Soar アーキテクチャにおける強化学習によるルール生成機構の実装

保知 良暢† 大囿 忠親†† 新谷 虎松††
Yoshinobu BOCHI Tadachika OZONO Toramatsu SHINTANI
† 名古屋工業大学大学院 工学研究科 †† 名古屋工業大学 知能情報システム学科

1 はじめに

不完全または記述困難な知識が存在する状況下においても、ユーザの負担を最大限軽減し、帰納的に知識を獲得しつつ問題を解決するアーキテクチャに関する研究は多く行われており、実アプリケーションにおいても応用範囲は広いと考えられる。

Soar プロジェクトは知能一般のアーキテクチャの構築を目指して始められ、様々な研究がなされてきた [Laird 87]。汎用的問題解決器として提案された Soar は現在も多くのアプリケーションで利用されている [Rosenbloom 85][堀 94]。Soar の決定サイクルは情報収集段階と決定段階からなる。探索が行き詰まったらインパスに入り、新たなサブゴールを設定する。サブゴールが解かれるとチャンクを生成し、知識コンパイルによる問題解決の効率化が図れる。得られたチャンクは一般のプロダクションと同様に扱われ、以後の問題解決に利用できる。Soar の処理系はプロダクションシステムで構築されており、ユーザが予め与えたプロダクションまたはチャンキングにより獲得したプロダクションを利用し問題解決を図る。Soar では問題空間を予め記号で定義するため、オペレータや世界に関するモデルが既知である問題に適用でき、抽象的なレベルでのプランニングが可能である。Soar は記号で閉じた世界において高速なプランニングが可能だが、帰納的な学習機構を持たないため状態遷移が未知である環境や物体を含むような実世界には適用が困難である。

本研究では、Soar アーキテクチャの欠点を補うために強化学習 [Sutton 98] を利用する。強化学習は試行錯誤を繰り返し望ましい状態に対して報酬という数値化された信号を受け取ることで学習する。世界のモデルが未知である問題で高い性能を示すため、Soar に組み込むことでヒューリスティックスが不十分な場合でも帰納的に学習できる問題解決器になると考えられる。以下では強化学習による情報収集スレッドを追加することにより、帰納的な学習帰納を備えたシステムを提案する。Soar では強化学習により得られた知見を用いることで適切なプランニングができる。また提案システムを Eaters ゲームにおいて実装し評価する。

2 強化学習による情報収集スレッドを持つ Soar アーキテクチャ

図 1 に提案システムで実行する決定サイクルを示す。情報収集段階ではとりうる全ての行動や遷移先の状態に対する選好性を生成する。提案するシステムは従来の Soar の機能以外に強化学習による情報収集スレッドを持つ。2 つの情報収集スレッドは並列して実行することができる。いずれかのスレッドにおいて適切な選好性を生成 · 収集した場合決定段階に移る。決定段階では従来の Soar と同様に、文脈スロットを選択する。選択するものが決まらなければインパスに入る。従来のシステムでは選好性解析においてインパスに入るが、提案システムでは未知の環境を知覚した場合においてもインパスに入る。この場合強化学習による情報収集スレッドにおいて学習が収束した場合においてインパスが解消される。

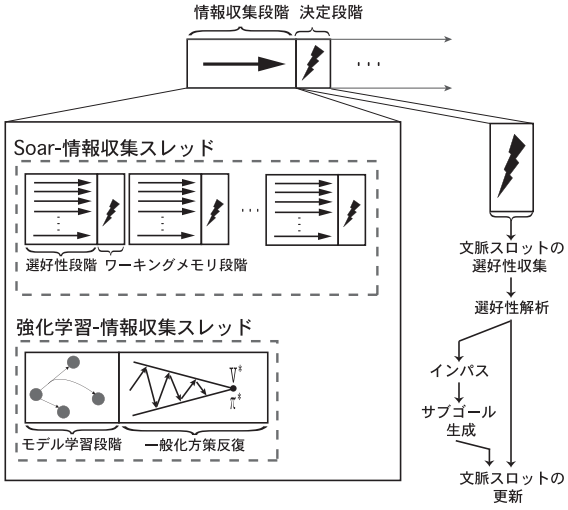

図 1: 強化学習スレッドを持つ決定サイクル

図 2 に強化学習による情報収集スレッドを用いた学習サイクルを示す。強化学習による情報収集スレッドでは内部的にモデルを生成する。本論文でモデルとは状態遷移に関する確率, 規則を指す。学習の初期段階では環境とのインタラクションを盛んに行い、その経験からモデルの学習を行う。獲得したモデルに基づき、強化学習アルゴリズムによる一般化方策反復が行われて目標に対する最適政策を学習する。このスレッドは未知の環境や物体を含む世界で有効と考えられる。Soar の重要な機能であるチャンキングはインパス発生から解消までの知識コンパイルだけでなく、強化学習による情報収集スレッドにより得られた知識に対しても適用する。学習するモデルと状態 · 行動価値関数は実装上木構造またはテーブル型となっている。それぞれには全状態, 行動に対して遷移確率や価値が付加されており、大きな記憶容量が必要となる。学習が収束した時点でプロダクションの形に置き換え、変数化を行うことで知識コンパイルを行う。チャンキングにより帰納的な学習が可能である。内部モデルと状態 · 行動価値関数から獲得した知識を Soar システムに反映することで適切なプランニングが可能となる。

Soar は、ワーキングメモリに知覚した状態オブジェクトを保持し、定義されたオペレーションを実行する。図 2 のサイクルにも同様の状態オブジェクトとオペレーションを用い

