

4th SICE Symposium on Computational Intelligence
December 4-5, 2013, Fukuoka

第4回

コンピューターショナル・インテリジェンス研究会
講演論文集

会 期 : 2013年12月4日 (水), 5日 (木)

会 場 : 九州大学伊都キャンパス

SICE[®]

主 催 : 計測自動制御学会 システム・情報部門

企 画 : ニューラルネットワーク部会, 知能工学部会

協 賛 : 情報処理学会, システム制御情報学会, 電子情報通信学会, 電気学会, 日本神経回路学会, 日本機械学会, 人工知能学会, 日本知能情報ファジィ学会, ヒューマンインタフェース学会, 進化計算学会, Japan Chapter of IEEE Computational Intelligence Society, Japan Chapter of IEEE Systems, Man, and Cybernetics Society

著作権 © 2013

公益社団法人計測自動制御学会(SICE)
〒113-0033 東京都文京区本郷 1-35-28-303

カタログ番号 13 PG 0014

著作権は、計測自動制御学会がもっている
ので、個人の使用のための複製以外の
目的で掲載の記事の一部または全文を
複製する場合には、著作権者に許可を求
め規定の複製料を支払うこと。

発行日：2013年12月4日

発行者：公益社団法人計測自動制御学会 システム・情報部門
ニューラルネットワーク部会

プログラム

A : 20分発表, 5分討論 B : 18分発表, 7分討論 C : 15分発表, 10分討論

12月4日 (水)

13:00-14:27 行動決定・行動獲得

司会 : 小林邦和 (愛知県立大学)

1. B 非対称近傍関数を持つG-SOMおよびそれを用いたロボットの命令学習システム
○大谷拓寛, 呉本 堯 (山口大学), 小林邦和 (愛知県立大学), 間普真吾, 大林正直 (山口大学)
... 1
2. B 経験に基づく情動の再構成を導入したロボットの行動選択システム
○綿田将悟, 大林正直, 呉本 堯, 間普真吾 (山口大学), 小林邦和 (愛知県立大学)
... 9
3. B CPGと強化学習を用いた多脚ロボットの行動獲得の試み
石倉裕貴, 岸本良一, ○堀内匡 (松江工業高等専門学校)
... 16

14:40-16:07 高次元ニューラルネットワーク1 ※

司会 : 新田徹 (産業技術総合研究所)

※ SICE先端融合システムズアプローチ創出委員会応用クリフォード代数WGと東北大学電気通信研究所
共同プロジェクト研究「高次元ニューラルネットワークの情報処理能力」の共同提案によるテーマセッション

4. A 極変数複素ニューロンにおけるプラトー
○新田 徹 (産業技術総合研究所)
... 20
5. A パラメータの自動調整が可能なカオス複素多方向連想メモリ
千野拓也, ○長名優子 (東京工科大学)
... 24
6. A 四元数ニューラルネットワークの歩行者数推定への適用
○村本憲幸, 磯川悌次郎, 西村治彦, 松井伸之 (兵庫県立大学)
... 30

16:20-17:47 進化・免疫システム

司会 : 長名優子 (東京工科大学)

7. B 粒子群最適化における探索の停滞現象の解消に関する検討
○安田大悟, 畠中利治 (大阪大学)
... 36
8. C 組合せ最適化問題におけるラベルの再割振りに関する研究
○船木亮平, 高野浩貴, 村田純一 (九州大学)
... 40
9. A スマートフォンによるユーザ参加型センシングシステムにおける
観光写真のLeivenshtein距離を免疫システムによる分類手法
○鎌田 真, 市村 匠 (県立広島大学)
... 46

18:00-19:30 技術交流会 (九州大学ビッグオレンジ・レストラン)

A : 20分発表, 5分討論 B : 18分発表, 7分討論 C : 15分発表, 10分討論

12月5日 (木)

10:00-11:27 マルチエージェントシステム

司会 : 黒江康明 (京都工芸繊維大学)

10. B 報酬の解釈を導入したマルチエージェント強化学習システムにおける協調行動の獲得
○椿本樹矢, 小林邦和 (愛知県立大学) . . . 54
11. A 複数の餌場がある軍隊アリシミュレーションシステムにおける
エージェントの挙動からの利他行動の分析
○上本拓也, 市村 匠 (県立広島大学) . . . 59
12. B マルチエージェント環境における協調度を用いた行動選択法
○河村将範, 小林邦和 (愛知県立大学) . . . 67

12:30-13:57 高次元ニューラルネットワーク2 ※

司会 : 佐藤茂雄 (東北大学)

※ SICE先端融合システムズアプローチ創出委員会応用クリフォード代数WGと東北大学電気通信研究所
共同プロジェクト研究「高次元ニューラルネットワークの情報処理能力」の共同提案によるテーマセッション

13. A 四元数ネットワークインバージョンのための正則化
○井浦 翼, 小川毅彦 (拓殖大学) . . . 72
14. A 大規模fan-inを有するニューロンを実現する多数決回路
○秋間学尚 (東北大学), 片山康弘 (東芝), 佐藤茂雄, 櫻庭政夫, 中島康治 (東北大学) . . . 76
15. A リカレントクリフォードニューラルネットワークのダイナミクス
ーエネルギー関数とその存在条件ー
○黒江康明, Vagif Hasanov, 飯間 等 (京都工芸繊維大学) . . . 80

14:10-15:37 ネットワーク・インテリジェントシステム

司会 : 小川毅彦 (拓殖大学)

16. A ユーザの性格特性に基づく学習適応的感情指向型インタフェース
○田邊幸祐, 市村 匠 (県立広島大学) . . . 87
17. C サポートベクターマシンによる導関数データを利用した関数近似法とその評価
○三木翔登, 黒江康明, 飯間 等 (京都工芸繊維大学) . . . 95
18. A 遺伝子ネットワークにおけるアイソクロンの簡単な計算法
○森 禎弘, 黒江康明 (京都工芸繊維大学) . . . 102

非対称近傍関数を持つ G-SOM およびそれを用いたロボットの 手形状命令学習システム

○大谷拓寛 呉本堯 (山口大学) 小林邦和 (愛知県立大学) 間普真吾 大林正直 (山口大学)

A hand shape instruction learning system using a G-SOM with asymmetric neighborhood function

* T. Otani and T. Kuremoto (Yamaguchi University) and K. Kunikazu (Aichi Prefectural University) and S. Mabu and M. Obayashi (Yamaguchi University)

Abstract — A Hand shape instruction learning system using PL-G-SOM which is a kind of self-organizing map of Khonen has been proposed in our previous work. However, by increasing the number of times of learning, the twisting phenomenon of distance topology between units often occurs in PL-G-SOM. In this paper, we introduce an improved asymmetric neighborhood function to G-SOM, and build a new hand shape instruction learning system using the proposed AGSOM.

1 はじめに

現在, IT 技術の進歩により次世代のロボット開発は急激に発達し, エンタテインメント自律型ロボットなどの開発, 商品化が盛んに行われている. しかし, 市販ロボットは, 予め設定された音や画像, センサーなどによる命令認識機能を持つもの, 人間から音声や画像による命令の理解, 学習, およびロボットの内部状態を表す感情などの機能に関する開発は不十分である.

本研究では, 人間と自律型ロボットにおけるコミュニケーションを行い, 人間の指令をロボットに学習させる知的システムの開発を行う. これまで, 人間の「手形状」画像を命令として利用する手法が提案された^{1),2)}. 「手形状」を識別する方法として, 自己組織化マップ(SOM)³⁾を用いる. 少ない計算量でより多くの命令数を学習可能にさせるため, “記憶機能”を持つ T-SOM⁴⁾や, ノードの枯渇や学習の忘却といった問題を改善した PL-G-SOM^{1),2),5)}を用いた手形状による命令学習システムが提案されている. しかし, ユニット間の距離トポロジーのねじれ現象が学習過程で生じることや, ノード数の増加に伴い計算量が大幅に増加するという問題が生じる.

そこで本論文では, まず, 青柳と青木に提案され

た非対称近傍関数⁶⁾の導入及びその非対象近傍関数の改良をした AGSOM を提案し, 従来法より学習性能の向上を図る. また, 秩序率という特徴マップの状態を定量的に表すための指標を新たに導入する. 最後に, 実際の手形状指令による計算機実験を行い, 学習の収束や学習後のマップのトポロジーについて, 従来システムとの比較を行う.

2 PL-G-SOM を用いた手形状指令学習システム

従来のコホネンの自己組織化マップ(SOM: self-organizing map)の改良モデルである Parameter-Less-Growing-SOM (PL-G-SOM)³⁾は, Growing-SOM^{7),9)}の成長構造と PL-SOM¹⁰⁾の学習率と近傍関数の適応調整法を用いる. 前者の G-SOM は, 学習が進みマップをある程度構築した後に, 未学習のデータの入力を確認できた場合, ノードを増加させることによって(成長)させることで, 入力されるデータの種類が増加してもマップ上で分類することを可能にした SOM である. また, 後者の PL-SOM は過去に学習したデータの忘却や追加学習過程で学習するデータと既学習データ間の位相が破壊されるという問題を解決している. これらの手法を組み合わせた PL-G-SOM は, ノードの枯渇や追加学習による結合荷重の忘却問題を改善でき, 実機ロ

ボットによってその有効性が確認されている^{1), 2), 5)}.

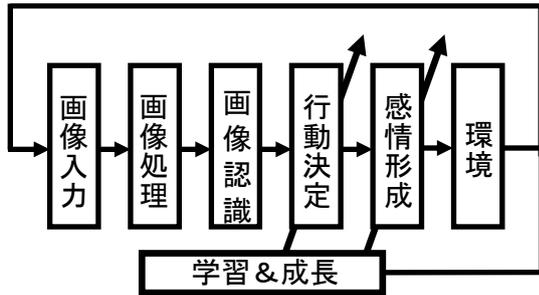


Fig.1: A hand shape instruction learning system

本研究で提案する手形状指令学習システムを Fig.1 に示す. まず, ロボットがカメラから, 画像を得る (画像入力). 次に, 画像処理部で得た画像から手部分 (肌色) を抽出し, 画像から得た特徴量を元に SOM を用い画像認識部で命令認識を行う. さらに, 行動決定部で学習を行い, 認識した命令に従った行動を実現する. 学習過程において, 教示者が報酬, または, 罰をタッチセンサーなどを用いて与える. 受けた報酬と罰により感情形成部でロボットの感情マップを形成し, 感情の表現はロボットの顔表情で表し, 教示者に学習達成度を示す.

2.1 画像処理部

カラー画像 (Fig.2 参照) から RGB の R 値, HSV の H,S 値に着目し閾値を設けて肌色を抽出する (Fig.3 参照). その後, ロバスト性を持たせるため画像の回転・拡大を行う (Fig.4 参照).



Fig.2: The original image Fig.3: Skin color extraction



Fig.4: Enlarged image

2.2 命令認識部

手の領域を抽出後, そのパターンの幾何学的特徴を識別するため, SOM を用いる. 特徴量の抽出法として, SOM 分類法に有効な輪郭線法を用いる (Fig.5 参照).

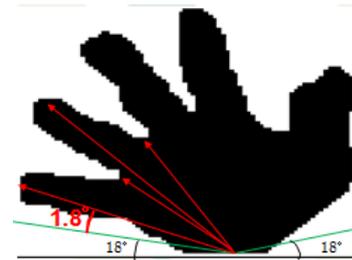


Fig.5: Feature extraction method

この手法を用い以下に特徴量を抽出した結果を示す. Fig.6 のグラフの縦軸は手首から輪郭線までの距離, 横軸は手首から3時の位置を 0° とし, 0° から 18° の間と 162° から 180° の間は考えないものとする. つまり, 18° 目を1とし, 1.8° ずつ時計回りに 2, 3, ..., 80 と番号を振っていったものである. ただし, 距離の最高値を1とし, 正規化を行った. その結果を Fig.6 に示す.

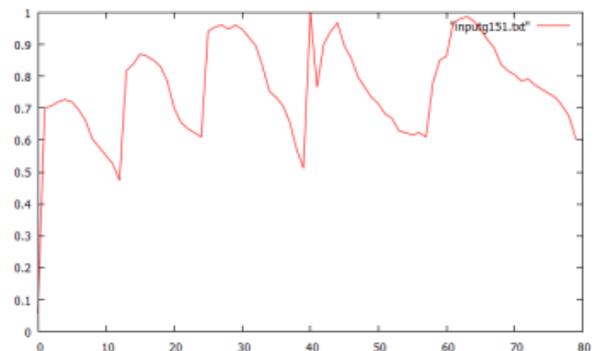


Fig.6: Graph of the feature quantity obtained when the amount of the feature contour

従来の PL-G-SOM^{1), 2), 5)}では, 適合度 $\varepsilon(t)$ と近傍関数 $hc_i(t)$ によるアニーリング計画を用いて, 以下の式より重みベクトル W_i の更新を行う.

$$r(t) = \max(\|x(t) - w_c\|^2, r(t-1)) \quad (1)$$

$$h_{c,i}(t) = e^{-dis(i,c)^2 / \theta(\varepsilon(t))} \quad (2)$$

$$\varepsilon(t) = \|x(t) - w_c\|^2 / r(t) \quad (3)$$

$$w_i(t+1) = w_i(t) + \varepsilon(t) \cdot h_{c,i}(t)(x(t) - w_i(t)) \quad (4)$$

2.3 行動決定部

Table.1 Action learning map

似た画像	行動1	行動2	行動3	行動4
1	6	2	-8	0
2	10	1	0	1
n×m	0	2	7	2

行動学習の報酬：
正しい行動 +10
(褒められた場合)；
誤った行動 -10
(叱られた場合).

Table.1 を元に、命令認識部の手形状分類マップにより選択した勝者ノード番号より、番号に対応した行動マップから行動価値関数を読み込み、行動をボルツマン分布

$$\pi_t(a_t(i) | s_t) = \frac{e^{\frac{Q_t(s_t, a_t(i))}{T}}}{\sum_{j=1}^A e^{\frac{Q_t(s_t, a_t(j))}{T}}} \quad (5)$$

に従って、確率的に決定する。ここで、 s_t は手形状の t 番目のパターン、 $a_t(i)$ は i 番目の行動、 $Q_t(s_t, a_t(i))$ は状態—行動価値関数である。Q の初期値は乱数で与える画、教示者の評価により Q 値を更新する。このように行動選択に関する価値関数の学習により、画像分類に連動した動作を学習することができる。

2.4 成長機能

ロボットが行動に失敗し、かつ行動価値が一定値以上（本システムでは 10 以上）の時、新しい命令データが入力されたと判断し、新しいノードを生成する。この成長機能により、既学習データを忘れる（初期化）ことなく新しい命令データの追加学習を可能にする。以下に、生成過程を示す。

STEP.1:ノードの挿入方向の決定

勝者ノード c の最近傍ノード $i \in h$ の中から、最も類似度の低いノード d を探す。

$$w_d = \max_{i \in h} (\|w_c - w_i\|^2) \quad (6)$$

STEP.2:ノード生成

勝者ノード c とノード d の間に新たなノード r を以下の式に従い生成する。ただし、2次元の場合、行または列で生成し、新ノード r 以外の行および列ノードを R とし、隣接ノードの平均の特徴ベクトルとする。

$$w_r = x_i \quad (7)$$

$$w_R = 0.5 \cdot (w_E + w_D) \quad (8)$$

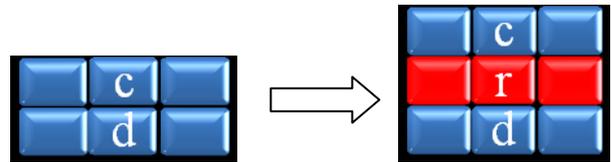


Fig.7: Growth process of Growing-SOM

2.5 感情形成

命令の習得状況を利用者に示すための感情値を以下の式で更新する。この感情値により命令習得状況を教示者に示すことで、人間とのコミュニケーションを実現する。

褒められた場合、

$$Feel = Feel + \alpha * Continue - \beta * SE \quad (9)$$

叱られた場合、

$$Feel = Feel - \alpha * Continue - \beta * SE \quad (10)$$

Continue:褒められ（叱られ）続けた回数、

SE:入力ベクトルと勝者ベクトルの二乗誤差、

$$0 < \alpha < 1, 0 < \beta < 1$$

である。

褒められ続けるほど、ロボットの感情（Feel）が急激に喜び（悲しみ）に近づき、入力ベクトルと勝者ベクトルとの二乗誤差が大きいほど悲しみに近づく。逆に、入力ベクトルと勝者ベクトルとの二乗誤

差が小さいほど常に高い喜び度を持つことになる。

また、 α と β に関しては、今回は二乗誤差の係数 β を0.05と定め、報酬や罰の係数 α を0.2とした。

3 PL-G-SOM の問題点

しかし、PL-G-SOM を用いた手画像命令学習システムには依然、いくつかの問題点が存在する。

- ・手画像の特徴量が多い
- ・ノードの無駄が多い
- ・ノードに位相不整合点が存在する

本論文では、三点目の位相不整合点の問題に着目する。近傍関数を対象型から非対称型に変更することで、マップのねじれを解消し、学習速度の向上や学習の精度の向上を図る。

4 非対称近傍関数

自己組織化マップを応用する上で、マップの学習時間、すなわち皮質が構造を持たない状態から正しいトポグラフィックマップを持つ状態まで変化するのに時間が短いことが望ましい。しかし、実際には学習時間を大きく増大させる要素があり、そのうちのひとつが“位相不整合点(ねじれ)”である。Fig.8は典型的なマップな状態及び、位相不整合点が1ヶ所または、複数発生したマップの典型例である。

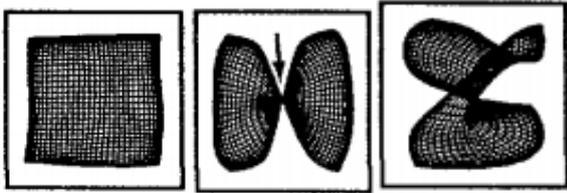


Fig.8: Examples of a map with no twist (left), one twist (center), and more twists (right)

SOM の結合荷重を更新する際に、従来の対称近傍関数(Gaussian 関数)の代わりに、以下の非対称近傍関数を用いてノードの位相不整合現象を解消することが提案されている⁴⁾。

$$h_{\beta}(\tilde{r}_{ic}) = 2\left(\frac{1}{\beta_0} + \beta_0\right)^{-1} \exp\left(-\frac{\tilde{r}_{ic}}{2\sigma^2}\right), \quad (11)$$

$$\tilde{r}_{ic} = \begin{cases} \sqrt{\left(\frac{r_{ic} \cdot k}{\beta_0}\right)^2 + \|r_{\perp}\|^2}, & \text{if } r_{ic} \cdot k \geq 0, \\ \sqrt{(\beta_0 \cdot r_{ic} \cdot k)^2 + \|r_{\perp}\|^2}, & \text{if } r_{ic} \cdot k < 0. \end{cases} \quad (12)$$

$$r_{\perp} \equiv r_{ic} - (r_{ic} \cdot k)k \quad (13)$$

ここで、 $0 < \beta_0 \leq 1$ は経験による固定パラメータである。

5 位相不整合とその解消の例

ここでは、対称近傍関数と非対称近傍関数の効果を比較するため、入力とベクトルと出力ユニットが共に2次元の[0, 1]の実数値の場合のシミュレーションを行い、学習過程におけるマップのトポロジーの変化を観測した。Fig.9はねじれを一つ以上持たせた初期の参照ベクトル(マップ)である。また、Fig.10は学習回数24,000回後のマップである。結果より、非対称近傍関数を用いると位相不整合点を除去できているのがわかる。

