報酬を獲得する（すなわちゴール状態に到達する）状態を発見し，これを状態クラス C_1 として一般化する（**b**）．その後さらに他のアクションによって既に定義された状態クラスに到達する状態が一般化され，新たな状態クラスが次々に生成されて行く（**c**）．そしてそのような過程が続けられ，最終的にセンサー入力空間全体が定義された状態クラスによって覆われる（**d**）．

このときの**状態クラスの数の変化**を表したものが図6.5であり，エージェントの**行動のパフォーマンス変化**を表したものが図6.6である．ただし，ここでは行動パフォーマンスを“**目標物に到達するまでに要する平均のアクション数の少なさ**”によって定義している．エージェントが新しい状態クラスを獲得していくにつれて，その行動が洗練されゴール到達までに要するアクションの数が減って行くことがわかる．

6.3 実験1：冗長度の異なるセンサー構成同士の比較

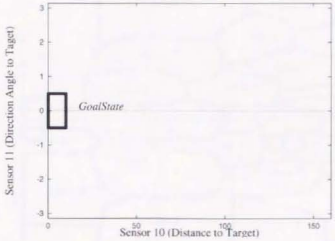
最初の実験としてまず，先に説明した11のセンサー入力の組合せから，5つの異なるセンサー構成を考え，それぞれのセンサー組合せを用いたときのエージェントの学習過程と結果の違いを調べた．想定した5つのケースで使用可能なセンサー群は次の通りである．ここで，下に挙げているものほどセンサー系全体としての冗長性が増すようになっている．

Case 1：*OC*のみ（S_1-S_3）
Case 2：*CC*のみ（S_{10}-S_{11}）
Case 3：*OC*，*TS*，*SN*（S_1-S_9）
Case 4：*OC*，*SN*，*CC*（S_1-S_3，S_8-S_{11}）
Case 5：全てのセンサー（S_1-S_{11}）

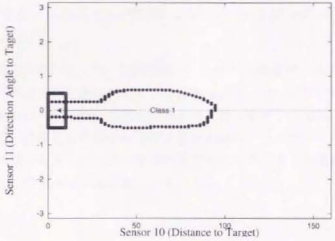
図6.7と図6.8は，学習が進むにつれて，**状態クラスの数**と，**ゴールに到達するまでに要する平均のモーターステップ数**がそれぞれのセンサー構成においてどのように変化していくかを表したものである．

これらのグラフから次の2つのことが観察される．

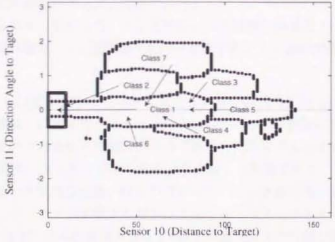
- 学習が完了した後，すなわち状態クラスの数がそれ以上増加しなくなった後の最終的な状態クラス数は，*Case 4*や*Case 5*などのセンサーを多く持つ（すなわち冗長性の高い）構成の方が多く，最終的なパフォーマンスも高い．
- 一方，学習が完了するまでに要する訓練例の数は，*Case 1*や*Case 3*などセンサー数の少ない構成の方が少ない．すなわち，冗長性の低い構成ほど状態空間の構成が早く完了する．



(a) 訓練例 0



(b) 訓練例 2000



(c) 訓練例 5000

図 6.3: 単純ベイズ分類器による状態クラス生成の様子

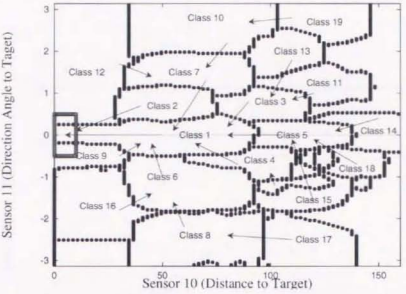


図 6.4: 単純ベイズ分類器によるセンサー空間分割の様子

これらの現象は各構成間でのセンサー空間の広さ，あるいは表現力（representation power）の差に起因するところが大きいと考えられる．すなわち，センサーが少ない場合，探索すべき空間が狭いので全体をカバーするのに必要な状態クラス数は少なくて済み，学習は早く完了するが，状態記述力はその代償として低くなるので最終的なパフォーマンスも劣る．逆にセンサーが多い場合は探索空間が大きくなるので学習が完了するまでに多くの訓練例と状態クラス数を要するが，最終的なパフォーマンスは高くなる．

6.4 実験2：センサー機能低下時における頑強性の比較

次に，実験1で状態抽象化が完了した後のエージェントについて，センサーの機能低下に対する頑強性を調べるテストを行った．ここで想定した3種類の機能低下は，一部のセンサーについて，1）ノイズの増大，2）正常値からの一定のずれ，3）値の欠落 が，生じるというものである．

表6.2は，*Case 1*（*OC*のみ）と *Case 5*（全センサー使用可）の2つのセンサー構成において，これらの機能低下に対するパフォーマンスの低下（ゴール到達に要するステップ数の増加）がどのようなものであるかを比較したものである．同様に表6.3は *Case 2*（*CC*のみ）と *Case 5* において同様のフォールトが生じた場合を比較した結果である．これらの表より，センサー数の多い，すなわち冗長度の高い構成の方がそうでない構成に比べて，同じセンサー機能低下による行動パフォーマンスの低下の度合が小さいということがわかる．例えば，表6.2において機上カメラ（*OC*）のみが使用可能な *Case 1* で機上カメラの機能低下によって著しいパフォーマンスの低下が生じているのに対し，全センサーが使用可能な *Case 5* では，比較的緩

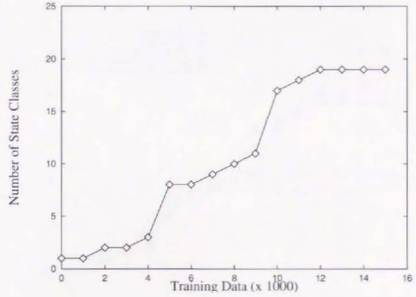


図 6.5: 生成される状態クラス数の変化

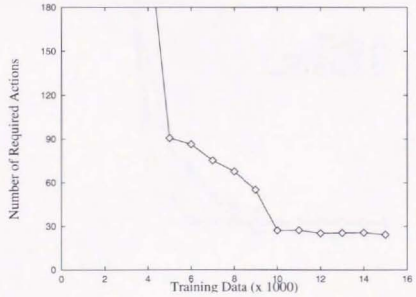


図 6.6: 行動パフォーマンス（ゴール到達に要する平均アクション数）の変化

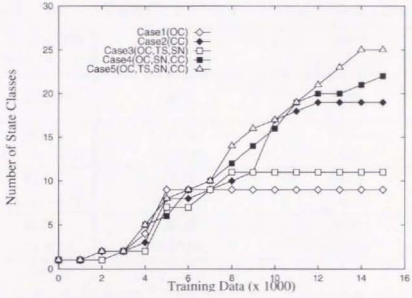


図 6.7: 異なるセンサー構成間の比較（状態クラス数の変化）

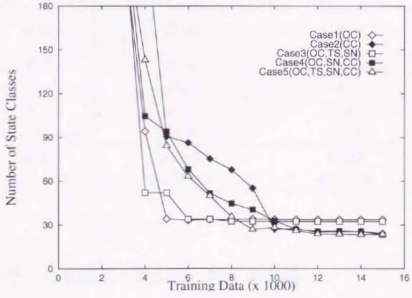


図 6.8: 異なるセンサー構成間の比較（行動パフォーマンスの変化）

やかな低下で済んでいる．これはフォールトによって性能が低下した OC の働きの一部を，扱う情報の性格が近い CC などが“肩代り”しているためであると考えられる．また，表 6.3 で比較している 2 ケースについても同様の現象が観察されている．

表 6.2: センサーの機能低下によるパフォーマンス変化の比較（Case 1,5）

センサー機能 低下パターン	Case 1 ($S_1 \sim S_3$)	Case 5 ($S_1 \sim S_{11}$)
正常時	33.11	23.38
低下 1 ($\sigma_{1,2,3} \times = 4$)	43.17 (+10.06)	27.44 (+4.06)
低下 2 ($S_2 + = 0.4$)	88.70 (+55.59)	33.13 (+9.75)
低下 3 (S_1) N/A	90.49 (+57.38)	25.47 (+2.09)

表 6.3: センサーの機能低下によるパフォーマンス変化の比較（Case 2,5）

センサー機能 低下パターン	Case 2 ($S_{10} \sim S_{11}$)	Case 5 ($S_1 \sim S_{11}$)
正常時	24.26	23.38
低下 4 ($\sigma_{10,11} \times = 4$)	31.18 (+6.92)	27.60 (+4.22)
低下 5 ($S_{11} + = 0.4$)	33.07 (+8.81)	25.67 (+2.29)
低下 6 (S_{11} N/A)	287.06 (+262.80)	65.22 (+41.84)

6.5 センサーの有用度・類似度に関する評価

4.6 節での議論に基づき，ある状態クラス集合 $\mathcal{C}$ に関する**各センサーの重要度** $Imp_{\mathcal{C}}(S_j) = I(S_j;\mathcal{C})$ と，ある 2 つのセンサー間の**類似度** $Sim_{\mathcal{C}}(S_j, S_k) = I(S_j; S_k;\mathcal{C})$ を，この実験における訓練例集合 $\mathcal{D}_{beh}$ について計算した．

図 6.9 は 11 個のセンサー情報について $Imp_{\mathcal{C}}(S_j)$ の大きさを示したものである．これによれば，$Imp_{\mathcal{C}}(S_{10})$ および $Imp_{\mathcal{C}}(S_{11})$ の値が他と比べて大きく，続いて $Imp_{\mathcal{C}}(S_1)$，$Imp_{\mathcal{C}}(S_2)$，

$Imp_C(S_3)$ の値が大きい．このことは天井カメラ（CC）や機上カメラ（OC）から得られるセンサー情報が他のセンサー情報に比べて，状態分類において重要な役割を果たしていることを示しており，実験1の結果と一致する．

図6.10は同じく $Sim_C(S_j, S_k)$ の大きさをマトリクス状にして表したものである．対角成分，すなわち $Sim_C(S_j, S_j)$ はその定義上 $Imp_C(S_j)$ に等しいことに注意されたい．この図から読み取れる顕著な点としては，S_1，S_2，S_3（これらはいずれも機上カメラから得られる），S_7（前方ソナー），そして S_{11}（天井カメラから得られるロボットの向きに対する目標物の方向角）間の類似度の高さである．S_1，S_2，S_3 と S_{11} との間の類似性は，実験2で一方の機能低下による影響をもう一方が抑制するという結果を裏付けている．これに対して，S_1，S_2，S_3 間の類似性が高いことは一見意外のように思われる．しかし，これらのセンサーはいずれも機上カメラが目標物を捉えているときのみそれぞれ値を持ち，そうでないときはそろって値を持たないという点での類似性がこの結果に反映されていると言える．

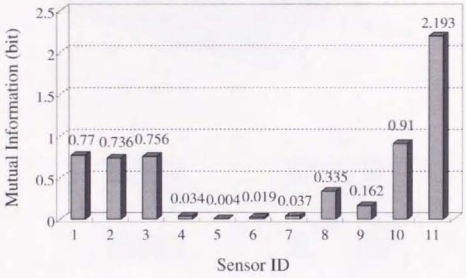


図 6.9: センサーの重要度（$Imp_C(S_j)$）

6.6 実験3：他の状態抽象化法との比較

最後に，3.3節で述べた代表的な状態抽象化手法のうち，**マハラノビス楕円体を用いた手法（MAH）**，および**決定木を用いた手法（DT）**と，**提案手法（SBC）**との比較実験を行った．この実験ではこれら3つの手法が，基本的な学習戦略に関してはいずれも4.3節に従うとし，経験データから状態クラスを一般化する過程，および新しいセンサー入力から状態認識を行う過程に関して，それぞれマハラノビス距離に基づく楕円体領域による表現，情報量ゲインに基づく決定木による表現，単純ベイズ分類器による表現を用いてそれぞれを比較する．ただし，マ

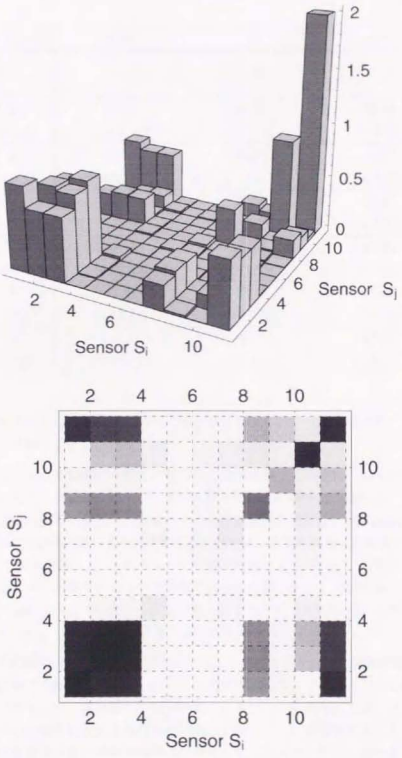


図 6.10: センサー間の類似度（$Sim_C(S_j, S_j)$）

表 6.4: センサーフォールト時のパフォーマンス低下に関する他の状態一般化・表現手法との比較

センサーの状況	MAH	DT	SBC
正常時	27.42	28.29	23.38
低下 1	38.68	28.38	27.44
($\sigma_{1-3}\times=4$)	(+11.26)	(+0.09)	(+4.06)
低下 2	41.80	28.33	33.13
$S_2+=0.4$	(+14.38)	(+0.04)	(+9.75)
低下 3	29.93	31.77	25.47
S_1 N/A	(+2.51)	(+3.48)	(+2.09)
低下 4	59.43	32.84	27.60
($\sigma_{10-11}\times=4$)	(+32.01)	(+4.55)	(+4.22)
低下 5	43.81	42.24	25.67
($S_{11}+=0.4$)	(+16.39)	(+13.95)	(+2.29)
低下 6	117.31	442.90	65.22
(S_{11} N/A))	(+89.89)	(+414.61)	(+41.84)

ハラノビス楕円体を用いた状態クラス表現では，表6.1のセンサー入力のうち，0，1の離散値を取る S_4-S_7 は除いている．

それぞれの手法によって状態の一般化が行われた後，実験2と同様に，いくつかの種類のセンサー機能低下が起こった場合のパフォーマンス低下の様子を調べた（表6.4）．最終的な状態クラスの数およびその獲得に要した経験データの数は，**提案手法**が**25**と約**14000**であったのに対し，**マハラノビス超楕円体による方法**が**19**と**13000**，**決定木を用いた方法**が**15**と**10000**であった．図6.11および図6.12はマハラノビス超楕円体による手法と，決定木による手法を用いた場合に，センサー空間がどのように分割されるかの様子を示した図である．この図では S_{10}（目標物への距離），S_{11}（目標物への方向角に対するロボットの向き）以外のセンサー値は一定値に固定し，S_{10}，S_{11} で構成される平面上の分布を示している．

マハラノビス楕円体を用いた手法については，一部のセンサー値の**誤差の増大**（低下ケース1，4）および**値のずれ**（低下ケース2，5），**値の損失**（低下ケース6）のいずれのフォールトに対してもパフォーマンスの著しい低下が見られる．この原因としては，各状態クラスのセンサー空間中での分布を他次元正規分布で近似的に表すことに無理があること，またその分布の特徴量である分散共分散行列の非対角成分を通じて，あるセンサーにおけるフォールトが他のセンサーの値にも影響を及ぼす、などの理由が考えられる．

一方，**決定木を用いた状態クラス学習・表現**では，一部のセンサー値の**定常的ずれ**（低下ケー

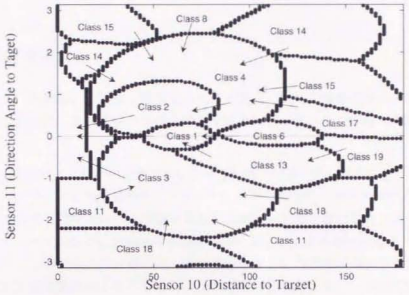


図 6.11: マハラノビス楕円体によるセンサー空間分割の様子

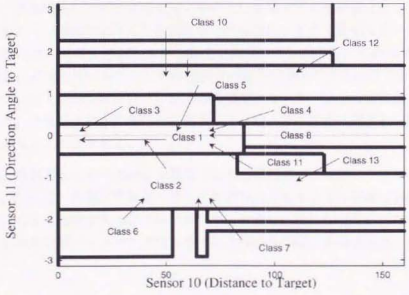


図 6.12: 決定木によるセンサー空間分割の様子

ス5），および**値の損失**（低下ケース6）によって大きな機能低下が生じている．これは決定木による手法がより多くの情報量ゲインが得られる一部のセンサー（この場合は S_{10},S_{11}）を優先してツリーを構成するため，それらのセンサーにフォールトが生じた場合に，クラス分類精度が著しく低下するのだと説明できる．

