

自律学習における CITTA による部分知識の結合

山川 宏^{*1}、宮本 裕司^{*2}、馬場 孝之^{*1}、岡田 浩之^{*1}

本報告では、自律学習というフレームワークにおいて、タスク学習において事前学習で獲得した部分知識をモジュールとして組み合わせて利用するために開発した、マルチモジュール問題解決アーキテクチャである CITTA (Cognition based Intelligent Transaction Architecture) について述べる。自律学習のフレームワークでは、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まっていない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う。

事前学習で獲得する部分知識にはセマンティクスが与えられていないので、CITTA は本来同一の意味を持つ情報についてはスタック上に積み上げて扱う。また、マルチモジュール環境では各モジュールが取り扱う変数の制御可能性が動的に変化するので、CITTA はセンサ入力と行動出力を統一的に扱う一般化状態を用いる。

最後には「RWC 2001 Final Exhibition & Symposia」で行ったナビゲーション課題によるデモンストレーション「ドア&鍵課題の自律学習による解法」において、自律学習機能により、事前学習により予め獲得した部分環境知識を、タスク学習で CITTA によって部分環境知識を組み合わせて利用することで、新たな場面への適応が可能となることをお見せする

1 はじめに

我々は、経済産業省の国家プロジェクトであるリアルワールドコンピューティングにおいて、図 1 に示すような、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まっていない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う自律学習というフレ

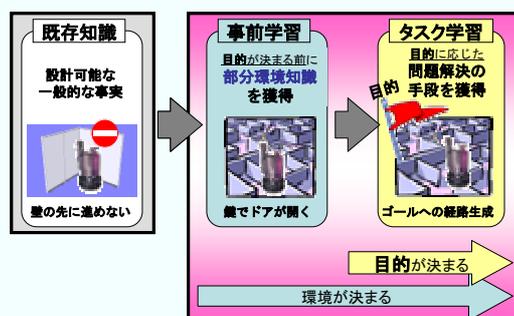


図1 自律学習の枠組みの提唱

ームワーク[1]を提唱してきており、これまでに、状況分解[2]と強化学習を組み合わせたナビゲーション実験[3]などを行っている。

我々は自律学習の実現のために、3つの要素技術を開発した。1つ目に事前学習において、部分環境知識を抽出するための Matchable 状況分解手法[2]で、本技術ではスプレッドシート状に整形されたデータに対して、特徴量とイベントの同時選択により関係の強い複数の部分情報を自動的に抽出する。

2つ目の要素技術は、個々のモジュールにおける認知距離学習による問題解決器である。学習より獲得した任意の状態間の行動によって結び付けられた認知距離を用いて、実行時の計算コストを抑えながらの与えられたゴールに応じた行動決定が可能である[4]。

3つ目の要素技術は、マルチモジュールアーキテクチャとしての CITTA がある。タスク学習では、CITTA を用いて事前学習で獲得した部分知識をモジュールとして組み合わせて利用する。

2 自律学習というフレームワーク[1]

実世界環境からの情報を扱う従来システムでは、人間があらかじめ獲得し、設計した環境に関する知識を利用してタスク学習を行っていたが、その知識獲得に要する設計コストが大きく問題となっていた。なぜなら、システムは環境ごとに知識獲得を行う必要があるためである。

Combining Partial Knowledge using CITTA in Autonomous Learning Framework

Hiroshi Yamakawa, Takayuki Baba, Hiroyuki Okada,
(株)富士通研究所, FUJITSU LABORATORIES LTD
Yuji Miyamoto*2, (株)富士通ビー・エス・シー研究所,
FUJITSU BROAD SOLUTION & CONSULTING Inc.

