Profit SharingとRestricted Boltzmann Machineを用いた空間の分節化による学習手法の提案

Learning method based on Restricted Boltzmann Machine with segmentation of space by Profit Sharing

大久保 将博
Masahiro Ookubo
県立広島大学大学院
総合学術研究科経営情報学専攻
Email:q422002xr@pu-hiroshima.ac.jp

市村 匠
Takumi Ichimura
県立広島大学
経営情報学部経営情報学科
Email:ichimura@pu-hiroshima.ac.jp

鎌田 真
Shin Kamada
広島市立大学大学院
情報科学研究科情報科学専攻
Email:da65002@e.hiroshima-cu.ac.jp

***Abstract*—Hierarchical Modular Reinforcement Learning (HMRL) consists of 2 layered learning where Profit-Sharing works to plan a target position in the higher layer and Q-learning trains the state-action pair to the target in the lower layer. The method can divide a complex task into subtasks, and it reduces to state dimension and improves learning capability. In order to solve this problem, we propose the learning method based on Restricted Boltzmann Machine (RBM) with subspace divided by Profit Sharing. In this paper, to verify the effectiveness of the proposed method, the assignment problem of taxies was investigated.**

I. はじめに

近年，人間の学習能力をコンピュータで実現する様々な学習アルゴリズムが提案されている．強化学習はその中の一つの手法であり，観測された状態から一連の行動を通じて，最も高い報酬を得るための方策を学習する手法である [1]. ただし，強化学習では状態と行動の組に対する評価値のテーブルが指数関数的に増大し，計算機のメモリや計算量が爆発的に増大してしまう「次元の呪い」を招く恐れがある．

このような問題に対して，学習環境の複雑度を緩和するために，$Profit\text{-}Sharing$ を用いて学習空間を部分空間に分節化し，各部分空間において $Q\text{-}learing$ を用いて学習する階層型の強化学習手法 [2] が提案されている．ところが $Q\text{-}learning$ ではタスクが複雑な場合に特徴を抽出することができないことがある．

そこで，本研究では上記手法のサブタスクの学習を $Restricted\ Boltzmann\ Machine(RBM)$ によって学習する．RBM は $Deep\ Learning$ における事前学習として用いられ，特徴抽出の分野で高い性能を持つことが知られている [3]．そのため，サブタスクの学習を特徴を抽出しながら学習でき，特徴点の多いサブタスクには既に学習した結果を利用し，学習効率の向上を図る．本論文では提案手法の有効性を示すため，提案手法をタクシーの配送問題 [4] へ適用し，最短経路の学習を行う．

II. $Profit\text{-}Sharing$

この節では $Profit\text{-}Sharning$ について説明する．$Profit\text{-}Sharing$ は状態における行動の価値を学習する強化学習手法の一つであり，エピソードと呼ばれる単位で学習が行われる．ある状態 s において実行可能な行動 a はルール (s,a) として表され，エージェントは観測された状態 s に対し，各ルールに割り当てられた評価値 $u(s,a)$ に基づき，ルール (s,a) を選択することで次の状態 s' を得る．報酬が得られるまでのルールの選択系列 $((s_{-m},a_{-m}),(s_{-m+1},a_{-m+1}),\cdots,(s_{-1},a_{-1}),(s_0,a_0))$ がエピソードとして記憶される．ここで, ステップ 0 は報酬が得られる直前のルールが選択された時刻を示す．報酬が得られると, エピソードに含まれているルールに対して一括で報酬が分配され，各ルールの評価値が更新される．報酬分配の方法は，強化関数 $f(r,i)$ によって定義され，r は報酬であり，i は報酬獲得からのステップ数を示す．一般的に報酬獲得時点 $(i=0)$ のルールに最大の強化量が与えられ，ルールを過去に遡るにつれて強化量を減衰させていくように更新させるために，強化関数には等比減少関数が用いられる [5].

$Profit\text{-}Sharing$ の更新式は,

$$u(s_i,a_i) \leftarrow u(s_i,a_i) + f(r,i) \quad (i=0,-1,\cdots,-m) \qquad (1)$$

である．以下に $Profit\text{-}Sharing$ のアルゴリズムを示す．$u(s,a)$ はルール (s,a) の評価値を示し，$f(r,i)$ は強化関数を示す.