6.7 結果の考察

本章において，実験1，2では，異なるセンサー構成，特に必要最小限のセンサー入力しか持たないという構成と，互いに似た情報を提供するセンサーをいくつか持つ冗長な構成との比較を行った．

実験1では，全てのセンサーが正常であるという条件下において，提案手法が冗長なセンサー情報を用いてエージェントの最終的なパフォーマンスを向上させる効果を持つことが示された．しかし，その一方で冗長なセンサー情報が存在する場合，そうでない場合と比べてセンサー入力空間全体をカバーするまでに生成される状態クラス数，および必要な経験データ数も増加し，その結果行動パフォーマンスの改善のペースが遅くなるという現象が観察された．この問題については，はじめは少数のセンサーを用いて比較的粗い抽象化を行い，徐々に冗長なセンサーを加えていきながら状態クラス集合を精密化していくことによって，抽象化のスピードと最終的なパフォーマンスを両立させるように抽象化のアルゴリズムを改良することが考えられる．

実験2では，提案手法によって冗長なセンサー入力を用いて状態抽象化がなされた場合，抽象化終了後にセンサー機能が低下した状況においても性能が大きく悪化しないということ，すなわち不確定性に対してエージェントの行動決定がロバストになることが示された．ただし，どのような種類の不確定性に対して，どの程度ロバストになるのかということは，4.6節で見たようにエージェントが持つセンサーの個別の重要度，センサー同士の類似度（冗長度）に深く関わっており，今後この関係に基づいて冗長なセンサー群を最適に選択し，配置するための方法論に発展することが望まれる．

実験3では，提案手法の単純 Bayes 分類器による状態表現が，決定木やマハラノビス楕円体を用いた方法に比べて，異種冗長なセンサー入力が与えられるような状況においては実環境の不確定性に対してより頑強であることが示された．ただし，これらの手法に比べて最終的な状態数が多く，必要な経験インスタンス数も多くなる傾向も同時に観察された．

第7章　行為結果のばらつき最小化による状態・行為抽象化に関する実験

7.1　実験の目的

本章では5章で提案した行動結果のばらつき最小化による状態・行為抽象化法の有効性や特徴を明らかにするために行った計算機シミュレーションによる実験結果を説明する．

この実験において検証すべきことを列挙すると，以下のようになる．

状態・行為抽象化の有効性

提案手法によって行動経験から自律的に抽象化された状態空間，行為空間を用いることによって，人間がヒューリスティックに定義した状態集合・行為集合用いる場合よりも，1）エージェントの（与えられたタスクにおける）行動パフォーマンスが向上するか．あるいは，2）状態数やアクション数が減り，その結果として行動学習や行動決定に要する計算コスト，記憶コストなどが減るか．

従来の状態／行為抽象化法との比較

センサー入力の変化，状態クラス遷移，（正負の）報酬獲得など，性格の異なる複数の行為結果のうち，いずれか一つの類似性（ばらつきの少なさ）だけを考慮して状態あるいはアクションの一般化を行う従来の手法と比較して，それらの行為結果全てを考慮に含める提案手法がどのような性格を持っているか．

行為政策学習と組み合わせた場合の挙動

提案手法による状態・アクションの抽象化と，Q-Learningなどの行動政策学習とを並行あるいは繰り返し適用した場合に，両者がどのように影響し合い，その結果エージェントの状態空間やアクション空間，そして行動政策がどのように変化して行くか．

事前に定義された状態空間・行為空間利用による効果

人間がヒューリスティックに定義した状態集合・アクション集合を用いてまず行動政策学習を行い，その後得られた行動経験に基づいて状態・行為抽象化を行ったとき，それらのヒューリスティックな状態空間・行為空間を用いないで抽象化を行った場合に比べてどのような効果が見られるか．

状態と行為を同時に抽象化したときの効果

従来の研究が状態の抽象化か行為の抽象化のどちらか一方のみを扱っていたのに対し，提

案手法ではそれらを同時に（あるいは交互に）行う枠組を与えているが，実際に2つの抽象化を同時に行うことによってどのような効果が得られるか．

環境変化に対する適応

行動政策学習をやり直すだけでは対応しきれないような環境やエージェント自身の変化に対して，状態集合やアクション集合自体を再構成することによってどのような効果が得られるか．

7.2 想定するエージェントおよびタスク

このシミュレーションでは，移動ロボットによるゴール到達タスク（**タスク1**），および車庫入れタスク（**タスク2**）を考える．この移動ロボットは図7.1のように実験用卓上ロボットKheperaを模擬したものであり，左右2つの車輪の回転数を与えることにより様々な移動コマンドを実現することができる．すなわち，このエージェントのモーター出力ベクトル $\boldsymbol{m}$ は2次元である．一方，このエージェントが利用できるセンサー入力ベクトル $\boldsymbol{s}$ としては，ゴール到達タスクでは**ロボットの重心位置と目標地点との距離**（S_1）と**ロボットの機軸に対する目標地点の相対方向**（S_2）の2つのセンサー入力が得られるものとする（図7.2，表7.1）．また，車庫入れタスクではこれらに加えてロボットのまわりに取り付けられた6個の物体距離センサーの値が得られるものとする（図7.3）．この物体距離センサーは，ある距離・角度範囲内に存在する物体までの距離を返すようになっている．したがって，このエージェントのセンサー入力は8次元となり，表7.2はその内容をまとめたものである．

まず**タスク1**（ゴール到達タスク）では，図7.4に示すように障害物が存在しないような領域[1]において，任意に与えられた初期地点から目標地点に到達することを目的とする．つまり，ロボットの重心位置が目標地点からある一定距離以内に入ったときに**正の報酬**（+10）が与えられ，決められた範囲の外に出てしまった場合に**負の報酬**（-10）が与えられるものとする．

一方**タスク2**は，図7.5に示すように任意に，与えられた初期位置から壁で囲まれた車庫に本体を収めることを目的とする“車庫入れタスク”である．このタスクでは前者のタスクと同様，目標地点からある一定距離以内に入ったときに正の報酬（+100）が与えられ，壁と衝突した場合に負の報酬（-10）が与えられるものとする．

比較的単純なタスクである**タスク1**では，人間による状態空間・行為空間の事前定義，すなわちセンサー入力／モーター出力空間の離散化はさほど困難ではなく，自律的に抽象化を行う意義というものはあまり大きくないが，数値による比較を通して提案手法の端的な性格を明らかにするには適している．一方，**タスク2**はタスク1と比べて難易度が増しただけでなく，センサー入力の数も大きく増えている．そのため，人間が事前に自律エージェントの状態空間を適切に定義することが困難であるようなドメインに対して本提案手法がどのように有効であるかということを示すのに適していると考えられる．

[1]実際には外壁は存在しているものとする

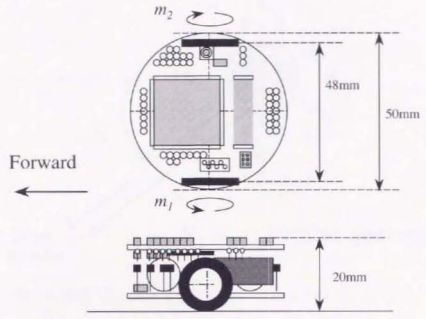


図 7.1: 想定する移動ロボットエージェントの外観

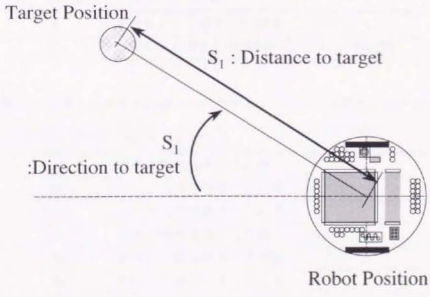


図 7.2: ゴール到達タスクで想定するエージェントのセンサー入力

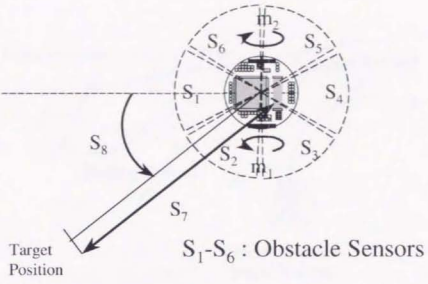


図 7.3: 車庫入れタスクで想定するエージェントのセンサー入力

表 7.1: ゴール到達タスクにおいて想定するエージェントのセンサー入力

センサー番号	センサー入力情報
S_1	ロボットと目標物との距離
S_2	ロボットの機軸と目標物への方向との相対角

表 7.2: 車庫入れタスクにおいて想定するエージェントのセンサー入力

センサー番号	センサー入力情報
S_1	前方の障害物までの距離
S_2	左前方の障害物までの距離
S_3	左後方の障害物までの距離
S_4	後方の障害物までの距離
S_5	右後方の障害物までの距離
S_6	右前方の障害物までの距離
S_7	ロボットと目標物との距離
S_7	ロボットの機軸と目標物への方向との相対角

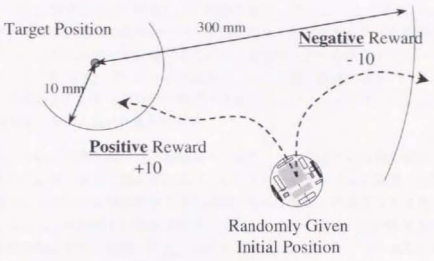


図 7.4: ゴール到達タスクと想定する報酬

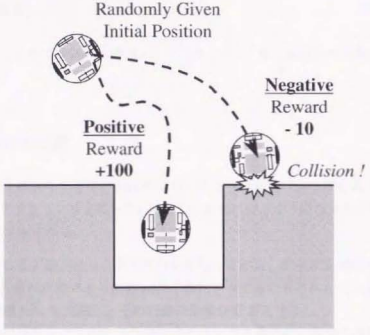


図 7.5: 車庫入れタスクと想定する報酬

より具体的に，タスク1とタスク2とで本質的にどのような違いが存在するのかを列挙すると，次のようになる．

- タスク1では障害物が全く存在しない環境を想定しているため，エージェントがランダムなアクションによって偶然にゴール状態に到達する可能性もある程度高い．それに対してタスク2では目標位置に達するためには障害物（壁）を回避するという行動が必要であり，ランダムなアクションでは滅多にゴール状態に到達できない．この違いは特に状態・行為空間を0から自律的に構成する場合に，ランダムアクションによって始めの行動経験を収集する場合に重要である．
- タスク1では正報酬を受ける目標地点への到達という状態と負報酬を受ける領域からの外出という状態とが遠く離れているのに対して，タスク2では正報酬（目標地点への到達）を受けるためには負報酬を受ける状態（壁への衝突）の極近傍を通過する必要がある．すなわち，正報酬と負報酬が近接している．このことは，正報酬を受ける状態（行為）と負報酬を受ける状態（行為）とを切り分けられないような不適切な状態空間および行為空間では学習が全く不可能になることを意味する．
- タスク1ではセンサー入力空間が2次元であるのに対して，タスク2ではタスクの複雑さを反映して必要なセンサー情報が増え，センサー入力空間は8次元になる．そのためいわゆる状態数爆発が容易に引き起こされ，人間が事前に適切な状態空間を定義することが困難である．

以下，実験1から4まではゴール到達タスク（タスク1）を，実験5では車庫入れタスク（タスク2）を用いる．

7.3 学習過程の概要

エージェントの全体的な学習は，5.10節で述べたように状態（および行為）の抽象化学習と行動政策（状態−アクションマッピング）学習とを交互に行うようになっている．すなわち，この過程は次のように表される．

1. エージェントはまず最初にランダムな行動を取りながら，アクション前のセンサー入力ベクトル $\boldsymbol{s}$，実行されたモーター出力 $\boldsymbol{m}$，それによってもたらされた行為結果ベクトル $\boldsymbol{r}$ から成る行動経験 beh_i を記録し，その行動経験集合を $\mathcal{B}$ とする[2]．
2. 次にこの行動経験集合 $\mathcal{B}$ を用いて**行為結果のばらつき最小化に基づく状態・行為抽象化**を行い，状態クラス集合 $\mathcal{C}$ と行為クラス集合 $\mathcal{A}$ を得る．

[2]ただし，5.11節で述べたように人間がヒューリスティックに与えた初期状態集合と初期行為集合を用いる場合には，ランダムに行動を決めるのではなく，この初期状態空間と初期行為空間を用いて行動政策学習を行いながら beh_i を記録する．

表 7.3: エージェントの行為結果ベクトル $\boldsymbol{w}$ の要素

行為結果要素	重み	内容
$r_1\ (=s_{post,1})$	$w_{s,1}$	行為後のセンサー入力 1（目標物への距離）の値
$r_2\ (=s_{post,2})$	$w_{s,2}$	行為後のセンサー入力 2（目標物への方向各）の値
$r_3\ (=rwd)$	w_r	行為によって獲得した報酬の値
$r_4\ (=c_{post})$	w_c	行為によって到達した状態クラス

3. この $\mathcal{C}$ と $\mathcal{A}$ を用いて **Q-Learning** による行動政策学習を行うとともに，このときの行動経験を新しい $\mathcal{B}$ として記録する．そして 2 に戻る．

この実験では，ステップ 2 において行動経験集合 $\mathcal{B}$ から状態クラス集合 $\mathcal{C}$ と行為クラス集合 $\mathcal{A}$ を得るのに，5.9 節で示した 2 つの具体的手法のうちの**アルゴリズム 2 ：決定木を用いた状態・行為抽象化法**を用いる．すなわち一般化された状態集合，アクション集合はプリミティブなセンサー入力，あるいはモーター出力に関する分類木によって記述される（図 5.6）．

また，5.3 節で述べたように提案手法では，エージェントの様々な種類の**行為結果要素に関するばらつき**（エントロピーの加重和）が状態クラス集合 $\mathcal{C}$ と行為クラス集合 $\mathcal{A}$ によってどれだけ減るかということを抽象化結果の評価基準としているが，この実験ではそのようような行為結果要素として，**アクション後のセンサー入力** $\boldsymbol{s}_{post}=[s_{post,1}, s_{post,2}]$，**直接獲得された報酬の値** rwd，および，**到達した状態クラス** c_{post} を考慮する（表 7.3）．すなわち，行為結果ベクトル $\boldsymbol{r}$ は，

$$\boldsymbol{r}=[\boldsymbol{s}_{post}, rwd, c_{post}]=[s_{post,1}, s_{post,2}, rwd, c_{post}] \tag{7.1}$$

と表され，これらの行為結果に対応する**重みベクトル** $\boldsymbol{w}$ は，

$$\boldsymbol{w}=[\boldsymbol{w}_s, w_r, w_c]=[w_{s,1}, w_{s,2}, w_r, w_c] \tag{7.2}$$

と表される．

ステップ 3 では，抽象化された状態空間 $\mathcal{C}$ と行為空間 $\mathcal{A}$ を用いて行動政策学習，すなわち $\mathcal{C}$ と $\mathcal{A}$ との間のマッピングの学習を行うのに，Q-Learning [82] を用いている．すなわち，エージェントはある時点 t で状態クラス $c\in\mathcal{C}$ においてアクション $a\in\mathcal{A}$ を選択実行する度に，次式によって状態 c におけるアクション a の価値関数 $Q_t(c,a)$ を更新する．

$$Q_t(c,a)=(1-\alpha)Q_{t-1}(c,a)+\alpha(r+\gamma\max_{a'\in\mathcal{A}}Q_{t-1}(c',a')) \tag{7.3}$$

ここで c' はこのアクションによってエージェントが遷移した次状態，r は直接獲得した（正負の）報酬値である．また，α は学習率，γ は減衰率である．

一方，各時点でのアクション決定は，現在の Q 値をもとに，Boltzmann 分布に基づく確率的なアクション選択法によって行う．つまり，この状態においてあるアクション a を選択する確

表 7.4: 実験1－aの設定

実験ケース	状態空間	行為空間	抽象化基準
Case 0	固定（格子状）	固定（格子状）	－

率を，

$$P(a|c) = \frac{exp(Q_t(c,a)/T)}{\sum_{a' \in A} exp(Q_t(c,a')/T)} \tag{7.4}$$

によって定義し，ルーレット戦略によりアクション a を決定する．ここで T は温度係数である．

7.4 実験1：ゴール到達タスクにおける状態空間構成

最初の実験として，ロボットのアクション集合 A についてはあらかじめ定義されているという状況のもとで，状態空間 $\mathcal{C}$ のみを抽象化する実験を行った．

ここでは，モーター空間は図7.6のようにあらかじめ格子上に9つに分割されており，それぞれの領域の中心点にあたるモーター出力ベクトルによって各アクションを実現されるものとする．すなわち，9つのアクションとそれを実現するモーター出力ベクトル $\boldsymbol{m} = [m_l, m_r]$ は具体的に，後退（A_1，$[-667, -667]$），右後退（A_2，$[-667, 0]$），左回転（A_3，$[-667, 667]$），左後退（A_4，$[0, -667]$），静止（A_5，$[0, 0]$），左前進（A_6，$[0, 667]$），右回転（A_7，$[667, -667]$），右前進（A_8，$[667, 0]$），前進（A_9，$[667, 667]$）である．