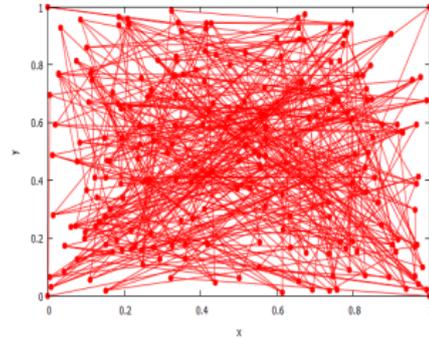


Fig.9: Initial map

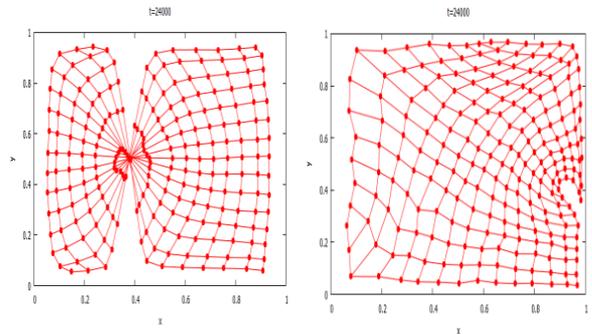


Fig.10: Maps after 24,000 training: symmetric neighborhood function (left), and asymmetric neighborhood function (right)

6 改良した非対称近傍関数

非対称近傍関数を導入すると、位相不整合点を学習の早期に除去できるが、ここで別の問題が生じている。

位相不整合点が解消され連続的な対応構造は形成できているものの、刺激ベクトルの分布の情報を歪めて表現してしまっている。そこで、マップの歪みを少しずつ軽減し最終的に均一にする処理を追加した。

$$\beta(t) = \begin{cases} 1 + (\beta_0 - 1) \left(1 - \frac{t}{T_{total}}\right), & \text{if } t < T_{total} \\ 1, & \text{if } t \geq T_{total} \end{cases} \quad (14)$$

すなわち、(11)~(13)式の固定値を持つ β_0 の代わりに、学習回数に応じて調整する $\beta(t)$ を用いる。この修正を加えた場合の結果を Fig.11 に示す。

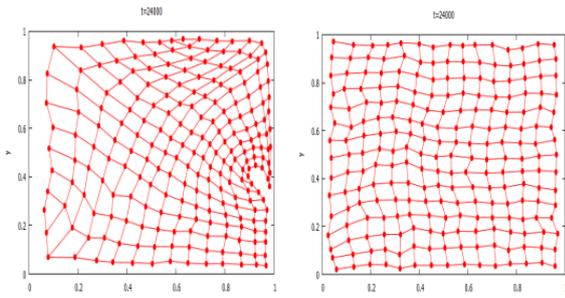


Fig.11: Maps after 24,000 training: asymmetric neighbor function (left) and improved function (right)

7 秩序率の設定

ここでは、現在のマップの状態を定量的に表して議論するために、秩序率という指標を導入する。秩序率 η ($0 \leq \eta \leq 1$) は、刺激とユニット列が連続的に対応し、参照ベクトルが一様に配置されている場合は $\eta=1$ をとり、位相不整合点の発生や偏った参照ベクトルの配置により 0 に近づく。 η を求める式は以下のように設定する。

$$\eta = \frac{\text{連続的対応が成立したトポロジー数}}{\text{総トポロジー数}} \quad (15)$$

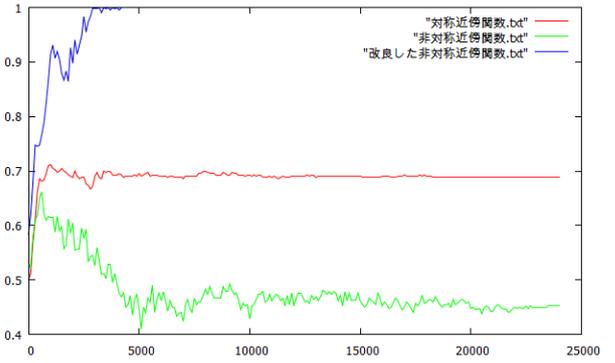


Fig.12: Transition graph of the order parameter in the 24,000 times number of learning

Fig.12 は、前節でのシミュレーションにおける学習回数 24,000 回での秩序率の推移グラフである。

Fig.12 より、改良した非対称近傍関数は常に $\eta=1$ であることから、刺激とユニット列が連続的に対応し、参照ベクトルが一様に配置されていることがわかる。

8 非対称 G-SOM による手形状画像認識

非対称近傍関数と改良した非対称近傍関数を導入して従来の PL-G-SOM との比較実験を行い、提案法の有用性を確認した。実験に用いる手画像は、8 種類の手形それぞれを 3 つずつ、計 24 個を SOM の入力としてランダムに入力する。

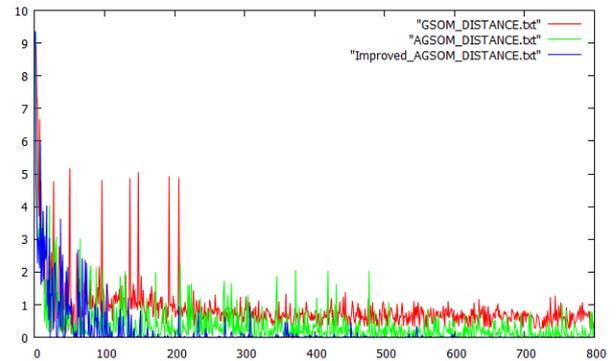


Fig.13: Vector distance between Best match units (BMUs) and the input features during training

Fig.13 は初期ノードが 3×3 の学習回数 800 回の入力画像と勝者ノードの誤差の推移グラフである。実験結果より、改良した非対称近傍関数を導入した結果は、対象近傍関数・非対称近傍関数の結果より早

い段階で誤差が小さくなっているため、提案システムのほうがより入力データ空間に適合したマップの構築を行っていることがわかる。

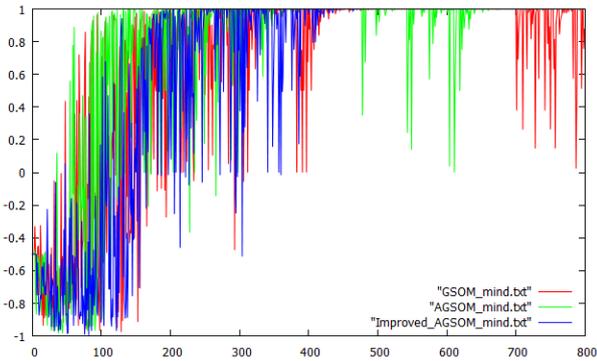


Fig.14: Transition graph of feeling value

Fig.14 は感情値の変化を表している。感情値は、命令習得度も表しており、-1に（悲しみ）に近いほど行動失敗を繰り返しているため命令の習得ができていないことを、+1（喜び）に近いほど行動が成功していること表している。Fig.14 より、いずれの手手法も学習初期は手画像による命令を学習できていないため、失敗して罰を与えられることが多く、感情値が負に向かう傾向がある。提案手法では、学習回数400回を過ぎたあたりから、常に感情値の値が+1になっていることから全ての命令が習得できていることがわかる。しかし、対象近傍関数・非対称近傍関数の結果は学習後期にも感情値が負に近づくことがみられたため、提案手法の有効性が確認できた。

2	1	1	5	0	4	1	7
0	3	1	0	0	4	4	4
6	1	1	7	5	5	2	2
6	6	7	3	5	5	2	2
6	6	7	1	5	1	1	3

(a)

0	1	1	-0.46959	1	1	0	0
0	1	1	1	0.794567	1	1	-0.5
1	0.25	0	1	-0.28951	-0.88593	0.125	-0.75
1	0.078594	-0.84281	1	1	1	1	-1
-0.15634	-0.24234	1	0	1	0	0	0

(b)

Fig.15: Action learning map (a) and emotion value map (b) at 24,000 times the number of learning when using

PL-G-SOM

2	1	1	1	1	7
0	1	1	1	3	0
8	0	5	5	7	3
8	1	4	5	7	3
6	1	4	0	6	1
6	1	4	4	6	1
6	1	1	1	6	2
2	1	1	1	1	5
2	1	1	1	1	3

(a)

0	0	0	0	0	0
0	0	-0.09301	-0.18602	1	1
0	0	0.5	1	1	1
0	0	1	1	1	1
0	0	0.75	1	1	1
0	0	1	1	1	1
0	0	-0.18857	-0.37714	1	1
0	0	-0.09429	-0.18857	-0.28286	1
0	0	0	0	0	0

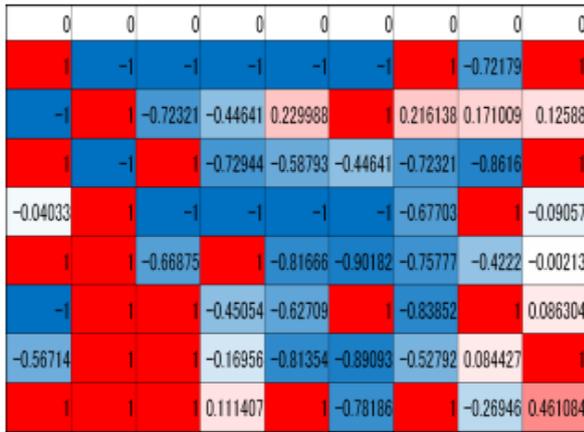
(b)

Fig.16: Action learning map (a) and emotion value map (b) at 24,000 times the number of learning when using

AGSOM

2	1	1	1	1	1	1	1	7
3	3	3	3	3	3	3	3	3
6	5	5	5	5	5	5	5	5
7	7	7	5	5	5	4	4	7
0	0	0	0	2	2	6	6	6
0	0	0	5	2	2	6	6	6
6	1	1	2	2	2	6	6	1
4	1	1	2	2	0	2	6	6
4	4	4	2	2	1	2	1	1

(a)



(b)

Fig.17: Action learning map (a) and emotion value map (b) at 24,000 times the number of learning when using an improved AGSOM

Fig.15 から Fig.17 はそれぞれ PL-G-SOM と非対称近傍関数, 改良した非対称近傍関数での感情値マップ (a) と行動学習マップ (b) である. 命令の習得度合いを感情値マップで表し, 8 種類の手の形状それぞれに対応した行動の数字を行動学習マップで表現した (感情値が+1 のユニットのみ着色した).

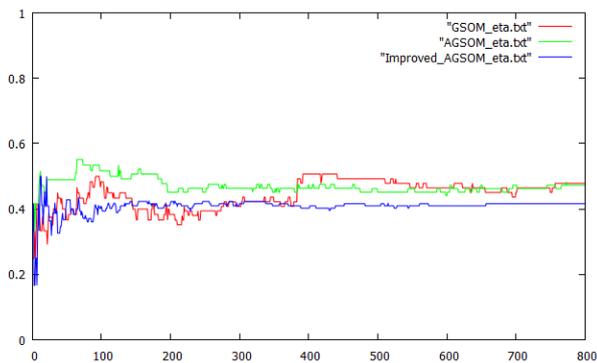


Fig.18: Transition graph of the order parameter in the 8,000 times number of learning

Fig.18 は, 手形状画像認識シミュレーションにおける学習回数 8,000 回での秩序率の推移グラフ である. 第 7 節でのシミュレーションにおいては, 改良した非対称近傍関数を導入した際は秩序率 $\eta=1$ となったが, 本シミュレーションではいずれの手法に関しても, $\eta=1$ に収束することはなかった. 導入したパラメータは, 秩序のあるマップであるかどうかを定量化するのに適切ではないと考え, 今後の課題としたい.

9 まとめ

本研究は, 非対称近傍関数及びその改良を G-SOM に導入し, AGSOM を提案した. AGSOM は学習過程におけるマップの位相不整合 (距離トポロジーねじれ) 現象を解消することができた. また, AGSOM を従来の PL-G-SOM を用いた手形状命令学習システムを導入し, 実験の結果より, AGSOM および改良 AGSOM がより入力データ空間に適合したマップ構築を行っていることが確認でき, 有効性を示した.

マップのユニット間の位相の整合性を評価するため, 秩序率の概念を導入し, シミュレーション結果より, PL-G-SOM より AGSOM, AGSOM より改良 AGSOM の学習結果がよりトポロジーのねじれや歪みを解消したことが明らかになった. しかし, 手形状命令学習システムの実験結果より, 秩序率により AGSOM と改良 AGSOM の効果は確認できず, 今後の課題に残る.

参考文献

- 1) Kuremoto T., Obayashi M., Kobayashi K., and L.-B. Feng: "Instruction learning systems for partner robots", *In Advances in Robotics - Modeling, Control and Applications*, (eds. Calin Ciufudean & Lino Garcia), *iConcept Press*, pp. 149-170, 2012 (Online Open Access)
- 2) 大谷拓寛, 呉本堯, 小林邦和, 大林正直: "PL-G-SOMを用いた手形状指令学習システムの構築", 第 21 回計測自動制御学会中国支部学術講演会論文集, pp. 168-169, 2012
- 3) T.コホネン: *自己組織化マップ*, シュプリンガー・フェアラーク東京出版, 1996
- 4) 羽野ともえ, 呉本堯, 小林邦和, 大林正直: "Transient-SOMを用いた手画像命令学習システム", *計測自動制御学会論文集*, Vol.43, No. 11, pp.1004-1006, 2007
- 5) Kuremoto T., Komoto T., Kobayashi K., Obayashi M.: "Parameterless-growing SOM and its application to a voice instruction learning system", *Journal of Robotics*, doi:

1155/2010/307293, 2010.

- 6) Aoki T., Aoyagi T.: “Self-organizing maps with asymmetric neighborhood function”, *Neural Computation*, Vol.19, No.9, pp.2515-2535, 2007
- 7) Fritzke B.: “Growing grid – a self-organizing network with constant neighborhood range and adaption strength”, *Neural Processing Letters*, Vol.2, No.5, pp.9-13, 1995
- 8) Bauer H.-U., & Villmann Th.: “Growing a hypercubical output space in a self-organizing feature map”, *IEEE Transaction on Neural Networks*, Vol.8, No.2, pp.218-226, 1997
- 9) Dittenbach M., Merkl D., & Rauber A.: “The growing hierarchical self-organizing map”, In *Proceedings of IEEE International Joint Conference on Neural Network (IJCNN '00)*, pp. 15-19, 2000
- 10) Berglund E., & Sitte J.: “The parameter-less self-organizing map algorithm”, *IEEE Transaction on Neural Networks*, Vol.17, No.2, pp.305-316, 2006

経験に基づく情動の再構成を導入したロボットの行動選択システム

○綿田将悟 大林正直 呉本堯 間普真吾 (山口大学) 小林邦和 (愛知県立大学)

A Decision Making System of Robots

Introducing Re-construction of Emotions Based on Experiences

* S. Watada, M. Obayashi, T. Kuremoto, S. Mabu (Yamaguchi University)
and K. Kobayashi (Aichi Prefectural University)

Abstract— We have already proposed a decision making system for autonomous robots, which is based on a Markovian emotional model proposed by Banik. In our proposed system, decision making of the robot is designed automatically by two processes. The first is a construction of emotions based on learning of the input stimuli. The second is optimization of behavior selection policy for emotions by optimization of system parameters which define behavior selection probability of each emotion. In our above previous studies, a construction of emotions based on learning of the input stimuli was processed using learning samples of input stimuli on offline in advance. In this study, we introduce an experiences-based reconstruction of emotions to our decision making system as an approach to an automatic construction of decision making online learning.

Key Words: Emotion model, Decision-making, Optimization problem

1 はじめに

自律的なコンピュータシステム開発のために、生物がもつ情動という概念・機能に注目し、情動のモデル化やその応用に関する研究が数多く存在する¹⁻⁴⁾。著者らは、Banikらによって提案されたマルコフ情動モデルを用いた、情動に基づく行動選択システムをすでに提案している⁵⁾。マルコフ情動モデルは有限オートマトンにより構成される確率モデルである⁴⁾。

提案システムにおいては階層型ニューラルネットワーク (Multilayer Perceptron: 以下 MLP) と自己組織化マップ (Self-Organizing Maps :以下 SOM) を用いた非線形変換の自己組織化の提案により、ロボットのタスクやスペックに適応した状況認知手法の自動的な構築を可能とした。また、パラメータ設計に依存する行動選択法を定義することにより、パラメータの最適化により適切な行動選択が可能となった。

提案システムは独自のシミュレーションにより評価を行い、タスクに沿った行動決定法が自動的に生成されることから提案手法の有用性が確認された。ただし以前の研究においては、入力刺激の学習による情動の形成は、予測される入力刺激サンプルを用いて事前にオフラインにより行われていた⁵⁻⁶⁾。そのため現実的でない入力刺激に対応するような無駄な領域が SOM に発生する。

本研究ではオンライン学習による移動ロボットの行動制御の自動構築へのアプローチのために、提案手法へ経験に基づく情動の再構成を導入した。つまりシステムにおける入力刺激に関する SOM を試行中にオンラインで更新することにより、実際の試行における入力刺激の実測値や頻度を考慮した、入力刺激とシステムパラメータとの対応付けを形成することが可能であると考えた。それに伴い、MLP を使用せず、SOM のみで内部パラメータを決定する方法へシステムの改良

を行った。

著者らは市販の Webots シミュレータ⁷⁾を用いて移動ロボット e-puck を対象としたシミュレーションを行い、我々の手法の有用性を確認した。タスクはロボットが通過した床面積を最大化する問題である。シミュレーションにより、タスク試行中に実際に得られた刺激入力を追加学習することにより SOM 中の不必要な刺激入力への対応付けが減少し、それにより生成されるロボットの行動決定法の性能を向上させることが示された。

本論文では、まず 2 章において、Banik らによって提案されたマルコフ情動モデルについて述べる。次に 3 章では、我々の提案手法であるマルコフ情動モデルに基づく行動選択法について述べる。4 章では、追加学習のための提案システムの改良点について述べる。5 章では提案システムに関する計算機シミュレーションについて述べ、最後に 6 章で考察と今後の課題を記す。

2 マルコフ情動モデル

Banik により提案されたマルコフ情動モデルは、複数の状態を示すオートマトンから構成される確率モデルであり、基本的な 4 つの情動 (Joy, Anger, Sadness, Fear) から成り立っている⁴⁾。マルコフ情動モデルの基本的な構造を Fig.1 に示す。このモデルの情動状態 Y の遷移は式(1)で行われる。

$$Y_{k+1} = CY_k \quad (1)$$

ここで Y_k は時刻 k における情動状態確率ベクトルであり、 Y は式(2)で表され、 T は転置である。

$$Y = [y_{joy} \quad y_{anger} \quad y_{sadness} \quad y_{fear}]^T \quad (2)$$

ただし、 Y の各要素は基本的な情動である Joy, Anger, Sadness, Fear に対応する各状態存在確率である。 C は状態遷移行列であり式(3)で表される。ただし、行列 C

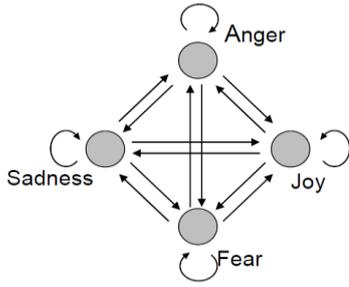


Fig. 1. Topology of Markovian Emotional Model.

