

LH-003

文脈に基づくモジュール切り替え型学習モデル A Module Selection Learning Model Based on Context

郷古 学[†]
Manabu GOUKO

菅谷 至寛[†]
Yoshihiro SUGAYA

阿曾 弘具[†]
Hiroto ASO

1. まえがき

モジュール構造を持つ制御システムでは、比較的単純な制御を実現するモジュールを複数用意し、それらのモジュールを組み合わせ、全体として複雑な制御を実現する。このようなシステムでは、モジュールの組み合わせを変えることにより、制御対象の変化への速やかな適応が期待できる。これまでモジュール構造を持つ制御システムを、学習により自動的に獲得する研究が行われている [1, 2]。これらの研究では、学習によって各モジュールに対応するように、状態 (入力) 空間をいくつかの領域に分割し (以下、分節化と呼ぶ)、それぞれの領域において必要な制御則を獲得している。

強化学習により環境の予測可能性に基づき環境を分割する複数モジュール競合アーキテクチャとして、強化学習 MOSAIC が絞島らにより提案されている [1]。強化学習 MOSAIC は複数の線形予測モデルの競合により、非線形・非定常な環境を時空間的に区分された線形な環境に分解する。さらに分解された環境に対応する制御則を強化学習により獲得する。また、西村らはモジュールとして複数の不完全制御器を用い、それらの切り替え法を強化学習で獲得することで、実アクロバット制御を実現した [2]。これらの研究ではモジュールの選択に関しては、単純マルコフ過程を前提としている。

人間の行う情報処理は、多くの場合、現在の観測状態に加えて、過去の情報 (文脈と呼ぶ) に依存している (n 重マルコフ過程)。そのため、同一の観測状態からでも、異なる判断 (出力) を行うことが出来る。そこで本論文では、このような柔軟な情報処理の実現を目指し、文脈に基づくモジュール選択が可能な学習モデルを提案する。提案モデルは、学習により分節化を行うと同時に、分節化された各領域において制御を担うモジュールを獲得する。さらにモデルは、文脈に基づき選択すべきモジュールを予測することで、観測状態だけでは適切なモジュール選択が困難な場合にも対応可能である。

2. 提案モデル

提案モデルは、ひとつのモジュール選択機構と複数のモジュールから構成される。提案モデルに入力を与えると、モジュール選択機構においてモジュールがひとつ選択され、その選択されたモジュール (以下、選択モジュールと呼ぶ) からの出力をモデル全体の出力とする。

[†] 東北大学大学院工学研究科電気・通信工学専攻

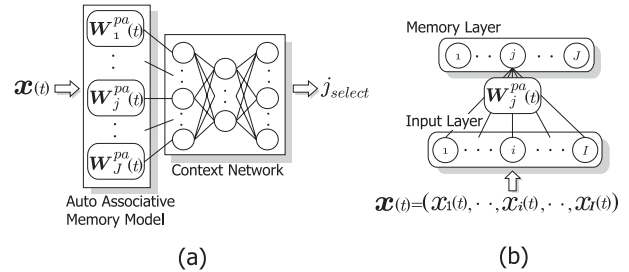


図 1: (a) モジュール選択機構, (b) 自己連想記憶モデル

提案モデルは、ある時刻 t における入力パターン $x(t)$ と出力 $y(t)$ の対から成る時系列データ $(x(0), y(0)), (x(1), y(1)), \dots, (x(T), y(T))$ を用いて学習を行う。学習により、モジュール選択機構は入力空間の分節化を実現し、各モジュールでは分節化された領域における入出力写像を学習する。

2.1 モジュール選択機構

モジュール選択機構では、入力パターン $x(t)$ から、選択モジュール番号 j_{select} を出力する (図 1(a))。モジュール選択機構は、逐次学習型時系列予測モデル ASLN (Adaptive and Sequential Learning Network) [3] を用いて構築する。