7.4.1 実験 1－a：事前定義された固定状態空間を用いた場合

まず，状態空間 $\mathcal{C}$ についても人間が事前に定義したものを用いて行動学習を行った場合（[**Case 0**]）の様子を調べた．すなわち，2つのセンサー入力 S_1（目標地点までの距離）と S_2（目標地点の方向）をそれぞれ等間隔に区切り，図7.7に示したような格子状に分割することによって定義された状態空間を用いる．ここで状態数は **30** である．図7.8および図7.9は，学習パラメータを $\alpha = 0.1$，$\gamma = 0.5$，$T = 0.2$ として，前述の Q-Learning とアクション選択を適用したときの**タスク成功率**（60ステップ以内に目標地点に到達することができた割合），および**ゴール到達に要する平均アクション数**[3]の変化を示したものである[4]．ここで横軸はタスク試行回数を表している．

このグラフによれば，試行回数150回程度で成功率が約60%に達するが，その後はいくら訓練を重ねても，ほとんどそれ以上成功率が向上しないことがわかる．この原因としては，

[3]ゴールに到達できなかった場合は60ステップとして計算している．
[4]同じ実験を10回行い，その平均を取っている．

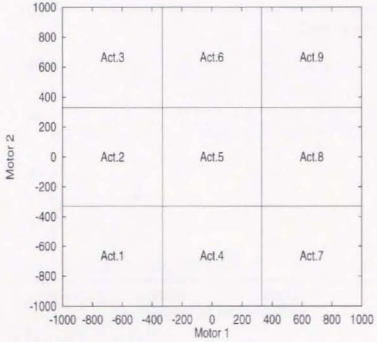


図 7.6: モーター出力空間を格子状に分割して定義された固定行為空間

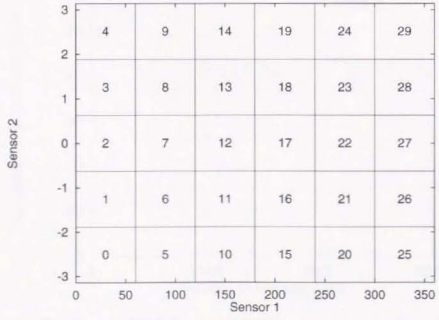


図 7.7: センサー入力空間を格子状に分割して定義された固定状態空間

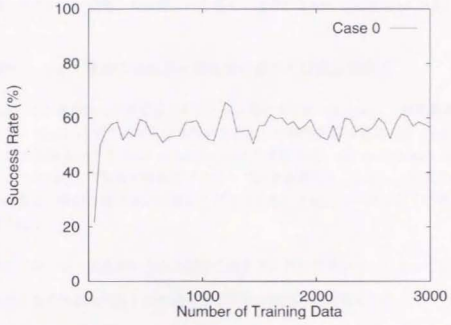


図 7.8: 格子状状態空間・行為空間を用いてQ学習を行った場合のタスク成功率変化

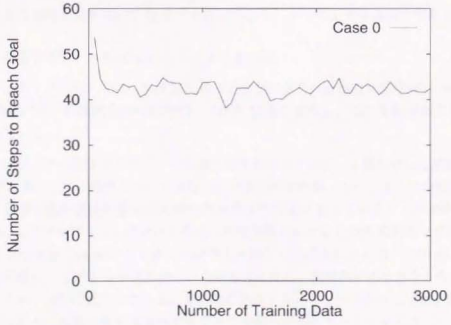


図 7.9: 格子状状態空間・行為空間を用いてQ学習を行った場合のゴール達成に要する平均アクション数の変化

1）定義した状態空間 $\mathcal{C}$ が適切でない 2）固定されたアクション空間 A が適切でない という2つの可能性が考えられる．以下の状態抽象化，および次節のアクション抽象化実験では，この状態空間，アクション空間とも固定された場合の結果が1つの比較対象 to なる．

7.4.2 実験 1－b：複数行為結果の類似性に基づく状態空間構成

次に，前述した3種の行為結果要素（アクション後のセンサー入力 s_{post}，直接獲得報酬 rwd，到達状態クラス c_{post}）の全てのばらつき具合を考慮して状態空間の構成を行った．すなわち，行為結果に対する重みベクトル $\boldsymbol{w} = [w_s, w_r, w_c]$ の各成分を，$w_s = [1.0, 2.0]$，$w_r = 20.0$，$w_c = 0.1$ のように設定して前述の抽象化アルゴリズムを適用する．ただし，生成される状態クラスの最大個数は，前述の格子状状態集合と同じく30とする．エージェントの学習過程は次のように表される．

（1）まずエージェントは最初の600回の行動をランダムアクションによって行う．

（2）その間に集められた行動経験に基づき1回目の状態空間構成を行う．

（3）得られた状態空間を用いてQ学習を行いながら，さらに600回の行動を行う．

（4）（3）によって集められた行動経験に基づき，2回目の状態空間構成（再構成）を行う．

（5）得られた状態空間を用いてQ学習を行いながら，さらに1800回の行動を行う．

このときの学習戦略ケースを [**Case 1**] とする（表7.5）．

ただし，ステップ（5）では，5.12節で述べた方法により，最初の状態空間について学習されたQ値の配列を，再構成後の状態空間におけるQ値に変換し，これを初期値として再びQ学習を行っている．

このときのタスク成功率およびゴール到達に要する平均アクション数を示したが図7.10および図7.11である．これらの図より，2回目の状態空間構成の後，エージェントの行動パフォーマンスが，事前定義の固定状態空間を用いた場合よりも良くなっていることがわかる．また，これらの図にはステップ（5）において最初の状態空間におけるQ値を再利用せずにQ学習を初めから行った場合（*Case 1'*とする）の結果も比較として示されている．この *Case 1'* では，状態空間を再構成した後にQ学習を始めからやり直すため，訓練例が約1200のあたりでパフォーマンスが一時的に低下している．最終的に到達するパフォーマンスは *Case 1* も *Case 1'* もほぼ同じであり，行動政策を再利用することの有利さが現れていると言える．

また，図7.12と図7.13は，それぞれ1回目と2回目の状態抽象化の後の状態空間の様子を表したものである．これによると，1回目の方は状態クラスが比較的センサー入力空間全体において均等に分布しているのに対し，2回目ではゴール状態（正報酬）の周辺，すなわちセンサー1の値が0に近い領域ほど，状態クラスが細かく分布していることがわかる．これは，は

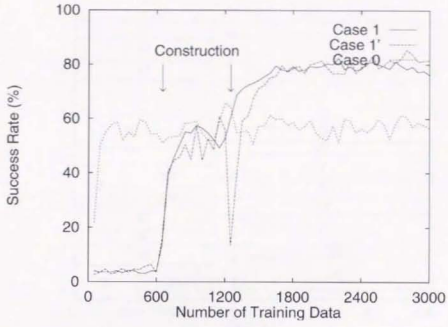


図 7.10: 格子状行為空間を用いて状態空間を0から自律構成した場合のタスク成功率変化

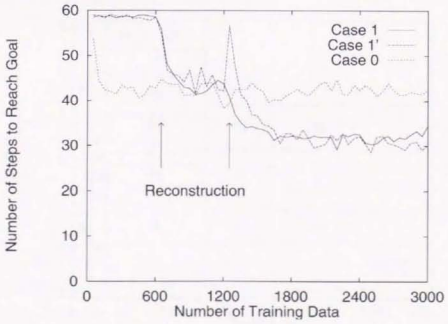


図 7.11: 格子状行為空間を用いて状態空間を0から自律構成した場合のゴール到達に要する平均アクション数変化

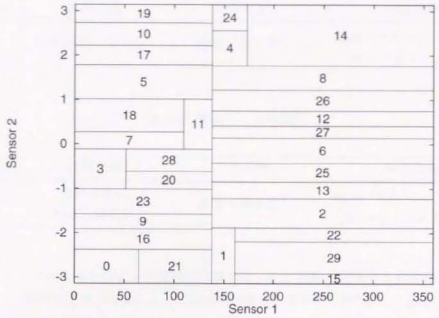


図 7.12: 実験1－b 1回目の状態空間構成

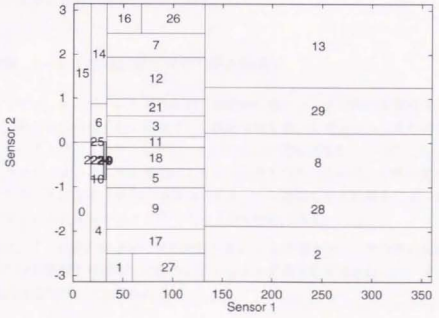


図 7.13: 実験1－b 2回目の状態空間構成

表 7.5: 実験1－bの設定

実験ケース	状態空間	行為空間	抽象化基準
Case 1	自律構成（0から）	固定（格子状）	s_{post}, rwd, c_{post} のばらつき $\boldsymbol{w} = [1.0, 2.0, 20.0, 0.1]$

表 7.6: 実験1－cの設定

実験ケース	状態空間	行為空間	抽象化基準
Case 2	自律再構成 （初期空間利用）	固定（格子状）	s_{post}, rwd, c_{post} のばらつき $\boldsymbol{w} = [1.0, 2.0, 20.0, 0.1]$

じめのランダムアクションによる行動経験収集では，正の報酬を獲得する経験数が非常に少ないため，主に“特定のアクションによって負報酬を獲得するかどうか”という基準と，“アクション後のセンサー入力が互いに近いか”という基準にしたがって状態空間の構成が行われるのに対し，2回目ではその前の行動経験において正報酬を獲得する訓練例が多く得られているため“特定のアクションによって正報酬を獲得するかどうか”という基準にしたがって状態分節が行われる傾向が強まるためであると考えられる．

7.4.3 実験 1－c：初期状態空間の利用と再構成

実験1－bでは，エージェントが過去の行動経験に基づいて自律的に状態空間を構成することによって，あらかじめ格子状に分割された状態空間を用いた場合よりも最終的なパフォーマンスが良くなることが示された．しかし，この方法では状態空間を0（ゼロ）から作るために，最初は全くランダムなアクション選択により行動経験を収集するので，その分だけ多くの訓練例を必要とすることになる．また，実際のロボットへの適用を考えた場合には，そのようなランダムアクションは大きなリスクを伴うという点も問題である．

そこで次に，まず前述の格子状の状態空間を用いてQ学習を行い，その間に記録された行動経験を用いて状態空間を再構成した場合にどうなるかを調べる実験を行った．すなわち，このときの学習過程は次のように表される．

（1）格子状状態空間を用いてQ学習を行いながら経験例を収集．

（2）集められた行動経験データに基づき状態空間構成．

（3）再構成された状態空間を用いてQ学習を行う．

表 7.7: 実験1－dの設定

実験ケース	状態空間	行為空間	抽象化基準
Case 3-a	自律再構成 （初期空間利用）	固定（格子状）	rwd, c_{post} のばらつき $\boldsymbol{w} = [0.0, 0.0, 20.0, 0.5]$
Case 3-b	自律再構成 （初期空間利用）	固定（格子状）	$\boldsymbol{s}_{post}$, c_{post} のばらつき $\boldsymbol{w} = [1.0, 2.0, 0.0, 0.1]$
Case 3-c	自律再構成 （初期空間利用）	固定（格子状）	$\boldsymbol{s}_{post}$, rwd のばらつき $\boldsymbol{w} = [1.0, 2.0, 20.0, 0.0]$

これを **Case 2** にとする（表 7.6）．なお，このときの各行為結果要素に対する重みづけは，先と同じものである．つまり，前述の3種の行為結果全てを考慮している．

図 7.14 および図 7.15 は，このときのタスク成功率およびゴール到達に要する平均アクション数の変化を示したものである．これによれば，この学習戦略を用いた場合，状態空間をゼロから構成した場合と比較して，最終的な行動パフォーマンスがほぼ同じレベルに達していることに加えて，学習の初期の段階，特に2回目の状態抽象化が行われる前までの段階（訓練例が1200回以前）において，著しく成績の改善がなされていることが観察される．すなわち，一定のパフォーマンスレベルに達するまでに必要な行動経験数が減っている．また，このときの再構成された状態空間の様子を表したのが図 7.16 であるが，初期状態空間を用いない場合の2回目の状態空間構成後の様子（図 7.13）に類似していることがわかる．

7.4.4 実験 1－d：行為結果への重みの違いによる影響

次に，この実験で抽象化基準として考慮されている3種の行為結果属性，アクション後のセンサー入力 $\boldsymbol{s}_{post} = [s_{post,1}, s_{post,2}]$，直接獲得された報酬の値 rwd，および，到達した状態クラス c_{post} それぞれのばらつき具合が抽象化結果にどのような影響を及ぼすかを調べるための実験を行った．ここで状態抽象化と行動政策学習の過程は前の実験と同じであり，行為結果に対する重みベクトル $\boldsymbol{w}$ の成分の値だけを変えて比較を行った．すなわち，

- 直接獲得報酬と到達状態クラスのみを考える場合（[**Case 3-a**]，$\boldsymbol{w} = [0.0, 0.0, 20.0, 0.5]$）
- 行為後のセンサー入力と到達状態クラスのみを考える場合（[**Case 3-b**]，$\boldsymbol{w} = [1.0, 2.0, 0.0, 0.1]$）
- 行為後のセンサー入力と直接獲得報酬のみを考える場合（[**Case 3-c**]，$\boldsymbol{w} = [1.0, 2.0, 20.0, 0.0]$）

の3通りについて，そのパフォーマンス変化を前述の，固定状態空間を用いた場合（[**Case 0**]），3つの行為結果属性全てを考慮して状態空間の再構成を行った場合（[**Case 2**]）と比較した．

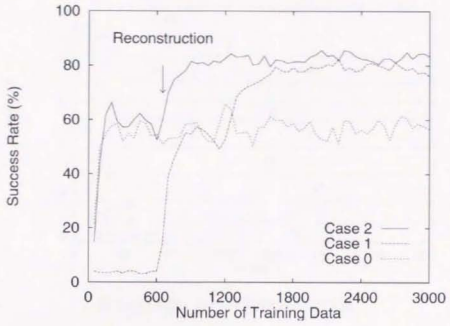


図 7.14: 状態空間を初期状態集合から自律再構成した場合のタスク成功率変化

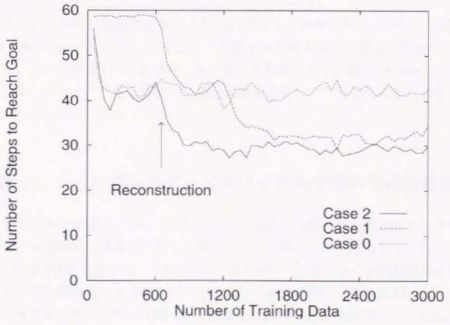


図 7.15: 状態空間を初期状態集合から自律再構成した場合のゴール到達に要する平均アクション数変化

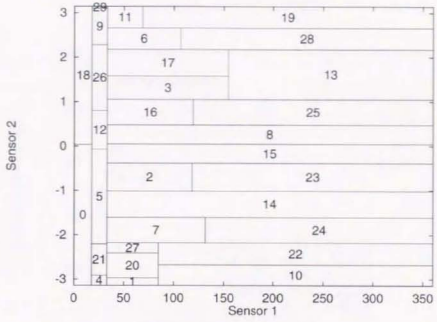


図 7.16: 実験1－c 再構成後の状態空間

図7.17がタスク成功率の変化を示したもの，図7.18がゴール到達に要する平均アクション数の変化を示したものである．

この結果より，3つの行為結果要素のうちのいずれかを抽象化基準から除いた場合（**Case 3-a**，**Case 3-b**，**Case 3-c**）は，いずれも，固定状態空間を用いた場合（**Case 0**）よりは最終的なパフォーマンスが良くなるものの，3つを全て考慮に入れた場合（**Case 2**）には及ばないことがわかる．つまり，異なる種類の行為結果属性のばらつき具合を抽象化基準に含めることの効果が現れているといえる．

7.5 実験2：ゴール到達タスクにおける行為空間の自律的構成

前節ではアクション空間については人間によって事前に定義された固定のものを使い，状態空間についてのみ自律的に抽象化を行った場合について調べたが，本節ではそれと反対に，状態空間については図7.7の格子状に分割されたものを用い，行為空間を行為結果のばらつき最小化に基づき自律的に抽象化する実験を行った．ここでも自律的に定義されるアクションの最大個数は固定行為空間の場合と同じ**9個**とし，抽象化における各行為結果に対する重みは，Case 2と同じ（$\boldsymbol{w} = [1.0, 2.0, 20.0, 0.1]$）に設定してある．

行為抽象化を実現する上での現実的な問題点の1つは，前述の行為抽象化アルゴリズムを適用するためには様々なモーター出力ベクトルを意識的に選択・実行し，その行為結果を記録しなければならないが，完全にランダムにモーター出力を決めてしまうのではQ学習などの行動政

