一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

状態空間の分割と状態遷移の学習に基づく Parallel POMDP の評価

山田 燿司† 滝口 哲也‡ 有木 康雄‡

†神戸大学大学院システム情報学研究科 〒 657–8501 兵庫県神戸市灘区六甲台町 1–1
E-mail: †y.yamada@me.cs.scitec.kobe-u.ac.jp, ††{takigu,ariki}@kobe-u.ac.jp

あらまし 部分観測マルコフ決定過程 (POMDP)を対話システムに適用する際の問題点として、状態数の増加に伴い、計算量が指数的に増加することが挙げられる。この問題を解決するために、我々はこれまで、この POMDP を階層的に配置した Hierarchical POMDP (H-POMDP)を対話システムに適用したが、POMDP 間の遷移が制限される問題があった。この点から、本研究では、POMDP を並列に配置することで、POMDP 間の遷移をより自由にした、Parallel POMDP (P-POMDP)を提案する。また、この POMDP 間の遷移の学習を行い、より適切なシステムの発話を選択する。従来の POMDP と H-POMDP、P-POMDP の 3 種類、それぞれのタスク達成率と計算量を比較することで、P-POMDP の有用性を証明する。

キーワード POMDP，音声対話システム，報酬関数，状態空間

Evaluation of Parallel POMDP based on State Space Partition and State Transition Learning

Yoji YAMADA† Tetsuya TAKIGUCHI‡ and Yasuo ARIKI‡

†Graduate School of System Informatics, Kobe University Rokkodaicho 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
E-mail: †y.yamada@me.cs.scitec.kobe-u.ac.jp, ††{takigu,ariki}@kobe-u.ac.jp

Abstract Significant problems with Partially Observable Markov Decision Process (POMDP) lie in a large number of states, causing a long computation time and difficulty of system development. In order to solve these problems, we have applied Hierarchical POMDP (H-POMDP) to the spoken dialogue system (SDS) so far. However, in H-POMDP, the transitions between the states are restricted because of the partition of state space and the hierarchical structure. In this paper, we propose Parallel POMDP (P-POMDP) in which small POMDPs are scattered and they work together in parallel to relax this restriction. In P-POMDP, the reward function is newly formalized to make the system reply optimally. In the experiment, we compared the proposed P-POMDP with H-POMDP and the conventional POMDP in terms of task achievement rates and computation times.

Keywords POMDP, Spoken Dialogue System, Reward Function, State Space

1. はじめに

近年、レストラン検索[1]やカーナビゲーションシステム[2]など、タスク指向型の音声対話システムを構築する際に、POMDP などの強化学習が多く用いられている[3][4][5]。しかし、POMDP の対話システムへの適用において、状態数の増加に伴い、方策の計算量も指数的に増加するという問題が挙げられる。この問題を解決するために、我々はこれまで、この POMDP を階層的に配置した、Hierarchical POMDP (H-POMDP)を対話システムに適用した。しかし、H-POMDP の階層構造が POMDP 間の遷移を制限する問題が生じた。

そこで、階層構造の代わりに、POMDP を並列に配置した Parallel POMDP を提案する。これにより、POMDP 間の遷移をより自由に行うことができる。

また、この P-POMDP に対し、商品の検索を目的とした対話タスクを適用する。商品とは、食品や CD、本などの売り買いの対象となる物品を指すが、今回は本の検索のみをタスクとして設定する。システムはユーザとの対話を通して、ユーザの意図に適した本を提示する。また、報酬の学習を本のラベルデータを用いて行い、最適なシステムの発話を選択する。

さらに、この P-POMDP の有用性を証明するために、従来の POMDP と H-POMDP、P-POMDP の 3 種類、それぞれのタスク達成率と計算量を比較し、システムの評価を行う。

Sys : どんな本をお探しでしょうか? (<value>="", <slot>="カテゴリ")
Usr : 小説が読みたい. (<value>="小説", <slot>="カテゴリ")
Sys : ジャンルは何でしょうか? (<value>="", <slot>="ジャンル")
Usr : 推理がいいな. (<value>="推理", <slot>="ジャンル")
Sys : 年はいくつですか? (<value>="", <slot>="年齢")
Usr : 22歳です. (<value>="20代", <slot>="年齢")
Sys : では、こちらがおすすめです.