Algorithm 1 Profit-Sharing アルゴリズム

1: $u(s,a)$ の任意の初期化
2: **repeat:(すべての反復に対して)**
3: s の初期化
4: **repeat:(すべての反復のエピソードに対して)**
5: 方策を用いて状態 s から行動 a を選択
6: 行動 a を実行, s' を観測, エピソードを記憶
7: $s \leftarrow s'$
8: **until:(報酬 r が得られるまで)**
9: **一括更新:**$u(s_t,a_t) \leftarrow u(s_t,a_t) + f(r,i)$

III. *Restricted Boltzmann Machine*

この節では RBM について説明する．RBM はエネルギーに基づいた確率的モデルであり，図 1 のように可視層と隠れ層の 2 層で構成される．

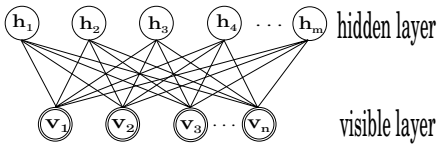

図 1. RBM の構造

通常のボルツマンマシンでは，全てのノードで結合があるが，RBM では各層ごとの結合がなく，隠れ層で独立した特徴を学習できる [6]．可視素子のベクトル $\mathbf{v} = [v_1, \cdots, v_n](v_i \in \{0,1\})$ と隠れ素子のベクトル $\mathbf{h} = [h_1, \cdots, h_m](h_j \in \{0,1\})$ としたとき，同時生起確率 $p(\mathbf{v},\mathbf{h})$ 及びエネルギー $E(\mathbf{v},\mathbf{h};\theta)$ は，式 (2)(3) のように示される．

$$p(\mathbf{v},\mathbf{h};\theta) = \frac{\exp(-E(\mathbf{v},\mathbf{h};\theta))}{Z(\theta)} \tag{2}$$

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j \tag{3}$$

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\theta)) \tag{4}$$

ここで，$\theta = \{W, b, c\}$ はパラメータであり，それぞれ可視層と隠れ層間の重み，可視層のバイアス，隠れ層のバイアスを表す．また，$Z(\theta)$ は式 (4) で表される分配関数である．

RBM の学習の最終目標は与えられた訓練データを $p(\mathbf{v},\mathbf{h})$ に加えた際に，データ点と当てはまりが尤もよくなる分布を作成する．つまり RBM のそれぞれのパラメータ θ は，この確率変数の対数尤度 $L = \log p(\mathbf{v};\theta)$ を最大化するように推定される．この対数尤度をそれぞれのパラメータで偏微分した式 (5)〜(7) により，パラメータを更新し学習する．

$$\frac{\partial L}{\partial W_{ij}} = \frac{1}{N}\sum_k v_i^k \sigma(c_j + \sum_i v_i^k W_{ij}) - \frac{1}{N}\sum_k \widehat{v}_i^k \sigma(c_j + \sum_i \widehat{v}_i^k W_{ij}) \tag{5}$$

$$\frac{\partial L}{\partial b_i} = \frac{1}{N}\sum_k v_i^k - \frac{1}{N}\sum_k \widehat{v}_i^k \tag{6}$$

$$\frac{\partial L}{\partial c_j} = \frac{1}{N}\sum_k \sigma(c_j + \sum_i v_i^k W_{ij}) - \frac{1}{N}\sum_k \sigma(c_j + \sum_i \widehat{v}_i^k W_{ij}) \tag{7}$$

ここで，N は訓練データ数，$\widehat{v}_i^k$ は可視素子のサンプル v_i^k を 1 回だけ遷移させた場合についてのサンプルを表している．また，$\sigma(x)$ は [0-1] を出力するシグモイド関数である．

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{8}$$

IV. *Profit-Sharing* と *RBM* を用いた学習手法

提案手法では，$Profit\text{-}Sharing$ で学習空間を分節化し，部分空間内で学習できる手法である．分節化された空間ではある部分空間からある部分空間へとサブタスクを順に行う必要があり，達成すべき目標までの一連のサブタスクを RBM で特徴を抽出しながら学習し，サブタスクごとの行動を決定する．

A. タクシー配置問題

タクシー配置問題では，ある地点をスタートとするタクシーが乗客の待機場所へと移動し，乗客を乗せた状態でより距離の短い経路でゴールすることを目的とする．本実験では実験環境における空間の分節化を取り入れた手法の学習精度を測定するため，10×10 のグリッド上での学習を行った．エージェントのスタート位置は左下端 (0,0) に設定し，乗客の待機場所は中央 (5,5)，最終的な目的地は右上端 (9,9) とし，これらは固定である．また，本実験ではタクシー台数は 1 台，乗客は 1 人とした．エージェントの行動は一回につき「上下左右のいずれかに一マス進む」の 4 つの行動と,「現在のグリッドに留まる」を合わせた 5 つの行動パターンの中から一つを選択することができる．エージェントは目的地に到達する前に乗客の待機場所を一度通過しなければならない．この上記の行動を $Profit\text{-}Sharing$ と RBM を用いた手法により学習し，各状態に合わせた行動を選択するか，また目的地への最短経路を学習するかをシミュレーションした．