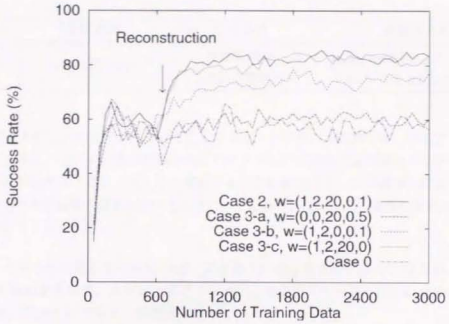


図 7.17: 異なる抽象化基準に従って状態空間を自律再構成した場合のタスク成功率変化の比較

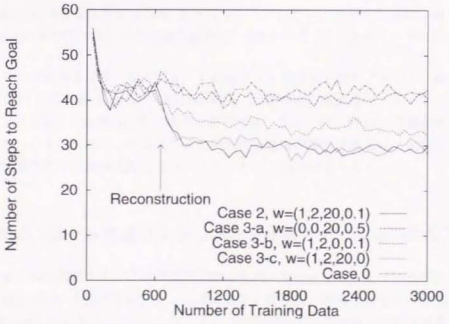


図 7.18: 異なる抽象化基準に従って状態空間を自律再構成した場合のゴール到達に要する平均アクション数変化の比較

表 7.8: 実験2（行為空間自律抽象化）の設定

実験ケース	状態空間	行為空間	抽象化基準
Case 4	固定（格子状）	自律再構成 （初期空間利用）	s_{post}, rwd, c_{post} のばらつき $\boldsymbol{w} = [1.0, 2.0, 20.0, 0.1]$

策学習と両立することができないということである．そこで，本実験では，あるアクションが選択されたときに，モーター出力空間中のこのアクションに該当する領域からランダムにモーター出力を選び出すことによって，Q学習による行動政策獲得と，行為抽象化に必要な様々なモーター出力の実行結果の収集とを両立させた．すなわち，このときの全体の学習過程（[**Case 4**]）は次のように表される．

（1）モーター出力空間を格子状に分割して定義された行為空間を用いてQ学習を行いながら，行動経験を収集．ただし，各アクションに相当するモーター出力はそのアクションの定義領域内からランダムに選択される．

（2）集められた行動経験データに基づきモーター出力空間を分割することによって行為空間を再構成する．

（3）得られた行為空間を用いて再度Q学習を行う．ただし，各アクションに相当するモーター出力はそのアクションの定義領域の中心点を用いる（すなわちランダムにはしない）．

このときのタスク成功率変化，およびゴール到達までに要する平均のアクション数の変化を示したのが図7.19と図7.20である．これによると，行為空間の再構成後，エージェントの行動パフォーマンスが著しく改善されていることがわかる．また，図7.21は，再構成された行為空間の1例を示したものである．行為クラスの非対称な分布が特徴であるが，試行による結果のばらつきが状態空間の自律構成時に比べて大きいという傾向が見られた．

7.6 実験3：ゴール到達タスクにおける状態とアクションの交互抽象化

前2節では，行為空間あるいは状態空間のどちらか一方は固定し，もう一方だけを自律的に抽象化した場合について調べたが，ここでは両者とも自律的に構成した場合の効果について調べた．5.9節では，センサー入力空間とモーター出力空間を，行動経験中の行為結果に関するエントロピーゲインの大きさに従って**同時**に分割していくアルゴリズムを提案した．しかし，本章で考えるタスクでは，センサー入力空間分割によるエントロピーゲインの方がモーター出力空間分割によるそれよりも一般的に大きく，実質的には状態空間を先に自律構成した後に行為空間を自律構成するのと変わりがない．そこで，本実験では以下のように，状態空間構成と行為空間構成を**交互**に行う戦略（[**Case 5**]）を用いた．

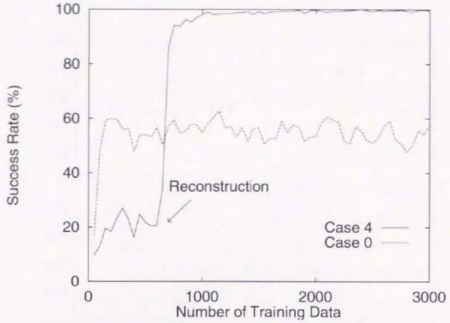


図 7.19: 行為空間を自律再構成した場合のタスク成功率変化の比較

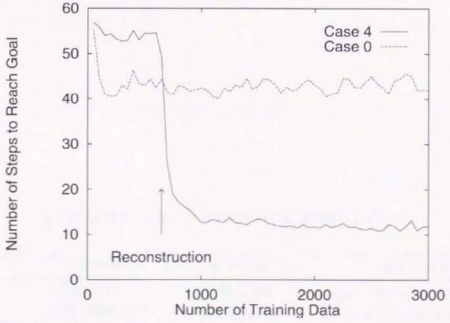


図 7.20: 行為空間を自律再構成した場合のゴール到達に要する平均アクション数変化の比較

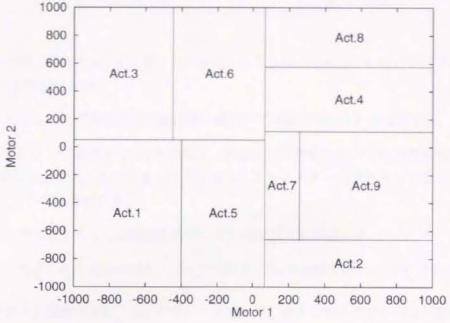


図 7.21: 実験2 再構成後の行為空間

表 7.9: 実験3（状態・行為空間の交互自律抽象化）の設定

実験ケース	状態空間	行為空間	抽象化基準
Case 5	自律再構成 （初期空間利用）	自律再構成 （初期空間利用）	$\boldsymbol{s}_{post}$, rwd, c_{post} のばらつき $\boldsymbol{w} = [1.0, 2.0, 20.0, 0.1]$

表 7.10: 実験4 想定したセンサー故障

センサー	故障の内容
S_1（目標物への距離）	正常値 + 100mm
S_2（目標物への方向角）	正常値 + 1.5rad

（1）状態空間，行為空間ともに格子状に定義されたもの（Case 0）を用いてQ学習を行いながら行動経験を記録する．

（2）（1）によって収集された行動経験に基づいて状態空間のみを再構成する．

（3）（2）によって獲得された状態空間と，格子状行為空間を用いてQ学習を行いながら行動経験を記録する．このとき，前述の方法によりモーター出力を各アクション定義領域内でランダムに選択する．

（4）（3）で収集された行動経験に基づいて行為空間を再構成する．

（5）（2）で得られた状態空間と，（4）で得られた行為空間を用いてQ学習を行う．

このときの**タスク成功率変化**，および**ゴール到達までに要する平均のアクション数の変化**を示したのが図7.22と図7.23である．この結果より，状態空間と行為空間をともに自律的に抽象化した場合，どちらか一方だけを自律構成した場合（状態空間のみ − Case 2，行為空間のみ − Case 4）よりも最終的な行動パフォーマンスが良くなることが示された．また，この実験において自律的に構成された状態空間と行為空間の例を示したのが図7.24と図7.25である．

7.7 実験4：ゴール到達タスクにおけるセンサー故障時における状態空間の再構成

最後に，エージェントが持つセンサーに故障が発生した場合に，状態空間を再構成することによって，パフォーマンスがどのように改善されるかを調べる実験を行った．ここで想定した故障は，S_1（ゴールへの距離）とS_2（ゴールへの相対方向角）の値が，それぞれ正常時よりも100mm，1.5 rad 大きくなる“定常ずれ”である．

図7.26と図7.27は，先のCase 2の実験において得られた状態空間を用いて，この故障が生じた後に（1）**状態空間の再構成を行わずにQ学習による行動政策獲得を行った場合**と，（2）600回の行動経験後，**状態空間を再構成し，さらにQ学習を行った場合**のパフォーマンス変化の違いを示したものである．これによると，センサー故障後に状態空間の再構成を行った方が，若干ながら最終的なパフォーマンスが改善されることが認められた．また，この再構成後の状態空間の様子を示したものが図7.28であるが，これを再構成前の状態空間（図7.16）と比較し

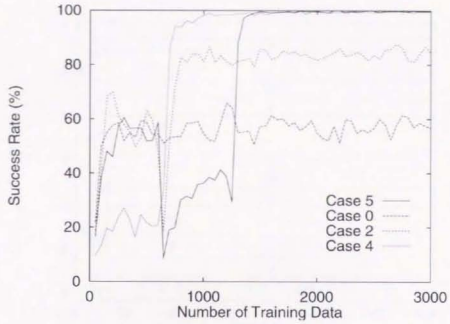


図 7.22: 状態空間と行為空間を交互に自律再構成した場合のタスク成功率変化

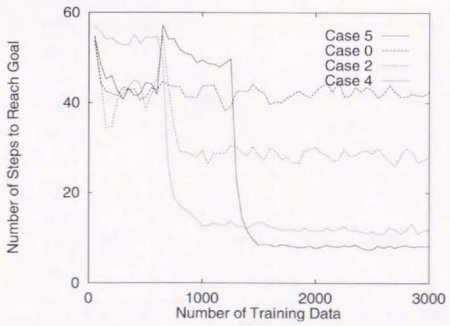


図 7.23: 状態空間と行為空間を交互に自律再構成した場合のゴール到達に要する平均アクション数変化

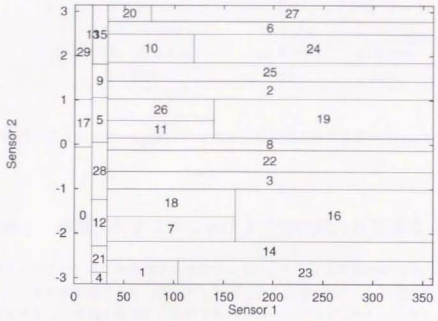


図 7.24: 実験3 再構成後の状態空間

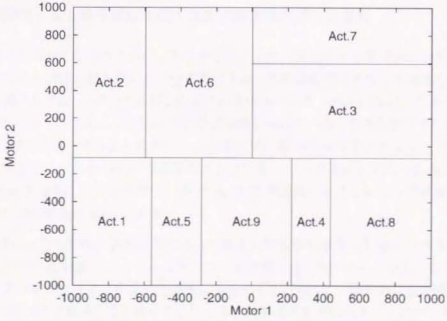


図 7.25: 実験3 再構成後の行為空間

表 7.11: 格子状に定義された状態空間

センサー	*Fix 1*	*Fix 2*
S_1 - S_6（障害物センサー）	分割なし	2 分割（$\theta = 30$）
S_7（目標物への距離）	10 等分割	10 等分割
S_8（目標物への方向角）	20 等分割	10 等分割
全状態数	200	6400

てみると，故障によるセンサー入力のずれを反映した状態空間再構成が行われていることがわかる．

7.8 実験5：車庫入れタスクにおける状態空間の自律構成

上の実験1～4では，ゴール到達タスクをテストベッドに提案手法の有効性がある程度示されたが，タスク自体が簡単であり，またセンサー入力も少ないために，状態空間や行為空間を自律的に構成すること自体のありがたみが少ないという点は否定しがたい．そこで，ここでは先に述べた車庫入れタスクを用いて状態空間の自律構成実験を行った．

7.8.1 実験5－a：格子状に定義された状態空間を用いた場合

まず実験1と同様に比較のために，あらかじめセンサー入力空間を格子状に分割することによって定義された状態空間を用いて（Q学習による）行動政策学習を行った場合を調べた．格子状状態空間としては，表7.11に示したように2つのケース（*Fix-1*,*Fix-2*）を考えた．*Fix-1* では，障害物センサーである S_1-S_6 については分割を行わず，S_7（目標位置までの距離）と S_8（目標位置の相対方向）のみをそれぞれの定義域を10，20等分割することによって状態空間を定義する．すなわち，この場合の全状態数は200である．一方 *Fix-2* では，S_1-S_6 についてはそれぞれ値30を境界にして2分割し，S_7 と S_8 は10等分割することによって状態空間を定義する．よって全状態数は6400である．

Fix-1 と *Fix-2* それぞれの状態空間を用いて車庫入れ行動を学習した場合のタスク成功率の変化を示したのが図7.29である．ここでタスク成功率とは，ランダムに与えられた地点から，負の報酬を受けることなく（すなわち側壁に衝突することなく），最終的に正の報酬を受ける（すなわち目標地点に到達する）率を表している．結果はそれぞれのケースについて20回の試行の平均によって表されている．また，Q学習において用いられたパラメータの値は，$T = 1.0$（温度係数），$\alpha = 0.2$（学習率），$\gamma = 0.5$（割引率）である．

このグラフからわかるように，*Fix-1* では訓練例をいくら増しても全く学習が行われず，タスク成功率はほとんど0のままである．これはこの S_7，S_8 のみによって記述された状態空間が，

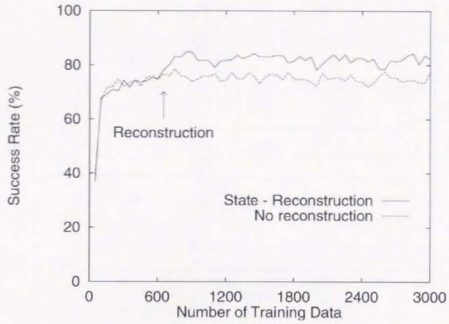


図 7.26: センサー故障時に状態空間を自律再構成した場合のタスク成功率変化

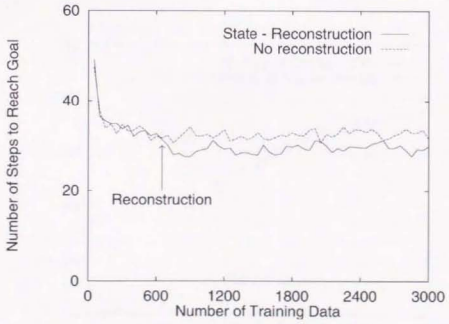


図 7.27: センサー故障時に状態空間を自律再構成した場合のゴール到達に要する平均アクション数変化

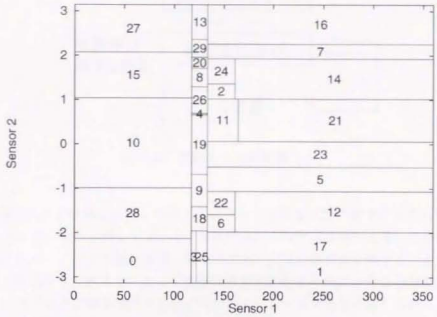


図 7.28: 実験4 センサー故障時における再構成後の状態空間

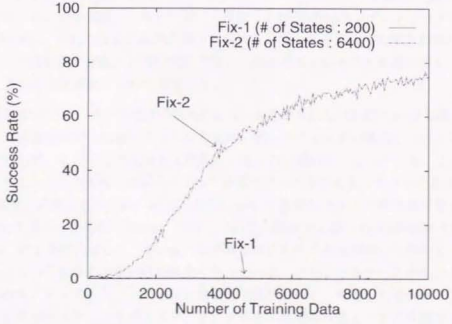


図 7.29: 実験5－a：格子状状態空間を用いた場合のタスク成功率変化

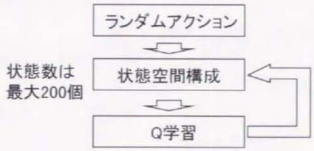


図 7.30: 実験5－bの学習フロー

（ゴール到達タスクとは異なり）このタスクにとって十分な状態の切り分けを行っていないことを意味している．それに対して *Fix-2* の方は徐々にタスク成功率が上昇しており，学習が行われている．しかし，学習曲線は非常に緩やかであり大量の訓練例を要することがわかる．これは *Fix-1* とは異なり，タスクにとって重要な状態を同定するのに十分な分割粒度になっているものの，一方では不必要な細分化も多く行われているためであると考えられる．これは全状態数6400という数字があまり現実的ではないことからも容易に想像がつく．

7.8.2 実験5－b：自律的に状態空間を構成した場合

次に，エージェントが行動結果のばらつき最小化に基づき状態空間を自律的に構成する場合について調べた．ここでは実験1－bと同様に，最初の1000試行はランダムアクションによって行動経験を収集し，それをもとに状態空間を構成する．そしてその状態空間を利用してQ学習を行いながら行動経験を記録し，再び状態空間を再構成するということを繰り返す（図7.30）．このとき，最大の全状態数を200に固定する．