の各要素は状態間の遷移確率であり、例えば P_{SR} は状態 R から状態 S への遷移確率である。これらの要素 P は入力刺激の影響を受けてオンラインで変化する。 P_{SR} の決定法の例として、 R を joy として、 joy から他の状態への遷移確率の決定法を式(4)に示す。 $\alpha, \beta, \gamma, \delta$ は情動誘発因子であり、基本的な情動状態 $Joy, Anger, Sadness, Fear$ を其々喚起する機能を持ち(例えば α は joy の情動状態確率を発生させるパラメータである)、内部状態を反映して変動する。 $q_{s/r}$ は情動遷移の事前確率であり、事前に設計されるべき値である。

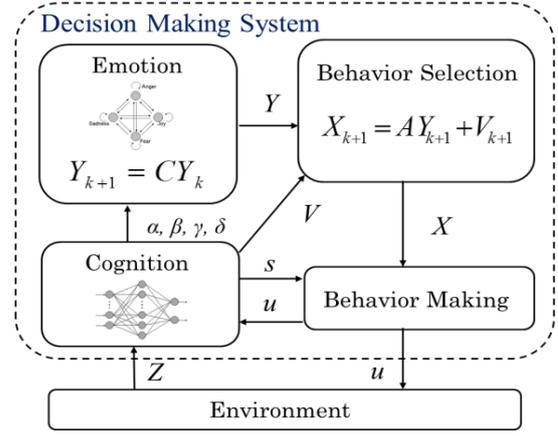
以上の過程により、情動確率ベクトル Y は情動誘発因子 $\alpha, \beta, \gamma, \delta$ により、内部状態を反映してリアルタイムに更新される。

$$C = \begin{bmatrix} P_{joy/joy} & P_{joy/anger} & P_{joy/sad} & P_{joy/fear} \\ P_{anger/joy} & P_{anger/anger} & P_{anger/sad} & P_{anger/fear} \\ P_{fear/joy} & P_{fear/anger} & P_{fear/sad} & P_{fear/fear} \\ P_{sad/joy} & P_{sad/anger} & P_{sad/sad} & P_{sad/fear} \end{bmatrix} \quad (3)$$

$$\left. \begin{aligned} P_{anger/joy} &= q_{anger/joy} + (\beta - \alpha)q_{anger/joy} \\ P_{sad/joy} &= q_{sad/joy} + (\delta - \alpha)q_{sad/joy} \\ P_{fear/joy} &= q_{fear/joy} + (\beta - \alpha)q_{fear/joy} \\ P_{joy/joy} &= 1.0 - (P_{anger/joy} + P_{sad/joy} + P_{fear/joy}) \end{aligned} \right\} \quad (4)$$

3 マルコフ情動モデルに基づくロボットの行動選択システム

マルコフ情動モデルに基づく従来システム⁴⁾では、情動生成の機能のみ実装されたため、それに基づくロボットの行動決定法は明確に定義されていなかった。そこで我々は、行動決定手法を含む総合的なシステムを提案した⁵⁾。提案システムの全体構造をFig.2に示す。システムは4つのモジュール、Cognition, Emotion, Behavior Selection, Behavioral Making から構成される。システムの一部(Cognition, Emotion)はBanikらによって提案された情動生成システム⁴⁾を利用している。ここで、 Z は刺激入力、 $\alpha, \beta, \gamma, \delta$ は情動誘発因子、 Y は情動状態確率ベクトル、 X は行動選択確率ベクトル、 V はロボットに設計者が先天的に与える行動選択を促す本能パラメータベクトル、 s はロボットの姿勢、 u はロボットの制御入力である。ここで k は時刻を表す。



Z : Measurement Y : Emotional state V : Instinctual parameter

C : Emotional state transition matrix A : Behavioral state transition matrix

s : Situation Behavior state $\alpha, \beta, \gamma, \delta$: Emotion-inducing factors

Fig. 2. A proposed behavior selection system

各モジュールの機能は、作用順に以下の通りである。Cognition モジュールは外部環境の状態やロボットの内部状態等の刺激入力 Z を認識し、情動誘発因子 $\alpha, \beta, \gamma, \delta$ を決定する。Emotion モジュールはBanikらによって提案されたマルコフ情動モデルによって構成されている。このモジュールはCognition モジュールが生成した情動誘発因子 $\alpha, \beta, \gamma, \delta$ を用いて情動状態確率ベクトル Y を生成し、Behavior Selection モジュールへ出力する。Behavior Selection モジュールはベクトル Y 及び本能パラメータベクトルである V を利用して行動選択確率ベクトル X を決定する。Behavior Making モジュールはロボットの姿勢である s を考慮して、ベクトル X に基づき決定された行動を実現するロボットの制御入力 u を出力する。

3.1 マルコフ過程に基づく行動選択手法

提案システムにおいて、行動決定はマルコフ過程に基づいて Behavior Selection モジュールで行われる。Behavior Selection モジュールはベクトル Y 及び本能パラメータベクトルである V を利用して行動選択確率ベクトル X を決定する。本研究ではFig.2における V を0とし、行動選択確率の決定は式(5)で表される。

$$X_{k+1} = AY_{k+1} \quad (5)$$

ここで X_k は時刻 k での行動選択確率ベクトルであり、ベクトルの各要素は行動選択確率である。例えば、ロボットが Drive, Left, Right, Back の4つの行動を取りえる場合、 X は式(6)で表される。また、 A は遷移行列であり、同様に式(7)で表される。ここで行列の各要素は遷移確率であり、 $P_{behavior/emotion}$ は情動状態 $emotion$ から行動 $behavior$ への遷移確率である。

本研究において V を0とした理由を次に述べる。 V を使用する理由は、単純な情動遷移モデルだけでは逐次的な行動決定を行う従来法のような形式的な行動選

択の実現が困難となることによる。例えば、ロボットがバッテリーを充電中、内部状態の変化により特定の探索行動を選択しやすくなるが、実際には、充電が完了するまで待機の方が効率的である。ただし、本研究ではロボットに先天的知識を与えないことを条件とし、 V を0とした。

$$X = [x_{drive} \ x_{left} \ x_{right} \ x_{Back}]^T \quad (6)$$

$$A = \begin{bmatrix} P_{Drive/joy} & P_{Drive/anger} & P_{Drive/sad} & P_{Drive/fear} \\ P_{Left/joy} & P_{Left/anger} & P_{Left/sad} & P_{Left/fear} \\ P_{Right/joy} & P_{Right/anger} & P_{Right/sad} & P_{Right/fear} \\ P_{Back/joy} & P_{Back/anger} & P_{Back/sad} & P_{Back/fear} \end{bmatrix} \quad (7)$$

3.2 遺伝的アルゴリズムによる設計パラメータの最適化

この行動選択手法においては、行動の選択確率行列である A の要素である各パラメータの設定次第で、様々な行動選択法を設計可能であることが特徴である。従って、これらのパラメータの設定に最適化手法を用いることで、タスクに適切な行動選択法を自動的に構築することが可能である。本論文ではこれらのパラメータ設定法として、近似解を探索するメタヒューリスティックアルゴリズムである遺伝的アルゴリズム (Genetic Algorithm : 以下 GA) を使用した。

3.3 情動誘発因子のニューラルネットワークを用いた表現

Cognition モジュールは、MLP により構成される (Fig.3)。出力層に2つのノードを持ち、其々Joyを喚起する α と Anger を喚起する β に対応する。Sadness、Fear を喚起するパラメータ γ, δ はそれぞれ対立する情動 Joy、Anger を喚起するパラメータの1に対する補確率、即ち $\gamma=1.0-\beta, \delta=1.0-\alpha$ で求められる。入力層のノード数は刺激入力数 N とし、中間層のノード数はタスクによって調整可能とする。

MLP は高い近似能力と汎化能力を持つが、学習には適切な教師信号が必要である。そこで、MLP の学習における教師データの生成に SOM を使用する。SOM を用いた学習の様子を Fig.5 に示す。SOM を使用する目的は、似た刺激入力を似た情動誘発因子に対応するクラスタリングを実現することである。提案システムでは、SOM の競合層におけるノードの位置を情動誘発因子の値に対応させる。即ち、SOM の2次元競合層のノード位置を2個の0から1の範囲に正規化された情動誘発因子 (Fig.4 では α と β) で構成される2次元空間と考えると、その刺激入力ベクトルに対する情動誘発因子の値が一意に決まり、学習後の2次元 SOM の競合層各ノードのコードベクトル (結合荷重ベクトル) はそれぞれ類似した入力ベクトル群が圧縮されたものと考えられることができる。

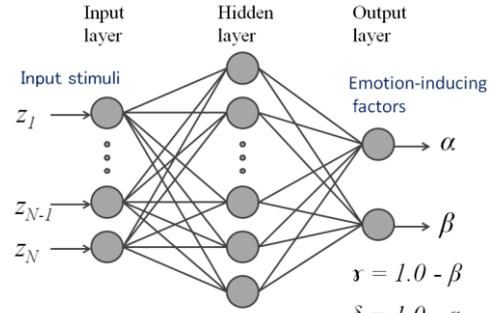


Fig. 3. Multilayer Perceptron used in Cognition module.

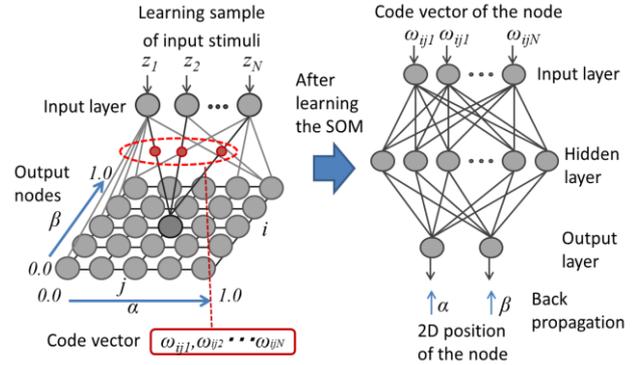


Fig. 4. Learning process of MLP using SOM.

次に、離散化されたコードベクトルと対応する情動誘発因子を連続値として取り扱えるように MLP の汎化能力を利用する。すなわち、コードベクトルを MLP の入力、正規化された2次元座標 (情動誘発因子の値) を教師データとして、BP法により MLP を学習する。学習の結果、刺激入力、情動誘発因子の値をそれぞれ連続値入出力とする非線形変換を行う MLP が生成される。

4 追加学習のための Cognition モジュールの改良

以前の提案システムにおいては、Cognition モジュールは MLP により構成され、MLP の学習における教師データの生成に SOM を使用していた。MLP を使用することで SOM の各ノードに離散的に保管されたコードベクトルを、連続的に扱うことができた。しかし MLP の教師有学習は多大な時間を要し、本論文のテーマである試行中にオンラインで状態空間を再構成する場合には不向きである。そこで本研究では MLP を使用せず、SOM のみを用いて Cognition モジュールを構成する。

本研究における Cognition モジュールの構造を Fig.5 に示す。SOM の入力層の各ノードは刺激入力に対応し、ノード数は刺激入力数 N である。また、従来の Cognition モジュール同様に SOM の競合層におけるノードの位置を情動誘発因子の値に対応させる。

つまり Cognition モジュールはロボットが得られる刺激入力を入力信号として競合層へクラスタリングす

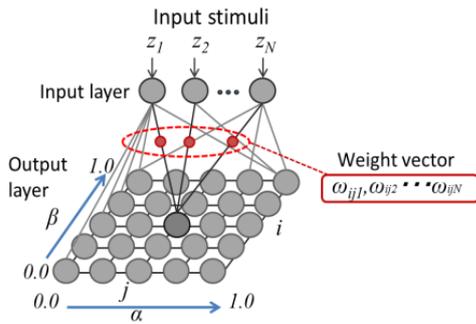


Fig. 5. New cognition module proposed in this paper.

ることにより、刺激入力と情動誘発因子との対応付けを構成する。また、刺激入力を入力信号として決定された勝者ノードの位置情報を 0 から 1 の範囲に正規化することで、刺激入力を情動誘発因子へ変換することが可能である。

本研究における改良により、Cognition モジュールが出力する情動誘発因子の値が離散的になり、複雑な刺激入力を扱う場合は性能が低下する恐れがある。しかし出力値の解像度は SOM のノード数に依存するため、SOM のサイズを調整することで性能低下を回避可能であると著者らは考える。

5 計算機シミュレーション

提案システムの有用性の検証のために、移動ロボットが通過した床面積の最大化に関するシミュレーションを行った。家庭用掃除ロボットなどにおいては、装備するセンサーなどの限られた情報をもとに障害を回避し、環境内のより多くの領域を通過することが要求される。そこで本研究では、距離センサーを備えた二輪移動ロボットが一定時間内により多くの領域を通過するための行動決定法が、提案手法を用いて自動構築される様子をシミュレーションにより示す。移動ロボットは e-puck を対象とし、シミュレータは現実に近い環境でシミュレーションが可能な Webots を使用した。

5.1 共通の問題設定

本研究では Fig.6 に示す 4 つの環境を用いてシミュレーションを行った。図中の 4 つの環境はそれぞれ一

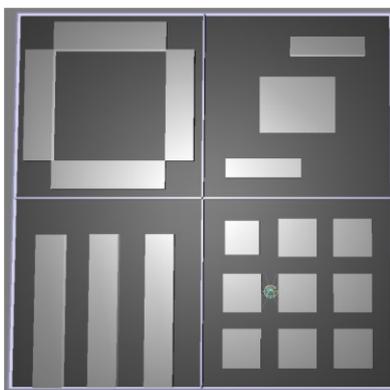


Fig. 6. Environments used in simulations.

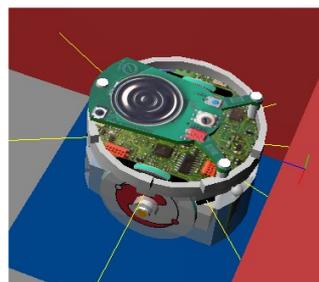


Fig. 7. E-puck robot in Webots simulator.

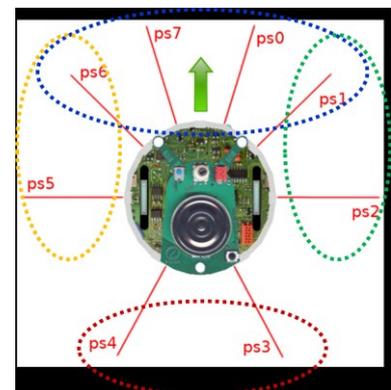


Fig. 8. Sensors of e-puck

辺 1m の正方形で、白い領域が壁、黒い領域がロボットが通過できる床である。対象ロボットの e-puck は、直径約 70[mm]高さ約 50[mm]の小型二輪移動ロボットであり 8 つの赤外線センサーを備えている。Webots における e-puck の様子を Fig.7 に示す。各赤外線センサーの測定レンジは 4cm である。2 つの車輪にはステッピングモーターを使用しており、移動距離や回転角の計算が可能である。

著者らはロボットの行動状態として直進、左折、右折、後退の 4 つを定義し、それぞれに対応する左右のモータートルクを次のように設定した：直進[100, 100]、左折[-50, 50]、右折 [50, -50]、後退 [-100, -100]。Behavior Making モジュールでは各行動に設定された左右のトルクを行動選択確率ベクトル X で重み付け合成することで出力トルクを計算する。

また、Cognition モジュールにおける SOM の競合層のノード数は 30×30 、入力層のノード数は 6 とし、対応する刺激入力を次の 4 つとした： z_0 ：前方 4 つのセンサー距離の平均 (Fig.8 参照)、 z_1 ：右側 2 つのセンサー距離の平均、 z_2 ：左側 2 つのセンサー距離の平均、 z_3 ：後方 2 つのセンサー距離の平均、 z_4 ：12 秒前からの直線移動距離の累計、 z_5 ：12 秒前からの回転角の累計。

5.2 改良型提案システムによる刺激入力と情動の対応付けと行動選択法の自動構築

まず SOM による刺激入力の学習について、事前学習のみを行い、試行による追加学習は行わない場合を考える。事前学習はセンサーの測定レンジ等の事前知識を基にランダムに作成した学習サンプルを用いる。

各刺激入力の取りえる値域を z_0, z_1, z_2, z_3 は $[0:4.0]$ 、 z_4, z_5 は $[-4.0:4.0]$ と設定し、ランダムに生成された学習サンプルを用いて 1000 回学習を行った。この時、近傍関数パラメータの初期値 $\sigma(0)=24.0$ 、学習係数 $\eta=0.1$ とした。学習後の SOM の競合層のノードが保管する結合荷重の様子を Fig.9 に示す。6 つの 2 次元マップは SOM の競合層に対応し、刺激入力別に競合層の各ノードが持つ結合荷重の値を色により表している。すべての刺激入力の組み合わせが表現されるように色が分布されていることがわかる。これらの 2 次元マップの横軸は

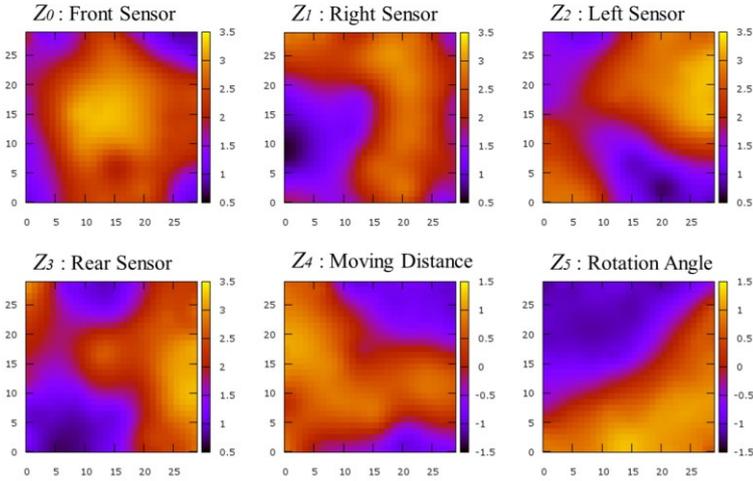


Fig. 9. Value of input stimuli stored by SOM.

情動誘発因子 α に、縦軸は β に対応しているため、刺激入力の組み合わせを情動マップ上に表現することで可読性を高めることが出来る。生成した情動マップを Fig.10 に示す。マップ上の大きな9つの円の中には、円が位置する情動を喚起する刺激入力の組み合わせが表されている。刺激入力の組み合わせは、障害物との近さを円の大きさで、直進距離と回転角を矢印の大きさで表現している。例えば、Joy は周辺に障害物が少なく直進移動距離が大きい場合に喚起されることが図より読み取れる。

次に事前学習後の SOM を用いて、GA によるパラメータ A の最適化によりロボットの行動決定法を構築した。GA では、各個体において一定時間タスクを実行し、タスク中に通過した床面積を個体の適応度とする。遺伝的進化処理を設定世代数行くとシミュレーション終了とする。まず、シミュレーション環境は Fig.6 左上の障害物が無い狭い環境のみを用いるとし、各個体のタスク開始時にロボットは所定の位置（中央）に設置される。世代数は 25、個体数は 12、1 個体におけるタスクの制限時間は 300s とした。

GA 実行時における各世代の最大適応度の推移を Fig.11 に示す。グラフより適応度が世代数の増加に従って増加していることがわかる。次に、最終世代における優秀個体のタスクの様子を観察した。個体のパラメータである確率行列 A を式(8)に、環境のロボットの軌跡を白色で示した図を Fig.12 に示す。また、図中には経路の一部を矢印で示している。ロボットは開始直後に少し右に曲り直進した。その後、周辺に壁が無い空間では直進、左前方に壁を検知すると急に右に曲がる、の二つの動作を繰り返した。

式(8)の A では、まず Drive の選択確率に関しては $P_{Drive/joy}$ と $P_{Drive/fear}$ が大きくなっており、Fig.10 の情動マップを参照すると、Joy は周辺に壁が無い状態（図中の中段右）、fear は後方と左に壁がある状態（下段中央）に対応している。また、Right の選択確率に関し

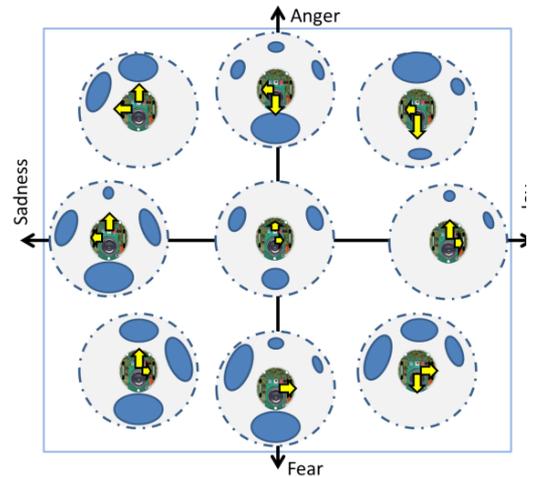


Fig. 10. Emotional map.

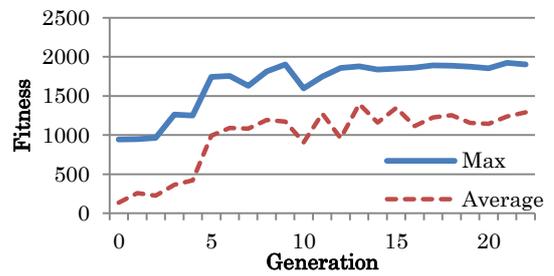


Fig. 11. Transition of fitness value.