ASLN は自己連想記憶モデルと文脈ネットワークからなる教師なし学習モデルで入力時系列に対して次の入力を予測するように学習し続ける。モジュール選択機構への入力は I 次元の数値ベクトルとする。自己連想記憶モデルは、 I 個のニューロンを持つ入力層とモジュール数以上のある値 J 個のニューロンを持つ記憶層からなる (図 1(b))。入力層では入力ベクトル $x(t)$ を正規化する。以下では、入力 $x(t)$ は正規化されたものを表すとする。記憶層ニューロン j はすべての入力層ニューロンと結合しており、その結合重みをベクトル $W_j^{pa}(t) = (W_{j1}^{pa}(t), \dots, W_{ji}^{pa}(t), \dots, W_{jI}^{pa}(t))$ で表す。この重みベクトルを記憶パターンとも呼ぶ。入力 $x(t)$ と $W_j^{pa}(t)$ の類似度を $O_j^{me} = W_j^{pa}(t) \cdot x(t)$ とする。また、記憶層ニューロン j は、角度しきい値 $\theta_{th,j}(t)$ ($\theta_{min,j}(t) \leq \theta_{th,j}(t) \leq 90$) を持っていて、 $O_j^{me} \geq \cos(\theta_{th,j}(t))$ のとき発火する。入力に対して O_j^{me} が最大となるニューロンで発火したものを選択ニューロンと呼び、 j_{me} と表す (発火するものがないとき、 $j_{me} = 0$ とする)。ついで、逐次学習として、選択ニューロンの記憶パターンと角度しきい値を更新する。この更新は、記憶パターンがより入力パターン

に近付くようになされ、角度しきい値は小さくなるようになされる (詳細は [3] 参照) . 角度しきい値 $\theta_{th,j}(t)$ は学習が進むと $\theta_{min,j}(t)$ に一致するか近い値をとるようになる . 記憶パターンの更新には忘却更新も付随していて、低頻度の入力は記憶し続けられなくなり、高頻度の入力だけが記憶パターンとして記憶されることになる .

文脈ネットワークは状態層、中間層、想起層の 3 層からなる階層型ニューラルネットワークで、状態層、想起層は入力層、出力層に相当する . 状態層、想起層は記憶層と同じ数のニューロンからなる . 状態層の状態は状態ベクトル $s(t) = (s_1(t), \dots, s_j(t), \dots, s_J(t))$ で表す . 時刻 t で状態は $s(t-1)$ となっているが、選択ニューロン j_{me} の情報に応じた学習の後、 $s(t) = \alpha s(t-1)$ 、 $s_{j_{me}}(t) = 1$ と更新される . $\alpha (0 \leq \alpha < 1)$ を減衰定数と呼ぶ . この更新は、時刻 t より前の時刻 $t-l$ の選択ニューロンが j_l であったとき、 $s_{j_l}(t) = \alpha^l$ という値をとることを意味する . つまり、自己連想記憶モデルで選択されてきた選択ニューロンの時系列に関する情報を α のべき乗の形で記憶することになる . この文脈ネットワークの学習は、想起層が 1 時刻後の入力を予測するようになされる . すなわち、 $s(t-1)$ に対して、時刻 t の選択ニューロン j_{me} を教師として誤差逆伝播法で状態層-中間層間、中間層-想起層間の結合重みを更新する . 学習が進むことで、 $s(t-1)$ に対しては予測値が j_{me} となることが期待され、 $s(t)$ に対する想起層の出力は j_{me} を与えた入力の次の入力に対して選択されるものを予測したものになる .