この実験を20回行い，成功率変化の平均をとったものを示したのが図7.31の実線である．これによると，状態空間が1回目に構成された後は，明らかに *Fix-2* の場合と比べて急速に学習が行われているが，その一方で成功率は最高で75%程度で頭打ちになっている．この理由を詳しく調べたところ，この実験では試行によって結果のばらつきが非常に大きいことがわかった．図7.31の破線と点線はそれぞれ，20回の試行において最終的なタスク成功率が最も良かったもの（*Best*）と悪かったもの（*Worst*）を示している．*Best* では約2000の訓練例ですでに成功率が約90%に達するのに対して，*Worst* では何度かはじめのうち成功率が一時的に上がるものの，その後は0のまま全く学習が行われなくなっている．このようなケースが生じる原因としては，最初のランダムアクションによる行動経験収集において，前述した理由から正の報酬を獲得する行動経験がほとんど集められず，またタスク・環境の性格として正報酬の分布と負報酬の分布とが近接しているために，不適切な，すなわち正報酬を受ける状態と負報酬を受ける状態とがうまく切り分けられてないような状態空間が構成されてしまい，その不適切な状態空間を用いてQ学習をいくら行ってもやはり正報酬の獲得経験が得られないという悪循環に陥っ

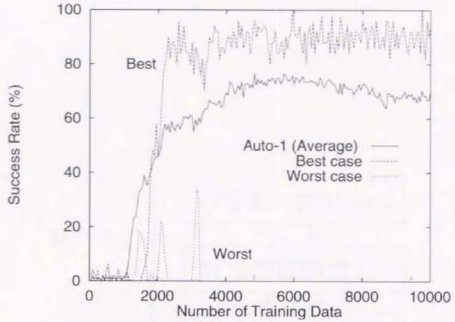


図 7.31: 実験5－b：自律的に状態空間構成を行った場合のタスク成功率変化

表 7.12: 実験5－cにおける3つのタスクレベルとロボットの初期位置範囲

タスクレベル	初期位置・向きの範囲
レベル1	極近傍かつ向きは固定
レベル2	やや近傍から，向きはランダム
レベル3	位置，向きともにランダム

てしまうということが考えられる．

7.8.3 実験5－c：発達的学習アプローチによる改善

実験5－bにおいて失敗ケースが生じた原因が，最初のランダムアクションによる行動経験収集において正報酬獲得経験がほとんど得られないことにあると推察し，人為的に正報酬獲得経験が増えるように学習環境をコントロールすることによって改善を図る実験を行った．

具体的には，エージェントにランダムに与えられる初期位置の範囲を調整することによって，表7.12のように難易度の異なる3つのタスクレベルを用意する．そして図7.32に示すように，まず最も容易なタスクレベルにおいてランダムアクションを行って行動経験を収集して状態空間構成を行い，次にタスクレベルをやや難しくしてQ学習を行った後に再び状態空間の再構成を行い，最終的なタスクレベルに移行していく．

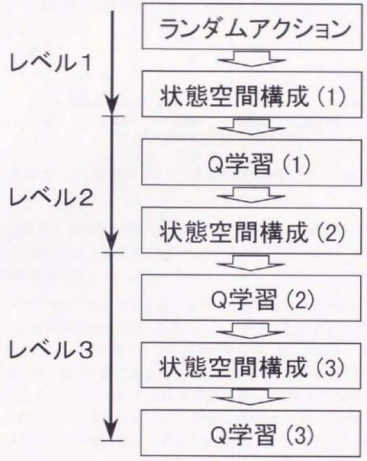


図 7.32: 実験5－c：発達的アプローチを用いた場合の学習フロー

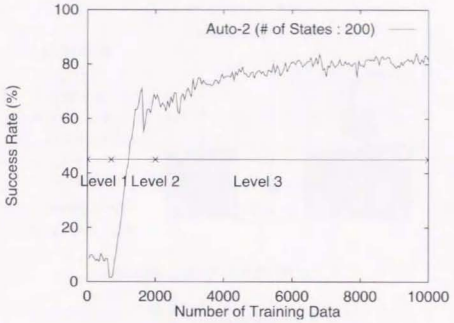


図 7.33: 実験5－c：発達的アプローチを用いた場合のタスク成功率変化

図7.33はこのときのタスク成功率の変化を表したものである．発達的な学習アプローチを採用することによって，前述のような学習不能ケースが発生しにくくなり，実験5－bと比べて平均のタスク成功率が改善されている．

この車庫入れタスクにおいても，提案手法によって（定性的に）どのように状態空間が構成されたかを調べることは非常に興味深い．しかし，センサー入力が2次元であった先のゴール到達タスクと異なり，このタスクではセンサー入力が8次元になっており，図7.12や図7.13のような可視化を行うのは一般に困難である．そのためここでは，切り分けられた**状態**（State）の中で典型的なものを3つ示すに留める（図7.34-図7.36）．これらの図では，(1) ロボットが実空間でどのような状況にあるか，(2) その状態が実際のセンサー入力によってどのように定義されているか，(3) その状態において最も強化されたアクション，および弱化されたアクションは何か，が示されている．これらからエージェントの状態がセンサー入力空間を効率的に分割することにより定義されていることがわかる．

7.9 結果の考察

本章では移動ロボットのゴール到達問題という比較的単純な問題，および車庫入れ問題というやや複雑な問題をテストベッドとして，5章で提案した**行為結果のばらつき最小化に基づく状態と行為の抽象化法**の有効性を検証した．以下にこの実験で得られた各結果に対する考察を述べる．

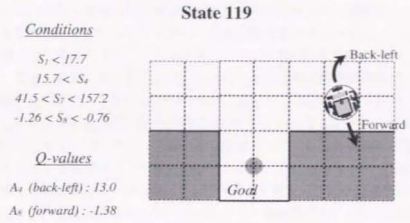


図 7.34: 定義された状態の例 （1）

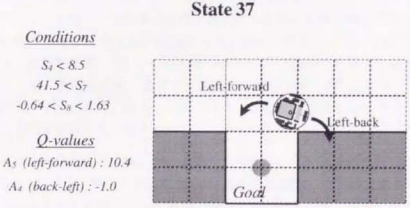


図 7.35: 定義された状態の例 （2）

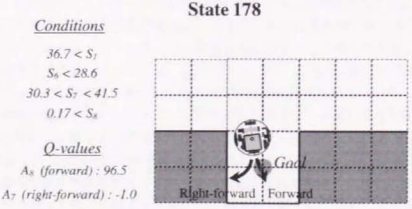


図 7.36: 定義された状態の例 （3）

実験1では，行為空間はあらかじめ人間が定義したものを用い，状態空間のみを3種の行為結果属性についてのばらつき最小化基準基づいて自律的に構成した．実験1－bでは，状態空間を0から作るために，ランダムアクションによる行動経験収集から始め，その後状態抽象化とQ学習とを繰り返すことによって，徐々に適切な状態空間が構成されていくことが示された．この実験において最も重要と思われる点は，1回目と2回目の状態空間構成で，全く同じ抽象化基準（行為結果属性に対する重み $\boldsymbol{w}$ が同じということ）を用いてるにも関わらず，抽象化過程の結果として得られる状態空間の様子が異なっているということである（図7.12，図7.13）．この現象は，5.10節で述べたように，学習の初期段階においては正の報酬を獲得する（ゴール状態に到達する）ような経験があまり得られないために，主に**センサー入力変化の類似性と負の報酬獲得に基づく状態抽象化**，言い替えれば，**"データ指向／リスク回避型の状態空間構成"** が行われるのに対し，行動政策学習が進んでからは正の報酬獲得や，その前の状態（サブゴール状態）への遷移の経験が増加するため，**ゴール／サブゴール到達に基づく状態／行為抽象化**，すなわち **"ゴール指向型の状態空間構成"** へと移行するために生じると考えられる．

実験1－cでは，人間が初期知識として与えた状態空間を始めに用いてこれを自律的に再構成することによって，実験1－aのようにランダムアクションを行う場合に比べて最終的なパフォーマンスに達するまでに要する訓練例が大きく節減されることが示された．このことは，自律エージェントの行動学習において，人間が与えられる範囲内での事前知識を用いて行動政策の獲得を行い，その後行動経験に基づいて適応的に行動知識を修正していくことによって，学習のスピードアップと，学習に伴うリスクやコストの軽減を図ることができることを示したという点で，非常に意義があると考えられる．

実験1－dでは，各行為結果要素に対する重みを変えて上と同じ実験を行い，その結果を比較することによって，**"センサー入力の変化"** や **"直接獲得報酬"**，**"遷移状態クラス"** など，異なる種類の行為結果要素を考慮に入れて状態の抽象化を行った方が，従来のようにそのうちのどれか1つか2つを考慮して抽象化を行う場合よりも，より良い状態空間が得られることが示された．

実験2では，状態空間はセンサー入力空間を格子状に分割して定義された固定のものを用い，行為空間についてのみ自律的な抽象化を行った．この行為の自律抽象化は，先の状態空間のみの自律構成に比べて大きなパフォーマンスの改善をもたらすことが示されたが，その結果の解釈には若干の注意を要する．というのは，このパフォーマンス向上には，行為結果に関するばらつき（エントロピー）最小化に基づいて行為空間が適切に構成されたということ自体よりも，行為空間が**非対称**に定義されたということの方が大きく寄与していると考えられるからである．つまり，図7.6のように格子状に定義された行為空間では，左回転（A_3）と右回転（A_7），前進（A_9）と後退（A_1）とがそれぞれ**対称**なアクションになっているため，Q学習の結果次第ではこれらの対称な行為を繰り返すデッドロックに陥る危険性があるのに対し，図7.21のように非対称な行為空間ではそのようなデッドロックに陥ることはない．また，もう本実験におけるもう一つの問題点は，5.9節で述べたような手法によって漸次的にモーター空間を分割し，行為の抽象化を行った場合，各分割における重みつきエントロピーゲイン（行為結果のばらつき

減少）が状態抽象化の場合と比べてとても小さいということである．そのため，同じ実験を繰り返した場合の行為空間の抽象化結果の再現性が低く，図7.21を含めていくつかのパターンが観察された．

実験3では行為結果のばらつき最小化に基づく状態空間の再構成と，行為空間の再構成とを続けて行うことによって，そのどちらかを単独で行うよりも，行動パフォーマンスがより改善されることができることが示された．しかし，実験2の行為空間のみの自律構成の場合と比べてその性能向上は小さく，しかも前述のように性能の向上が状態空間・行為空間の自律的抽象化によるものだけでなく，行為空間が非対称であることに大きく依存している点が問題である．

実験4では，状態空間の自律構成と行動政策獲得学習が一旦完了した後にセンサー系に故障が起きた場合，従来の研究のように行動政策のみを学習し直すのではなくて，状態空間自体から再構成することによって，より良い性能回復を実現できることが示された．しかし実験3と同様に，本実験におけるその効果はあまり大きなものではなかった．これはこの実験において想定したタスクが非常に単純なものであったということが大きな理由として考えられる．つまり，この程度の易しい行動獲得問題の場合，状態空間の定義の違いがそれはどクリティカルに全体のパフォーマンスに影響せず，Q学習などによって行動政策を学び直すだけである程度の性能回復が実現されてしまうということである．

実験5では，実験1～4におけるゴール到達タスクに比べてややレベルの高い車庫入れタスクについて，自律的な状態空間構成を行った．本章のはじめにも述べたように，このタスク2はタスク1と比べてただ複雑になっているだけでなく，正報酬と負報酬の存在領域が近接していること，ランダムなアクションでは滅多にゴール状態に到達しないこと，必要なセンサー入力が必然的に増えるためにセンサー入力空間も遥かに大きくなり，人間が事前に適切な定義を行うことが難しい，などの特徴を有していた．実際，実験5－aで見たように，センサー入力空間を単純に格子状に分割することによって定義した状態空間では，分割が粗い場合には全く適切な学習が行われず，細かく分割した場合には状態数が非常に多くなってしまい，行動政策学習に膨大な時間やメモリーコストが要され非現実的であると言える．

これに対して実験5－bではランダムアクションによる行動経験収集から始め，行動結果のばらつき最小化に基づく状態空間の自律構成プロセスと，Q学習による行動政策獲得プロセスとを交互に行うことによって，これらの問題が大きく改善されることを示した．しかし，その一方でこの車庫入れタスクのように正報酬を獲得するような経験が偶然には集まりにくいタスクでは，最初の状態空間構成で正報酬が得られる領域と負報酬が得られる領域とが正しく分離されないような不適切な状態空間が構成され，結果として悪循環に陥ってしまうケースが存在することも確認された．

そこで実験5－cでは，タスクを易しいものから徐々に難しくしていきながら状態抽象化と行動政策学習とを繰り返し行う**発達的学習アプローチ**によって，この問題を改善する方法の有効性を示した．ここで重要なのは，このような発達的学習アプローチが単なるその場しのぎ的な対策であるというよりは，むしろ，エージェントが複雑で高度な行動を学習する際における本質的な意味を持つものであると考えられることである．これは人間などの高等動物が高度な

行動を獲得するようになる過程と対比してみても容易に想像が付く．この実験で採用した発達的学習アプローチは，言い替えれば，「環境を人為的にコントロールすることによってエージェントの学習を助ける」ことであって，広い意味で人間の事前知識とエージェントによる自律的学習とを統合した形であると言える．同様の効果が期待され，またより直接的に人間の事前知識を利用するアプローチとしては，実験1－cで示した，人間が与えた状態空間を用いて行動学習を行い，ある程度の正報酬獲得経験が得られた段階で状態空間の再構成を行う方法，また，サブゴール（副報酬）を与えることによって複雑なタスクを容易なタスクの集合に分割して学習を行わせる方法などが考えられるが，いずれの場合もそのような事前知識をどのように得るのかということが最大の問題点でもある．

このように本実験では，“複数行為結果のばらつき最小化に基づく状態・行為抽象化”という提案手法の有効性をある程度証明することができた．しかしその一方で，想定したタスクがいずれも依然単純であるために，自律的な状態・行為抽象化によってもたらされる効果があまり明確でなかったり，抽象化以外の要因による影響を大きく受けるという問題点も指摘された．したがって今後は，タスクがより難しくエージェントのセンサーやアクチュエータの種類や個数が多いために，人間が直観的に状態空間や行為空間を定義することが困難であるような場合を想定して実験を行う必要がある．

第8章　提案手法の評価

本章では6章，7章における実験結果を踏まえ，本研究で提案した単純ベイズ分類器による状態クラスの一般化・表現法および行為結果のばらつき最小化に基づく状態・行為の抽象化の評価を行う．

8.1　単純ベイズ分類器による状態クラスの一般化・表現法

8.1.1　異種冗長センサー情報の統合

6章における実験では，4章において提案した**単純ベイズ分類器に基づく状態一般化・表現法**が，機上・天井カメラ画像から得られる距離や角度などの連続量から，ソナーセンサーによって得られる物体への距離，On-Offの離散値として得られる接触センサーの値に至るまで，様々な種類のセンサー入力情報をその異種性をほとんど意識することなく統合でき，またそれらの異種情報がロバストな状態認識に有効に利用されることが示された．これは，プリミティブなセンサー入力群の情報を，センサーの種類に依存する距離や角度，接触の有無などといった物理量の次元で統合するのではなく，確率，あるいは確率密度の比という無次元量のレベルで統合しているからであり，原理的にはこの実験では想定しなかった性格を持つセンサー（例えば，順序関係のある離散値を取るセンサーなど）に対しても適用可能である．

これらの点に関して，実験において比較した従来の他手法を見てみると，まず**マハラノビス楕円体による状態一般化・表現法**では，全センサー入力が連続値を取ることが前提となっている．したがって，この実験において想定した接触センサーのように（特に順序関係を持たない）離散値を取るセンサー入力を扱うことができない．さらに，この手法はセンサー入力空間中での**マハラノビス距離**がエージェントの状態の本質的な距離を表現するという大前提に成り立っている．すなわち，各状態クラスに中心点が存在し，その中心点からのマハラノビス距離が近ければ近い程そのクラスに属する確率が高く，センサー入力空間中で同じ距離にある点は**同程度にその状態クラスらしい**ということになる．このことは，より直観的には，あるセンサー入力に関してある状態クラスに属する確からしさをグラフにした場合，図8.1のように中心値において唯一の極大点を持った左右対称の形になることを意味している．しかし，実際には全てのセンサー入力がこのような性格を持つことは期待できない．例えば，実験で用いた**ソナーによる物体までの距離**というセンサー入力は，ある状態クラスに関して図8.2のようにこの前提に著しく反する性格を持っている．また，より一般的には図8.3のように，状態クラス中心点

からのマハラノビス距離が，状態クラスらしさと一致しない場合が存在する．このような場合，マハラノビス楕円による手法は，**線形判別関数**や**最近傍法**を用いた状態一般化・表現法と同様に，エージェントの状態空間を適足に構成することができないと考えられる．これに対し，ベイズ分類器とノンパラメトリックな確率密度推定を組み合わせた提案手法では，あるセンサー入力軸に関する分布が図8.3のようになっている状態クラスでも，一般化・表現が可能である．