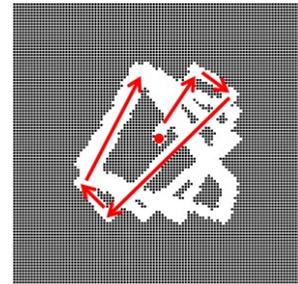


Fig. 12. Trajectory of the robot.

て $P_{Right/sad}$ と $P_{Right/ang}$ が大きく、Sad と Right が高い状態は左側と前方に障害物が近い状態（上段左）に対応している。これらによって前述したタスク中のロボットの行動が説明できる。

$$A = \begin{bmatrix} P_{Dri/joy} & P_{Dri/ang} & P_{Dri/sad} & P_{Dri/fea} \\ P_{Lef/joy} & P_{Lef/ang} & P_{Lef/sad} & P_{Lef/fea} \\ P_{Rig/joy} & P_{Rig/ang} & P_{Rig/sad} & P_{Rig/fea} \\ P_{Bac/joy} & P_{Bac/ang} & P_{Bac/sad} & P_{Bac/fea} \end{bmatrix} = \begin{bmatrix} 0.80 & 0.13 & 0.00 & 0.40 \\ 0.20 & 0.07 & 0.00 & 0.26 \\ 0.00 & 0.37 & 0.73 & 0.11 \\ 0.00 & 0.43 & 0.26 & 0.22 \end{bmatrix} \quad (8)$$

5.3 SOM の追加学習による情動の再構成

以前の提案システム⁵⁻⁶⁾では刺激入力の事前学習により刺激入力と情動の対応付けを構成した。しかし実際のタスク中に発生しない刺激入力に関しても対応付けが行われる。例えば、ロボットが四方を壁に囲まれている状態はタスク中には発生しない。よって SOM 上に参照されにくい無駄なノード領域が多数発生すると考えられる。そこで実際のタスク中に SOM の追加学習をすることにより、タスク中に発生しうる刺激入

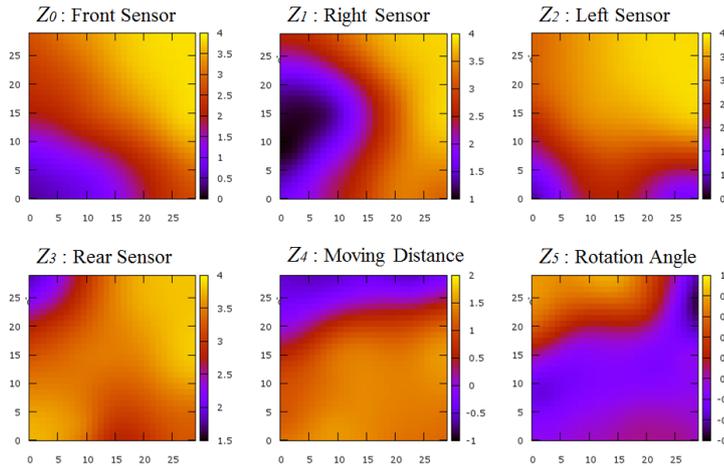


Fig. 13. Value of input stimuli stored by SOM after update.

力や発生頻度を考慮した、より効率的な対応付けが行われはざである。

本シミュレーションでは事前学習後の GA による最適化において、序盤の世代のタスク試行中に SOM の追加学習を導入する。つまり終了世代を 50 世代とし、そのうち最初の 10 世代において、試行中に発生した刺激入力を用いて SOM をオンラインで更新する。

追加学習前の SOM は 5.2 節で用いた Fig.9、Fig.10 で表される SOM を使用した。GA の個体数は 24、1 個体の 1 環境におけるタスクの制限時間は 600s とし、Fig.6 に示す 4 つの環境全てでタスクを実行した。各環境での試行における通過床面積の合計を個体の適応度とした。最初の 10 世代において SOM の更新は 1s 毎に行われ、よって SOM の追加学習の回数は 576,000 (=600×4×24×10) 回である。この時、近傍関数パラメータの初期値 $\alpha(0)=24.0$ 、学習係数 $\eta=0.0001$ とした。

追加学習により得られた SOM の競合層のノードが保管する結合荷重の様子を Fig.13 に示す。追加学習前の Fig.9 と比較して、各刺激入力に関して値の分布に偏りが生じている。すべての刺激入力の組み合わせを保有していた追加学習前の SOM に比べて、表現される刺激入力の組み合わせが減り、よりシンプルになっている。また、Fig.13 の結果を用いて作成した情動マップを Fig.14 に示す。Joy や Anger に対応する刺激入力では周辺に壁が少ない状態に対応しており、タスク中に頻繁に発生される刺激入力である。また、左側へのカーブや前方への移動が大きい刺激入力の組み合わせが多く見られる。これらの観察結果より、刺激入力のタスク中における発生頻度などが SOM に反映されていることがわかる。

次に環境のロボットの軌跡を白色で表した図を Fig.15 に示す。ロボットはタスク中、壁が無い空間ではやや左に曲がりながら直進し、正面に壁があるときに左へ大きく曲がり、左に壁がある時は右へ大きく曲

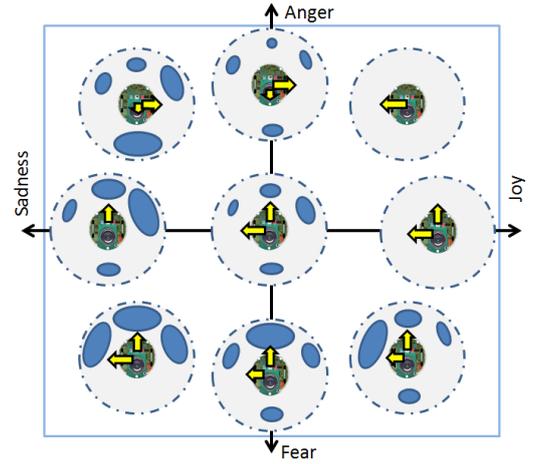


Fig. 14. Emotional map.

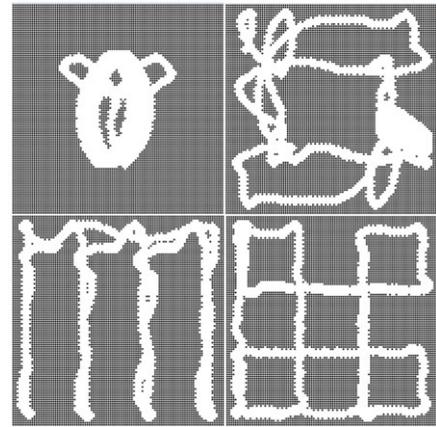


Fig. 15. Trajectory of the robot.

がった。また、左右を壁に囲まれている状態では直進した。また、個体のパラメータである確率行列 A を式 (9) に示す。5.2 節と同様に A によりロボットの行動決定法が説明できる。例えば、周辺に壁が存在しない状態に対応する Joy からの行動選択確率は Drive が 0.64、Left が 0.23 であり、やや左に曲がりながら直進の動作となる。また、右前方に壁がある状態に対応する Sadness には Left が、左に壁がある状態に対応する Fear には Right の選択確率が高くなっている。

$$A = \begin{bmatrix} P_{Dri/joy} & P_{Dri/ang} & P_{Dri/sad} & P_{Dri/fea} \\ P_{Lef/joy} & P_{Lef/ang} & P_{Lef/sad} & P_{Lef/fea} \\ P_{Rig/joy} & P_{Rig/ang} & P_{Rig/sad} & P_{Rig/fea} \\ P_{Bac/joy} & P_{Bac/ang} & P_{Bac/sad} & P_{Bac/fea} \end{bmatrix} = \begin{bmatrix} \underline{0.64} & 0.44 & 0.00 & 0.34 \\ \underline{0.23} & 0.16 & \underline{0.80} & 0.12 \\ 0.12 & 0.21 & 0.00 & \underline{0.42} \\ 0.01 & 0.19 & 0.20 & 0.10 \end{bmatrix} \quad (9)$$

次に追加学習を行わない従来の提案システムとの比較を行った。GA における各世代の最大適応度の推移を Fig.16 に示す。ここでグラフ中の垂直線は追加学習が終了する世代数 10 を示している。結果において最終的な適応度は追加学習を行ったシステムの方が高くなった。次に、タスク中に SOM の各ノードが勝利ノードとして参照された回数を調査し、ヒストグラムを作成した。Fig.17 にそれぞれのシステムにおける参照回数のヒストグラムを示す。事前学習のみのシステムで

は参照されるノード領域が集中しており、タスク中の情動誘発因子の変化が少ないことがわかる。一方、それに比較すると追加学習を行ったシステムでは参照されたノード領域が分散し、タスク中の情動誘発因子の変化が大きい。つまり、追加学習を行ったシステムではシミュレーション環境内で発生する刺激入力の変化に情動誘発因子が影響を受けやすい、情動誘発因子と刺激入力とのより効率的な対応付けが行われたことがわかる。

これらの結果より、タスク試行中に実際に得られた刺激入力を追加学習することにより SOM 中の不必要な刺激入力への対応付けが減少し、生成されるロボットの行動決定法の性能を向上させることが示された。

6 結言

本論文では、著者らが提案したマルコフ情動モデルに基づく行動選択システムを追加学習可能なシステムに改良し、タスク試行において刺激入力をオンラインで学習することを提案した。従来の提案システムでは刺激入力は事前知識によりオフラインで行われていた。そのため実際のタスクでは発生しない刺激入力に対しても情動が対応付けられていた。そこで、実際のタスクで得られた刺激入力をを用いた追加学習により不必要なノード領域が減少し、より効率的なパラメータ決定が可能であると考えた。

以前の提案システムにおける Cognition モジュールでは、MLP による近似学習に多大な時間を要するため、刺激入力の追加学習には不適切であった。そこで MLP を使用しない SOM のみで構成される Cognition モジュールへと改良した。

改良型提案システムは Webots シミュレータを用いて、移動ロボット e-puck を対象とした通過床面積最大化問題に関するシミュレーションを行った。結果ではまず、従来の提案手法を用いたロボットの行動決定の自動生成について示した。また、学習後の SOM の様子や獲得された行動選択確率を解析することにより、生成されたロボットの行動決定法について説明が可能であることを示した。次に追加学習を導入した改良型提案システムについての結果を示した。適応度の推移と各ノードの選択回数のヒストグラムにより、従来の提案手法との比較を行った。結果では追加学習によって SOM の不必要なノード領域が減少し、それによりロボットの行動決定法が改善されたことが示された。

ただし結果より、追加学習によるヒストグラムの改善に比べて行動決定法の改善はわずかであった。また、類似した刺激入力に対応するノード領域が大きくなると、刺激入力の変化に対して情動誘発因子が過剰に反応することが考えられる。これらの点から、追加学習における学習パラメータや学習方法の改善等を検討する必要があると考えられる。

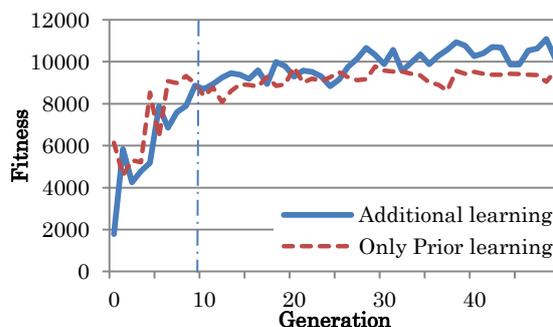


Fig. 16. Transition of fitness value.

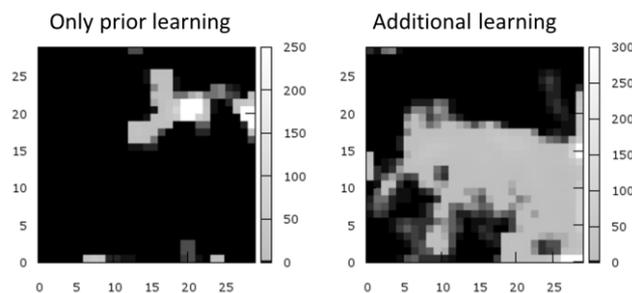


Fig. 16. The histogram of the number of times each node is selected.

また、今後は行動決定法のオンライン学習に向けたシステムの改善を行っていく。現システムではシステムパラメータの決定に GA を用いているため、実機による行動決定法の獲得は困難と考えられる。そこで、試行中に逐次的にパラメータを最適化する方法や、逐次的なシステムの評価が可能なタスクを考案していきたい。

参考文献

- 1) 久保田直行, 脇坂史帆, 小嶋宏幸: “情動モデルを用いたパートナーロボットに関する研究: 仮想現実空間の構築と人間との相互作用”, 知能と情報 (日本知能情報ファジィ学会誌) Vol.20, No.4, pp.449-460
- 2) J.Moren, C.Balkenius: “A computational Model of Emotional Learning in the Amygdla”, Cybernetics and System32(6), pp.611-636, 2001
- 3) M. Obayashi, T. Takuno, T. Kuremoto, K. Kobayashi: “An Emotional Model Embedded Reinforcement learning System”, Proc. of IEEE International Conference on Systems, Man, and Cybernetics, (SMC2012),pp.1058-1063, 2012
- 4) Sajal Chandra Banik, Keigo Watanabe, Kiyotaka Izumi: “Improvement of group performance of job distributed mobile robots by an emotionally biased control system”, Artif Life Robotics(12),
- 5) 綿田将悟 大林正直 呉本堯 小林邦一 間普真吾: マルコフ情動モデルを備えたロボットの行動決定法, 電気学会 論文誌 C, Vol.134, No.1 (2014) 掲載予定
- 6) 綿田将悟 大林正直 呉本堯 小林邦一 間普真吾: 情動形成に基づく行動選択システムとその実機ロボットへの応用, 第 22 回計測自動制御学会中国支部 学術講演会論文集 (2013)
- 7) “Webots: robot simulator“, <http://www.cyberbotics.com/>

CPG と強化学習を用いた多脚ロボットの行動獲得の試み

石倉裕貴 岸本良一 ○堀内 匡 (松江工業高等専門学校)

A Study on Behavior Acquisition of Multi-Legged Robot by CPG and Reinforcement Learning

Y. Ishikura, R. Kishimoto and *T. Horiuchi (Matsue College of Technology)

Abstract— Reinforcement learning is one of behavior learning methods by which autonomous agents acquire action rules to adapt to unknown environments. The objective of the agent is to maximize the sum of the received reward. Many studies in reinforcement learning have used only computer simulations in their experiments. In this study, reinforcement learning is applied to a multi-legged robot which has Central Pattern Generator (CPG) in the real environment. CPG parameters to generate efficient walking patterns are optimized using reinforcement learning, instead of learning the goal angle of each motor at each time step. The objective for the robot is to acquire the efficient and smooth forward movement to the goal direction, by itself. Through the real robot experiment, we confirm the effectiveness of our method using CPG and reinforcement learning.

Key Words: Reinforcement learning, CPG, Multi-legged robot, Behavior acquisition, Real robot environment

1 はじめに

人間とロボットの共存に向けて、ロボットが自律的に行動を獲得するための技術が必要不可欠であり、強化学習はその技術として有望な手法である。我々は、実機ロボットを対象とした強化学習に関する研究を進めている。これまでに、六脚ロボットに価値反復を用いた強化学習を適用し、前進行動が獲得できることを示した¹⁾。しかしながら、獲得した前進行動は、1ステップ毎に静止する行動であり、滑らかな行動パターンではなかった。

これに対して動物は、柔軟に環境に適応するだけでなく、滑らかでエネルギー効率の良い歩行をしていることが知られている。その歩行パターンは、CPG (Central Pattern Generator) によって生成されていることが示唆されている²⁾³⁾。我々は、CPG と多目的最適化を用いたロボットの歩行パターンの獲得に関する研究を進めている。これまでの研究において、ROBOTIS 社の Bioloid で組み立てた犬型ロボットにおいて、CPG と多目的 GA を用いて滑らかな前進行動の獲得を実現した⁴⁾⁵⁾。その際、犬型ロボットのシミュレーションモデルを用いて CPG のパラメータを最適化することで、滑らかな前進行動を獲得した。

しかしながら、上記の手法はシミュレーションモデルのないロボットでは適用できない。そこで本研究では、シミュレーション環境のない多脚ロボットを対象として、CPG と強化学習を組み合わせた手法により、多脚ロボットが滑らかな前進行動パターンを獲得することを実現する。

2 六脚ロボット

本研究では、ROBOTIS 社のロボット製作キット Bioloid⁶⁾ で組み立てた六脚ロボット (Fig. 1 参照) を用いる。このロボットはそれぞれの脚に 2 個ずつ、全部で 12 個のモータで構成されている。これら 12 個のモータを q_1, \dots, q_{12} と表記する。

本研究では、12 個のモータすべてを学習対象とするのではなく、以下の式 (1) の拘束条件を設けることに

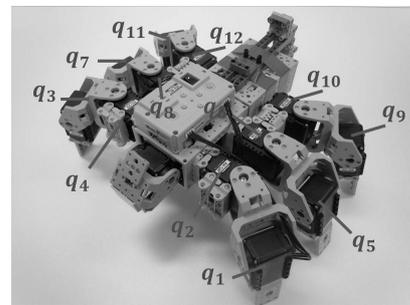


Fig. 1: Six-legged robot

より、5 個のモータ $q_1, q_2, q_3, q_9, q_{10}$ を CPG が駆動する対象とする。

$$\begin{cases} q_3 = -q_4 = -q_7 = q_8 = q_{11} = q_{12} \\ q_5 = q_6 = 0 \end{cases} \quad (1)$$

3 CPG を用いた歩行パターン生成

3.1 三角関数を利用した CPG

CPG モデルには、松岡モデル⁷⁾ やファンデルポール振動子モデル⁸⁾ など幾つかのモデルが提案されている。本研究では、Cohen らがモデル化した三角関数を利用した CPG モデル¹¹⁾ を用いる。そのダイナミクスを以下に示す。

$$\dot{\theta}_i = u_i^r + \sum \alpha_{ij} (\sin(\theta_j - \theta_i - u_i^n)) \quad (2)$$

$$\zeta_i = u_i^A \sin(\theta_i) \quad (3)$$

ここで、 θ_i, ζ_i はそれぞれ振動子 i の状態を与える角度、振動子 i の状態であり、 u_i^r と u_i^n と u_i^A は CPG の内部パラメータである。 u_i^A はモータ i の振幅を制御する変数に対応しており、 u_i^r は CPG の目標角速度を制御する変数、 u_i^n は CPG 間の目標位相差を制御する変数にそれぞれ対応している。

3.2 六脚ロボットの歩行パターン生成

本研究では、式 (1) の拘束条件によって、5 個のモータ $q_1, q_2, q_3, q_9, q_{10}$ に振動子をそれぞれ割り振り、制御

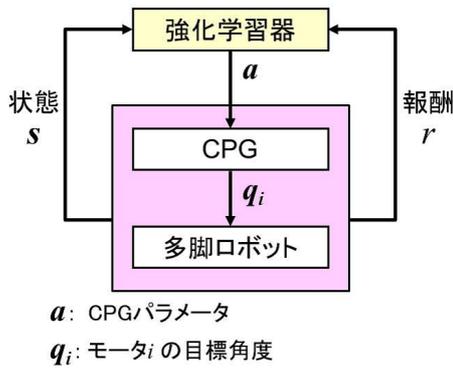


Fig. 2: Framework of reinforcement learning with CPG

することにする. CPG のダイナミクスおよびモータへの出力を以下に示す.

$$\begin{aligned}
 \dot{\theta}_1 &= u_1^r + \sin(\theta_3 - \theta_1 - u_1^n) \\
 \dot{\theta}_2 &= u_2^r + \sin(\theta_1 - \theta_2 - u_2^n) \\
 \dot{\theta}_3 &= u_3^r + \sin(\theta_1 - \theta_3 - u_3^n) \\
 \dot{\theta}_9 &= u_9^r + \sin(\theta_1 - \theta_9 - u_9^n) \\
 \dot{\theta}_{10} &= u_{10}^r + \sin(\theta_1 - \theta_{10} - u_{10}^n) \\
 \zeta_i &= u_i^A \sin(\theta_i) \\
 q_i &= \zeta_i
 \end{aligned} \quad (4)$$

本研究では, ロボットの各モータの各時刻における目標角度を学習するのではなく, 上記の CPG により各モータの出力パターンが生成されるものとして, CPG のパラメータのうち u_i^A を強化学習により最適化する. この枠組みを Fig. 2 に示す.