以上が ASLN の概要であるが、モジュール選択機構としては次の操作を付加している . まず、モジュール選択においては、同一のモジュールに対して多数の入力パターンが対応することを考慮する必要がある . 入力によるモジュールの選択結果は選択ニューロンになるが、選択ニューロンの発火は、入力パターンと記憶パターンの成す角度 $\theta_j(t)$ が角度しきい値以下であることを要請している (類似度は $\cos(\theta_j(t))$ となる) . 従って、角度しきい値が小さすぎると記憶パターンに非常に似ている入力パターンしか受付けなくなる . しかし、より幅広い入力パターンに対しても同一モジュールを選択させたい場合があるはずで、これを実現するために、角度しきい値の収束先であるその下限値 $\theta_{min,j}(t)$ を更新することとした . この値は、ASLN では定数パラメータとして設計者が与えていたが、モジュール選択機構としては、変動するパラメータとして扱い、学習で更新させる . 更新規則は次のものである .

$$\theta_{min,j}(t) = \begin{cases} \theta_{min,j_{me}}(t-1) - \epsilon_1 O_{j_{me}}^{me}(t) & \text{if } j = j_{me} \\ \theta_{min,j}(t-1) + \epsilon_1 (1 - (O_j^{me}(t) - O_{j_{me}}^{me}(t))^2) & \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, J) \quad (1)$$

ただし、 ϵ_1 はある定数である . この更新は、選択ニューロン j_{me} が持つ角度しきい値の下限値を j_{me} の類似度に応じて小さく (許容範囲をせまく) し、それ以外のニューロン j では j と j_{me} の類似度の差に応じて下限値を大きく (許容範囲を広く) している .

複数の記憶パターンにおいて、入力パターンに対する類似度が同程度の場合、選択ニューロン j_{me} が適切でない場合が生じる . この時、各想起層ニューロンの中で出力が最大となる j_{pred} (想起出力) は、過去の記憶を反映して、現在選択されるであろうニューロンを予測しており、この予測情報の方が適切な選択となる可能性がある .

モジュール選択機構の最終出力 j_{select} は次のように決定する .

$$j_{select} = \begin{cases} j_{pred} & \text{if } O_{j_{pred}}^{me}(t) > d O_{j_{me}}^{me}(t) \text{ かつ} \\ & \theta_{j_{pred}}(t) < \theta_{th,j_{pred}}(t) \\ j_{me} & \text{otherwise} \end{cases} \quad (2)$$

式 (2) において、 d は $0 \leq d \leq 1$ のある定数であり、文脈 (予測結果) を考慮する度合いに相当する . つまり d が大きければ、モジュールの選択は入力パターンのみから決定し、逆に d が小さければ、予測に依存したモジュールの選択が優先される .

提案モデルでは、角度しきい値の下限値を式 (1) により変化させることで、予測情報を用いた適切なモジュール選択が期待できる .

2.2 モジュール機構

各モジュールは学習により分節化された領域における入出力の写像を実現する . j 番目のモジュールは入力パターンの各要素に対応する I 個のガウス関数 $G_{ji}()$ 、($i = 1, 2, \dots, I$) と、出力パラメータ $b_j(t)$ を持つ . ガウス関数 $G_{ji}()$ は中心値 $c_{ji}(t)$ が可変で、分散は定数である . 入力パターン $x(t)$ に対する、 j 番目のモジュールの出力は $y_j(t) = \prod_{i=1}^I G_{ji}(x_i(t)) b_j(t)$ である .

各モジュールは中心値 $c_{ji}(t)$ と出力パラメータ $b_j(t)$ を、 $x_i(t)$ 及び $y(t)$ に近づけるように、以下の式で学習させる .

$$\begin{aligned} c_{ji}^{new}(t) &= c_{ji}^{old}(t) + \epsilon_2 (x_i(t) - c_{ji}^{old}(t)) \\ b_j^{new}(t) &= b_j^{old}(t) + \epsilon_2 (y(t) - b_j^{old}(t)) \end{aligned} \quad (i = 1, 2, \dots, I, j = 1, 2, \dots, J) \quad (3)$$

ただし、 ϵ_2 はある定数である .