一方，決定木を用いた状態クラスの一般化・表現法では，各属性軸（センサー入力変数の軸）に対して垂直な面でセンサー入力空間の分割が行われるので，それらの属性が連続値と離散値のどちらを取るかということには依存しない．しかしこの手法では，異なるセンサー入力同士が（ある状態クラスに関して）互いに近い情報を返す場合，より大きな情報ゲインをもたらすセンサー入力に基づく（決定木中の）分割が行われ，他のセンサー入力がもたらす同種の情報が用いられなくなる可能性が高い．実際に6.6節における比較実験では，状態クラスの決定木が常に大きな情報ゲインをもたらす天井カメラや機上カメラ画像によるセンサー入力（S_1–S_3，S_{10}，S_{11}）によって構成され，あまり大きなゲインをもたらさないソナーセンサーや接触センサーの値がほとんど参照されていないことがわかる．この点で，決定木による手法もまた提案手法と比較して，状態抽象化における異種冗長なセンサー情報の利用には不向きであることが示された．

8.1.2 不確定性要因に対する頑強性

6章の実験では，異種冗長なセンサー入力の情報を用いた状態認識が行われることによって，ノイズの増大やフォールトの発生など，実環境エージェントにとってはほとんど不可避な不確定性要素に対して頑強な行動決定を行えることも示された．この結果では，**異種**な冗長センサー系による頑強性・耐故障性の実現ということに特に意義がある．というのは，同種センサーだけからなる冗長系の場合，その種のセンサー固有の問題 (pitfall) に陥る可能性があるのに対し，異種センサーによる冗長系はそのような状況に対しても強いと考えられるからである．例えばある種の画像センサーは理想的な環境ではエージェントにとって非常に質の高い情報を提供するが，十分な明るさが得られない状況では性能が著しく劣化してしまうという場合には，同種のセンサーが予備としてあってもあまり意味がなく，むしろソナーセンサーや触覚センサーなどによる異種冗長性の方が効果的である．2章ではエージェントの**自律性**を，「様々な状況に柔軟に対応し，合理的な行動を取れること」として定義したが，異種冗長性によって環境の不確定性，多様性に対応することはまさにその一部であることが言える．

単純ベイズ分類器に基づく提案手法がこのような頑強性を有することの一つの理由として，状態クラスに関して各センサーがある値を取る条件つき確率が独立であるとする単純化仮定 (naive assumption) によって，あるセンサーに生じたフォールトが他のセンサーの情報に影響を与えないということが挙げられる．このことは全センサー間の依存関係を明示的に扱うマハラノビス楕円体による状態一般化・表現手法において，一部のセンサーのフォールトによって著しくパフォーマンスが劣化している実験結果からもわかる．また，このセンサー間の独立仮

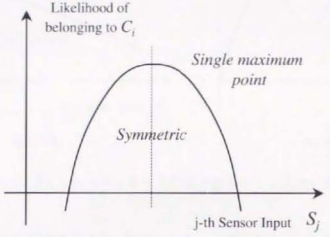


図 8.1: マハラノビス楕円体による状態表現手法が想定するセンサー入力と尤度の関係

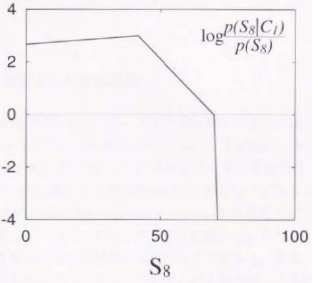
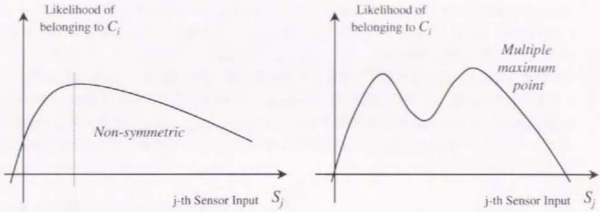


図 8.2: 実際のセンサー入力値と状態クラス尤度の関係



(a) 非対称　　(b) 複数の極大値

図 8.3: 考えられるセンサー入力値と状態クラス尤度の関係

定は，センサー入力から状態クラスを一般化，あるいは決定するのに必要な計算量をセンサーの数の 2 乗のオーダーから線形のオーダーに下げるという効果も持つが，全センサーの依存関係や誤差のモデルが完全に同定されている場合には，その知識を利用して状態認識を行う方が精度は良くなる．この点でセンサー情報間の独立を仮定するかどうかには，トレードオフが存在していると言える．

8.1.3 条件付確率（密度）分布の推定法

ベイズ分類器に関する研究分野では，インスタンスがあるクラスに属するという条件下での各属性の確率（密度）分布をどのように推定するかということが極めて重要な実際的問題になっている．この点に関して提案手法では，**ノンパラメトリックな推定法の一つである k_n-最近傍推定法**によって各状態クラスに属するときの各センサーの条件つき確率（密度）分布 $P(S_j|C_i)$ もしくは $p(S_j|C_i)$ を過去の行動経験から求めている．k_n-最近傍推定法のようなノンパラメトリックな推定法を用いることのメリットは，データが無限に与えられるという極限状態においては，任意の分布形状を推定することができるということであり，また，分布のモデルという一種の事前知識を使用しないということは，マハラノビス楕円体による状態表現法などにおいて暗黙に用いられている「センサー空間において定義される（マハラノビス距離やユークリッド距離などの）**距離**が，状態間の真の近さ・遠さを線形的に表す」という仮定を排していることを意味する．言い替えれば，この手法では状態間の距離尺度自体も帰納的に学習されるということになる．先の実験においても，ソナーセンサー入力（S_8，S_9）などに関しては，センサー値軸状での距離とある状態クラスへの近さとが必ずしも線形な関係にないことが示されている．

この性質は各状態クラスのセンサー入力空間における分布の形態があらかじめ得られないよ

うな様々なセンサー情報を統合する場合には，非常に有利であると考えられる．しかしその反面，この方法は推定結果が信頼できるようになるためには多くの経験データを必要とするという性格も持つ．6章における実験でも一つの状態クラスを定義するのに，約100個の経験インスタンスを要した．この経験データ数が不十分な場合には，状態分類の精度が著しく低下することが予想される．この問題に対しては，任意の分布系をうまく近似表現できるような分布モデル，例えば多数のガウス分布の**混合分布**（mixture）を仮定し，データに最も一致するような（すなわち尤度が最大になるような）パラメータの値を**EM法**（expectation maximamization method）などによって推定するということが今後の拡張として考えられる．

8.1.4 冗長性の偏りに関する問題

前述したように，単純ベイズ分類器におけるセンサー入力間の独立条件は冗長なセンサー系を有するエージェントにおいて，状態の一般化と表現に要するコストを低減するとともに一つのセンサーに生じたフォールトの影響を遮断する効果を持つ一方，全てのセンサーが正常である場合の最適性が保証されないという問題もある．特にセンサー間の冗長性に著しい偏りがある場合には「**不平等な多数決**」が生じることによって，状態分類のパフォーマンスが著しく落ちる可能性がある．つまり，6章における実験において提案手法が好ましい結果を示した背景には，このエージェントが有するセンサー系が，ここで想定したタスクに対して比較的「偏りのない」冗長性を有するように選ばれているということが深く関わっていると考えられる．

この問題に対するアプローチとしては，大きく2つの可能性が考えられる．1つは4.6節で提案した各センサー入力の**有用性基準** $Imp_C(S_j)$ および任意のセンサー入力間の**類似性基準** $I(S_j;S_k;C)$ などを用いて「バランスの良い」冗長センサー系を組むことである．これに関してはセンサー系における冗長性の「バランスの良さ」とこれらの基準との間の関係を明確にし，その定量的指標を考える必要がある．もう1つは単純ベイズ分類器における独立仮定（naive assumption）を取り除き，センサー間の依存関係（相関）を明示的に扱うより一般的な分類器を用いることである．ベイズ分類器による教師つき学習に関する研究分野では，より一般的なベイジアンネットワークをインスタンスデータから自動構築しこれを分類器として用いる方法（Bayesian network classifier）や，属性間の独立仮定を「限定的に」緩和することによって単純ベイズ分類器の単純さを維持しつつ，性能を向上させる方法などが提案されている[34]．しかし，これらの拡張によってセンサー入力間の相関，冗長性を明示的に扱った状態分類を行った場合，フォールトトレランスなどの性格がどのように変化するかということなどについては，現在のところ明らかでない．

8.2 行為結果のばらつき最小化に基づく状態・行為の抽象化

8.2.1 行為結果のばらつき最小化による状態・行為抽象化基準

従来の状態抽象化・行為抽象化における研究では，“同じアクションによって同じ報酬を獲得するような前状態におけるセンサー入力を一般化して状態クラスを定義する”[80][45][16]とか，“同じセンサー入力変化をもたらすようなモーター出力を一般化して行為クラスを定義する”[61][44]という極めてヒューリスティックな状態一般化法，行為一般化法が別々に提案され，しかもそれらの相異なる抽象化ポリシーの間にどのような理論的関係が存在するのか，十分に議論されることがなかった．これに対して本研究では，このような従来のヒューリスティックな抽象化法を包含する一般的な状態抽象化，行為抽象化の概念的枠組として，5章において“エージェントの行動経験集合における行為結果のばらつき（各行為結果属性の重み付き情報エントロピー和）”という状態・行為抽象化の統一的な評価基準を提案し，これを最小化するようなセンサー入力空間，あるいはモーター出力空間の分割を求める過程として一般的な状態・行為抽象化問題を定義し直した．前章で示した実験結果では，この新たな状態・行為抽象化の枠組が一般的なものであることが明らかになった．すなわち，この抽象化基準は状態空間，行為空間のどちらの自律的構成にも適用可能であり，かつ，人間が常識的にそれらの内部表現を定義する場合よりも，遥かに良い行動パフォーマンスをエージェントにもたらすことが示された．

8.2.2 異種行為結果属性の考慮

また，従来の研究では，“何をもって異なる状態および行為の類似性あるいは距離を定義するか”という問題に対して，同じ報酬を獲得するかどうかとか，センサー入力変化が似ているかどうか，など，様々なヒューリスティックな基準を用いているが，それらの抽象化基準は決して統合されることが無かった．これに対して本研究で提案した行為結果のばらつき最小化に基づく抽象化手法では，これらの異種の行為結果属性を，各属性に関する情報エントロピーの重み和という形で1つの抽象化基準に統合し，またこれによってより一般的で効率の良い状態・行為抽象化が実現されることを示した．

特に，前章の実験1－bで示されたように，これらの複数の行為結果属性を考慮することによって，エージェントの状態・行為抽象化が，行動政策が十分に獲得されていない段階では“データ指向／リスク回避型の状態・行為空間構成”を行い，適切な行動政策が得られるにつれて“ゴール指向型の状態・行為空間構成”へと徐々に変化させていくことが明らかになった．このような抽象化ポリシーの学習進度に伴った動的変化は非常に理にかなっており，大きな意味を持つ結果であると言える．

8.2.3 行動政策学習との連係

さらに，提案手法では過去の一定期間における行動経験のデータを入力として，抽象化評価基準を（局所的に）最適化する状態分割および行為分割を出力とするように定式化されているので，Q学習などの強化学習に代表される行動政策学習との統合が容易である．すなわち，強化学習によって行動政策を学びつつ，その間の行動経験に基づいて適応的に状態空間・行為空間を再構成し，再び強化学習を行うことができる．その結果，前章の実験1－cで明らかになったように，人間があらかじめ与えた状態空間を適応的に再構成することによって状態抽象化と行動学習のスピードアップを実現したり，実験1－dで示されたように，環境の変化やエージェント自身の身体的変化に対して臨機応変に状態空間を組み直すことが可能になった．従来の状態一般化手法では，状態の抽象化と行動政策の獲得とが一体化しているために一度状態空間が構成されるとそれで固定されてしまったり [5][80][85]，あらかじめ与えられた状態空間を状態遷移の不確かさに基づいて分割するだけであるために不必要な細分化が行われてしまう [59] という問題が存在しており，本提案手法の適応的状態・行為抽象化実現への貢献は大きいと考えられる．

8.2.4 より高度なタスク・行動クラスへの適用

7.8節では，(障害物が存在しない環境での) ゴール到達タスクよりもより高レベルなタスクとして，移動ロボットの車庫入れタスクを取り上げ，これに提案手法を適用した．このタスクでさえも，将来的に知能ロボットに期待される様々な複雑な行動タスクからは依然遠いということは否定しがたいが，それでもこの実験はそのような複雑なタスクに提案手法をどのように応用すれば良いのかということに対して，いくつかの重大なヒントを与えている．

その中でも最も重要な事項は，「タスクがある程度複雑になった場合，エージェントが現実的な時間やコストで適切な行動を学習するためには，何らかのかたちで人間の事前知識が必要になる」ということである．本実験においても，全く事前知識を仮定せず，ランダムアクションによって行動経験を収集することから始めた場合，一定量の試行数では十分な正報酬獲得経験が得られず，そのために不適切な状態空間が形成され，行動学習も進まないという悪循環が観察された．

そのため次の実験では，この車庫入れタスクに対する行動学習を，始めから最終的な環境で行うのではなく，ごく簡単な状況から車庫に入れる行動をまず学習させ，その後徐々に問題を難しくしていくという，「発達的学習アプローチ」を採用し，一定の効果があることを示した．ここで「ある（難しい）タスクに対して，そのような（易しい問題から徐々にもとの問題に近付けて行く）学習スケジュールを立てるか」ということは，広い意味で人間が与えた「事前知識」である．しかし，ここで重要なことは「事前知識」といってもエージェントが取るべき行動を直接規定するような膨大なルールの集合を与えるわけではなく，エージェントが自律的に適切な状態・行為空間，および行動ルールを学ぶことを容易にするために学習環境をコントロー

ルするような知識を用いているということである．言い替えれば，エージェントが適応的に行動を学習していく意義を失わない範囲内で人間のヒューリスティクスを利用し，学習の効率化を図ろうとするものである．前述したようにこれに類する他のアプローチとしては，複雑なタスクをサブゴールあるいは副報酬を与えることによって小さなタスクの集合に変換して学習させる方法，デフォルト的な状態・行為空間を初期知識として与えて行動決定／政策学習を行わせた後，それらを再構成する方法などが挙げられる．

第9章　結論と今後の課題

本章では，本研究の成果をまとめるとともに，今後の課題と展望を述べる．

9.1　本研究の成果

本論文では，反射的自律エージェントの行動獲得問題において，エージェントの内部表象である状態空間および行為空間をあらかじめ人間が与えるのではなく，エージェント自身がその目的や行動経験に基づいてプリミティブなセンサー入力，モーター出力から自律的に抽象化することが非常に重要であることを示すとともに，この状態・行為抽象化問題において従来の研究が扱ってこなかった2つの重要な未解決テーマ，**異種冗長センサー情報からの状態一般化と表現**，および**複数抽象化基準の統一的扱い**という問題に対する新しいアプローチを示した．

まず，「エージェントが持つ様々なセンサーから得られる異種冗長な情報をどのように統合し，状態の一般化と表現を行うか」という問題に対しては，**単純ベイズ分類器に基づく状態一般化・表現法**を提案した．この手法では，エージェントの行動経験データから統計的に推定された各センサー入力ごとの対数尤度の分布関数によって各状態の一般化と表現が行われ，新しいセンサー入力ベクトルがどの状態クラスに属するかは各センサー入力ごとに計算された対数尤度の和によって決定される．この提案手法は，以下のような実環境学習エージェントにとって望ましい性格を有することが示された．

- 各種センサーから得られる異種冗長な情報を，その入力信号の連続／離散性や確率分布の型などの性格に関係なく，柔軟かつ効率的に統合しつつ，状態の一般化と表現を行うことが可能である．
- またその結果として，従来の状態一般化・表現手法と比較して，センサーノイズや機能低下など，実環境にとって不可避な不確定性に対する頑強性が大きく改善される．

次に，「エージェントの状態や行為を何に基づいて抽象化するべきか」という問題に関しては，従来の諸研究において十分な議論なく用いられてきたヒューリスティックな抽象化基準を整理・統合し，**行動結果のばらつき（エントロピー）最小化**という新たな抽象化基準に基づく状態・行為抽象化の定式化を提案した．この新たな抽象化枠組が反射的自律エージェントの行動獲得にもたらす貢献として，以下の3点が挙げられる．

- ゴール・サブゴール状態への到達，報酬の獲得，センサー入力変化など，複数の行動結

果属性の類似性に基づいた状態および行為の抽象化が可能である．

- 従来の状態・行為の抽象化法では状態と行為のどちらかを固定した上でもう片方の自律的抽象化を行っていたために困難であった，**状態と行為の同時抽象化**がこの枠組によって容易に実現されるようになった．
- さらに，提案した方法に基づく状態・行為抽象化と，強化学習による行動政策獲得とを交互に繰り返し行うことによって，抽象化された状態／行為空間の性格が（データ指向からゴール指向へと）徐々に変化していき，エージェントの行動性能が大きく改善されることが示された．