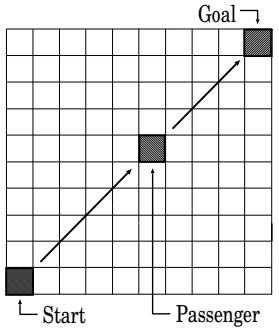

図 2. 実験環境

Algorithm 2 タクシー配送問題の学習アルゴリズム

1: **For:(学習回数分繰り返す)**
2: エージェント，乗客の待機場所，目的地の初期化
3: **For:(目的地に到達するまで繰り返す)**
4: エージェントの移動経路のサブゴールを $Profit\text{-}Sharing$ により決定
5: 乗客の待機場所・目的地へ到達時のみエージェントは報酬を獲得
6: 報酬が得られれば $Profit\text{-}Sharing$ で求めたサブゴールの経路を強化
7: **For:(全てのサブゴールを訓練データの 1 セットとし，学習回数分繰り返す)**
8: 現在位置から目的地までの経路を RBM により学習
9: **For:(目的地に到達するまで繰り返す)**
10: **For:(サブゴールに到達するまで繰り返す)**
11: 現在位置と行動後の位置をテストデータとし，学習後の RBM により行動を選択

B. $Profit\text{-}Sharing$ による分節化

本手法では，$Profit\text{-}Sharing$ で分割された状態ごとに評価値を与え，その値が高いサブゴールを順次決定し，経路を作成する．その後，乗客の待機場所，あるいは目的地へ到達時に評価値を更新する．タクシー配置問題における評価値の更新は式 (9) のようになる．

$$u(e, g(i), h_e(i)) = u(e, g(i), h_e(i)) + k(e, g(i), h_e(i)) \quad (9)$$

$$k(e, g(i-1), h_e(i-1)) = \rho k(e, g(i), h_e(i)) \ \ (i = 0, -1, \cdots, -m) \quad (10)$$

$u(\cdot)$ はサブゴールの評価値を求める関数を示し，$k(\cdot)$ は強化関数を示す．e はエージェント，$g(i)$ は報酬が得られた時点を 0 としてステップ i 時点での乗客の待機場所又は目的地，$h_e(i)$ はエージェント e がステップ i 時点にいた位置を示す．また式 (10) は，評価値が強化される割合が報酬獲得ステップを遡るごとに減衰することを示している (i は負値)．$\rho(\rho < 1)$ は忘却係数，m は更新ステップ数である．エージェントのサブゴールは，式 (11) により求められる．

$$\theta_e = arg \max_{v} \sum_{\epsilon} \frac{u(e, g, v)}{\mu^{|h_e - v|}} \quad (\mu \geq 1) \quad (11)$$

$u(\cdot)$ はサブゴールの評価値，e はエージェント，g は乗客の待機場所又は目的地，h_e はエージェントの現在地，v はサブゴールの候補を示している．θ_e は，エージェントとサブゴールの距離が遠ければ遠いほど，評価値は低くなる．このルール選択方法により，現在の位置に応じてルールの評価値が変化するため，探索的な行動を取れると考える．

$Profit\text{-}Sharing$ は報酬獲得時にエピソードを一括して更新するため学習速度は速いが，学習精度に欠けることがある．そこで本研究では，$Profit\text{-}Sharing$ は 50 回の試行で一連のサブゴールを学習すると仮定し，この過程を 10 回繰り返す．この結果のうち，最適なサブゴールの経路を分節化された空間とした．

C. RBM によるサブタスクの学習

サブゴールを決定後，現在地とサブゴールを組み合わせた訓練データをもとに RBM により，現在地から進むべきサブゴールを学習する．本論文では可視層への入力は $\{0, 1\}$ とし，xy 座標をそれぞれ 4 桁の 2 進数に変換した．ここで 4 桁としたのは，本実験は 10 × 10 のグリッド空間上の問題であり，最も大きい数値の 9 を 2 進数で表現するためには $\{1, 0, 0, 1\}$ と 4 桁になり，結果として全ての入力数を合わせることができる．したがって，16 次元の訓練データを用いて学習する．また，訓練データ数は $Profit\text{-}Sharing$ で求めたサブゴール数となる．