2.3 提案モデルの学習及び動作

学習用の時系列データ $(x(0), y(0)), (x(1), y(1)), \dots, (x(T), y(T))$ を考える . $x(t)$ は入力パターンで、 $y(t)$ はそれに対する望ましい出力である . モジュール選択機構では $x(t)$ だけを用いて学習させる . また、モジュールの学習は選択ニューロン j_{me} に対応するモジュールだけに

注目して、各パラメータを式 (3) により更新する (学習が d の値に依存しないようにするためである)。

次に、提案モデルの動作について述べる。時刻 t において、パターン $x(t)$ が入力されると、モジュール選択機構は、選択ニューロン j_{me} と文脈ネットワークの想起出力 j_{pred} の決定をもとに、式 (2) で j_{select} を決定する。モジュール j_{select} では入力パターンをもとに $y(t) = y_{j_{select}}(t)$ を出力する。

3. 実験結果及び考察

提案モデルの有効性を確認するため、計算機実験として、移動ロボットによるナビゲーションタスクシミュレーションを行った。実験では、データの分節化とモジュールの形成について確認するとともに、予測によるモジュール選択の有効性について調査した。

ロボットナビゲーションタスクは、移動ロボットが壁に衝突することなく、スタートからゴールまで移動するタスクである。実験で用いる移動ロボットを上から見た図を図 2(a) に示す。移動ロボットは直径 20cm の円形で、進行方向に対し左右 30 度、200cm 先までの物体を検出できるセンサを持つ。ロボットへの入力は、最短の壁面と進行方向の相対角度 $x_1(t)$ ($-30 \leq x_1(t) \leq 30[\text{degree}]$) (角度は、進行方向に対し時計回りの方向を正としている)、壁との最短距離 $x_2(t)$ ($0 \leq x_2(t) \leq 200[\text{cm}]$) の 2 つとした。これらの情報をもとにロボットのステアリング角度 ($y(t)$) が出力される。最大操舵角は進行方向に対して $\pm 10[\text{degree}]$ である。ロボットはセンサからの入力をもとにステアリング角を決定し、その方向に 10cm 進む。この入力から移動までを 1 ステップとする。実験に用いたコースを図 2(b) に示す。

本タスクにおいて、図 2(a) のような状況 ($x_1(t) > 0$) は、左右どちらのコーナーでも観測される状況である。このときに、ロボットが $x_1(t)$ が正であるため、自身の右側に壁があると判断し、左にステアリングをきくとすると、右コーナーで同様の状態が観測された場合には、壁に衝突してしまう。つまり、このタスクでは、単に入力だけでは、適切な行動を決定出来ない場合が存在する。

ロボットはスタートからゴールへと向かうタスクを繰り返す、ゴールに到達もしくは壁に衝突するまでを 1 試行とした。スタート位置及びスタート時の向きは、一定範囲内で試行毎にランダムに決めた。ゴール到達回数を全試行数で割ったものをタスク達成率 (achievement rate) とする。

提案モデルへの学習データは、ルール数が 2 の簡単なファジィ推論モデル [4] (以下、学習用モジュールと呼ぶ) を用いて取得した。学習モジュールは入出力が一对一で対応する制御モデルであり、タスク達成率は 52% であった。この学習用モジュールを用いて、タスクを実行させ、タスク成功時のデータをモデルの学習用データとした。

実験では、 $I = 2$, $J = 4$, $\epsilon_1 = \epsilon_2 = 0.01$ とした。ま

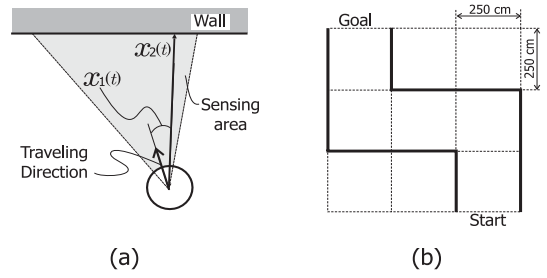


図 2: (a) 移動ロボット. (b) コース.