9.2 今後の課題

本研究の今後の主な課題としては，以下に挙げるものが考えられる．

9.2.1 異種冗長センサー入力およびモーター出力の統合に関する課題

行為クラスの表現法の拡張

本論文では，エージェントの状態空間の表現については，4章においてベイズ分類器による方法を提案し，その性質を詳細に明らかにしたが，行為（アクション）をどのように一般化し表現するかという問題に関しては十分な回答を示していない．というのは，本研究ではエージェントのある行為をモーター出力ベクトル空間中の“点”として表現しているが，このモーター出力空間の各軸はプリミティブな各モーター出力の大きさ（スカラー）に過ぎず，その中に時間的な概念は明示的には含まれていない．つまり，複数のモーター出力が**非同期的**に実行されることによって実現されるような行為を1つのアクションとして表現することができない．しかし，実際の人間やロボットの行為は，複数のモーター出力が必ずしも同時に実行されて実現されるわけではなく，それらの出力が時間的に組み合わされることによって構成されている．したがって，今後はこのようなモーター出力の時間的パラエティーを明示的に記述できるような行為の表現法を取入れ，その記述空間の中における抽象化を考える必要がある．この拡張によって，いわゆる“マクロ”的な行為も自律的に抽象化されるようになることが期待される．

また，4章では異種冗長なセンサー情報を利用して状態を一般化することによる効果について述べたが，行為の一般化についても同様のことが期待される．すなわち，異種冗長なアクチュエーターを利用し，エージェントの行為選択にバラエティを持たせることによって，行為における不確定性や故障に対しても頑強な行動が実現されるものと考えられる．

状態・行為一般化アルゴリズムのオンライン化

5.10 節において示した状態・行為抽象化と行動政策獲得との統合法は，バッチ的な状態・行為一般化と，強化学習による行動政策の獲得と経験収集とを**交互**に行うというものであり，厳密な意味でのオンライン化にはなっていない．つまり，この場合の完全なオンライン化とは，エージェントの（1）行動決定（実行），（2）行動政策の獲得（更新），（3）状態・行為抽象化（または状態・行為クラス集合の更新），の3つの過程が同時並列的に行われることであるが，これを実現するためには4章で提案した状態の一般化アルゴリズム，および5.9.2 節，5.9.1 節で例を示した行為の一般化アルゴリズムを，バッチ的なものからインクリメンタルなものに改良する必要がある．

最適な冗長系の設計

4章では異種冗長に構成されたセンサー系をどのようにして効率的に用いるかということを議論したが，工学的な観点に立つと，今後は「エージェントのセンサー系やアクチュエータ系にどのように異種冗長性を持たせれば効果的であるか」という問題，すなわち設計論へと拡張していく必要がある．さらに，これと関連した問題としては，本テーマでもっぱら考えたような「比較的単純な異種センサーを多数集めてロバストな状態の認識を行う」というアプローチではなく，「少数の高性能かつ高信頼度のセンサーによって状態認識を行う」という工学的アプローチに対しても，提案手法が何らかの意義を持つか，ということが挙げられる．これらの意味で，4.6 節で提案したような，情報量基準に基づいたセンサー入力，モーター出力の重要性・類似性尺度を用いることはとても有用であると考えられるが，本論文ではその具体的な方法を明らかにするまでには至っていない．また，センサー入力やモーター出力の**異種性**をこの設計論の中でどのように表現し扱うか，ということも重要な課題である．

他の状態表現手法との比較

本研究では，4章および6章において，提案するベイズ分類器による状態一般化・表現法と，決定木による方法，マハラノビス楕円体による方法などと，定性的にあるいは実験的に比較を行い，その特徴を明らかにした．しかし，当然のことながらこれまでに研究されてきた知識の一般化法，表現法はこれらだけには留まらない．特に本研究ではあまり取り上げなかった，様々な種類の人工ニューラルネットワーク（Artificial Neural Network）に代表されるコネクショニストアプローチや自己組織化アプローチなどとの同様な比較は今後の重要な課題である．

9.2.2 状態および行為の統一的抽象化基準に関する課題

事前知識の利用による学習の高速化

本研究で扱っているような行動学習理論をロボットなどの実環境エージェントに適用する上での最大の問題は，適切な行動政策獲得や状態・行為の抽象化を行うために膨大な訓練例，行動経験データが必要とされるという点である．この現実的な問題の解決のために，様々な工夫・研究が行われているが，事前知識（apriori knowledge），背景知識（background knowledge）の利用，すなわち何もかもをエージェント自身が0（ゼロ）から学び始めるのではなくて，一般性が失われない範囲内で人間が与えた組み込み知識を利用し学習の高速化を図るというアプローチもその一つである．本研究においても，人間が一時的に定義した状態空間，行為空間を用いて行動政策学習を行い，その際の行動経験に基づいて状態空間，行為空間を再構成するという方針を採用した方が，ランダムアクションによって行動経験を収集して学習を行う場合に比べて大幅に効率が改善されることが示された．しかし，それでも依然多くの行動経験を要することには変わりなく，今後は状態遷移の因果則に関する知識など，より一般的に事前知識を効率的かつエージェント自身による学習の長所を損なわないように用いる方法について研究を行う必要がある．

行為結果のばらつき最小化アルゴリズムの改善

5章では，状態と行為の抽象化を複数行為結果のばらつき最小化問題として定式化するとともに，具体的な最小化アルゴリズムを2つ示した（5.9.1節，5.9.2節）．しかしこれらはあくまでも1つの例に過ぎず，最良のものであるとは限らない．実際，これらのアルゴリズムでは探索コストを低減するために，センサー空間あるいはモーター空間をエントロピーゲインの大きさに基づいて漸次的に分割しているが，これは一種の山登り法（hill-climbing method）であり局所最適解に陥る危険性が非常に大きい．したがって，今後は最終解の最適性と探索コストのトレードオフを考慮した上で最善の最小化アルゴリズムを考える必要がある．

抽象化基準として考慮すべき行為結果に関する検討

5章では，性格の異なる複数の行為結果属性のばらつき最小化を抽象化基準として用いることを提案し，7章の実験ではそのような行為結果属性として，「（直接）獲得報酬」，「到達した状態クラス」，「行為後のセンサー入力」の3つを考慮している．しかし，この実験で用いている各行為結果属性に対する重み係数は，経験的に（良い結果をもたらすものとして）決めたものであり，またエージェントの行為結果として考えられるのはこの3つに限らない．そこで今後は，異なる行為結果属性としてどのようなものを考慮すべきか，また，それらの行為結果属性に対してどのような重みを付加して状態・行為抽象化を行うのが良いのかを理論的，実験的に検討する必要がある．

センサー空間とモーター空間の直積空間における抽象化

本論文では，従来の状態・行為抽象化に関する研究と同様，エージェントの状態集合$\mathcal{C}$と行為集合$\mathcal{A}$とが互いに独立に，それぞれセンサー入力空間$\mathcal{S}$，モーター出力$\mathcal{M}$空間を分割することによって定義されるものとして扱って来た．つまり，$\mathcal{S}$と$\mathcal{M}$の直積空間$\mathcal{S}\times\mathcal{A}$は，図5.4のように格子状に分割され，各格子はある状態クラスC_iとある行為クラスA_jとの組合せとなることを想定している．しかし，現実的には状態クラスの集合と行為クラスの集合とは独立していない．例えば，"ボールを押す"というアクションは"目の前にボールが存在する"という状態において初めて意味を持つものであり，"ドアの前にいる"という全く関係のない状態においてこのアクションが選択肢として定義されていること自体不自然であると考えられる．すなわち，より一般的には状態と行為の抽象化は独立して行われるのではなく，図5.5のように，センサー入力空間とモーター出力空間の直積空間$\mathcal{S}\times\mathcal{A}$の分割として行われるべきである．この場合，分割された各部分空間を$\mathcal{S}$に投影したものがある状態クラスであり，$\mathcal{A}$に投影したものがある行為クラスということになる．

抽象化粒度の自動決定

7章における実験では，定義される状態クラス，行為クラスの最大個数をあらかじめ与え，この上限に達するか，あるいは分割（新しいクラスの定義）によるエントロピーゲインが一定値を下回るまで抽象化を行うようにしている．しかし，本来は状態クラス，行為クラスの個数，すなわち抽象化の粒度も"最適"に決まることが望ましい．この問題に関して，5章では抽象化基準の関数に"表現の複雑度に関するペナルティー項"を含めることを提案しているが，実際にはこの項(関数)をどのように決めるかという問題がある．このような粒度選択に関する問題は，帰納学習や統計モデル決定などの分野においても存在し，最小記述長（MDL），赤池情報量（AIC）などの利用が提案されているが，これらの基準を本問題にそのまま適用することはできない．というのは，状態・行為抽象化における"最適性"とは，その状態クラスや行為クラスの集合を用いて行為政策学習を行った結果初めて評価されるからである．すなわち，MDLやAICなどの情報理論的基準との理論的関係が全く明らかでない．したがって，どのようにして状態・行為集合の最適な粒度を決定するかという問題は今後の重要な課題の一つである．

9.2.3 状態・行為抽象化の応用に関する課題

記号的プランニングシステムとの統合

本論文では，提案した状態・行為抽象化法により構成された状態空間，行為空間を，反射的エージェントにおける行動政策学習（主に強化学習）に用いることを想定し，両者の統合法なども示したが，1章でも述べたように，自律エージェントがプリミティブなセンサー入力やモーター出力を抽象化し，一般的な状態／行為集合を獲得することは，一種のシンボル生成過程の

一部として考えることもできる．すなわち，状態・行為の抽象化は，従来の記号に基づくプランニングシステムと，反射的システムとの融合という，人工知能学における重要問題の鍵となる可能性がある．

マルチエージェントへの適用

本論文では，1個体の反射的エージェントが，外部環境とのインタラクションに基づいて自律的に状態および行為の抽象化を行うという問題を扱ったが，この延長として，複数個体のエージェントすなわちマルチエージェントシステムへの応用という問題が考えられる．この場合，各エージェントはそれぞれの身体性や目的，タスクなどに基づいて自分にとって適切な固有の内部表現（状態・行為空間）を獲得する一方で，複数のエージェントが互いに知識を共有したり，コミュニケーションを行うことが考えられる．そのためには，エージェント同士のインタラクションが各個体エージェントの内部表現獲得にどのような影響を及ぼすか，または及ぼすべきかという問題を今後考えていく必要がある．

謝辞

本研究を遂行するにあたり，本当に多くの方々の御指導，御協力，御助言を頂きました．ここにこれらのお世話になった方々に対して，心から感謝の意を表したいと思います．

まず最初に，筆者の指導教官であり，また，本論文の主査である堀浩一教授に対し，厚くお礼を述べたいと存じます．

堀教授には，筆者が本学の修士課程に進学し，知能工学研究室に所属するようになって以来，5年間にわたって一貫した親身の指導を頂いて参りました．その間の研究会や個別のディスカッションの場において頂いた，教授の数え切れない程多くの適切な助言は，ときに思考の袋小路に迷い込んで途方に暮れている筆者に新たな光明を与え，また，逆に筆者の思考が雑多なアイデアで発散しつつあるようなときには，極めてシャープで現実的な解決のヒントを与えて下さいました．また，研究以外の面でも，教授の豊かな人間性やユーモアのセンスは，知能工学研究室全体に潤いと活気を与え，その快適な環境の中で筆者はのびのびと研究に従事することができました．

航空宇宙工学科の中須賀真一助教授には，筆者が学部4年生のときから，常に力強い御指導を頂いて参りました．著者が本論文で取り組んだ研究テーマに進むきっかけを与えて下さったのも先生でした．また，先生とのミーティングではいつも議論が大きく発展しましたが，先生は時間が経つのも気にせず，次の研究段階として何が面白そうかを一緒になって考えて下さいました．これらのディスカッションの中で先生から分け頂いた発想のテクニックは，著者にとって貴重な財産となっています．

佐藤知正教授、鈴木真二教授、伊庭斉志助教授には，大変ご多忙であるにも関わらず，本論文の副査として，多大なる御指導を頂きました．各分野において第一人者である先生方は，非常に的確で建設的なコメントの数々によって，とかく一点に集中してしまいやすい著者の視野を大いに広げるとともに，今後の研究方針を考えて行く上で重要なヒントを提供して下さいました．

知能工学研究室の山内平行助手には，研究面での御助言を数多く頂くとともに，計算機などの研究環境整備においても大変な御尽力を頂きました．山内助手のこれらの協力なくしては，本論文の完成はあり得ませんでした．

中須賀研究室の花岡照明助手，技官の田中明氏には，研究室所有のロボットによる実験や，計算機の利用などに関して，多大な便宜をはかって頂きました．

知能工学研究室，中須賀研究室それぞれの秘書である，二木晶子さんと木村美津子さんには，事務手続きをはじめ様々な面で研究を支えて頂きました．

そして，航空宇宙工学科中須賀研究室，先端学際工学知能工学研究室（堀研究室）の両研究室の諸先輩や同僚の方々には，研究の面ではミーティングやディスカッションを通して様々な助言を頂き，研究以外の場での交流も著者にとっては大変有意義な経験でした．

同じ工学系博士課程に在籍した航空宇宙工学専攻の藤原健氏，先端学際工学専攻の渡部聡彦，石野洋子両氏，機械情報工学専攻の村川正宏氏，機械工学専攻の江口郁子氏には，研究における議論から日常の雑談まで，様々な形での交流によって多大なる良い刺激を受けました．著者が本研究に対して最後まで意欲を失わずに遂行できたのは，まさにこれらの交流があったからこそです．

また，日本学術振興会にも多大な感謝の意を表したいと存じます．著者は大学院博士課程の３年間，特別研究員として採用され，研究および生活の両面において強力な支援を頂きました．ここに深く感謝致します．

最後に，これまでの著者の生活を精神的経済的に支え，温かく見守ってくれた父と今は亡き母に深く感謝いたします．

参考文献

[1] P. E. Agre. Computational research on interaction and agency. *Artificial Intellingence*, Vol. 72, No. 1-2, pp. 1–52, 1995.

[2] J. Albus, A. Lacaze, and A. Meystel. Multiresolutional intelligent controller for baby robot. In *Proceedings of the 10th International Symposium on Intelligent Control*, 1995.

[3] M. Arbib and J. Liaw. Sensorimotor transformation in the worlds of frogs and robots. *Artificial Intellingence*, Vol. 72, No. 1-2, pp. 53–79, 1995.

[4] R. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. In P. Maes, editor, *Designing Autonomous Agents*, pp. 105–122. MIT/Elsevier, 1991.

[5] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorization for robot learning. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1502–1509, 1996.

[6] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, Vol. 23, pp. 279–303, 1996.

[7] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 1999.

[8] E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in Bayesian networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[9] R. Beer, H. Chiel, and L. Sterling. A biological perspective on autonomous agent design. In P. Maes, editor, *Designing Autonomous Agents*, pp. 169–186. MIT/Elsevier, 1991.

[10] S. Bennett and G. Dejong. Real-world robotics: Learning to plan for robust execution. *Machine Learning*, Vol. 23, pp. 121–161, 1996.

[11] A. Berler and S. Shimony. Bayesian networks for sensor fusion in occupancy grids. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[12] R. Brooks. Artificial life and real robots. In *Toward A Practice of Autonomous Systems : Proceedings of the First European Conference on Artificial Life*, pp. 3–9, 1991.

[13] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Trans. Robotics and Automation*, Vol. RA-2, No. 1, pp. 14–23, 1986.

[14] R. A. Brooks. Elephants don't play chess. In P. Maes, editor, *Designing Autonomous Agents*, pp. 3–15. MIT/Elsevier, 1991.

[15] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, Vol. 47, pp. 139–159, 1991.

[16] D. Chapman and L. P. Kaelbling. Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. In *Proceedings of Twelfth International Joint Conference on Artificial Intelligence*, pp. 726–731, 1991.

[17] P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, chapter 6. AAAI Press, 1995.

[18] P. Chen, J. Mills, and K. Smith. Performance improvement of robot continuous-path operation through iterative learning using neural networks. *Machine Learning*, Vol. 23, pp. 191–220, 1996.

[19] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, Vol. 9, pp. 309–347, 1992.

[20] S. Davies, A. Ng, and A. Moore. Applying online search techiniques to continuous-state reinforcement learning. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, pp. 753–760, 1998.

[21] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1990.

[22] R. Dearden, N. Friedman, and S. Russell. Bayesian q-learning. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, pp. 761–768, 1998.

[23] G. Dejong and S. Bennett. Permissive planning: Extending classical planning to uncertain domains. *Artificial Intelligence*, Vol. 89, pp. 173–217, 1997.

[24] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, Vol. 29, No. 2/3, pp. 103–130, 1997.

[25] M. Dorigo. Alecsys and the autonomouse: Learning to control a real robot by distributed classifier systems. *Machine Learning*, Vol. 19, pp. 209–240, 1995.

[26] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In A. Prieditis and S. Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*. Morgan Kaufmann, 1995.

[27] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley-Interscience, N.Y., 1972.