そこで我々は個別の環境に依存しない普遍的な知識をシステムに設計して与えた後に、環境に関する知識獲得を事前学習として行う、自律学習の枠組みを提唱した。これによれば、環境に関する知識の設計コストが大幅に低減できる。

自律学習という考え方は、人間の認知発達からヒントを得たものです。つまり子供の時には特定の目的を持たない行動を行いながら環境に関する知識を蓄積してゆき、大人になったときには既に蓄えていた知識を利用して目的に応じた効率的に学習やタスクの遂行を行える能力を実現し様としています。そういう意味では、認知発達の過程では学習は多段階で行われており、単純に 2 段階に分けられるものではありません。先行する学習が後の学習を効率的に行えるようにサポートするブーツストラップな関係を作り上げることが重要であると考えています。しかし、行動の目的の有無により事前学習とタスク学習を切り分けは、最もわかりやすいと考えています。

事前学習とタスク学習の具体的な手法としても、様々な選択肢があるとは思いますが、現在我々は環境に関する知識獲得の手法として Matchable 状況分解手法[2]を利用し、これにより得られた部分環境知識を組み合わせ用いれば、タスク学習においては計算コストの削減や、より少ない経験からの汎化能力が実現できます。

3 マルチモジュールアーキテクチャ(CITTA)

CITTA により、事前学習で得られた部分環境知識を組み合わせ利用する。今回提案する CITTA(図2参照)では、今まで我々が提案してきた CITTA[5,6]に対して、他の二つの技術と組み合わせるため、情報交換に利用する共有メモリのスタック化やコントローラの導入などの改良を行った。

3.1 CITTA の構成と協調動作

CITTA はその内部に共有メモリ、モジュール群、コントローラを持ち、全体として環境からの現在の状態を入力し、ゴール解決のための処理を行い、ゴールへ移動するための次の状態を出力する。CITTA は環境と交互に動作し、これを 1 サイクルとする。外部環境およびモジュール間との情報交換には共有メモリが用いられる。

CITTA は複数のモジュールを管理し、ある現在の状態において処理が行なえるモジュールを使用して問題解決を行なう。

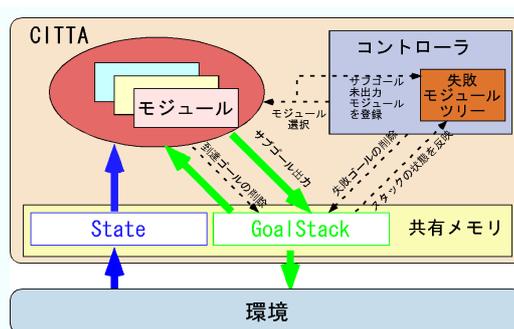


図2 CITTA の基本構成

この図では、現在の状態(State)のベクトルとゴールnスタック(Goal Stack)を分けて書いてあるが、実際にはノードごと現在の状態とゴールのスタックが存在する。

使用するモジュールはコントローラによって選択される。

共有メモリは情報の最小単位であるノードのベクトルで、ノードごとに一つの現在の状態と、実現したいゴールのスタックを持つ。

コントローラではモジュールの制御とスタックの制御を行なう。1 サイクルごとに、実行処理を行う 1 つのモジュールを選択する。一度選択され実行処理を行なったモジュールはサブゴールを出力できる間は連続して選択され、実行処理を行う。サブゴールを出力できない場合に、サブゴールを出力できるモジュールが見つかるまで、ランダムにモジュール選択を行う。

また、スタック上のゴールで、全てのモジュールが実行できない(全モジュールがサブゴール未出力)ゴールはスタックから削除する。

各モジュールは、接続されているノードとの情報の受け渡しとして、現在の状態、ゴールの取得と、サブゴールの出力を行う。処理としては、学習処理、到達ゴールの削除処理、実行処理の3つがある。学習処理、到達ゴールの削除処理は全てのサイクルで行う並列的な処理であるが、実行処理についてはコントローラによって選択されたモジュールのみ行う(逐次処理)。