図 1 : 対話例

2. 商品検索型音声対話システム

膨大な数の商品の中から、ユーザが求めているものを、対話を通じて条件を絞り、検索を行う対話システムについて考える。

例えば、本は小説や雑誌、参考書などのカテゴリに区分される。さらに、小説は推理小説や歴史小説などのジャンルに分けられる。また、読者の年齢層や性別により、好まれる本が異なってくる場合もある。このように、本は“カテゴリ”や“ジャンル”などの slot、それら slot に対する“小説”や“推理”などの value に分類することができる。ユーザがシステムからの質問に答えることで、これら value を特定し、slot を埋め、本の検索の条件を絞る。こうして、ユーザの意図を推定し、最終的におすすめの本を一冊紹介してくれるシステムを構築する。3 つの value、“小説”、“推理”、“20代”が 20 代向けの推理小説を特定している例を図 1 に示す。

3. Hierarchical POMDP

POMDP の問題点として、状態数が増大するにつれて計算量が指数的に増えることが挙げられる。そこで図 2 のように、POMDP を階層的に構築し、状態空間を分割した Hierarchical POMDP (H-POMDP) を用いた研究がある[6][7][8]。タスクを複数の部分問題に分割し、階層的に統合することで、従来よりも複雑なタスクを実現することが可能となる。

まず、最上位の POMDP1 から開始してその内部で状態の遷移が生じ、POMDP1 における最終目的となる状態にたどり着く。その後、次の POMDP2 もしくは POMDP3 に遷移する。こうして POMDP 間の遷移を続け、最下位の POMDP において最終目的の状態にたどり着いた場合、タスク全体のゴールとなる目的の状態に到達する。

この H-POMDP を商品検索タスクに適用した例を示す。まず、最上位に位置する POMDP は“本の種類”を決定するための POMDP で、その構成として、状態を{“小説”, “雑誌”}の集合、観測値を{“小説”, “雑誌”}、行動を{“確認する”, “POMDP2 へ遷移”, “POMDP3 へ遷移”}と設定する。“確認する”とはユーザの発言を促す

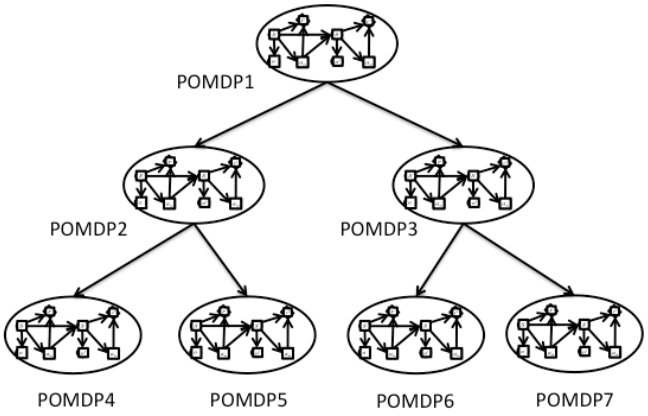

図 2 : H-POMDP の構造

行動を示す[9]。例えば、最上位の POMDP において、システムが“確認する”という行動として、本の種類を尋ねる。そして、“小説”というユーザ発話を観測して信念状態 b(s′)に更新され、“POMDP2 へ遷移”という行動を選択したとする。その後、POMDP2 の行動として、システムは小説のジャンルを尋ねる。こうして POMDP 間の遷移を続け、最下位の POMDP において、“本を提示する”という行動が選択されると、システムがユーザに本を一冊紹介し、タスク完了となる。

4. Parallel POMDP

しかし、この H-POMDP の問題点として、階層構造のため、最上位の POMDP から対話を開始する必要があり、また、一つ下位の POMDP にしか遷移ができないなど、POMDP 間の遷移が制限される問題がある。これらの問題を解決するために、図 3 のように POMDP を並列に配置した、Parallel POMDP (P-POMDP)を提案する。これにより、さらに多くの POMDP 間の遷移が可能となり、より柔軟な対話システムが実現できる。

4.1. P-POMDP の構成

システムはどの POMDP からでも対話を開始できる。例えば、図 3 において、POMDP2 から対話を開始した場合を考える。POMDP2 はユーザの発話により信念状態を更新し、POMDP2 の目的の状態にたどり着く。その後、その状態に応じて POMDP1、POMDP3、もしくは POMDP4 に遷移する。このような POMDP 間の遷移を、タスクが完了するまで続ける。