4 強化学習の適用

本研究では, 六脚ロボットが目標の方向に早く前進する行動を強化学習により獲得する問題を考える. 本研究では, 強化学習の手法として Q 学習を適用する.

4.1 Q 学習

Q 学習は, 離散的な状態行動空間を対象とした強化学習アルゴリズムである. Q 学習では, 状態と行動の組に Q 値という評価値を与える. したがって, Q 値は状態行動価値 $Q(s, a)$ である. 状態 s において行動 a を選択し, 次状態 s' に遷移したとき, 報酬 r を得たとする. このとき, 以下の式により Q 値を更新することで学習を進める.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (5)$$

なお, α は学習率, γ は割引率と呼ばれるパラメータである.

Q 学習では行動選択手法を決める必要がある. 今回は ϵ の確率でランダムな行動を行い, $1 - \epsilon$ の確率で Q 値の最も高い行動を行う ϵ -greedy 法を用いる

4.2 状態行動空間の構成

Q 学習を適用するためには, 状態と行動を定義する必要がある. 本研究では, 4 個のモータ q_1, q_2, q_9, q_{10} の振幅 $u_1^A, u_2^A, u_9^A, u_{10}^A$ を決めることを行動とする. 左前脚先のモータ q_3 の振幅 u_3^A は $8.8[deg]$ に固定する. 振

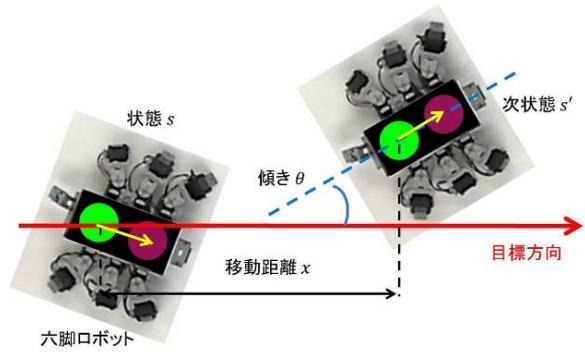


Fig. 3: Problem setting

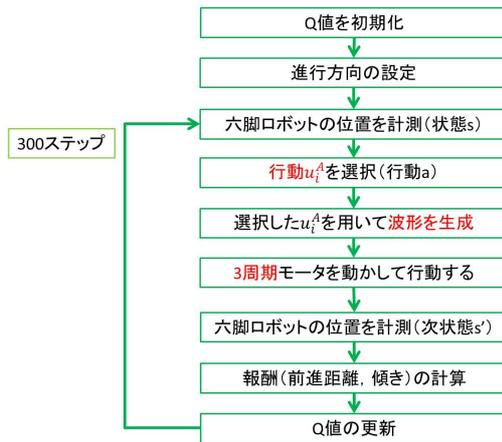


Fig. 4: Procedure of experiment

幅 $u_1^A, u_2^A, u_9^A, u_{10}^A$ は, $0[deg]$ もしくは $20[deg]$ のいずれかの値をとるものとする. 状態は, 六脚ロボットの向きと目標方向のずれの角度 θ を考え, この角度 θ を $-54[deg] \sim 54[deg]$ の範囲を 9 等分して離散化する. したがって, 離散的な状態行動空間となる.

4.3 報酬関数

CPG が生成した波形で 3 周期分だけ六脚ロボット動かすことを 1 ステップとして, 1 ステップで進んだ目標方向への移動距離 $x[cm]$ およびロボットの向きと目標方向のずれの角度 $\theta[deg]$ をもとにした報酬 r_t を設定する (Fig. 3 参照). そこで, 報酬関数を以下の式で定義する.

$$r = \begin{cases} \alpha x + |\cos \theta| - 1.5 & (-54 \leq \theta \leq 54) \\ -10 & (\text{otherwise}) \end{cases} \quad (6)$$

ここで, α は比例定数であり, 第一項と第二項のバランスを考慮し, $1/9$ とした. また, 第三項の -1.5 は 1 ステップの行動に対するコストとして与えている.

なお, 本研究では 4 つのモータの振幅 $u_1^A, u_2^A, u_9^A, u_{10}^A$ のみを学習対象とするため, その他の CPG パラメータは以下の通りとした.

$$\begin{aligned}
 u_1^r &= u_2^r = u_3^r = u_9^r = u_{10}^r = 1[rad/s] \\
 u_1^n &= u_9^n = 0[deg] \\
 u_2^n &= u_{10}^n = 180[deg] \\
 u_3^n &= 90[deg]
 \end{aligned} \quad (7)$$

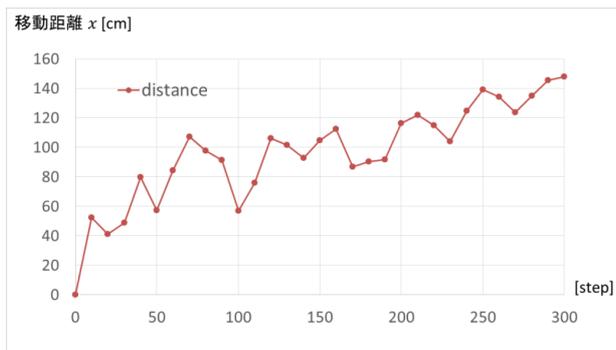


Fig. 5: Experimental result

5 実験

5.1 実験方法

Q 学習におけるパラメータとして、学習率 α と割引率 γ がある。本実験では、学習率 α は 0.5, 割引率 γ は 0.9 とした。行動選択には ϵ -greedy 法を用いる。 ϵ の値は 150 ステップまでは $\epsilon = 0.5$ に固定し、その後は 250 ステップで $\epsilon = 0$ になるように線形で減少させた。

前節で述べたように、CPG が生成した波形で 3 周期分だけ六脚ロボット動かすことを 1 ステップとして、300 ステップまで実験を行う。この実験を 3 セット行う。実験の手順を Fig. 4 に示す。

六脚ロボットの位置と向きは、天井付近に設置した Web カメラを用いて計測する。具体的には、ロボットの背中に紫色と緑色の 2 色のマーカ (Fig. 3 参照) を取り付け、天井近くに設置した Web カメラでそれらのマーカを追跡することにより、ロボットの位置と向きを算出する。これは OpenCV を用いたプログラムにより実現されている。

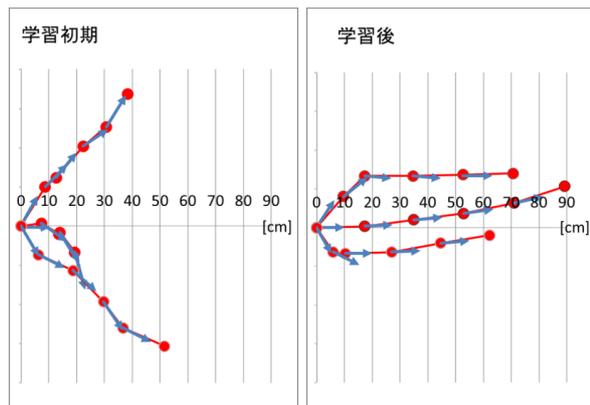
5.2 実験結果

まず、10 ステップ毎のロボットの目標方向への移動距離 $x[\text{cm}]$ の変化を Fig. 5 に示す。これは 3 セットの実験の平均をとったものである。Fig. 5 より、ロボットの目標方向への移動距離が着実に増えていることがわかる。

また、学習初期と学習後におけるロボットの移動軌跡の例を Fig. 6 に示す。Fig. 6 より、学習初期では目標方向でない方向に進んでいるのに対して、学習後では目標方向に進むことが学習できていることがわかる。

6 まとめ

本研究では、六脚ロボットを対象として、CPG と強化学習を組み合わせた手法により、六脚ロボットが目標の方向に向かって滑らかに前進する行動パターンを獲得することを実現した。この六脚ロボットにはシミュレーションモデルが存在しないため、シミュレーション環境ではなく実機環境において CPG のパラメータを学習する必要がある。そこで、強化学習を用いて CPG のパラメータを実機環境下で学習し、滑らかな前進行動パターンを獲得した。このとき、ロボットの各モータの各時刻における目標角度を学習するのではなく、CPG により各モータの出力パターンが生成されるものとし



(a) Initial stage of learning

(b) After learning

Fig. 6: Examples of robot movement: changes of robot position



Fig. 7: Four-legged robot

て、CPG のパラメータを強化学習により学習している。具体的には、三角関数を利用した CPG モデルを用いて歩行パターンを生成し、その CPG パラメータを Q 学習により最適化することを試みた。

現在、六脚ロボット以外の多脚ロボットとして、Fig. 7 に示す四脚ロボットへの適用も進めている。四脚ロボットは六脚ロボットに比べて安定性を維持しにくい構造であるため、倒れにくい歩行動作を獲得する必要がある。四脚ロボットを対象とした実験結果については、発表当日に報告することを考えている。

今後の課題として、1) 学習対象の CPG パラメータを増やすこと、2) 連続値の状態行動空間を扱う強化学習手法の導入、3) 目標の方向と地点を与えてロボットが目標地点に到達して止まる行動の獲得、などが挙げられる。

参考文献

- 1) 石原弘二, 大村佑貴, 堀内 匡, モデルベース強化学習を用いた六脚ロボットの前進行動獲得, 計測自動制御学会論文集, **49-3**, 408/410 (2013)
- 2) 多賀徹太郎, 脳と身体の動的デザイナー—運動・知覚の非線形力学と発達, 金子書房 (2002)
- 3) A.J. Ijspeert, Central Pattern Generators for Locomotion Control in Animals and Robots: a Review, *Neural Networks*, **21-4**, 642/653 (2008)

- 4) 石原弘二, 堀内 匡, 多目的 GA を用いた四脚ロボットの歩行パターン獲得, 電気学会研究会資料, システム研究会 ST-12-017, 35/40 (2012)
- 5) 萩原涼太, 石原弘二, 堀内 匡, CPG と多目的 GA を用いた四脚ロボットの歩行パターン獲得, 第 57 回システム制御情報学会研究発表講演会講演論文集, 347-4 (2013)
- 6) <http://www.robotis.com/xenobioid>
- 7) K. Matsuoka: Sustained oscillations generated by mutually inhibiting neurons with adaptation, *Biological Cybernetics*, **52**, 367/376 (1985)
- 8) 泉 清高, 田島彰人, 渡辺桂吾, 劉 光磊, ファンデルポール振動子を用いた CPG ネットワークの構成法, 第 16 回インテリジェント・システム・シンポジウム講演論文集, 365/368 (2006)
- 9) R.S. Sutton and A.G. Barto: *Reinforcement Learning: An Introduction*, The MIT Press (1998) (三上貞芳, 皆川雅章 (訳): 強化学習, 森北出版, 2001)
- 10) 牧野研司, 中村 泰, 柴田智広, 石井 信, CPG-Actor-Critic 法によるミミズ型ロボットの推進運動の獲得, 電子情報通信学会技術研究報告, NC2006-152 (2007)
- 11) A.H. Cohen, P.J. Holmes, and R.H. Rand: The nature of coupling between segmental oscillators of the lamprey spinal generator for locomotion, a mathematical model. *Journal of Mathematical Biology*, **13**, 345/369 (1982)

極変数複素ニューロンにおけるプラトー

○新田 徹 (産業技術総合研究所)

Plateau in the Polar Variable Complex-Valued Neuron

*Tohru Nitta (National Institute of Advanced Industrial Science and Technology, AIST)

Abstract - In this paper, the plateau of the polar variable complex-valued neuron model is investigated via computer simulations. The main results are as reported below. (a) There could be the cases that a quasi-plateau consists of several sub-plateaus, reducing the learning error albeit only slightly. (b) A theoretical result is confirmed by computer simulations: the speed of change of the amplitude is faster than that of the phase around singular points.

Key Words: Neural network, Singular point, Complex number, Polar coordinate, Plateau

1 はじめに

複素ニューラルネットワークは、複素データや2次元データを情報処理するのに適している^{1, 2, 3}。本稿では、実数値の重みと閾値を持つ通常のニューロンを実ニューロンと呼び、実ニューロンから構成されるニューラルネットワークを実ニューラルネットワークと呼ぶことにする。一般に、複素数の表現方法には、直交直線座標による方法と、極座標による方法の2種類がある。パラメータ(重みと閾値)が直交直線座標で表現されるような複素ニューロンを“直交直線変数複素ニューロン”と呼び、パラメータが極座標で表現されるような複素ニューロンを“極変数複素ニューロン”と呼ぶことにする。極変数複素ニューロンから構成される複素ニューラルネットワークモデルとその応用に関しては、文献^{1, 4, 5, 6}を参照されたい。

ところで、近年、学習モデルが特異点との関係で調べられつつある^{7, 8, 9, 10}。たとえば、階層型ニューラルネットワークや混合正規分布といった、階層構造や結合荷重の交換に関する対称性を持った学習モデルは概ね特異点を持っている。特異点は学習モデルの学習ダイナミクスに影響を与え、学習の停滞を招く原因であることなどが分かってきた。

極変数複素ニューロンの学習ダイナミクスについては、特異点と関連付けて、文献^{11, 12}において、既に報告した。本稿では、その後行ったプラトーに関する分析結果について報告する。論文を自己完結させるために、第2章に文献¹¹の解析結果を記す。

2 解析

2.1 極変数複素ニューロンの識別不能性

極変数複素ニューロンが識別不能性を持つことを示す。識別不能性の厳密な定義は^{7, 9}を参照されたい。本稿では、極変数複素ニューロンが同一の出力値を取るようなパラメータ値から成る連結集合を危集合(critical set)と呼び、危集合上の点を特異点と呼ぶ。連結集合だけを解析の対象としたのは、非連結集合は学習ダイナミクスに悪い影響を与えないと思われるからである。

次のような N 入力の極変数複素ニューロンを考

える。出力値 v は次のように定義される:

$$v = f_C \left(\sum_{k=0}^N r_k \exp[i\theta_k] \cdot z_k \right) \in \mathcal{C}, \quad (1)$$

ここで、 \mathcal{C} は複素数全体の集合、 $z_k \in \mathcal{C}$ は k 番目の入力信号、 $r_k \exp[i\theta_k] \in \mathcal{C}$ は k 番目の入力信号に対する重み($r_k \in \mathbf{R}$ は振幅、 $\theta_k \in \mathbf{R}$ は位相)($1 \leq k \leq N$)、 $i = \sqrt{-1}$ 、 $z_0 \equiv 1$ 、 $r_0 \exp[\theta_0] \in \mathcal{C}$ は複素ニューロンの閾値($r_0 \in \mathbf{R}$ は振幅、 $\theta_0 \in \mathbf{R}$ は位相)、 \mathbf{R} は実数全体の集合である。また、 $f_C : \mathcal{C} \rightarrow \mathcal{C}$ は活性化関数である。

上記の極変数複素ニューロンにおいて、ある $0 \leq k \leq N$ に対して、 $r_k = 0$ ならば、 $r_k \exp[i\theta_k] \cdot z_k = 0$ となり、 θ_k がどのような値を取っても、複素ニューロンの出力値 v に影響を与えない。故に、 θ_k は識別不能なパラメータであり、極変数複素ニューロンは識別不能性を持っていることが分かる。次に、上記極変数複素ニューロンの危集合を具体的に求める。まず、

$$M \stackrel{\text{def}}{=} \{(\mathbf{r}, \Theta) \in \mathbf{R}^{N+1} \times \mathbf{R}^{N+1}\}, \quad (2)$$

$$\mathbf{r} \stackrel{\text{def}}{=} \begin{bmatrix} r_0 \\ \vdots \\ r_N \end{bmatrix} \in \mathbf{R}^{N+1}, \quad (3)$$

$$\Theta \stackrel{\text{def}}{=} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_N \end{bmatrix} \in \mathbf{R}^{N+1} \quad (4)$$

とおく。 M は、上記の極変数複素ニューロンを規定するパラメータ空間である。そこで、任意の $(\mathbf{r}', \Theta') \in M$ と任意の $0 \leq k \leq N$ に対して、

$$C_k(\mathbf{r}', \Theta') \stackrel{\text{def}}{=} \{(\mathbf{r}, \Theta) \in M \mid r_0 = r'_0, \dots, r_{k-1} = r'_{k-1}, r_k = 0, r_{k+1} = r'_{k+1}, \dots, r_N = r'_N, \theta_0 = \theta'_0, \dots, \theta_{k-1} = \theta'_{k-1}, \theta_{k+1} = \theta'_{k+1}, \dots, \theta_N = \theta'_N\} \quad (5)$$

とおくと、上記極変数複素ニューロンの危集合 $C(\mathbf{r}', \Theta')$ は、

$$C(\mathbf{r}', \Theta') = \cup_{k=0}^N C_k(\mathbf{r}', \Theta') \quad (6)$$

で与えられる.

2.2 極変数複素ニューロンの特異点近傍における学習ダイナミクス

極変数複素ニューロンの特異点近傍における学習ダイナミクスを文献⁸⁾の解析方法を用いて調べる.

2.1節で定義した極変数複素ニューロンを解析対象とする. ただし, 活性化関数 f_C は, 簡単のために, 線形関数とする:

$$f_C(z) = z, \quad z = x + iy. \quad (7)$$

2乗誤差を $E = (1/2)|t - v|^2$ と定義する ($t \in \mathbf{C}$ は教師信号, $v \in \mathbf{C}$ は実際の出力値).

最急降下法を用いると, 学習則は次のようになる: 任意の $0 \leq k \leq N$ に対して,

$$\begin{aligned} \Delta r_k(n) &\stackrel{\text{def}}{=} r_k(n+1) - r_k(n) \\ &= -\varepsilon \cdot \frac{\partial E}{\partial r_k} \\ &= \varepsilon \cdot \text{Re} [\bar{\delta} \cdot z_k \cdot \exp[i\theta_k(n)]], \end{aligned} \quad (8)$$

$$\begin{aligned} \Delta \theta_k(n) &\stackrel{\text{def}}{=} \theta_k(n+1) - \theta_k(n) \\ &= -\varepsilon \cdot \frac{\partial E}{\partial \theta_k} \\ &= -\varepsilon \cdot r_k(n) \cdot \text{Im} [\bar{\delta} \cdot z_k \cdot \exp[i\theta_k(n)]]. \end{aligned} \quad (9)$$

ただし, $\delta \stackrel{\text{def}}{=} t - v$, \bar{z} は複素数 z の複素共役, n は学習回数を表す変数であり, たとえば, $r_k(n)$ は n 回目の学習終了後のパラメータ r_k の値を表す.

ここで, 任意の $0 \leq k \leq N$ に対して,

$$M_{r_k} \stackrel{\text{def}}{=} \{ (\mathbf{r}, \Theta) \in M \mid \Delta r_k = 0 \}, \quad (10)$$

$$M_{\theta_k} \stackrel{\text{def}}{=} \{ (\mathbf{r}, \Theta) \in M \mid \Delta \theta_k = 0 \} \quad (11)$$

と定義すると, 学習則 (式 (8), (9)) から,

$$M_{r_k} = \{ (\mathbf{r}, \Theta) \in M \mid \text{Re} [\bar{\delta} z_k \cdot \exp[i\theta_k]] = 0 \}, \quad (12)$$

$$M_{\theta_k} = \{ (\mathbf{r}, \Theta) \in M \mid r_k \cdot \text{Im} [\bar{\delta} z_k \cdot \exp[i\theta_k]] = 0 \}. \quad (13)$$

そこで, 特異点近傍の学習の振る舞いを調べる. 特異点 $r_k = 0$ ($k = 0, \dots, N$) の近傍においては, 式 (8), (9) から, $k = 0, \dots, N$ に対して,

$$\Delta r_k = \varepsilon \cdot \text{Re} [\bar{\delta} \cdot z_k \cdot \exp[i\theta_k]], \quad (14)$$

$$\Delta \theta_k \approx 0. \quad (15)$$

故に, 振幅 r_k ($k = 0, \dots, N$) の変化の速さは, 位相 θ_k ($k = 0, \dots, N$) の速さに比べて速く, 状態は部分多様体 $\bigcap_{k=0}^N M_{r_k}$ に引き付けられる ($\Delta r_k \approx 0$ ($k = 0, \dots, N$) という状態に近づく). つまり, $\bigcap_{k=0}^N \{M_{r_k} \cap M_{\theta_k}\}$ という平衡状態に入る. そして, パラメータ $(\mathbf{r}, \Theta) \in M$ はほとんど変化しなくなる. これが学習曲線におけるプラトー現象である. この現象は文献⁸⁾において示された実ニューラルネットワークの特異点近傍における学習ダイナミクスと同様のものである.