学習処理は、取得した現在の状態を利用してモジュールごとの学習を行う。到達ゴールの削除処理は、自ら出力したサブゴールに到達していればスタックから削除する。

実行処理は取得した現在の状態とゴールについての問題解決を行い、次の状態をサブゴールとして出力して共有メモリ上のゴールスタックにプッシュする処理を行なう。しかしながら、以下のような3つサブゴール未出力条件を満たす場合には、モジュール内の処理を行う以前に実行動作は中止される。

- 現在のゴールに対して既に失敗している
- 現在の状態が既にゴール到達している
- 前サイクルに出力したサブゴールに到達していない

また、モジュール内の処理をした後に以下のような条件になった場合にもサブゴールの出力を行わない。

- 内部処理でサブゴールを出力できない
- 前回と同じサブゴールを出力しようとした
- ゴールと同じサブゴールを出力しようとした

以上のような動作により、スタックの最上部のゴールを監視しているモジュールの中から動作可能なモジュールの一つが動作を行うことにより全体として協調して問題解決が図られる。

3.2 CITTA の特徴

このアーキテクチャは、一種の黒板モデルと考えられるが、事前学習により獲得した部分知識を利用するために以下のような点で特徴がある。

- **セマンティクスラベルが存在しない**

部分知識にはセマンティクスをあらわすラベルが無いので、本来同一の意味を持つ情報を、スタック上に積み上げるなどして、その同一性を保持する。このために、システム内に同じ意味を持つ情報は分散して存在しない。

- **一般化状態の導入**

マルチモジュール環境では各モジュールが取り扱う変数の制御可能性が動的に変化し、それを事前学習の段階で知ることは困難なので、CITTA はセンサ入力と行動出力を統一的に扱う一般化状態を用いる。

4 デモンストレーション

図3に示す、「RWC 2001 Final Exhibition & Symposia」で行ったナビゲーション課題によるデモンストレーション「ド

ア&鍵課題の自律学習による解法」により、自律学習の効果を示す。

4.1 課題の基本設定

環境中に置かれたエージェントは以下の情報の入出力を行う

- 位置: 部屋内の 2 次元座標位置
- 行動: 前後左右方向への移動
- 存在物: 現在位置のもの (ドア、鍵置場、電話置場、エサ)
- 所持品: 現在持っている物 (なし、鍵、電話)
- 部屋: 部屋の ID
- ドア条件: ドアが開く所持品を示す ID

これらは、一般化状態であるから特にセンサと行動は区別されていない。

4.2 事前学習

事前学習では、エージェントが 9 個の異なる環境でランダムウォークし、そのデータを用いて Matchable 状況分解します。ここでの状況分解では 2 時刻をひとまとめにして扱う時系列状況分解の技術が用いられ、これにより部分マルコフ遷移を取り出すことができます。

状況分解は属性とイベントの同時選択により、よく起こる出来事の部分集団 (部分状況) を抽出します。ここで、各部分状況は内部に規則性に基づいて評価されます。) よって、状況分解による事前学習を行うことで、「鍵とドアの関係」や、「位置と移動の関係」などについての部分環境知識が獲得されます。この例では合計 14 個の部分環境知識 (部分状況) が抽出されました。その結果を図4に示します。この中では、位置、行動、存在物、部屋を含む部分状況を取り出されており、これは移動に関するモジュールとして動作します。また、存在物、所持品、ドア条件を含む部分状況はドアを開けるためのモジュールとして動作します。

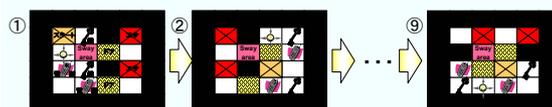


図 3 事前学習を行う環境

部屋ごとにマップが異なり、鍵でドアが開く部屋と電話でドアが開く部屋などがある。

No	位置	行動	存在物	持ち物	部屋	ドア条件
1	場所1	↑	空	空	1	1
2	場所4	→	鍵	鍵	1	1
3	場所3	↓	空	鍵	1	1
4	場所5	↓	ドア	鍵	1	1
5	場所6	↓	空	鍵	1	1
6	場所9	←	エサ	鍵	1	1
...
1000	場所7	↓	空	空	1	1
1001	場所1	↑	空	空	2	1
...
9000	場所1	↑	電話	電話	3	3