この P-POMDP を商品検索タスクに適用した例を、図 4 に示す。“カテゴリ”や“ジャンル”など、slot に相当する POMDP が存在し、それぞれ“小説”や“雑誌”、“推理”や“歴史”などの value に相当した状態を持つ[10]。また、“カテゴリ”という POMDP から、状態に応じて、“ジャンル”や“年齢”、“雑誌”、“性別”の POMDP に遷移することを表す。POMDP 間の遷移を

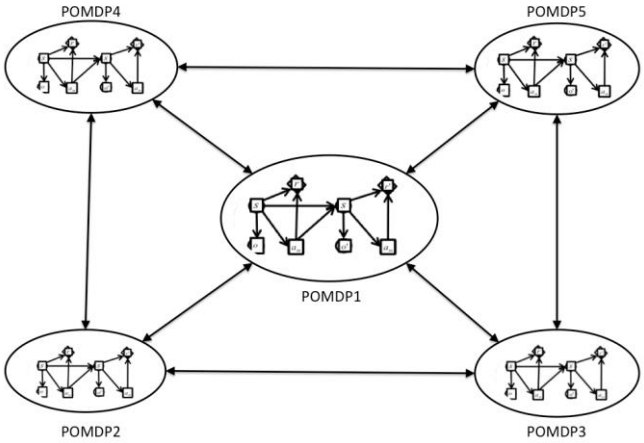

図 3：P-POMDP の構造

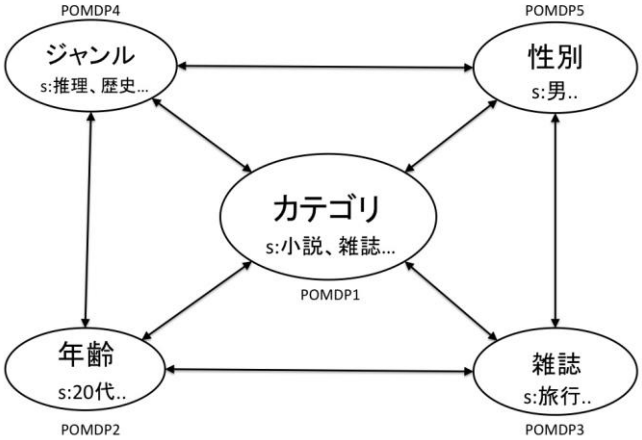

図 4：P-POMDP の適用例

続け、それぞれの value を特定し、特定された value を全て含む本が見つかれば、システムはそれを提示する。

しかし、これら POMDP 間の遷移において、どの POMDP から POMDP への遷移が適切かを決定する必要がある。そこで、本のラベルデータを基に、POMDP 間の遷移を学習することを考える。

4.2. POMDP 間の遷移の学習

表 1 のように、“状態(=value)”と“POMDP(=slot)”の組のラベルデータを持つ本がある。例えば、Book 1 は“小説”、“推理”、“20 代向け”の 3 つの状態を持つ。これらはそれぞれ、“カテゴリ”、“ジャンル”、“年齢”の POMDP に対応する。この本を検索するためには、これら 3 つの POMDP に遷移し、それぞれの value を特定する必要がある。そのため、この 3 つの POMDP には関連があり、これらの POMDP 間での遷移が可能であることが言える。例えば、“カテゴリ”の POMDP に含まれる“小説”という状態において、システムは行動として、“ジャンルの POMDP に遷移”もしくは“年齢の POMDP に遷移”を選択する。このように、本のラベルデータを基に POMDP 間の遷移を決定し、さらにその遷移の数を合計することで、遷移関数を作成する。

表 1：本のラベルデータの例

Book1		Book2	
状態 s	POMDP	状態 s	POMDP
小説	カテゴリ	小説	カテゴリ
推理	ジャンル	歴史	ジャンル
20 代	年齢	男性	性別

表 2：遷移関数 p(a|s)の例

現在の POMDP	状態 s	行動 a：次の POMDP に遷移			
		カテゴリ	ジャンル	年齢	性別
カテゴリ	小説	-	0.5	0.25	0.25
ジャンル	推理	0.5	-	0.5	0
:	歴史	0.5	-	0	0.5
年齢	20 代	0.5	0.5	-	0
性別	男	0.5	0.5	0	-