Table 1: Training pattern used in the experiment

	入力	出力
学習パターン	1.0	0.5i

Table 2: Initial values of amplitude of weight. Case 1: Learning is started from near the singular point. Case 2: Learning is started from off the singular point.

	r_1
ケース 1	0.00001
ケース 2	1.0

2.3 実験

簡単のために, 1 入力の極変数複素ニューロンを用いる. ただし, 閾値は恒等的に 0 とする. 活性化関数 f_C は, 線形関数とする:

$$f_C(z) = z, \quad z = x + iy. \quad (16)$$

学習パラメータは, 重み $w_1 = r_1 \cdot \exp[i\theta_1]$ である. 学習は, 通常最急降下法 (式 (8), (9)) を用いた. 学習率は 0.5 とした. 学習パターンを表 1 に示す. 学習誤差 $(1/2)|t - v|^2$ が, 0.0001 以下になったときに収束したとみなして, 学習を終了させた (t は教師信号, v は極変数複素ニューロンの実際の出力値). 上記極変数複素ニューロンの特異点は, $r_1 = 0$ (重み w_1 の振幅がゼロ) である. そこで, 特異点 $r_1 = 0$ の近傍から学習を始める場合を想定して, r_1 の初期値を 0.00001 とした (表 2 のケース 1). また, 特異点 $r_1 = 0$ から離れたところから学習を始めることを想定して, $r_1 = 1.0$ という初期値を用いた (表 2 のケース 2). また, 重み w_1 の位相 θ_1 の初期値は, 表 3 に示す 8 種類とした.

表 4 に, 収束に要した学習回数を示す. 特異点近傍から出発した場合の平均学習回数は, 特異点からある程度離れた位置から出発した場合の平均学習回数の 1.52 倍 ($\approx 83.88/55.13$) だった.

特異点の近傍から学習を始めた場合, ケース 5 においてプラトー現象が見られた (図 1). 図 1 を見る限り, 110 回までの学習では誤差が全く変わらずにプラトーが起きているように見えるが, 実際のデータを見ると状況はそうでないことが分かる. 確かに,

Table 3: Initial values of phase θ_1 of weight w_1 .

ケース	1	2	3	4	5	6	7	8
初期値	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$

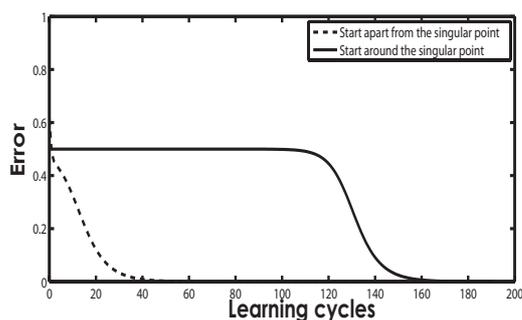


Fig. 1: A Learning curve (Case 5). An example of a plateau phenomenon.

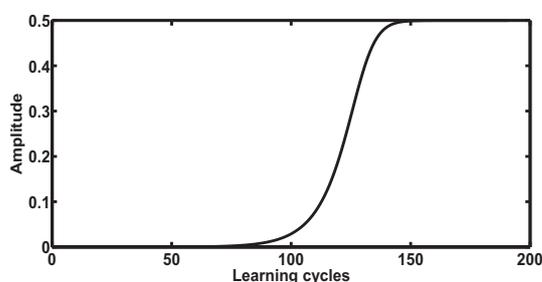


Fig. 2: Transition of the amplitude r_1 of the weight w_1 (Case 5, starting from near the singular point)

1~40回、42~46回、48~49回、54~55回の間はそれぞれ学習誤差が全く変化していない。つまり、これらの間はそれぞれプラトーが起きていると言える。そして、それら以外の学習ではわずかながらも学習誤差は減少している。しかし、大雑把に言えば、110回までは疑似プラトーが起きていると言える。

重み w_1 の振幅 r_1 と位相 θ_1 の推移を図2と図3にそれぞれ示す(ケース5, 特異点の近傍から学習を開始)。振幅 r_1 と位相 θ_1 は、ともに90回目の学習まで(つまり、疑似プラトー現象の間)ほとんど変化していない: $\Delta r_1 \approx 0$ かつ $\Delta \theta_1 \approx 0$ 。

また、図4は、振幅と位相が変化するにつれて、どのように学習誤差が変化したかを示す図である。学習誤差が0.5から0.4になるまでの間は、振幅の変化は位相の変化よりも速く ($\Delta r_1 > \Delta \theta_1$)、学習誤差が0.4から0.17までの間はほぼ等しく ($\Delta r_1 \approx \Delta \theta_1$)、それ以降は、振幅の変化は位相の変化よりも遅くなっている ($\Delta r_1 < \Delta \theta_1$)。つまり、特異点に比較的近い所では、振幅の変化は位相の変化よりも速い。この観察結果は、2.2節において示した数理的な結果: 「特異点付近では $\Delta r_1 > \Delta \theta_1$ となっている」に一致している。

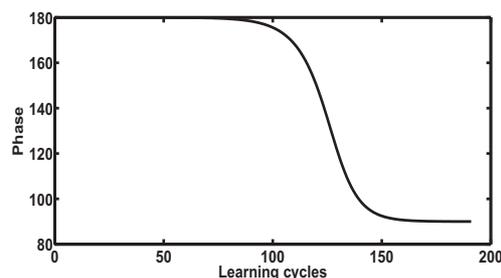


Fig. 3: Transition of the phase θ_1 of the weight w_1 (Case 5, starting from near the singular point)

3 おわりに

本稿では、極変数複素ニューロンのプラトーについて調べ、次の結果を得た。(a) 一見すると全く学習誤差が変わらずにプラトーが起きているように見えるが、厳密には、わずかながらも学習誤差が減少していることがある。ただし、部分的にはプラトーが起きている。(b) 「特異点に比較的近い所では、振幅の変化は位相の変化よりも速い」という数理的な結果が、コンピュータシミュレーションにより確かめられた。

参考文献

- 1) 廣瀬明: 複素ニューラルネットワーク, SGC ライブラリ 38, サイエンス社 (2005).
- 2) T. Nitta (Ed.): Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters, Information Science Reference, Pennsylvania, USA (2009).
- 3) A. Hirose (Ed.), Complex-Valued Neural Networks: Advances and Applications, IEEE Press/Wiley (2013).
- 4) A. Hirose, C. Tabata and D. Ishimaru: Coherent Neural Network Architecture Realizing a Self-Organizing Activeness Mechanism, in Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, N. Baba, L. C. Jain, and R. J. Howlett (Eds.), 576/580, IOS Press, Tokyo (2001).
- 5) S. Kawata and A. Hirose: A Coherent Optical Neural Network that Learns Desirable Phase Values in Frequency Domain by Using Multiple Optical-Path Differences, Opt. Lett., Vol.28 No.24, 2524/2526 (2003).
- 6) A. Hirose, Y. Asano and T. Hamano: Mode-Utilizing Developmental Learning Based on Coherent Neural Networks, in Int'l Conf. on Neural Inform. Proceedings (ICONIP) 2004 Calcutta (Lecture Notes in Computer Sciences 3316), 116/121, Berlin, Springer (2004).

Table 4: Number of training cycles needed to converge. Case number implies those presented in Table 3 (the initial value of the phase θ_1 of the weight w_1).

ケース	1	2	3	4	5	6	7	8	平均
Start around the singular point ($r_1 = 0.00001$)	191	66	12	66	192	66	12	66	83.88
Start apart from the singular point ($r_1 = 1.0$)	74	61	12	61	74	73	13	73	55.13

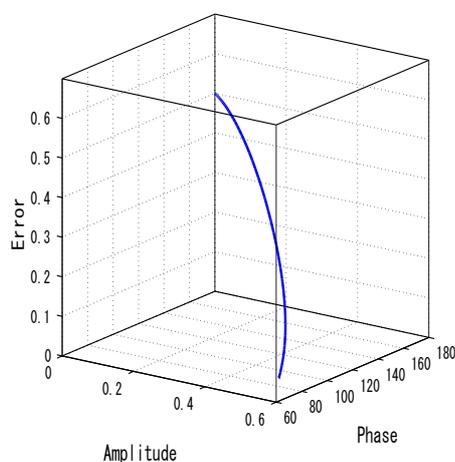


Fig. 4: Transition of the error with respect to the amplitude r_1 and the phase θ_1 of the weight w_1 (Case 5, starting from near the singular point).

- 7) 渡辺澄夫, 福水健次, 萩原克幸, 甘利俊一: 特異モデルの学習理論, 信学論 (D- \square), Vol.J88-D- \square , No.2, 159/169 (2005).
- 8) S. Amari, H. Park, and T. Ozeki: Singularities Affect Dynamics of Learning in Neromanifolds, Neural Computation, Vol.18, No.5, 1007/1065 (2006).
- 9) 渡辺澄夫: 代数幾何と学習理論, 森北出版 (2006).
- 10) T. Nitta.: Local Minima in Hierarchical Structures of Complex-Valued Neural Networks. Neural Networks, Vol.43, 1/7 (2013).
- 11) 新田徹: 複素ニューロンの特異性について, 信学論 (D), Vol.J93-D, No.8, 1614/1621 (2010).
- 12) 新田徹: 極変数複素ニューロンにおける特異点, 第3回コンピューターショナル・インテリジェンス研究会 講演論文集, 97/101 (2013).

パラメータの自動調整が可能なカオス複素多方向連想メモリ

千野拓也 ○長名優子 (東京工科大学)

Chaotic Complex-Valued Multidirectional Associative Memory with Adaptive Parameters

T. Chino and *Y. Osana (Tokyo University of Technology)

Abstract— In this research, we propose a Chaotic Complex-valued Multidirectional Associative Memory (CCMAM) with adaptive scaling factor and investigate its generalization ability for network size. The proposed model is based on the conventional CCMAM with variable scaling factor and can realize one-to-many associations of M -tuple multi-valued patterns. In the proposed model, the scaling factor of refractoriness is determined based on not only the time but also the internal states of neurons. We carried out a series of computer experiments and confirmed that the proposed model can determine the scaling factor of refractoriness automatically in various size networks.

Key Words: One-to-many association, Chaotic complex-valued multidirectional associative memory, Adaptive scaling factor of refractoriness

1 はじめに

ニューロンの入出力や内部状態として複素数を扱うことのできる複素ニューロンモデル¹⁾では、複素数値を多値に割り当てることで多値パターンを扱うことができる^{2)–6)}。また、カオスニューロンモデル⁷⁾から構成される連想メモリでは、記憶したパターンを動的に想起できることが知られている⁷⁾⁸⁾。

複素ニューロンモデル⁶⁾とカオスニューロンモデル⁷⁾を組み合わせたカオス複素ニューロンモデル⁹⁾では、多値パターンの動的な想起を行うことができ、この性質を利用したカオス複素連想メモリ⁹⁾やカオス複素多方向メモリ¹⁰⁾¹¹⁾などが提案されている。また、カオス複素多方向連想メモリ¹²⁾では、多方向連想メモリ¹³⁾に複素ニューロンモデルとカオス複素ニューロンモデルを導入することで3つ組(以上の)データの多値パターンの1対多の連想を実現している。カオス複素多方向連想メモリでは、カオス複素ニューロンのパラメータ、特に不応性のスケーリングファクタによって1対多の想起能力が大きく変わってくることが示されている¹⁴⁾¹⁵⁾。また、時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリでは、不応性のパラメータを時間的に変化させることで、1対多の想起能力を向上させている^{14)–16)}。このモデルでは、不応性のパラメータを時間的に変化させることで1対多の想起能力は向上するが、不応性のパラメータを変化させる適切な範囲はネットワークの層の数や入力される信号の大きさによって変わってくると考えられる。

本報告では、パラメータの自動調整が可能なカオス複素多方向連想メモリを提案する。提案モデルは、時間的に変化する不応性のパラメータを有するカオス複素連想メモリ¹⁴⁾¹⁵⁾に基づいたモデルであり、不応性のパラメータを時間的に変化させるだけでなく、ニューロンの内部状態に応じて自動調整することで1対多の想起能力の向上を実現している。

2 カオス複素ニューロンモデル

ここでは、提案するパラメータの自動調整が可能なカオス複素多方向連想メモリで用いるカオス複素ニューロンモデル⁹⁾について説明する。カオス複素ニューロ

ンモデルは、カオスニューロンモデル⁷⁾を内部状態や出力として複素数値を扱えるように拡張したモデルであり、カオス複素ニューロンから構成されるカオス複素連想メモリでは、記憶した複数の多値パターンを動的に想起できることが知られている⁹⁾。

カオス複素ニューロンのダイナミクスは以下のように表される。

$$x(t+1) = f \left(A(t) - \alpha \sum_{d=0}^t k^d x(t-d) - \theta \right) \quad (1)$$

(ただし、 $A(t), x(t), \theta \in \mathbb{C}$ $k, \alpha \in \mathbb{R}$)

ここで、 $x(t)$ は時刻 t におけるニューロンの出力、 $A(t)$ は時刻 t における外部入力、 α は不応性に関するスケーリングファクタ、 k は時間減衰定数($0 \leq k < 1$)、 θ はニューロンの閾値である。また、 $f(\cdot)$ は出力関数であり、以下のように与えられる。

$$f(u) = \frac{\eta u}{\eta - 1.0 + |u|} \quad (\text{ただし、}\eta \in \mathbb{R}) \quad (2)$$

ここで、 η は $\eta > 1$ を満たす定数である。

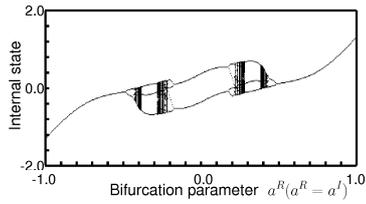
式(1)は、外部入力 $A(t)$ を一定($A(t) = A$)であるとすると以下のように変形することができる。

$$\begin{aligned} x(t+1) &= f(u(t+1)) \\ &= f(ku(t) - \alpha f(u(t)) + (A - \theta)(1 - k)) \\ &= f(ku(t) - \alpha f(u(t)) + a) \end{aligned} \quad (3)$$

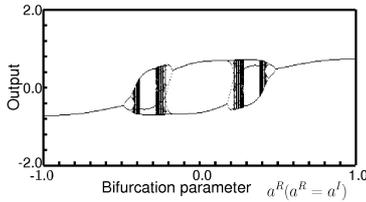
ここで、 $u(t)$ は時刻 t におけるニューロンの内部状態、 $a = (A - \theta)(1 - k)$ は分岐パラメータである。 $\alpha = 1.0$ 、 $k = 0.8$ として分岐パラメータを変化させたときのカオス複素ニューロンの内部状態と出力の実部の分岐図を図1に示す。なお、図1では出力関数のパラメータ η は1.1としている。また、分岐パラメータ a は複素数値であるが、図1では実部と虚部に同じ値($a^R = a^I$)を用いている。

3 パラメータの自動調整が可能なカオス複素多方向連想メモリ

ここでは、提案するパラメータの自動調整が可能なカオス複素多方向連想メモリ (Chaotic Complex-Valued



(a) 内部状態 (実部)



(b) 出力 (実部)

Fig. 1: カオス複素ニューロンモデルの分岐図

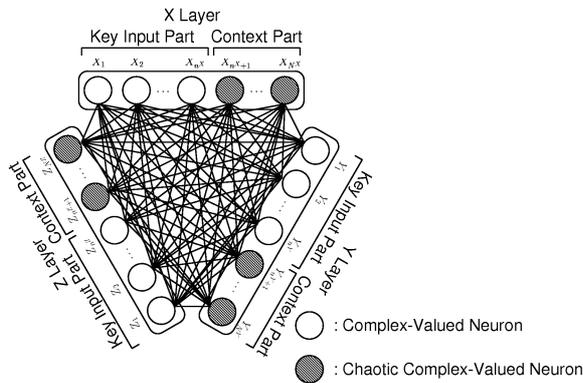


Fig. 2: 提案モデルの構造

Multidirectional Associative Memory (CCMAM with adaptive scaling factor) について説明する。このモデルは時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリ¹⁴⁾¹⁵⁾に基づいたモデルであり、 M 個のデータの組からなるパターンの1対多の連想を実現することができる。

3.1 構造

パラメータの自動調整が可能なカオス複素多方向連想メモリは、3つ以上の層から構成されている。図2の例ではX層、Y層、Z層の3つの層から構成されており、各層は図2に示すようにキー入力部 (Key Input Part) と文脈情報部 (Context Part) の2つに分けられている。キー入力部は複素ニューロンモデル⁶⁾から、文脈情報部はカオス複素ニューロンモデル⁹⁾から構成されている。提案モデルでは、文脈情報部のカオス複素ニューロンの状態がカオスによって変化することで1対多の連想を実現している。

3.2 学習過程

提案モデルでは、直交学習を用いて学習を行う。 M 層から構成される提案モデルにおいて、 x 層から y 層

への結合重みは

$$w^{yx} = X_x(X_y^*X_y)^{-1}X_y^* \quad (4)$$

で与えられる。ここで、 $*$ は共役転置、 -1 は逆行列を表す。また、 X_x と X_y は x 層と y 層に学習させる学習パターン行列であり、

$$X_x = \{X_x^{(1)}, \dots, X_x^{(p)}, \dots, X_x^{(P)}\} \quad (5)$$

$$X_y = \{X_y^{(1)}, \dots, X_y^{(p)}, \dots, X_y^{(P)}\} \quad (6)$$

で与えられる。ここで $X_x^{(p)}$ は x 層に記憶させるパターン p 、 $X_y^{(p)}$ は y 層に記憶させるパターン p 、 P は学習組の数である。

直交学習では1対多の関係にある学習組を記憶することはできない。提案モデルでは、時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリ¹⁴⁾¹⁵⁾ などと同様に1対多の関係にある学習組に文脈情報を付加することで擬似的に1対1の関係に変換し、1対多の関係にある学習組の記憶を実現している。付加する文脈情報パターンはランダムに生成する。

3.3 想起過程

パラメータの自動調整が可能なカオス複素多方向連想メモリでは、文脈情報を付加した学習組を用いて学習を行うが、想起時には文脈情報は未知であると考えるのが自然であるため、キー入力のみからの想起を行う。例えば3層から構成される提案モデルに

$$\{(X_1, Y_1, Z_1), (X_1, Y_2, Z_2), (X_2, Y_3, Z_3)\} \quad (7)$$

のような1対多の関係にある学習組を学習させる場合には

$$\begin{aligned} &\{(X_1-C_{X1}, Y_1-C_{Y1}, Z_1-C_{Z1}), \\ &(X_1-C_{X2}, Y_2-C_{Y2}, Z_2-C_{Z2}), \\ &(X_2-C_{X3}, Y_3-C_{Y3}, Z_3-C_{Z3})\} \end{aligned} \quad (8)$$

のように文脈情報を付加したパターンを用いて学習を行うが、想起を行う際には例えば X_1 がパラメータの自動調整が可能なカオス複素多方向連想メモリに与えられる入力となる。パラメータの自動調整が可能なカオス複素多方向連想メモリにおいて X_1 が入力として与えられると、文脈情報に相当する部分のカオス複素ニューロンの状態がカオスによって遷移することにより、

$$\begin{aligned} (X_1-0, ?, ?) \rightarrow \dots \rightarrow (X_1-C_{X1}, Y_1, Z_1) \rightarrow \dots \\ \rightarrow (X_1-C_{X2}, Y_2, Z_2) \rightarrow \dots \end{aligned} \quad (9)$$