図4 事前学習で抽出される部分状況

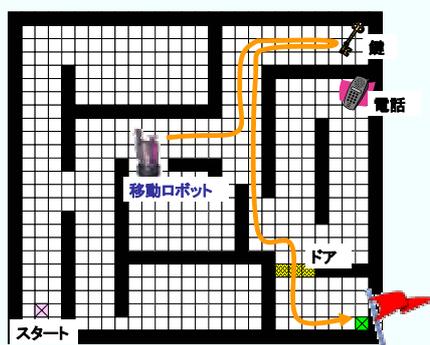


図5 タスク学習に用いられた部屋の例

4.3 タスク学習

この場合の CITTA の動作は、エサが存在する場所に行くことがゴールとして与えられた場合には、そのゴールを実現可能であると判断した「位置と移動の関係」をもつモジュールがまず動作することがおこる。

部分環境知識を利用すれば、ゴールを与えられた以降のタスク学習での行動決定に必要な経験の量を小さくできる。このため必要に応じて鍵を取りながらゴールへ向かう経路生成を少ない経験で行えるようになった。

また自律学習では、タスク学習の経験も分解して蓄えるので、環境が変化しても部分的な経験を組み合わせることで、新たな環境に素早く対応できる。

5 おわりに

我々が提案している、学習を特定の目的に対する行動を獲得する「タスク学習」とまだ目的が決まってない段階で環境に関する知識を獲得する「事前学習」の二つの段階に分けて取り扱う自律学習というフレームワークについて述べた。

事前学習としての状況分解で獲得した部分知識を、タスク学習においてモジュールとして組み合わせて利用するために、改良したマルチモジュールアーキテクチャである CITTA の提案を行った。ナビゲーション課題において、事前学習により予め獲得した部分環境知識を、タスク学習で CITTA によって部分環境知識を組み合わせて利用することで、新たな場面への適応能力が向上することが示された。

我々の研究の目的は、柔軟な学習能力を実現であり、そのためにシステムの内部構造を学習により獲得することを目指してきた。今回の成果はそんなことがある程度可能になってきたことをデモンストレーションすることが目的である。そのため、マルチ・エージェント技術の点から見ると、エージェント間の協調や競合解消などに関しては、既存の技術が十分に盛り込まれていない。一方で学習によりシステムの内部構造を獲得しようとする場合には、「セマンティックスラベルを利用できない」、「センサと行動を区別しづらい」などの困難があり、それらに適応しうるアーキテクチャとして CITTA 提案を行っている。この研究は学習システムとしての工学的応用のみならず、人間の脳機能を理解する上でも有用であると考えているので、それらとの関係付けも探ってゆきたい。

なお本研究の一部は RWC 予算により行われた。

参考文献

- [1] N. Watanabe, H. Okada, H. Yamakawa and T. Mohri. (2000). "Situation Decomposition Algorithm for Autonomous Learning," Proc. RWC2000, pp. 219-222.
- [2] 山川,岡田, 渡部. (1999). "規則性を持つ部分データを抽出するアルゴリズムの提案," Proc. IBIS'99, pp.75-80.
- [3] 伊藤, 岡田,山川,大森. (2000). "Matchable 状況分解に基づく Multi-module 強化学習による移動ロボットの経路探索," 信学技報, NC99-99, pp. 47-54.
- [4] 山川, 岡田 馬場. (2000). "認知距離を用いた問題解決器の提案," Proc. MACC2000.
- [5] 山川, 岡田, 渡部, 松尾. (1998). "一般化状態における協調と競合のためのエージェント内部機構," Proc. MACC'98.
- [6] 山川, 岡田, 渡部, 松尾. (1998). "実世界で自律学習を行うための分散知能アーキテクチャ -CITTA-, " 第 12 回人工知能学会全国大会, pp.455-456.