た、 $\theta_{min,j}(t)$ の最大値を $50[\text{degree}]$ とし、これを初期値とした。各入力変数及び出力は値域 $[0,1]$ に収まるように正規化された値を用い、全てのモジュールの持つガウス関数の中心値及び出力変数の初期値を 0.5 とした。

まず、学習によりモジュール選択機構が入力パターンをどのように分節化するのかを確認した。表 1 に学習を 3000 回行ったモデルの各モジュールの持つパラメータ及び角度しきい値を示す。モデルの記憶層ニューロンは 4 つであったが、 $j = 4$ の角度のしきい値が大きいことから、学習が進まず、記憶パターンが形成されていないことがわかる。このように、ASLN では入力頻度に応じて記憶パターンが形成されるため、与えられるデータに対して、記憶層のニューロンの数が冗長な場合には記憶パターンが形成されない。

図 3 は、軌道上で観測される入力パターン ($x_1(t), x_2(t)$) を学習後のモデルに対して入力し、選択ニューロン毎にプロットしたものである。これを見ると、記憶層ニューロン 1 はコースの直線部分、ニューロン 2 は右コーナー、ニューロン 3 は左コーナーで観測されるパターンに対して、それぞれ発火している。これは、モジュール選択機構が入力されるパターンを直進、右折、左折の三つの部分に分節化していることを意味する。なお、スタート直後においては、いずれのニューロンの発火も確認できるが、これはスタート直後のロボットの向きがランダムで決定されており、スタート直後に壁を回避 (右左折) する可能性があるためである。

表 1 と図 3 を見ると、分節化された領域に対応する各モジュールが妥当な入出関係を獲得していることがわかる。例としてモジュール 2 を見ると、ロボットに対して壁が左側 ($c_{21}(t) = 0.073$) にあり、壁との距離が離れている ($c_{22}(t) = 0.878$) 場合に、右へステアリングをきる ($b_1(t) = 0.994$) ことを意味している。図 3 より、モジュール 2 に対応する記憶層ニューロン 2 は、右コーナーにおける観測値に対し反応することから、妥当なモジュールを獲得できていると考えられる。

次に、学習したモデルを用いて実際にタスクを実行し、達成率を調査した。図 4 のグラフは学習後のモデルを用い、パラメータ d を変化させ、同様のタスクを 100 回ずつ行った達成率である。同図中の予測割合 (prediction

表 1: 各モジュールのパラメータ及び角度しきい値

j	$c_{j1}(t)$	$c_{j2}(t)$	$b_j(t)$	$\theta_{th,j}(t)$
1	0.500	1.000	0.500	0.357
2	0.073	0.878	0.994	35.826
3	0.908	0.880	0.053	27.149
4	0.977	0.208	0.012	90.000

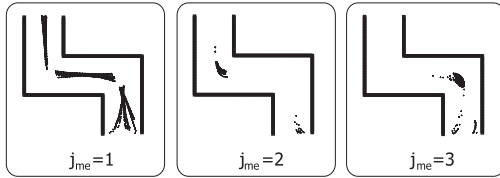


図 3: データの分節化

ratio) は, 100 試行中のタスク達成時のステップ数の合計のうち, $j_{me} \neq j_{select}$ となる割合であり, 予測の影響により, 入力パターンのみから選択されたモジュールとは異なるモジュールを選択した割合である.

図 4 を見ると, $d = 1$ の時, すなわち予測を考慮せず, 入力パターンのみからモジュール選択を行った場合の達成率は 54% であった. d を徐々に小さくすると予測割合がゆっくりと増加し, 達成率も増加する. これは, 左右コーナー部分において, その直前は共に直線部分であるにも関わらず, 適切な選択モジュールの予測が実現しており, その予測結果をある程度考慮することで, より適切なモジュール選択を実現していることを意味している.