Implementation of A Rule Generator based on Reinforcement Learning in The Soar Architecture
†Graduate School of Engineering, Nagoya Institute of Technology
††Dept. of Intelligence and Computer Science, Nagoya Institute of Technology

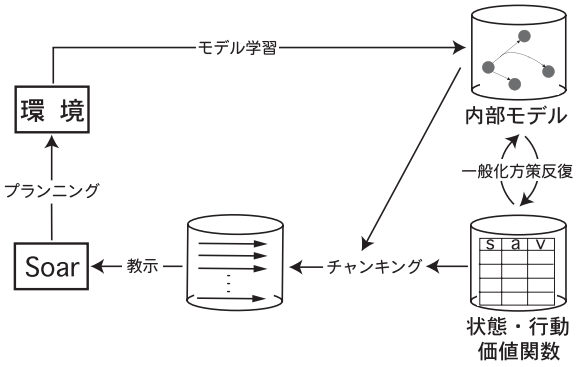

図 2: 強化学習-情報収集スレッドを用いた学習サイクル

る。つまり Soar と強化学習によるスレッドに共通の状態オブジェクトとオペレーションを定義し、実装する。図 2 ではある状態とオペレータにより次状態に遷移したとき、これを事例として内部モデルの状態遷移確率を更新する。内部モデルに基づき強化学習アルゴリズムを実行する際にも、知覚する状態と実行可能な行動は Soar と同様のものを用いる。

3 Eaters ゲームにおける実験

Eaters ゲームにおいて提案するアーキテクチャを実装した。図 3 はシミュレータを示す。eater は、画面上に点で表されている food を得るか他の eater と衝突することで得点または失点する。一定時間後に最も点数の高い eater が勝者となる。eater は自身が存在する位置とその周辺を知覚する。しかし本シミュレータではグレー部分はセンサーから情報が得られない。よってこの部分では実際に行動し、得られた経験から壁の位置等を学習しなければならない。

このシミュレータ上で提案システムを実装した。グレー部分では強化学習による情報収集スレッドで学習される。内部モデルの初期状態は行動実行によりランダムに隣接位置に移動かその場に止まる。内部モデルが不十分の間は外界に対して実際に行動し、得られた経験をモデル学習に用いる。一般化方策反復では food で獲得した得点を報酬とし、グレー部分から外に出ると 1 エピソードが終了する。また本シミュレータに特化した機能として、他の eater の戦略を帰納的にモデル学習するスレッドを実装した。チャンキング時に変数化を適切に行うことで戦略が分かる。獲得した知識はプランニングに用いることができる。実験では強化学習による情報収集スレッドによりモデルと方策が学習され、またそれは外界に対して適切であることを示した。

4 おわりに

強化学習による情報収集を行うことにより、試行錯誤で学習した結果をプランニングに利用するシステムを構築した。この情報収集スレッドは Soar に対して教示する働きを持つ。Soar は自身の知識により抽象レベルでの高速なプランニングが可能となる。

強化学習には問題空間が大規模になると状態空間の爆発が起こるという学習上の困難性がある。Soar と同様の状態オブジェクトを扱う場合、学習に不要な情報は削除しなければならない。提案システムでは予めそのような情報は削除して強化学習を行うように実装した。自律的にどの情報を用いて学習するかを決定するアルゴリズムが必要と考える。

提案システムにおけるチャンキングの目的は、単なる高速実行だけではない点が従来と大きく異なる。試行錯誤から得

図 3: Eaters ゲームシミュレータ

られた情報を、記憶容量の減少と規則性の発見を目的に知識コンパイルをチャンキングにより行う。そのためチャンキングメカニズムは従来に比べ複雑になる。実験では Eaters ゲームに特化しているため、汎用的なメカニズムを示す必要がある。

Soar のメカニズムでは新たな概念を学習することはできない。提案システムでは、未知のモデルについてその状態遷移に関する知識を帰納的に学習することはできるが、未知の物体や不完全な知識における対処は十分ではない。この問題に対しては人間からのガイダインスを用いる研究が盛んである。人間からシステムに対する一方向のガイダンスでなく、Soar が人間に対してどのような情報が知りたいかを提示する機能が必要ではないかと考える。環境とのインタラクションによりモデルが不完全な部分を特定する機能を実装する必要がある。

参考文献

[Laird 87] Laird, J. E., Newell, A., and Rosenbloom, P. S.: Soar: An architecture for general intelligence, *Artificial Intelligence*, Vol. 33, pp. 1–64 (1987).

[Laird 91] Laird, J. E., Yager, E. S., Hucka, M., and Tucc, C. M.: Robo-Soar: An integration of external interaction, planning and using Soar, *Robotics and Autonomous Systems*, Vol. 8, pp. 113–129 (1991).

[Morimoto 01] Morimoto, J. and Doya, K.: Acquisition of Stand-up Behavior by a Ral Robot using Hierarchical Reinforcement Learning, in *Proceeding of the 18th International Conference on Machine Learning*, pp. 623–630 (2001).

[Rosenbloom 85] Rosenbloom, P. S., Laird, J. E., McDermott, J., Newell, A., and Orciuch, E.: R1-Soar : An experiment in Knowledge-intensive programming in a problem solving architecture, *Pattern Analysis and Machine Intelligence*, Vol. 7, pp. 561–569 (1985).

[Sutton 98] Sutton, R. S. and Barto, A. G.: Reinforcement Learning—An Introduction—, The MIT Press (1998).

[堀 94] 堀, 池田：小特集「Soar プロジェクト」にあたって, 人工知能学会誌, Vol. 9, No. 4, pp. 478–504 (1994).