2H1-1

部分観測環境のパラメトリック記述に基づく高速モデルパラメータ逆強化学習プログラム

Efficient Model-Parameter Inverse Reinforcement Learning Program with Parametric Modeling of Partially Observable Environments

牧野 貴樹 [*1] 城 真範 [*2,*1] 合原 一幸 [*1]
Takaki Makino Masanori Shiro Kazuyuki Aihara

[*1]東京大学 生産技術研究所
Institute of Industrial Science, the University of Tokyo

[*2]産業技術総合研究所 ヒューマンライフテクノロジー研究部門
Human Technology Research Institute, National Institute of Advanced Industrial Science and Technology

The model-parameter inverse reinforcement learning, which estimates unknown parameters in partially observable environments from the demonstration of an expert, suffered from two difficulties, that is, heavy use of computational resource and lack of the framework for describing environments with unknown parameters. We present a new software for model-parameter inverse reinforcement learning LUKE (Learning Underlying Knowledge of Experts), which is aiming at solving the two difficulties.

1. はじめに

近年、強化学習 [Sutton 98] の問題設定を自動化する枠組みとして、徒弟学習 [Ng 00] が注目されている。強化学習を実用的問題に適用するときには、対象とするタスクに対し、どのような状態空間や遷移確率を考えるか、そして、何にどの程度の報酬を与えるか、といった適切な問題設定が必要であるが、実際の問題に対してどう問題設定するかは自明ではない。徒弟学習は、そのタスクのエキスパートがタスクを実行する様子 (演示) を利用して問題設定を行うことで、タスクの高度な模倣を実現しようというものである。

しかし、徒弟学習を利用して実用的な問題に取り組むためには、問題に合わせた適切なモデルとパラメータを設計し、実験するためのフレームワークが必要となる。例えば、医師の治療記録から最適な治療方針を抽出する研究 [Hauskrecht 00, 城 13] に逆強化学習を適用する [麻生 13] 場合、医師の複雑な治療をどのようなモデルで表現するかについて、様々なケースでテストしたうえで、最適な問題設定に対して大規模なデータを使って学習を実行したい。

筆者らは、計算効率の面、またモデル記述の 2 つの面で従来の手法を改良した、徒弟学習のフレームワークとなるソフトウェア LUKE (Learning Underlying Knowledge of Experts) を開発した。本ソフトウェアを利用することで、実用的な問題に対しても効率的な研究遂行が期待できる。本稿では、背景として徒弟学習によるモデルパラメータ推定の問題を概観したあと、従来ソフトウェアの問題と、それを解決した LUKE について説明する。

2. 背景

車の自動運転など、あるタスクをコンピュータで実行したいが、そのためのプログラムを記述するのが難しいという場合は少なくない。しかし、そのタスクを実際に実行した履歴（演示）を入手することは、多くの場合、容易である。エキスパートにより実行した履歴を使うことで、プログラムを記述せずにタスクを獲得できれば、非常に有効である。

単純には、演示に対して教師つき学習を適用することが考えられる。どの場面でどの行動を選択したかをそのまま模倣することで、タスクを再現することに相当するが、複雑なタスクにおいては、すべての場面を網羅した履歴を準備するためには膨大なデータ量が必要であり、事実上不可能である。

一方、プログラムを記述するかわりに、試行錯誤から最適な行動を学習することでタスクを獲得しようとする枠組みが、強化学習 [Sutton 98] である。行動の良さを表す報酬を定義することで、学習エージェント自身がさまざまな行動をためし、結果を観測することを通して、最適な行動を獲得することが可能であることから、多くの注目を集めている。

しかし、試行のコストが高い場合には適用が難しい。試行の代わりに演示データを使うことも可能であるが、演示データには適切な行動しか含まれない場合が多く、どの行動が不適切であるかを知ることは難しい。シミュレーションを利用して学習することで試行錯誤のコストを削減できるが、その場合には、シミュレーションを適切に実現できるような環境に関する知識が必要になる。

さらに、強化学習がうまく実行されるようにするには適切な問題設定や報酬関数の設定が必要であるが、実際にはそれも簡単ではない。車の運転を例にとって考えると、目的地に早く着くとよいという報酬関数を設定すれば、極限まで早く着く解が出力されるだろうが、実際には燃費や安全性といった多くの要素のバランスをとることが求められる。さらに、直接観測できないような要素 (物陰の歩行者など) を仮定する場合には、それらかどのようにふるまうか、といった内容も与えてやらなければ、適切に学習することができない。