表 3：報酬 r(s,a)の例

		行動 a				
		次の POMDP に遷移				
現在の POMDP	状態 s	カテゴリ	ジャンル	年齢	性別	確認する
カテゴリ	小説	-	5	2.5	2.5	-1
ジャンル	推理	5	-	5	-10	-1
:	歴史	5	-	-10	5	-1
年齢	20 代	5	5	-	-10	-1
性別	男	5	5	-10	-	-1

表 1 の 2 つの本データから、表 2 のような遷移関数 p(a|s)を作成する。例えば、現在選択されている POMDP が“カテゴリ”であり、状態が“小説”の場合を考える。このとき、“小説”という状態に対し、次に遷移する POMDP として、“ジャンル”を含む本が Book1 と Book2 の 2 つ、“年齢”を含む本が Book1 の 1 つ、“性別”を含む本が Book2 の 1 つである。これは、“ジャンル”の POMDP に遷移する確率、“年齢”の POMDP に遷移する確率、“性別”の POMDP に遷移する確率の 3 つ割合が、2:1:1 であることを表している。さらに、この割合について、総和が 1 になるように正規化を行うと、0.5:0.25:0.25 となる。このように、各状態に対して、可能な遷移をラベルデータに含んでいる本を数え、遷移の比率を算出する。これにより、状態 s において行動 a を選択する確率を表す、遷移関数 p(a|s)を作成することができる。そして、この遷移関数を基に、

報酬を設定する。

報酬の例を表 3 に示す。まず、“確認する”という行動は、何度も発話を行うユーザの煩わしさを考慮し、状態に関わらず-1 の報酬を与える[9]。また、p(a|s)=0 の値は、その状態 s における行動 a の選択が不可能であることを表しているので、それに対応する r(s,a)は -10 とする。そして、p(a|s)>0 の値を取る各状態 s と各行動 a に対しては、r(s,a) = 10*p(a|s)の式を適用し、正の報酬を与える。このような条件の下、報酬の設定を行う。

5. 評価実験

P-POMDP を評価するために、従来の POMDP (C-POMDP)と H-POMDP、P-POMDP の三つの POMDP を比較する。C-POMDP は状態数が 202、行動数が 59 の大きな単体の POMDP である。H-POMDP は、この C-POMDP を 57 個の小さな POMDP に分割し、それらを階層的に構築したもので、各 POMDP の平均状態数は 3.6、平均行動数は 2.5 である。P-POMDP は、分割された 57 個の POMDP を並列に配置したもので、各 POMDP の平均状態数は 3.6、平均行動数は 3.3 である。これら POMDP の報酬は、書店のウェブサイト上にある、200 冊の本のデータから設定され、各 POMDP の方策は Point Based Value Iteration (PBVI)を用いて学習する[9][11]。

5.1. 実験設定

今回、商品検索型の音声対話システムをスマートフォンアプリケーションとして開発した。このシステムの動作画面の例を図 5 に示す。システムはユーザに質問を行い、それに対する回答をもとに value を取得し、それら value を全て含む本が見つかれば、その本を提示する。また、5 回目の対話でその条件に適合する本が見つかっていなければ、それまでに得られた value をより多く含む本を提示する。

各 POMDP を適用した対話システムについて、それぞれのタスク達成率、平均対話回数、計算量を比較した。タスク達成率は、17 人のユーザから計算され、システムが提示した本に対し、ユーザが満足すればタスク成功とする。平均対話回数は、長い対話を行うユーザの煩わしさを考慮し、最大対話回数を 5 回とする。計算量は、POMDP の方策の計算にかかる時間を表し、H-POMDP と P-POMDP は 57 個の POMDP の集合なので、各 POMDP の平均、最大、最少、合計の計算時間を測定する[12]。C-POMDP は単体の POMDP なので、合計の計算時間のみ表示する。

図 5：動作画面例

5.2. 実験結果

実験の結果を表 4 に示す。平均対話回数が短いほどタスク達成率が高く、これは、システムが 5 回以内の対話で本を見つけ出すとタスクが完了となり、対話回数が短くなるためである。C-POMDP は多くの状態を含むため、多様なユーザの発話に対応でき、タスク達成率が最も高かった。しかし、状態数が多いと、計算量も指数的に増加するため、合計の計算時間が最も長かった。H-POMDP は、計算量が最も少なかったが、遷移が階層的に制限されているため、タスク達成率が最も低くなった。これらに対し、P-POMDP は C-POMDP に近いタスク達成率を示し、さらに、H-POMDP と同程度の計算量を示した。つまり、計算量が少なく、かつ、比較的に高い確率でタスクを成功させるシステムとなった。これは、並列構造が遷移の制限を緩和し、