本実験では，複数のサブゴールを訓練データとして学習するため，パラメータの更新は全ての訓練データを入力した後に一括で更新を行う．

$$W_{ij} = W_{ij} + \sum^{N} \frac{\partial L}{\partial W_{ij}} \quad (12)$$

$$b_i = b_i + \sum^{N} \frac{\partial L}{\partial b_i} \quad (13)$$

$$c_j = c_j + \sum^{N} \frac{\partial L}{\partial c_j} \quad (14)$$

上記の学習された RBM にテストデータを入力することで現在地からサブゴールまでの行動を決定する．ここでテストデータは現在と 1 つ後の位置を組み合わせたものとした．このように上下左右に 1 つ進む行動とどこにも進まないという 5 つの行動に対するテストデータを訓練後の RBM に入力し，得られた結果をもとに訓練データの結果に一番近い行動を選択し，サブゴールまで移動する．

V. 実験結果

本実験では学習環境を分節化するための $Profit\text{-}Sharing$ での学習とサブゴールまでの行動を学習するための RBM の 2 つの学習を行った．$Profit\text{-}Sharing$ で学習した経路を表 II と図 3 に示す．

表 I
PROFIT-SHARING のエピソード回数

試行回数	0	1	2	3	4	5	6	7	8	9	10	11	···	50
移動回数	252	44	52	30	22	18	18	18	18	18	18	18	···	18

表 II
サブゴールの遷移

移動回数	1	2	3	4	5	6	7	8	9	10
サブゴール	(1,0)	(2,0)	(2,1)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(4,5)	(5,5)
移動回数	11	12	13	14	15	16	17	18		
サブゴール	(5,6)	(6,6)	(6,7)	(7,7)	(8,7)	(9,7)	(9,8)	(9,9)		

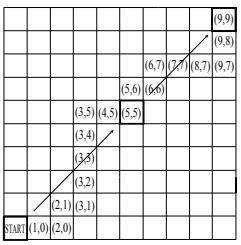

図 3. サブゴールの遷移例

エピソードとはスタート地点から最終的な目的地に到着するまでを探索した一連の経路であり，本実験の実験環境における最適解は 18 であるため，今回の結果では分節化が 1 マスずつ行われていることが分かる．今回の分節化で空間をより細かく分けた理由として，$Profit\text{-}Sharing$ でサブゴールを決定する式 (11) が関係している．式 (11) により，現在位置に近いほど評価値が高くなる選択方法を用いているため，分節化がエージェントに近い範囲で行われたと考えられる．

次に RBM では，現在位置と $Profit\text{-}Sharing$ で求めたサブゴールを組み合わせた訓練データを学習し，学習後の RBM を用いて現在位置からサブゴールまで行動選択を行った．

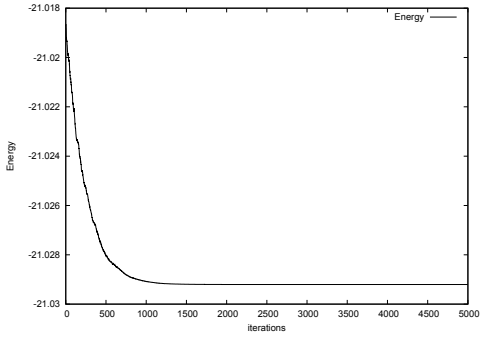

図 4.　RBM のエネルギー値

図 4 は上記の $Profit\text{-}Sharing$ で求めた 18 個のサブゴールをもとに学習を行った際のエネルギー値の推移である．図 4 では初期段階でパラメータの大幅な更新が行われ，試行回数 1000 回あたりからは微調整が行われエネルギー値が収束していることが分かる．

表 III と表 IV はそれぞれ実際の 2 進数で表した訓練データと学習後の RBM にテストデータを入力した際の出力値である．以下の結果より訓練データとテストデータの出力値が一致しているため，現在地から分節化して得られたサブゴールまでの行動が学習できていることが分かる．