逆強化学習 [Ng 00] は、エキスパートの学習結果から、強化学習の報酬関数を設定する手法の総称である。ここでは、エキスパートが報酬関数を知っており、その報酬を最大化するように行動を選択していると仮定することで、エキスパートの演示から未知の報酬関数を推定する。具体的には、エキスパートが、ある状態においてどの行動を選んだかという情報から、その状態における行動価値関数の大小関係を知ることができ、その情報をもとに、報酬関数を推定できるのである。特に、完全観測問題として定式化できるような環境である、自動車の運

連絡先: 牧野 貴樹, 東京都目黒区駒場 4-6-1 東京大学生産技術研究所 Ce-605, mak@sat.t.u-tokyo.ac.jp

転 [Abbeel 06]、ヘリコプターのアクロバット飛行 [Abbeel 07] といった問題において、目覚ましい成果を挙げているほか、自然言語の分析順序の学習 [坪井 13] の研究なども進められている。さらに、部分観測環境の報酬関数の推定に拡張することも研究されている [Choi 11]。

牧野らの提案 [Makino 12] は、この逆強化学習を一般化し、報酬関数だけでなく、環境モデルのパラメータ、すなわち遷移行列・観測行列の値に関しても推定しようとするものである。特に、部分観測問題においては、完全観測問題とは異なり、環境に多くの未知の要素が含まれるため、環境のモデルに適切なパラメータを設定することが困難である。しかし、エキスパートが報酬関数に加え、環境の未知のモデルに関しても正しい知識を持っており、その環境の上で報酬が最大になるよう行動を選択していると仮定することで、エキスパートの演示から、エキスパートが持っている環境のモデルのバラメータを推定することが可能になる。

3. 従来ソフトウェアの問題

しかし、この一般化した逆強化学習は、これまでの逆強化学習に比べ、計算コストが大きいことが課題となっていた。遷移行列や観測行列が既知で、報酬関数のみが未知であったときは、各行動の最適価値関数は報酬関数の値の線形和で書けるため [Abbeel 04]、エキスパートの行動から得られる最適価値関数の情報をもとに報酬関数の情報を推定することは容易であり、そのため大規模な問題に対しても適用することが可能であった。一方、遷移行列や観測行列の値は、価値関数と非線形に関連しているため、エキスパートの行動だけから未知の遷移行列・観測行列のパラメータを推定することは簡単ではない。

牧野らによるパラメータ推定 [Makino 12] においては、仮定されたパラメータ値であらわされる強化学習問題を解くことで、演示に含まれる行動列の尤度を計算し、観測列の尤度と合わせ、演示の事後確率が最大となるパラメータ値を探索するというプロセスで推定を行っていた。しかし、特に部分観測問題の場合、強化学習問題を解くこと自体が簡単ではなく、様々な研究により高速化したソルバーが開発されているものの、かなり計算がかかることに変わりはない。パラメータ最適化においては、相当数の反復計算を実行しなければいけないため、非常に時間がかかる。

また、もう一つの問題は、パラメータ値の変更がどのように結果に影響するかが自明ではないことである。部分観測問題ソルバーの多くが確率的挙動に基づいていること、また、ソルバーの速度の問題を解決するためには完全に収束する前に探索を中断しなければならないことから、ほとんど同じパラメータ値であっても、得られる結果が異なることが起こりうる。これは、目的関数の評価に大きなノイズが乗ることと同じであり、そのため、事後確率最大となるパラメータを探索することは非常に難しくなる。

また、実際の問題に適用するためには、もうひとつクリアしなければならない問題があった。それは、未知パラメータを含むモデルを適切に記述するための枠組みである。これまで、部分観測環境を記述するためのファイルフォーマットはいくつか提案されてきた [Cassandra 03] が、未知のパラメータを含むようなケースは記述できなかった。しかも、要求を詳細に検討すると、これまでのフォーマットでは対応が難しい多くの要素があることが明らかになった。(1) 未知パラメータの各々に対して、事前分布を記述できること、(2) 遷移行列・観測行列・報酬行列の各要素に、未知のパラメータで表される数式が記述できること、(3) 状態空間・行動空間・観測空間を、部分空間に分解した形で記述でき、遷移行列・観測行列・報酬行列がこれらの小空間に対して定義できること (pomdpX 形式 [Kurniawati 08] と同様)、(4) 複雑なパラメータ指定の際に、遷移行列・観測行列が満たすべき条件 (各列の和が 1 になる等) を守ることが簡単に実現できるようにすること、である。そこで、これらの要求を満たすために、モデルパラメータ徒弟学習のための新たなソフトウェアを開発することとなった。次節でこのライブラリの詳細を説明する。

4. LUKE

LUKE (Learning Underlying Knowledge of Experts) は、モデルパラメータを含めた徒弟学習を実現するためのソフトウェアである。