のように1対多の想起が行われることが期待できる。

Step 1 : x 層への入力

x 層のキー入力部にパターンを入力する。

Step 2 : x 層から他の層への伝搬

x 層のキー入力部にパターンが入力されると、他の層へ情報が伝搬される。時刻 t における y 層 ($y \neq x$) のキー入力部のニューロン k の出力 $x_k^y(t)$ は

$$x_k^y(t) = f \left(\sum_{j=1}^{N^x} w_{kj}^{yx} x_j^x(t) \right) \quad (10)$$

で与えられる。ここで、 N^x は x 層のニューロン数、 w_{kj}^{yx} は x 層のニューロン j から y 層のニューロン k への重み、 $x_j^x(t)$ は時刻 t における x 層のニューロン j の出力である。また、 $f(\cdot)$ は

$$f(u) = \frac{\eta u}{\eta - 1.0 + |u|} \quad (\eta \in \mathbb{R}) \quad (11)$$

で与えられる出力関数である。ここで、 η ($\eta > 1$) は定数である。

Step 3 : 他の層から x 層への伝搬

他の層から x 層へ情報が伝搬される。 x 層のキー入力部のニューロン j の出力 $x_j^x(t+1)$ は

$$x_j^x(t+1) = f \left(\sum_{y \neq x}^M \left(\sum_{k=1}^{n^y} w_{jk}^{xy} x_k^y(t) \right) + v A_j^x \right) \quad (12)$$

で与えられる。ここで、 M は層の数、 n^y は y 層のキー入力部のニューロン数、 w_{jk}^{xy} は y 層のニューロン k から x 層のニューロン j への重み、 v は外部入力からの重みを表している。また、 A_j^x は x 層のニューロン j への外部入力であり、

$$A_j^x = \begin{cases} 0, & (t < t_{in}) \\ \hat{x}_j^x(t_{in}), & (t_{in} \leq t) \end{cases} \quad (13)$$

$$t_{in} = \min \left\{ t \left| \sum_{j=1}^{n^x} (\hat{x}_j^x(t) - \hat{x}_j^x(t-1)) = 0 \right. \right\} \quad (14)$$

$$\hat{x}_j^x(t) = \underset{s}{\operatorname{argmin}} (\omega^s - x_j^x(t))^* (\omega^s - x_j^x(t)) \quad (s = 0, 1, \dots, S-1) \quad (15)$$

$$\omega = \exp \left(i \frac{2\pi}{S} \right) \quad (16)$$

で与えられる。ここで、 $\hat{x}_j^x(t)$ は時刻 t における x 層のニューロン j の量子化した出力、 S は状態数、 i は虚数単位である。

時刻 $t+1$ における x 層の文脈情報部のニューロン j の出力 $x_j^x(t+1)$ は

$$x_j^x(t+1) = f \left(\sum_{y \neq x}^M \left(\sum_{k=1}^{n^y} w_{jk}^{xy} \sum_{d=0}^t k_m^d x_k^y(t-d) \right) - \alpha(t, I) \sum_{d=0}^t k_r^d x_j^x(t-d) \right) \quad (17)$$

で与えられる。ここで、 k_m, k_r は時間減衰定数である。また、 $\alpha(t, I)$ は不応性の項を除いた内部状態の平均値 I のときの時刻 t における不応性のスケーリングファクタである。不応性の項を除いた内部状態の平均値 I は

$$I = \frac{1}{N^x - n^x} \sum_{j=n^x+1}^{N^x} \frac{|\operatorname{Re} u_j| + |\operatorname{Im} u_j|}{2} \quad (18)$$

$$u_j = \sum_{y \neq x}^M \left(\sum_{k=1}^{n^y} w_{jk}^{xy} \sum_{d=0}^t k_m^d x_k^y(t-d) \right). \quad (19)$$

Table 1: 高い想起率が得られたパラメータ a と b の組み合わせ

M	S	a	b
3	4	1.0	1.0
	8	1.0	0.5
	16	1.0	0.5
4	4	1.5	1.5
	8	1.5	1.0
	16	1.5	1.0
5	4	2.0	2.0
	8	2.0	1.0
	16	2.0	1.0

で与えられる。式 (17) において、 $\alpha(t, I)$ は

$$\alpha(t, I) = a(I) + b(a(I), S) \sin \left(c \cdot \frac{\pi}{12} \cdot t \right) \quad (20)$$

$$a(I) = 0.75 + 0.009(I - 1.0)^2 \quad (21)$$

$$b(a(I), S) = \begin{cases} a(I), & S = 4, 8 \\ a(I)/2, & S = 16. \end{cases} \quad (22)$$

で与えられる。式 (21), (22) におけるパラメータの決定方法は 3.4 に示す実験に基づき決定したものである。

Step 4 : 繰り返し

Step 2 と 3 を繰り返す。

3.4 内部状態と 1 対多の想起能力の関係

カオス複素ニューロンモデルでは、カオス応答が生成されるかどうかはニューロンの内部状態の値によって決まる。時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリ¹⁴⁾¹⁵⁾や提案モデルでは、各ニューロンの内部状態は他の層のニューロンからの入力とニューロン自身の不応性の影響を合わせたものと考えることができる。

従来時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリでは、時刻 t における不応性のスケーリングファクタ $\alpha(t)$ は

$$\alpha(t) = a + b \cdot \sin \left(c \cdot \frac{\pi}{12} t \right) \quad (23)$$

で与えられ、パラメータ a と b の組み合わせが 1 対多の想起能力に大きな影響を与えることが知られている¹⁵⁾¹⁶⁾。

ここでは、キー入力部のニューロン数が 100、文脈情報部のニューロン数が 50 の 3~5 層の時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリにおいて、 a と b の様々な組み合わせに対してニューロンの内部状態と 1 対多の想起能力の関係を調べる実験を行った。表 1 は各条件において高い想起率が得られた a と b の組み合わせである。また、図 3 は高い想起率が得られた a と不応性を除いた内部状態の関係を表している。提案モデルでは、これらの結果から式 (21), (22) のようにパラメータ a, b を決定するものとした。なお、図 3 において、曲線は提案モデルにおいて a の決定に用いている関数を表している。

4 計算機実験

提案するパラメータの自動調整が可能なカオス複素多方向連想メモリと従来時間的に変化する不応性の

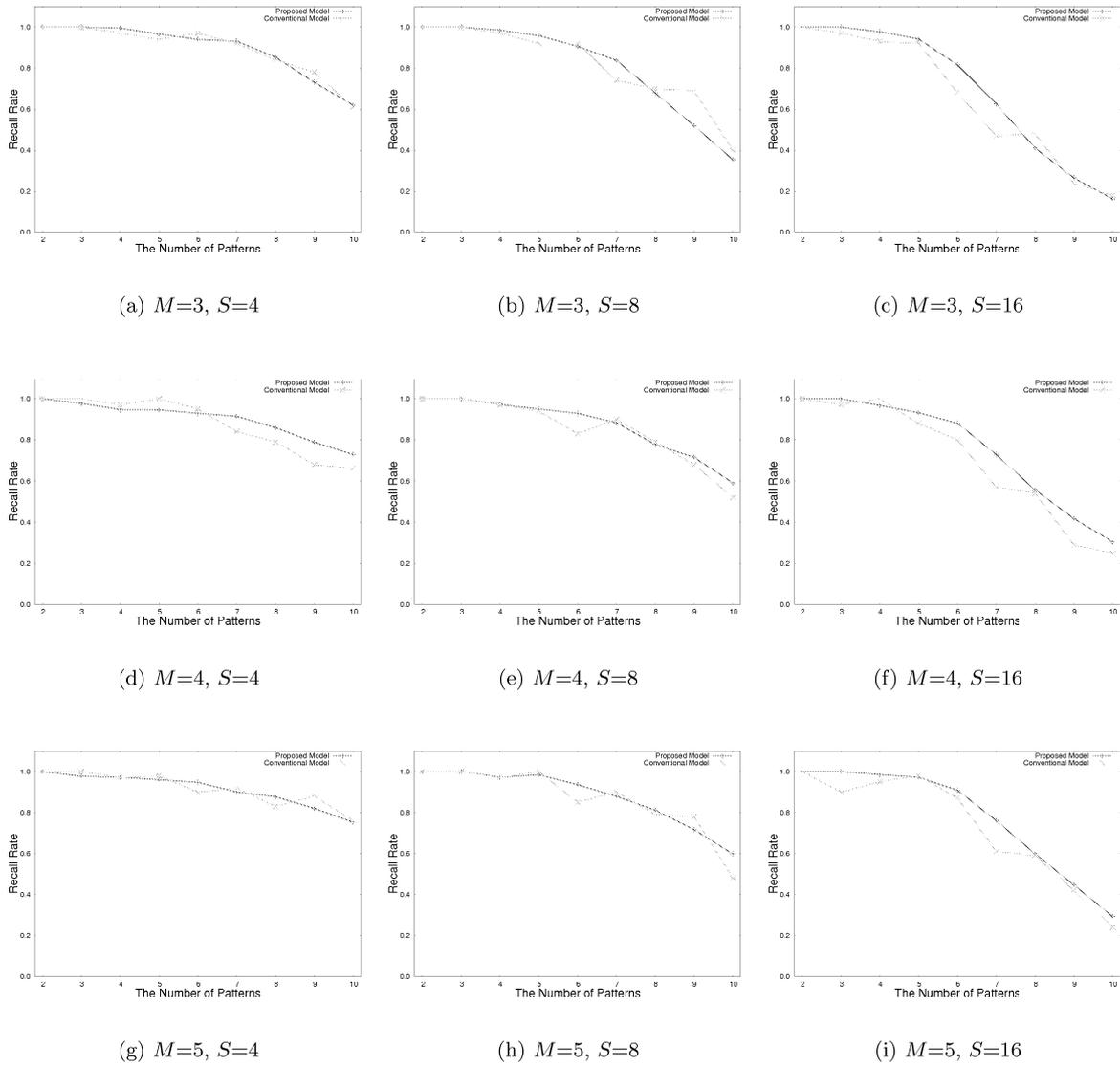


Fig. 4: 1 対多の想起能力の比較 ($N_K = 100$, $N_C = 50$)

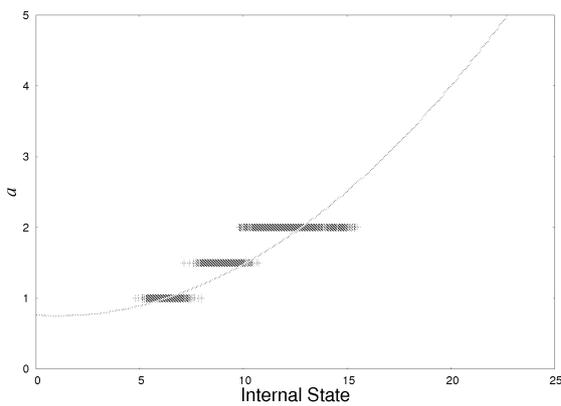


Fig. 3: 高い想起率が得られた内部状態とパラメータ a の組み合わせ

パラメータを有するカオス複素多方向連想メモリの想起能力¹⁴⁾¹⁵⁾の比較を行った結果を図4に示す。なお、従来の時間的に変化する不応性のパラメータを有する

カオス複素多方向連想メモリの想起能力¹⁴⁾¹⁵⁾は1対多の想起能力が最も高くなるようにパラメータを試行錯誤により調整したものを用いている。この実験では、キー入力部のニューロン数 (N_K) を100、文脈情報部のニューロン数 (N_C) を50とし、ランダムに生成した S ($S = 4, 8, 16$) 値のパターンを M ($M = 3 \sim 5$) 層のネットワークに記憶させている。図4より、提案モデルではパラメータの調整を手動で行った従来のモデルと同等の想起能力が得られていることが分かる。

また、図5にネットワークのサイズを様々に変化させた提案モデルにおける1対多の想起能力を示す。提案モデルにおいてパラメータの自動調整を行う方法を決めるためにはキー入力部のニューロン数 (N_K) が100、文脈情報部のニューロン数 (N_C) が50のネットワークを用いているが、図5(a)~(c)ではキー入力部のニューロン数 (N_K) を200、図5(d)~(f)ではキー入力部のニューロン数 (N_K) を300、図5(g)~(i)では文脈情報部のニューロン数 (N_C) を20としている。また、図5(j)~(l)は、6~8層のネットワークにおける結果である。いずれの

条件においても、図4と同程度の想起率が実現されていることが分かる。

5 おわりに

本報告では、パラメータの自動調整が可能なカオス複素多方向連想メモリを提案した。このモデルは時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリ¹⁶⁾に基づくモデルであり、不応性のスケールリングファクタのとり得る範囲を決定するパラメータである a と b を内部状態の値に基づいて自動調整することができる。なお、提案モデルにおいて a , b を決定する方法については、時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリにおいて a と b の組み合わせを変えて想起率を調べ、想起率の高くなる a と b の組み合わせのときの内部状態の値と a , b との関係に基づいて決定した。

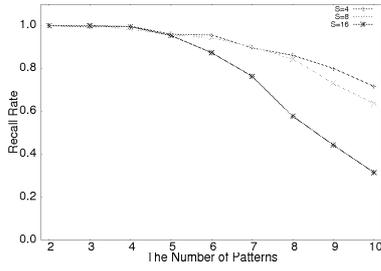
計算機実験を行い、提案するパラメータの自動調整が可能なカオス複素多方向連想メモリにおいて、以下のような性質があることを確認した。

- (1) パラメータの自動調整が可能なカオス複素多方向連想メモリは、時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリと同程度の想起能力を持つ。
- (2) パラメータの自動調整は層の数やニューロン数に関係なく行うことができる。
- (3) 提案モデルも従来モデルと同様に1対 N の N の値が大きくなると想起率が低下するが、従来モデルと比較すると想起率の低下はゆるやかである。

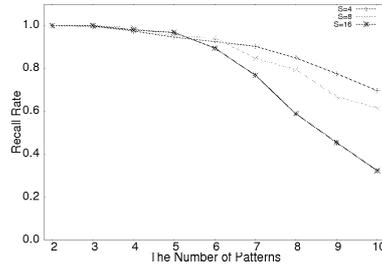
パラメータの自動調整が可能なカオス複素多方向連想メモリと時間的に変化する不応性のパラメータを有するカオス複素多方向連想メモリの想起能力の比較において、1対 N の N の値が大きい場合には提案モデルのほうが想起率が上回る場合が多かったものの、 N の値が小さい場合には従来モデルのほうが想起率が高くなることもあった。 N の値が小さい場合に想起率が低くなってしまうことがあるという問題への対策については今後検討していく必要がある。

参考文献

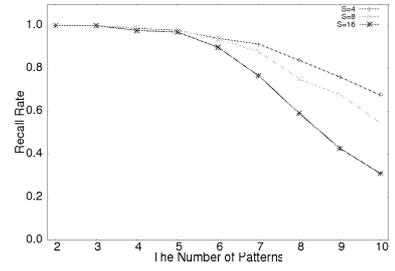
- 1) 根本 幾, 河野 知志: “神経回路網の複素数値化の試み,” 電子情報通信学会論文誌 D-II, Vol.J74, No.9, pp.1282–1288, 1991.
- 2) 青木 宏之, 小林 幸夫: “ペナルティ項を有する複素連想記憶モデルの性質,” 電子情報通信学会論文誌 A, Vol.J81, No.11, pp.1538–1546, 1998.
- 3) 青木 宏之: “複素数体上に拡張したホップフィールド連想記憶の平衡状態の解析,” 電子情報通信学会論文誌 A, Vol.J78, No.9, pp.1238–1241, 1995.
- 4) S. Jankowski, A. Lozowski and J. M. Zurada: “Complex-valued multistate neural associative memory,” IEEE Transactions on Neural Networks, Vol.7, No.6, pp.1491–1496, 1996.
- 5) 黒江 康明: “複素連想記憶のモデルとそのダイナミックス,” 電子情報通信学会学会誌, Vol.87, No.6, pp.454–459, 2004.
- 6) 橋本 直樹, 黒江 康明, 森 武宏: “複素ニューラルネットワークに対するエネルギー関数について,” 電子情報通信学会技術報告, No.NC98-152, pp.121–128, 1999.
- 7) K. Aihara, T. Takabe and M. Toyoda: “Chaotic neural networks,” Physics Letter A, 144, No.6, 7, pp.333–340, 1990.
- 8) Y. Osana and M. Hagiwara: “Separation of superimposed pattern and many-to-many associations by chaotic neural networks,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Anchorage, Vol.1, pp.514–519, 1998.
- 9) M. Nakada and Y. Osana: “Chaotic complex-valued associative memory,” Proceedings of International Symposium on Nonlinear Theory and its Applications, Vancouver, 2007.
- 10) Y. Yano and Y. Osana: “Chaotic complex-valued bidirectional associative memory,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Atlanta, 2009.
- 11) Y. Yano and Y. Osana: “Chaotic complex-valued bidirectional associative memory – one-to-many association ability –,” Proceedings of International Symposium on Nonlinear Theory and its Applications, Sapporo, 2009.
- 12) Y. Shimizu and Y. Osana: “Chaotic complex-valued multidirectional associative memory,” Proceedings of IASTED Artificial Intelligence and Applications, Innsbruck, 2010.
- 13) M. Hagiwara: “Multidirectional associative memory,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Washington D.C., Vol.1, pp.3–6, 1990.
- 14) A. Yoshida and Y. Osana: “Chaotic complex-valued multidirectional associative memory with variable scaling factor,” Proceedings of International Conference on Artificial Neural Networks, Espoo, 2011.
- 15) A. Yoshida and Y. Osana: “Chaotic complex-valued multidirectional associative memory with variable scaling factor – One-to-many association ability –,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, 2011.
- 16) M. Onagi and Y. Osana: “Pattern dependency of one-to-many association ability in chaotic complex-valued multidirectional associative memory with variable scaling factor,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Brisbane, 2012.



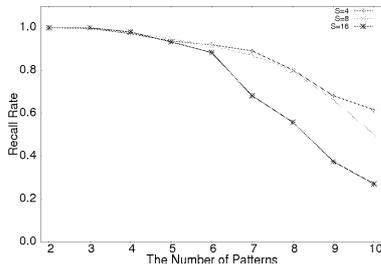
(a) $M=3, N_K=200, N_C=50$



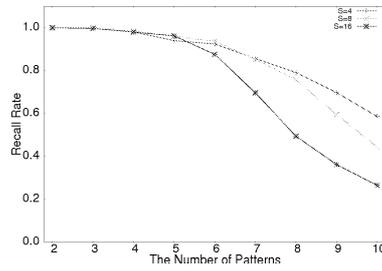
(b) $M=4, N_K=200, N_C=50$



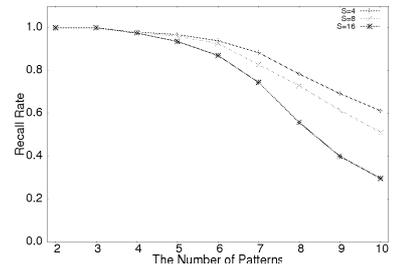
(c) $M=5, N_K=200, N_C=50$



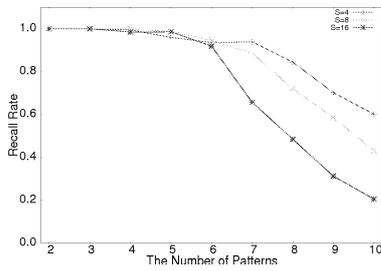
(d) $M=3, N_K=300, N_C=50$



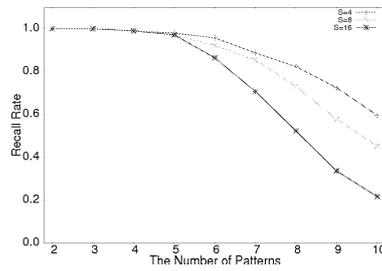
(e) $M=4, N_K=300, N_C=50$



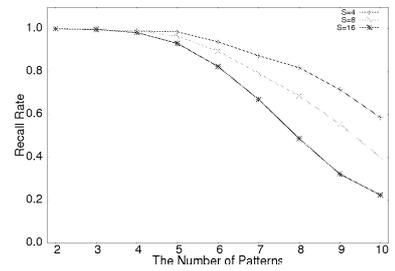
(f) $M=5, N_K=300, N_C=50$



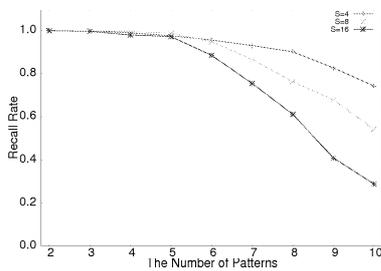
(g) $M=3, N_K=100, N_C=20$



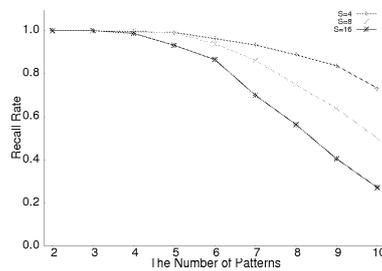
(h) $M=4, N_K=100, N_C=20$



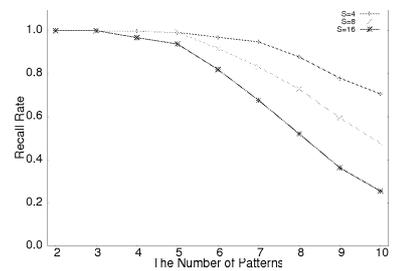
(i) $M=5, N_K=100, N_C=20$



(j) $M=6, N_K=100, N_C=50$



(k) $M=7, N_K=100, N_C=50$



(l) $M=8, N_K=100, N_C=50$

Fig. 5: 様々な条件のネットワークにおける1対多の想起能力

四元数ニューラルネットワークの歩行者数推定への適用

○村本憲幸 磯川悌次郎 西村治彦 松井伸之 (兵庫県立大学)

An Application of Quaternionic Multilayer Perceptron to Counting Pedestrians

*N. Muramoto, T. Isokawa, H. Nishimura and N. Matsui (University of Hyogo)

Abstract— This paper presents and explores a system for detecting and counting pedestrians from a sequence of video images. This system has capabilities of simultaneously detecting and tracking several groups of pedestrians. Groups are extracted by using the background subtraction, and a multilayer perceptron adopting quaternionic algebra estimates the number of pedestrians in each group. The performance of the proposed system is demonstrated through the experiments using sequences of images with a real scenery.