予測割合は $d = 0.88$ 付近で急激に増加し, その後さらに d を小さくしていくと, 再びゆっくりと増加し, 達成率が低下 (81%) した後, 共に一定値となった. d の減少による達成率の低下は, 予測によるモジュール選択が優先され, 適切なモジュールが選択されていないことが原因である. 実験では, $d = 0.88$ 付近を境に入力パターン及び予測のどちらに基づくモジュール選択を優先するかが大きく切り替わり, この時の達成率は 98% で最大となった.

達成率の比較のため, Fuzzy Inference Neural Network (FINN) [5] を用いて同様のタスクを行った. FINN は, ニューラルネットワークの学習能力を利用したルール自動生成型の学習型のファジィ推論モデルであり, 入出力が一对一で対応する. FINN を用いた場合の達成率は, 最大で 76% であった. この時 FINN が学習により獲得したルール数は 21 であった.

以上の結果, 提案モデルは入力パターンの分節化と, 対応するモジュールを学習により獲得可能なが分かった. また, 予測によるモジュール選択を行うことで, 多数のルールを用いた FINN よりも高い達成率を実現できることを確認した. 実験では, d の値の減少により予

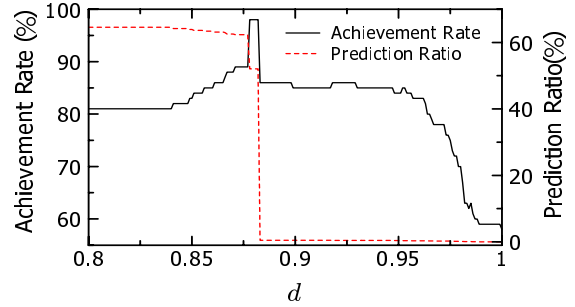


図 4: 達成率と予測割合の関係

測を考慮する割合が急激に増加し, ある特定のせまい範囲の d で最も高い達成率が得られた.

また, 本実験で扱った文脈によるモジュール選択を必要とするタスクを, MOSIC のように各モジュールが自身の予測器の結果に基づいて競合を行うモデルで扱う場合, 各モジュールでは, より複雑な (文脈を含む) 系列の予測を実現することが求められるため, モジュール自体の複雑化を招くことが考えられる.

4. むすび

本論文では文脈に基づくモジュール選択学習モデルを提案した. 提案モデルは, 教師なし学習により状態空間の分節化を行い, それらの分節化された領域における入出力写像を実現するモジュールを獲得する. また, 文脈を考慮した予測に基づくモジュール選択を実現する. 計算機実験としてロボットナビゲーションタスクへ適応し, 観測される状態だけでは適切なモジュールの選択が困難な場合にも, 予測によりモジュール選択を行うことで, タスクを達成することを確認した.

今後は, パラメータ d の自動設定及び文脈情報の再学習による, コースの変化への適応性を確認していくと共に, 提案モデルを実際の移動ロボットに実装し, オンラインでの学習を実現していく予定である.

参考文献

- [1] 鮫島和行, 片桐憲一, 銅谷賢治, 川人光男, “モジュール競合による運動パターンのシンボル化と見まね学習,” 電子情報通信学会論文誌 D-II, vol.J85-D-II, no.1, pp.90-100, 2002.
- [2] 西村政哉, 吉本潤一郎, 時田陽一, 中村泰, 石井信, “複数制御器の切替学習法による実アクトロボットの制御,” 電子情報通信学会論文誌 A, vol.J88-A, no.5, pp.646-657, 2005.
- [3] 郷古学, 菅谷至寛, 阿曾弘具, “逐次学習型時系列予測モデル,” 電子情報通信学会論文誌 A, vol.J89-A, no.3, pp.243-252, 2006.
- [4] M. Gouko, Y. Sugaya, and H. Aso, “Fuzzy inference model for learning from experiences and its application to robot navigation,” Proceedings of CIMCA 2005, vol.1, pp.577-582, 2005.
- [5] T. Nishina, and M. Hagiwara, “Fuzzy inference neural network,” Neurocomputing, vol.14, pp.223-239, 1997.