表 III
訓練データ一覧

入力値	現在位置 x				現在位置 y				サブゴール x				サブゴール y			
訓練データ 1 (0,0,1,0)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
訓練データ 2 (1,0,2,0)	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
訓練データ 3 (2,0,2,1)	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
訓練データ 4 (2,1,3,1)	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1
訓練データ 5 (3,1,3,2)	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0
訓練データ 6 (3,2,3,3)	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1
訓練データ 7 (3,3,3,4)	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0
訓練データ 8 (3,4,3,5)	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	1
訓練データ 9 (3,5,4,5)	0	0	1	1	0	1	0	1	0	1	0	0	0	1	0	1
訓練データ 10 (4,5,5,5)	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
訓練データ 11 (5,5,5,6)	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
訓練データ 12 (5,6,6,6)	0	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0
訓練データ 13 (6,6,6,7)	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
訓練データ 14 (6,7,7,7)	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
訓練データ 15 (7,7,8,7)	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1
訓練データ 16 (8,7,9,7)	1	0	0	0	0	1	1	1	1	0	0	1	0	1	1	1
訓練データ 17 (9,7,9,8)	1	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0
訓練データ 18 (9,8,9,9)	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1

VI. まとめ

学習を行う過程では多くの探索的な行動を必要とすることで，各状態に応じた挙動をより学習することができるが，実世界では学習回数や試行回数は制限されている場合が多い．そこで，本研究では学習空間を分節化し，分節化された空間の行動のみを学習することで，学習空間の大きさに関係なく部分的な問題として解くことができる手法を提案した．本手法では学習空間をより最適な分節化を行う学習を $Profit\text{-}Sharing$ によって行い，分節化された部分空間での行動を RBM により学習した．

本研究では提案手法をタクシーの配置問題へ適用した結果，最適なサブゴールを求めることができ，RBM によりサブゴールからサブゴールまでの行動を学習することができた．

今後の課題として，本実験では待機場所を固定しおり，目的地までの一連のサブゴールを 1 パターンしか学習しなかったため，待機場所を増やした環境での実験を行う．また，他の手法 ($Q\text{-}learning$) との学習性能を比較する．

表 IV
学習後の出力値

入力値	現在位置 x				現在位置 y				サブゴール x				サブゴール y			
テストデータ 1 (0,0,1,0)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
テストデータ 2 (1,0,2,0)	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
テストデータ 3 (2,0,2,1)	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
テストデータ 4 (2,1,3,1)	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1
テストデータ 5 (3,1,3,2)	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0
テストデータ 6 (3,2,3,3)	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1
テストデータ 7 (3,3,3,4)	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0
テストデータ 8 (3,4,3,5)	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	1
テストデータ 9 (3,5,4,5)	0	0	1	1	0	1	0	1	0	1	0	0	0	1	0	1
テストデータ 10 (4,5,5,5)	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1
テストデータ 11 (5,5,5,6)	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
テストデータ 12 (5,6,6,6)	0	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0
テストデータ 13 (6,6,6,7)	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
テストデータ 14 (6,7,7,7)	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1
テストデータ 15 (7,7,8,7)	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1
テストデータ 16 (8,7,9,7)	1	0	0	0	0	1	1	1	1	0	0	1	0	1	1	1
テストデータ 17 (9,7,9,8)	1	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0
テストデータ 18 (9,8,9,9)	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1

参考文献

[1] Sutton R.S. and Barto A.G., “Reinforcement Learning”,MIT Press,1998.

[2] 伊賀上大輔，市村匠,「階層型モジュラー強化学習による動的環境に適応した学習手法の提案」，proc. of 2012 IEEE SMC Hiroshima Chapter Young Researchers WorkShop，pp.9-12，2012.

[3] 岡谷貴之，齋藤真樹,「ディープラーニング」，情報処理学会研究報告，Vol.2013-CVIM-185，No.19，2013.

[4] Predrag D. Djurdjevic and Manfred Huber, , “Deep Belief Network for Modeling Hierarchical Reinforcement Learning Policies”，IEEE International Conference on Systems Man and Cybernetics，pp.2485-2491，2013.

[5] 宮崎和光，木村元，小林重信,「 ProfitSharing に基づく強化学習の理論と応用」，人工知能学会誌，Vol.14，No5，pp.800-807，1999.

[6] Hinton G.E., et al. “A fast learning algorithm for deep belief nets”，Neural Computation，Vol.18，No.7，pp.15271554，2006.

[7] Hinton G.E.，“Training products of experts by minimizing contrastive divergence”,Neural computation，Vol.14，No.8，pp.1771 － 1800，2002.

問い合わせ先

〒 734-0003
広島市南区宇品東一丁目 1 番 71 号
県立広島大学経営情報学部
市村 匠