特徴の第 1 は、新規に開発した方策反復型の POMDP ソルバーを搭載していることである。徒弟学習において、方策反復型の POMDP ソルバーを利用することのメリットには、次の 2 点がある [Makino 13]: (1) 尤度値の劣勾配が計算できることから、最急降下法等の効率的な最適化手法を適用可能となり、必要な評価回数が削減できること。(2) パラメータ値を変更した場合に、前回の問題の解である方策を探索開始地点として利用することで、探索に必要な時間を削減できる可能性があること。しかし、これまで公開されている POMDP ソルバーは、価値反復型のものが多く、徒弟学習の高速化に利用することができなかった。本ソフトウェアは、方策反復に基づく実用的な POMDP ソルバーとして、公開されている唯一のものであり、方策反復が有効となる様々な場面にも応用できるものと考えている。

特徴の第 2 は、記述のための要求を満たすために、未知パラメータを含む POMDP モデル記述のための新たなファイルフォーマットを定義したことである。基本的な行列の書式は、MATLAB/Octave のサブセットの形式が利用可能であるが、未知のパラメータに対するシンボルを（事前分布を指定することで）定義し、式中で自由に利用できる。さらに、複雑な式を含む行列でも各列の和を 1 にする作業を簡単にするため、特別なプレースホルダー変数を用意し、関数 `allot` を通すことで各列の和が 1 になるようなプレースホルダー変数の解を求め、割り当てる機能を提供している。

そして、畳み込み演算を利用することで、部分空間で定義された遷移行列・観測行列・報酬行列を統合することを実現している。これらは、実数密行列であれば MATLAB/Octave 上でも実行可能であることから、これらのソフトウェア上で確認しながらモデル記述を作成することができる。

図 1 で、4 × 4 の格子空間上で、一条一丁目にある塔を探すタスクを記述した例を示す。南北と東西の部分空間に分解することで、コンパクトにモデルを表現できていることがわかる。

5. 実験

実装したルーチンが効率的であることを確かめるため、単純なタスクである一般化 Tiger タスクに対して、徒弟学習計算速度とともに、学習結果に基づくタスク実行の平均報酬を比較した。実験設定は先行研究 [Makino 12] と同一で、100 ステップの演示からの学習を 100 回実行し平均を測定している。

表 1 に示す通り、従来ソルバーを利用した場合と比較して、LUKE を利用したほうが高速に解にたどりつくことが可能になっている。また、勾配を利用した最適化アルゴリズム (+ grad) を利用することで、さらに高速化するとともに、解の質

```
NS = StateSet('Ichijo', 'Nijo', 'Sanjo', 'Shijo');
                          % Subspace for N-S direction
EW = StateSet('Iccho', 'Nicho', 'Sancho', 'Yoncho');
                          % Subspace for E-W direction
STATES = Ichijo+Icchome:Shijo+Yonchome;  % Colon operator
ACTIONS = ActionSet('N', 'S', 'E', 'W');
OBSERVES = ObservationSet('SeeTower', 'SeeNoTower');

MoveOK = Beta( 5, 2 ); % parameter for successful move prob.
                       % with a prior of Beta distribution
SeeTowerAtOrigin = Beta( 4, 2 );
SeeTowerOnStreet = Beta( 3, 2 );
SeeTowerAtOther  = Beta( 2, 3 );
RewardForTower   = Normal(100, 50);
TransNS( NS, NS, N ) = [1-MoveOK 0 0 MoveOK ; MoveOK ...
  1-MoveOK 0 0 ; 0 MoveOK 1-MoveOK 0 ; 0 0 MoveOK 1-MoveOK ];
TransNS( NS, NS, S ) = TransNS( NS, NS, N )'; % Transpose
TransNS( NS, NS, E ) = eye(4);          % Identity Matrix
TransNS( NS, NS, W ) = eye(4);
TransEW( EW, EW, E ) = TransNS( NS, NS, N );  % Reuse
TransEW( EW, EW, W ) = TransEW( EW, EW, E )';
TransEW( EW, EW, N ) = eye(4);
TransEW( EW, EW, S ) = eye(4);
Trans = sconv2( TransNS, TransEW ); % Convolution Operator

Obser( SeeNoTower, :, : ) = PlaceHolder;
Obser( SeeTower, :, : ) = SeeTowerAtOther;
Obser( SeeTower, Ichijo+EW, : ) = SeeTowerOnStreet;
Obser( SeeTower, NS+Iccho, : ) = SeeTowerOnStreet;
                % states either on Ichijo or Iccho
Obser( SeeTower, Ichijo+Iccho, : ) = SeeTowerAtOrigin;
Obser = allot(Obser);  % Set every placeholder to be sum=1

Rewar( :, :, : ) = -1;
Rewar( Ichijo+Iccho, :, :) = RewardForTower;
Init(STATES) = 1.0 / 16;
Discount = 0.9;
```