表 4：実験結果

	タスク達成率	対話回数	計算時間(s)			
			平均	最大	最少	合計
C-P	38.24%	4.18	-	-	-	8659.32
H-P	26.47%	4.82	0.084	4.496	0.004	4.804
P-P	35.29%	4.45	0.205	11.33	0.004	11.713

システムがより最適な行動を選択できるため、タスク達成率が上昇したからである。

しかし、全ての POMDP について、タスク達成率が 40%以下とあまり高くなかった。この理由はタスクの複雑性にある。本には小説や雑誌、技術書などの種類があり、さらに歴史や医療、コンピュータなど様々な分類が考えられる。また、実験において、ユーザが特定の本のタイトルや著者名を発話する場合もあった。これらすべての状態に対応することは非常に困難であり、今回のタスクは、従来のシンプルなタスクよりも複雑であるため、タスクの成功率が低い結果となった。そこで、今後の課題として、より多くの本を用いて実験を行い、システムが対応できる状態の数を増やし、タスクの達成率を上げる。

6. おわりに

今回、H-POMDP における POMDP 間の遷移の制限を改善するため、POMDP を並列に配置した P-POMDP を提案した。さらに、P-POMDP の遷移をデータより学習し、報酬の設定を行った。これにより、さらに自由な POMDP 間の遷移を行うことができるようになり、状況に応じた、より適切なシステムの行動を選択できるようになった。また、実験結果より、P-POMDP はタスク達成率が高く、計算量が少ないことが証明され、P-POMDP の有用性が示された。しかし、今回の実験では、本のデータが 200 冊と少なかったため、あらゆるユーザの発話には対応できず、タスク達成率がすべての POMDP において低い結果となった。そこで、さらに多くのデータを用いた実験を今後行う。

文　献

[1] S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson and K. Yu, “The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management,” Computer Speech and Language, 24, 150-174, 2010.

[2] 岸本 康秀, 滝口 哲也, 有木 康雄, “階層的強化学習を適用した POMDP によるカーナビゲーションシステムの音声対話制御,” 電子情報通信学会信学技報, 110(143), 49-54, 2010.

[3] S. Young, M. Gasic, B. Thomson and J. Williams, “POMDP-Based Statistical Spoken Dialog Systems: A Review,” Proceedings of the IEEE, 101, 1160-1179, 2013.

[4] Jason D. Williams and Steve Young, “Partially observable Markov decision processes for spoken dialog systems,” Computer Speech and Language, 21, 393-422, 2007

[5] R.S. Sutton and A.G. Barto, ``強化学習,'' 森北出版社, 2000.

[6] 岸本 康秀, 滝口 哲也, 有木 康雄, “階層的強化学習を適用した POMDP による音声対話制御,” 電子情報通信学会信学技報, 110(356), 121-126, 2010.

[7] Joelle Pineau, Nicholas Roy, and Sebastian Thrun, “A Hierarchical Approach to POMDP Planning and Execution,” Proceedings of the ICML Workshop on Hierarchy and Memory in Reinforcement Learning, 110, 121-126, 2001.

[8] Heriberto Cuayahuitl, Ivana Kruijff-Korbayova, Nina Dethlefs, “Nonstrict Hierarchical Reinforcement Learning for Interactive Systems and Robots,” ACM Transactions on Interactive Intelligent Systems, 4(3), article 15, 2014.

[9] 南 泰浩, “部分観測マルコフ決定過程に基づく対話制御,” 日本音響学会誌, 67(10), 482-487, 2011.

[10] Heriberto Cuayahuitl: Hierarchical Reinforcement Learning for Spoken Dialogue Systems, University of Edinburgh (2009).

[11] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun, “Point-based value iteration: An anytime algorithm for POMDPs,” Proceedings of the International Joint Conferences on Artificial Intelligence, 1025-1032, 2003.

[12] Minlue Wang and Richard Dearden, “Run-Time Improvement of Point-Based POMDP Policies,” Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2408-2414, 2013.