[28] J. C. Eccles. 脳の進化, 第4章 ヒト科の進化における言語のコミュニケーション. 東京大学出版, 1990.

[29] C. Elkan. Boosting and naive bayesian learning. Technical Report CS97-557, University of California, San Diego, 1997.

[30] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2*, Vol. 2, pp. 139–172, 1987.

[31] D. Fisher and M. Pazzani. Computational models of concept learning. In D. H. Fisher, M. J. Pazzani, and P. Langley, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning*, chapter 1, pp. 3–44. Morgan Kaufmann, 1991.

[32] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

[33] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifier. *Machine Learning*, Vol. 29, No. 2/3, pp. 131–164, 1997.

[34] N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 1277–1284, 1996.

[35] N. Friedman and M. Goldszmidt. Sequential update of Bayesian network structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[36] N. Friedman, M. Goldszmidt, D. Heckerman, and S. Russell. Challenge: What is the impact of Bayesian networks on learning ? In *Proceedings of the Fifteenth International Joing Conference on Artificial Intelligence (IJCAI-97)*, pp. 10–15, 1997.

[37] H. Friedrich, S. Munch, R. Dillmann, S. Bocionek, and M. Sassin. Robot programming by demonstration (rpd): Supporting the induction by human interaction. *Machine Learning*, Vol. 23, pp. 163–189, 1996.

[38] E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proceedings of Tenth National Conference on Artificial Intelligence (AAAI-92)*, 1992.

[39] J. H. George and P. Langley. Estimating continuous distribution in Bayesian classifiers. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

[40] J. Gibson. *The Ecological Approach to visual perception*. Houghton Mifflin, 1979.

[41] J. Grefenstette. Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning*, Vol. 3, pp. 225–245, 1988.

[42] H. Guvenir and I. Sirin. Classification by feature partitioning. *Machine Learning*, Vol. 23, pp. 47–67, 1996.

[43] D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research Advanced Technology Division, 1995.

[44] W. Iba and J. H. Gennari. Learning to recognize movements. In D. H. Fisher, M. J. Pazzani, and P. Langley, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning*, chapter 13, pp. 355–386. Morgan Kaufmann, 1991.

[45] H. Ishiguro, R. Sato, and T. Ishida. Robot oriented state space construction. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1496–1501, 1996.

[46] J.H.Gennari, P.Langley, and D.Fisher. Models of incremental concept formation. *Artificial Intelligence*, Vol. 40, pp. 11–61, 1989.

[47] L. Kaelbling. An adaptable mobile robot. In *Toward A Practice of Autonomous Systems : Proceedings of the First European Conference on Artificial Life*, pp. 41–47, 1991.

[48] L. Kaelbling and S. Rosenchein. Action and planning in embedded agents. In P. Maes, editor, *Designing Autonomous Agents*, pp. 35–48. MIT/Elsevier, 1991.

[49] M. Kearns, Y. Mansour, and A. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pp. 282–293. Morgan Kaufmann, 1997.

[50] V. Klingspor, K. Morik, and A. Rieger. Learning concepts from sensor data of a mobile robot. *Machine Learning*, Vol. 23, pp. 305–332, 1996.

[51] J. Koza. Evolution of subsumption using genetic programming. In *Toward A Practice of Autonomous Systems : Proceedings of the First European Conference on Artificial Life*, pp. 110–119, 1991.

[52] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifire. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 223–228, 1992.

[53] P. Langley, G. Provan, and P. Smyth. Learning with probabilistic representations. *Machine Learning*, Vol. 29, No. 2/3, pp. 91–101, 1997.

[54] P. Langley and S. Sage. Induction of selective Bayesian classifier. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399–406. Morgan Kaufmann, 1994.

[55] P. Maes. Learning behavior networks from experience. In *Toward A Practice of Autonomous Systems : Proceedings of the First European Conference on Artificial Life*, pp. 49–70, 1991.

[56] P. Maes. Situated agents can have goals. In P. Maes, editor, *Designing Autonomous Agents*, pp. 49–70. MIT/Elsevier, 1991.

[57] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of Ninth National Conference on Artificial Intelligence*, pp. 768–773, 1991.

[58] T. M. Mitchell. Becoming increasingly reactive. In *Proceedings of Eighth National Conference on Artificial Intelligence (AAAI-90)*, pp. 1051–1058, 1990.

[59] A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, Vol. 21, pp. 199–233, 1995.

[60] J. P. Muller. An architecture for dynamically interacting agents. *International Journal of Intelligent and Cooperative Information Systems*, Vol. 3, No. 1, pp. 25–45, 1994.

[61] T. Nakamura and M. Asada. Motion sketch: Acquisition of visual motion guided behaviors. In *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence*, pp. 126–132, 1995.

[62] S. Nakasuka, T. Yairi, and H. Wajima. Autonomous generation of reflexion-based robot controller using inductive learning. *Robotics and Autonomous Systems*, Vol. 17, pp. 287–305, 1996.

[63] J. Oliver, R. Baxter, and C. Wallace. Unsupervised Learning using MML. In *Machine Learning: Proceedings of the Thirteenth International Conference (ICML 96)*, pp. 364–372, 1996.

[64] M. Pazzani. Learning causal patterns. In R. Michalski and G. Tecuci, editors, *Machine Learning IV - A Multistrategy Approach*, chapter 10, pp. 267–293. Morgan Kaufmann, 1994.

[65] M. J. Pazzani. Searching for dependencies in Bayesian classifiers. In *Proceedings of fifth International Workshop on Artificial Intelligence and Statistics*, 1995.

[66] J. Perl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[67] K. Poh, M. Fehling, and E. Horvitz. Dynamic construction and refinement of utility-based categorization models. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 11, pp. 1653–1663, 1994.

[68] J. R. Quinlan. Learning efficient classificaion procedures and their application to chess end games. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, Symbolic Computation*, chapter 15, pp. 463–482. Springer-Verlag, 1984.

[69] J. R. Quinlan. AIによるデータ解析. トッパン, 1995.

[70] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice - Hall, Inc., 1995.

[71] M. Salganicoff, L. Ungar, and R. Bajcsy. Active learning for vision-based robot grasping. *Machine Learning*, Vol. 23, pp. 251–278, 1996.

[72] T. Sawaragi, H. Sawada, and O. Katai. Reinforcement learning for autonomous mobile robots by forming approximate classificatory concepts. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1337–1344, 1996.

[73] A. Segre and J. Turney. Planning, acting, and learning in a dynamic domain. In S. Minton, editor, *Machine Learning Methods for Planning*, chapter 5, pp. 125–158. Morgan Kaufmann, 1993.

[74] L. Spector and J. Hendler. Planning and reacting across supervenient levels of representation. *International Journal of Intelligent and Cooperative Information Systems*, Vol. 1, No. 3-4, pp. 411–449, 1992.

[75] L. Steels. Emergent functionality in robotic agents through on-line evolution. In *Artificial Life IV*, pp. 8–14, 1994.

[76] J. Suzuki. A construction of Bayesian networks from databases based on an mdl scheme. In D. Heckerman and A. Mamdani, editors, *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 266–273, 1993.

[77] Y. Takahashi, M. Asada, and K. Hosoda. Reasonable performance in less learning time by real robot based on incremental state space segmentation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1518–1524, 1996.

[78] Ming Tan. Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, pp. 7–33, 1993.

[79] Jun Tani. Self-organization of symbolic processes through interaction with the physical world. In *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence*, pp. 112–118, 1995.

[80] A. Ueno, K. Hori, and S. Nakasuka. Simultaneous learning of situation classification based on rewards and behavior selection based on the situation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1510–1517, 1996.

[81] W. Uther and M. Veloso. Tree based discretization for continuous state space reinforcement learning. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, pp. 769–774, 1998.

[82] C. Watkins and P. Dayan. Tchnical note: Q-learning. *Machine Learning*, Vol. 8, No. 3-4, pp. 279–292, 1992.

[83] T. Yairi, S. Nakasuka, and K. Hori. Sensor fusion for state abstraction using bayesian classifier. In *Proceedings of IEEE International Conference on Intelligent Engineering Systems (INES'98)*, 1998.

[84] T. Yairi, S. Nakasuka, and K. Hori. State abstraction from heterogeneous and redundant sensor information. In *Proceedings of International Conference on Intelligent Autonomous Systems 5 (IAS-5)*, pp. 234–241, 1998.

[85] T. Yairi, S. Nakasuka, and K. Hori. Automatic reactive behavior acquisition for planetary rovers. *Journal of Space Technology and Science*, 1999. (To appear).

[86] 堀部安一. 情報エントロピー論. 森北出版, 1989.

[87] 神宮英夫. スキルの認知心理学. 川島書店, 1993.

[88] 石田亨. エージェントを考える. 人工知能学会誌, Vol. 10, No. 5, pp. 663–667, 1995.

[89] 中村恭之, 浅田稔. ステレオスケッチ：ステレオ視覚を持つ移動ロボットの行動獲得. 日本ロボット学会誌, Vol. 15, No. 4, pp. 533–541, 1997.

[90] 山西健司. データ圧縮と学習. 人工知能学会誌, Vol. 12, No. 2, pp. 204–215, 1997.

[91] 木村元, L. P. Kaelbling. 部分観測マルコフ決定過程下での強化学習. 人工知能学会誌, Vol. 12, No. 6, pp. 822–829, 1997.

[92] 鈴木宏昭. 類似と思考. 認知科学モノグラフ. 共立出版, 1996.

[93] 小橋康章. 決定を支援する. 認知科学選書. 東京大学出版会, 1988.

[94] 山崎弘郎, 石川正俊（編）. センサフュージョン － 実世界の能動的理解と知的再構成. コロナ社, 1992.

[95] 坂元慶行, 石黒真木夫, 北川源四郎. 情報量統計学. 共立出版, 1983.

[96] 山田誠二. 適応エージェント. 共立出版, 1997.

[97] 繁桝算男. 意志決定の認知統計学. 行動計量学シリーズ. 朝倉書店, 1995.

[98] 甘利俊一. 情報理論. ダイヤモンド社, 1970.

[99] 谷淳. ロボットにおける認知と自律性の構造：力学系の見地から. 日本ロボット学会誌, Vol. 14, No. 4, 1996.

[100] 鈴木譲, 大嶽康隆, 平澤茂一. 記述長最小基準と状態分割の立場からみた確率モデルの選択方法について. 情報処理学会論文誌, Vol. 33, No. 11, pp. 1281–1289, 1992.

[101] 森村英典. マルコフ解析. OR ライブラリー. 日科技連, 1979.

[102] 松原仁. 人工知能におけるロボットの役割. 日本ロボット学会誌, Vol. 14, No. 4, 1996.

[103] 佐々木正人. アフォーダンス － 新しい認知の理論. 岩波科学ライブラリー. 岩波書店, 1994.

[104] 大須賀節雄, 佐伯胖. 知識の獲得と学習. 知識工学講座. オーム社, 1987.

[105] 畝見達夫. 強化学習. 人工知能学会誌, Vol. 9, No. 6, pp. 830–835, 1994.

[106] 鳥脇純一郎. 認識工学 -パターン認識とその応用-. コロナ社, 1993.

[107] 大津展之, 栗田多喜夫, 関田巌. パターン認識 － 理論と応用. 行動計量学シリーズ. 朝倉書店, 1996.

[108] 日本ファジィ学会（編）. ファジィ測度. 講座ファジィ. 日刊工業新聞社, 1993.

[109] 日本機械学会（編）. 挑戦：知能化する機械. 養賢堂発行, 1997.

[110] 錦見美貴子. 言語を獲得するコンピュータ. 認知科学モノグラフ. 共立出版, 1998.

[111] 乾敏郎（編）. 知覚と運動. 認知心理学. 東京大学出版会, 1995.

[112] 石塚満. 知識の表現と高速推論. 丸善, 1996.

[113] 浅田稔. 強化学習の実ロボットへの応用とその課題. 人工知能学会誌, Vol. 12, No. 6, pp. 831–836, 1997.

[114] 矢入健久, 中須賀真一, 堀浩一. 異種センサー情報統合による自律ロボットの状態空間構成. 人工知能学会 合同研究会 ＡＩシンポジウム' 9 7 資料 SIG-J-9702, pp. 57–63, 1997.

[115] 鷲尾隆. エージェント. 広田薫（編）, 知能工学概論, 第 2 章, pp. 27–42. 昭晃堂, 1996.

[116] 宮崎和光, 山村雅幸, 小林重信. 強化学習における報酬割当の理論的考察. 人工知能学会誌, Vol. 10, No. 3, pp. 104–111, 1994.

発表文献リスト

学会誌論文

- S.Nakasuka, T.Yairi and H.Wajima, "Autonomous Generation of Reflexion-based Robot Controller Using Inductive Learning", Robotics and Autonomous Systems, Vol.17, pp.287–305, 1996
- T.Yairi, S.Nakasuka and K.Hori, "Automatic Reactive Behavior Acquisition for Planetary Rovers", Journal of Space Technology and Science, Vol.12, No.1, pp.17–26, 1996
- 矢入健久, 中須賀真一, 堀浩一, "異種冗長なセンサー情報に基づく自律的状態抽象化法", 人工知能学会誌, Vol.14, No.4, 1999

国際会議発表論文

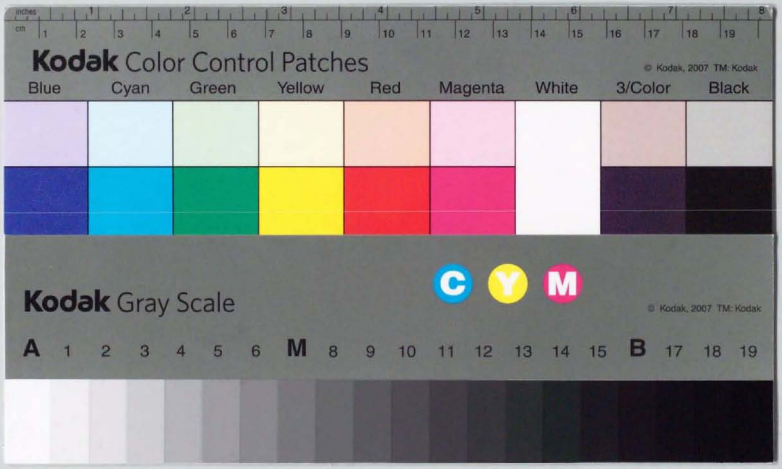
- T.Yairi, "On-board Reconfigurable Attitude Control System with Optimization", 19th International Symposium on Space Technology and Science, 1994
- T.Yairi, H.Wajima, S.Nakasuka and K.Hori, "A Novel Control Architecture for Autonomous Rover with Learning Subgoals in Attribute Space", Proc. of 11th International Astrodynamics Symposium, pp.289–294, 1996
- T.Yairi, S.Nakasuka and K.Hori, "Automatic Reactive Behavior Acquisition for Planetary Rovers", Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), pp.397–402, 1997
- T.Yairi, S.Nakasuka and K.Hori, "State Abstraction from Heterogeneous and Redundant Sensor Information", Proc. of International Conference on Intelligent Autonomous Systems 5 (IAS-5), pp.234–241, 1998
- T.Yairi, S.Nakasuka and K.Hori, "Sensor Fusion for State Abstraction Using Bayesian Classifier", IEEE International Conference on Intelligent Engineering Systems (INES'98), 1998

国内研究会・シンポジウム発表論文

- 矢入健久, 輪島裕之, 田中明, 中須賀真一, 堀浩一, "惑星上ローバーを想定した自律行動マネジメント系に関する研究", ロボティクス・メカトロニクス講演会'９６ (ROBOMEC'96), pp.612–615, 1996
- 矢入健久, 中須賀真一, 堀浩一, "異種センサー情報統合による自律ロボットの状態空間構成", 人工知能学会合同研究会 ＡＩシンポジウム'９７, pp.57–63, 1997

全国大会発表論文

- 矢入健久, 中須賀真一, 輪島裕之, "惑星上ローバーを想定した自律化アーキテクチュアの研究", 第３８回宇宙科学技術連合講演会, 1994
- 矢入健久, 堀浩一, 中須賀真一, 輪島裕之, "自律ローバーのための属性空間分割によるサブゴール列学習", 第１３回日本ロボット学会学術講演会予稿集, pp.553–554, 1995
- 矢入健久, 田中明, 中須賀真一, 堀浩一, "自律移動ロボットのための文脈依存型行動則体系", 第１４回日本ロボット学会学術講演会予稿集, pp.135–136, 1996
- 矢入健久, 田中明, 中須賀真一, 堀浩一, "ヘテロな機能冗長性を有する群ロボットシステムの協調行動学習", 第１５回日本ロボット学会学術講演会予稿集, pp.965–966, 1997


inches
cm
Kodak Color Control Patches
© Kodak, 2007 TM: Kodak
Blue
Cyan
Green
Yellow
Red
Magenta
White
3/Color
Black
Kodak Gray Scale
C
Y
M
© Kodak, 2007 TM: Kodak
A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19