図 1: `demo.model`: 4 × 4 空間上で塔を探すタスク

表 1: 実行速度とタスク実行の平均報酬の比較

ソルバー	計算時間 (秒)	平均報酬
SARSOP	2.55	1.074
LUKE	0.68	1.075 ± 0.044
LUKE + transfer	0.77	1.073 ± 0.045
LUKE + grad	0.34	1.084 ± 0.032
LUKE + grad + transfer	0.39	1.083 ± 0.032

がよりよくなっていることが分かる (勾配なしの場合と比べて有意差あり。不等分散の T 検定、$P < .05$)。このことは、方策反復法に基づくアプローチで性能を向上させられたことを示している。一方、前回の解の再利用 (+ transfer) に関しては、解の質という意味ではほぼ同じであったが、計算速度の低下がみられた。問題サイズが小さすぎるため、再利用のメリットよりオーバーヘッドのほうが大きくなったことが考えられ、より大規模な問題で比較する必要があると考えている。

6. まとめ

徒弟学習の研究について概観するとともに、モデルパラメータの徒弟学習を高速化し、より使いやすくするためのソフトウェア LUKE について紹介した。LUKE は GPL に基づきフリーウェアとして公開中である [Makino 14]。より大規模なモデル研究に利用できるよう、今後とも開発を継続していきたい。

謝辞 本研究は、科学研究費補助金 (25730128)、および総合科学学術会議により制度設計された最先端研究開発支援プログラム (FIRST 合原最先端数理モデルプロジェクト) により、日本学術振興会を通じて助成を受けた。

参考文献

[Abbeel 04] Abbeel, P. and Ng, A. Y.: Apprenticeship learning via inverse reinforcement learning, in *Proceedings of the 21st International Conference on Machine Learning*, ICML '04, pp. 1–8 (2004)

[Abbeel 06] Abbeel, P., Quigley, M., and Ng, A. Y.: Using Inaccurate Models in Reinforcement Learning, in *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 1–8 (2006)

[Abbeel 07] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y.: An application of reinforcement learning to aerobatic helicopter flight, in *Advances in Neural Information Processing Systems (NIPS) 19*, pp. 1–8 (2007)

[麻生 13] 麻生 英樹, 城 真範, 神嶌 敏弘, 赤穂 昭太郎, 興梠 貴英：逆強化学習による医療臨床データの分析, 電子情報通信学会技術研究報告, Vol. 112, No. 390, pp. 13–17 (2013), NC2012-96

[Cassandra 03] Cassandra, A. R.: `pomdp-solve` Input POMDP File Format, `http://www.pomdp.org/code/pomdp-file-spec.shtml` (2003)

[Choi 11] Choi, J. and Kim, K.-E.: Inverse Reinforcement Learning in Partially Observable Environments, *Journal of Machine Learning Research*, Vol. 12, pp. 691–730 (2011)

[Hauskrecht 00] Hauskrecht, M. and Fraser, H.: Planning treatment of ischemic heart disease with partially observable Markov decision processes, *Artificial Intelligence in Medicine*, Vol. 18, pp. 221–244 (2000)

[Kurniawati 08] Kurniawati, H., Hsu, D., and Lee, W.: SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces, in *Proc. Robotics: Science and Systems* (2008)

[Makino 12] Makino, T. and Takeuchi, J.: Apprenticeship Learning for Model Parameters of Partially Observable Environments, in *Proc. of the 29th International Conference on Machine Learning (ICML)*, pp. 1495–1502 (2012)

[Makino 13] Makino, T., Oda, Y., and Aihara, K.: New Optimizer Algorithm for Model Design in Partially Observable Environments, 生産研究, Vol. 65, No. 3, pp. 315–318 (2013)

[Makino 14] Makino, T.: LUKE: Software for Learning Underlying Knowledge of Experts, `http://www.snowelm.com/~t/research/software/luke.ja.html` (2014)

[Ng 00] Ng, A. Y. and Russell, S. J.: Algorithms for Inverse Reinforcement Learning, in *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670, San Francisco, CA, USA (2000)

[城 13] 城 真範, 神嶌 敏弘, 赤穂 昭太郎, 麻生 英樹, 興梠 貴英：強化学習による心疾患臨床データの分析, 第 28 回人工知能学会全国大会論文集, pp. 1E4–5 (2013)

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

[坪井 13] 坪井 祐太, 牧野 貴樹：自然言語処理における逆強化学習・模倣学習の適用, 計測と制御, Vol. 52, No. 10, pp. 922–927 (2013)