Key Words: Quaternion, Neural Network, Counting Pedestrians

1 はじめに

動画像から人の動きや流れを調べることはマーケティング調査やセキュリティシステム分野において有用である。現在、このような作業の多くは手作業によって行われているため、作業効率や人件費などの面から高コストな作業といえる。それゆえ、動画像から自動的に人の動きや流れを解析できる手法の開発が強く望まれており、これまでに様々な研究が行われてきた¹⁾。

単一画像から人を抽出する手法としては、テンプレートマッチング^{2,3)}やエッジ処理⁴⁾に基づいた方法が提案されている。しかしながら、これらの方法では、混雑時のような人が重なり合い体の一部が隠れてしまう場合には、正しく人を抽出することができない。この問題を解決するために、ステレオカメラやサーモカメラのような特殊なカメラを用いた識別法^{5,6,7,8)}や圧力センサを用いた歩行者ごとの検出⁹⁾、上空から鉛直下向きに設置したカメラによる人検出¹⁰⁾などが提案されている。しかしながら、これらの手法は装置の導入コストが高く、既に設置してある監視カメラの映像には適用することができない、あるいは屋外環境に柔軟に対応できないといった問題を抱えている。このように動画像中から自動的に歩行者を検出、計数する場合のように各歩行者を分別するかが重要であり、高精度に検出を行うためには費用面、計算時間、システムの設置方法のいずれかを犠牲にしなければならないのが現状である。

このような背景から、文献¹¹⁾において一般的に市販されているビデオカメラから歩行者数を推定する方法が提案されている。この方法では動画像から人物領域を抽出し、その領域に含まれる歩行者数をグループ単位で推定する。推定には歩行者グループごとの形状や色の特徴量を入力とするニューラルネットワークを用いている。

文献¹¹⁾では一般的な実数型のニューラルネットワークが用いられている。しかし、扱うデータが色や座標のような多次元データの場合においては文献¹²⁾で示されているように複素数に基づくニューラルネットワークの方がより効果的なデータ処理が可能であると期待される。

本論文では、四元数ニューラルネットワークを歩行者数推定へ適用する。使用するネットワークは局所解析的な活性化関数を有した四元数ニューラルネットワー

クであり、学習係数や重みなどのパラメータはすべて四元数によって記述されている。提案法の有用性は実際に撮影された動画像を用いて評価する。

2 人数推定システム

Fig. 1に今回提案するシステムにおける処理の流れを示す。まず、ビデオカメラによって得られた入力画像とシステム内にある背景画像との間で差分を取り、変化が大きい部分を人物が存在する領域として抽出する。次に、それぞれの歩行者グループに対して色や形状に関する特徴量を計算する。最後にこれらの特徴量をネットワークに入力し、その出力から抽出された各人物領域に含まれる人数を推定する。

2.1 人物領域の抽出

ビデオカメラによって得られた入力画像と背景画像の差分を計算し、あるしきい値で2値化を行う。このとき、差分値がしきい値を超えた画素を人物領域とみなす。Fig. 2に背景差分処理の例を示す。同図(a)は背景画像、(b)は入力画像であり、(c)が背景差分によって得られた2値画像である。なお、背景画像は環境の変化に対応するために一定の周期(本実験では30フレーム毎)で更新する。

続いて、得られた2値画像から歩行者グループを抽出するために、ラベリング処理によって同一連結成分を持つ領域毎に分けていく。本システムでは4連結でのラベリングを行い、連結数が少ない領域については人物領域でないと判断し、以降の処理は行わない。Fig. 2(c)に示すように、歩行者が連なって歩いている場合、複数の歩行者が一つのグループを形成する。そのため、歩行者グループ単位での特徴量算出および人数推定を行う。また、各歩行者グループに対してFig. 2(d)に示すように各グループに外接する矩形を付与する。この矩形は特徴量の算出と歩行者グループの追跡に用いられる。

歩行者グループの追跡はフレーム画像間での矩形の対応関係を求めることによって行われる。ここでは、時刻 t の矩形が時刻 $t-1$ の矩形の中心を含んでいる場合、二つの矩形を同一のものとして対応付けている。

2.2 特徴量の取得

抽出した各歩行者グループごとに以下に示す特徴量を計算する。

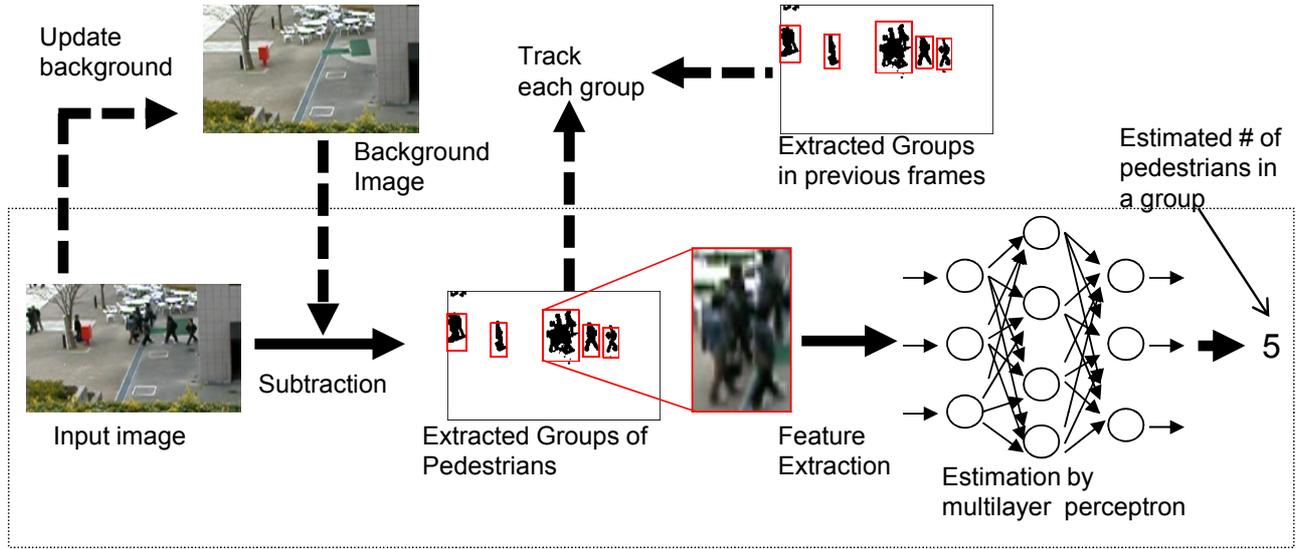


Fig. 1: Information flow in the proposed method



Fig. 2: Images used and processed in the proposed method. (a) The background image, (b) Input image, (c) The obtained image through the subtraction and binarization processes, and (d) The labeled groups of pedestrians with rectangles attached

1. 人物領域の重心位置と矩形の中心位置との差を表す2次元特徴量.
2. 人物領域を構成する画素のRGB成分各々の分散値を表す3次元特徴量.
3. 人物領域を構成する画素集合の円形度 $r = 4\pi S/l^2$. ここで、 l および S はそれぞれ画素集合の周囲長と面積を表し、円形度 r は対象の形状が円に近いほど1に近い値となる.
4. 矩形の面積に対する人物領域の割合.

以上の4つの特徴量、計7次元のデータをNNに入力し、各歩行者グループごとの含有人数を推定する.

2.3 人数の推定

歩行者グループに含まれる人数の推定には3層のニューラルネットワークによって行われる. 本システムで用いるニューラルネットワークの詳細については次章で説明する. このネットワークは前述した4つの特徴量を入力とし、各特徴量に対して一つのニューロンを割り当てる. そのため、入力層には4つのニューロンが必要になる. 出力層のニューロン数はネットワークが推定できる最大の人数を表しており、例えば推定結果が k 人であった場合、出力層の k 番目のニューロンが最も大きな値を出力する.

ネットワークの学習は、あらかじめ用意した歩行者グループから得られる特徴量を学習用データ、そのグループに含まれる人物の数を教師信号として、誤差逆

伝搬法によって行う. その際、教師信号は目測によって測定するものとし、差分処理によって抽出した領域に人物の半分程度が含まれていれば1名とみなす.

3 四元数ニューラルネットワーク

3.1 四元数の定義と演算規則

四元数は複素数を拡張した数体系であり、1つの実部と3つの虚部を持つ. 3つの虚数単位をそれぞれ i , j , k とすると四元数 \mathbf{x} は、

$$\mathbf{x} = x^{(e)} + x^{(i)}i + x^{(j)}j + x^{(k)}k \quad (1)$$

と表される. ここで、 $x^{(e)}$, $x^{(i)}$, $x^{(j)}$, $x^{(k)}$ は実数であり、四元数 \mathbf{x} の各成分を表す. したがって、四元数全体の集合 \mathbf{H} は1, i , j , k を基底とする4次元のベクトル空間を構成する. また、四元数はスカラー部分 $x^{(e)}$ とベクトル部分 $\vec{x} = \{x^{(i)}, x^{(j)}, x^{(k)}\}$ に分けることにより、

$$\mathbf{x} = (x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}) = (x^{(e)}, \vec{x}) \quad (2)$$

とも表記される. 四元数 $\mathbf{x} (\mathbf{x} \in \mathbf{H})$ の共役な四元数 $\mathbf{x}^* (\mathbf{x}^* \in \mathbf{H})$ は、

$$\begin{aligned} \mathbf{x}^* &= (x^{(e)}, -\vec{x}) \\ &= x^{(e)} - x^{(i)}i - x^{(j)}j - x^{(k)}k \end{aligned} \quad (3)$$

と定義される.

四元数の虚数単位間には次式の Hamilton 関係が成り立つ。

$$\begin{aligned} i^2 = j^2 = k^2 = ijk = -1, \\ ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j \end{aligned} \quad (4)$$

ここで、 $ij \neq ji$ が示すように、四元数は非可換の数体系である。

以上の定義に基づいて、四元数 $\mathbf{p} = (p^{(e)}, \vec{p})$ と $\mathbf{q} = (q^{(e)}, \vec{q})$ 間の演算は以下のようになる。加減算は、

$$\begin{aligned} \mathbf{p} \pm \mathbf{q} &= (p^{(e)} \pm q^{(e)}, \vec{p} \pm \vec{q}) \\ &= (p^{(e)} \pm q^{(e)}, p^{(i)} \pm q^{(i)}, p^{(j)} \pm q^{(j)}, p^{(k)} \pm q^{(k)}) \end{aligned} \quad (5)$$

となる。四元数 \mathbf{p} と \mathbf{q} の積 \mathbf{pq} は、

$$\mathbf{pq} = (p^{(e)}q^{(e)} - \vec{p} \cdot \vec{q}, p^{(e)}\vec{q} + q^{(e)}\vec{p} + \vec{p} \times \vec{q}) \quad (6)$$

となる。ここで、 $\vec{p} \cdot \vec{q}$ と $\vec{p} \times \vec{q}$ はそれぞれ 3 次元ベクトルの内積と外積を表す。四元数の積の共役と共役な四元数の積の間には次の関係が成立する。

$$(\mathbf{pq})^* = \mathbf{q}^* \mathbf{p}^* \quad (7)$$

四元数 \mathbf{x} のノルム $|\mathbf{x}|$ は、

$$|\mathbf{x}| = \sqrt{\mathbf{x}\mathbf{x}^*} = \sqrt{x^{(e)2} + x^{(i)2} + x^{(j)2} + x^{(k)2}} \quad (8)$$

と定義される。

スカラー $a = (a, \vec{0})$ と四元数 \mathbf{x} の間の積は次のようになる。

$$\begin{aligned} a\mathbf{x} &= (ax^{(e)}, a\vec{x}) \\ &= (ax^{(e)}, ax^{(i)}, ax^{(j)}, ax^{(k)}) \end{aligned} \quad (9)$$

3.2 四元数関数における局所解析性

四元数空間における関数の解析性は Cauchy-Riemann-Fueter (CRF) の方程式によって記述されるが、これを満たすものは線形関数と定数関数のみであることが知られており、これに代わる別の方法が模索されている。その中の 1 つが局所解析性である。局所解析性とは、CRF の大域的な解析性とは異なり、四元数空間のある点における局所的な解析性を示したものである。

四元数 \mathbf{x} は次のような形式で表現することができる。

$$\mathbf{x} = x^{(e)} + \mathbf{u}_x r \quad (10)$$

$$r = \sqrt{x^{(i)2} + x^{(j)2} + x^{(k)2}} \quad (11)$$

$$\mathbf{u}_x = \frac{x^{(i)}\mathbf{i} + x^{(j)}\mathbf{j} + x^{(k)}\mathbf{k}}{r} \quad (12)$$

式 (12) の定義より $\mathbf{u}_x^2 = -1$ となることがわかる。このことから \mathbf{u}_x を虚数単位とみなすことができる。

ここで、以下のように分解できる \mathbf{x} に関する微小量 $d\mathbf{x} = (dx^{(e)}, dx^{(i)}, dx^{(j)}, dx^{(k)})$ を導入する。

$$d\mathbf{x} = d\mathbf{x}_{\parallel} + d\mathbf{x}_{\perp} \quad (13)$$

ここで、

$$d\mathbf{x}_{\parallel} = \frac{1}{2} (d\mathbf{x} - \mathbf{u}_x d\mathbf{x} \mathbf{u}_x) \quad (14)$$

$$d\mathbf{x}_{\perp} = \frac{1}{2} (d\mathbf{x} + \mathbf{u}_x d\mathbf{x} \mathbf{u}_x) \quad (15)$$

である。この式より以下の関係が得られる。

$$d\mathbf{x}_{\parallel} \mathbf{x} = \mathbf{x} d\mathbf{x}_{\parallel}, \quad d\mathbf{x}_{\perp} \mathbf{x} = \mathbf{x}^* d\mathbf{x}_{\perp} \quad (16)$$

この関係より、 $d\mathbf{x}_{\parallel}$ と \mathbf{x} は可換であり、上記で定義した局所平面においては、 $d\mathbf{x}_{\parallel}$ と \mathbf{x} を複素数として取り扱うことができる。

\mathbf{x} ならびに $d\mathbf{x}$ を用いることにより、四元数関数 $\mathbf{F}(\mathbf{x} + d\mathbf{x})$ は次のように展開することが可能である。

$$\mathbf{F}(\mathbf{x} + d\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \mathbf{F}'(\mathbf{x}) + O(d\mathbf{x}^2) \quad (17)$$

$$\mathbf{F}'(\mathbf{x}) = \mathbf{F}'(\mathbf{x}) d\mathbf{x}_{\parallel} + \frac{\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x}^*)}{(\mathbf{x} - \mathbf{x}^*)} d\mathbf{x}_{\perp} \quad (18)$$

ここで、 $d\mathbf{x}_{\perp} = 0$ すなわち $d\mathbf{x} + \mathbf{u}_x d\mathbf{x} \mathbf{u}_x = 0$ とすると、 $\mathbf{u}_x d\mathbf{x} = d\mathbf{x} \mathbf{u}_x$ という関係が得られる。 \mathbf{u}_x は実部が 0 である四元数であるので、 $\mathbf{u}_x \times d\vec{x} = 0$ であることがわかる。これにより、 \mathbf{u}_x と $d\vec{x}$ は互いに平行であり、 $d\vec{x} = \delta \mathbf{u}_x$ と書くことができる。式 (11) より、次のような関係が得られる。

$$\begin{aligned} r d\mathbf{x} &= x^{(i)} dx^{(i)} + x^{(j)} dx^{(j)} + x^{(k)} dx^{(k)} \\ &= -\vec{x} \cdot d\vec{x} \\ &= r \delta. \end{aligned}$$

これにより、 $d\mathbf{x}$ は $d\mathbf{x} = d\mathbf{x}_{\parallel} + d\mathbf{x}^{(e)}$ と表すことができる。以下に示す微分演算子

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_{\parallel}} &= \frac{1}{2} \left(\frac{\partial}{\partial x^{(e)}} - \mathbf{u}_x \frac{\partial}{\partial x^{(r)}} \right), \\ \frac{\partial}{\partial \mathbf{x}_{\perp}^*} &= \frac{1}{2} \left(\frac{\partial}{\partial x^{(e)}} + \mathbf{u}_x \frac{\partial}{\partial x^{(r)}} \right) \end{aligned}$$

ただし、

$$\frac{\partial}{\partial x^{(r)}} \equiv \frac{x^{(i)}}{r} \frac{\partial}{\partial x^{(i)}} + \frac{x^{(j)}}{r} \frac{\partial}{\partial x^{(j)}} + \frac{x^{(k)}}{r} \frac{\partial}{\partial x^{(k)}}$$

および

$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}_{\parallel}} = \frac{\partial \mathbf{x}^*}{\partial \mathbf{x}_{\parallel}^*} = 1, \quad \frac{\partial \mathbf{x}^*}{\partial \mathbf{x}_{\parallel}} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}_{\parallel}^*} = 0$$

という関係を用いることにより、局所微分 $\mathbf{F}'(\mathbf{x})$ は次式にて表される。

$$\mathbf{F}'(\mathbf{x}) = \frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}_{\parallel}}$$

また、関数 $\mathbf{F}(\mathbf{x})$ における局所解析条件は対応する局所複素平面内において、

$$\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}_{\perp}^*} = 0$$

あるいは,

$$\frac{\partial \mathbf{F}}{\partial x^{(e)}} + \mathbf{u}_x \frac{\partial \mathbf{F}}{\partial x^{(r)}} = 0$$

と表される. この結果は, 常に $d\mathbf{x}_\perp = 0$ としている場合における¹³⁾の結果と一致する.

さらに, \mathbf{F} が \mathbf{x} と \mathbf{x}^* の関数である場合 (\mathbf{x} と \mathbf{x}^* は互いに独立である), その展開は次式にて表される.

$$\begin{aligned} & \mathbf{F}(\mathbf{x} + d\mathbf{x}, \mathbf{x}^* + d\mathbf{x}^*) \\ &= \mathbf{F}(\mathbf{x}, \mathbf{x}^*) + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_\parallel} d\mathbf{x}_\parallel + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_\parallel^*} d\mathbf{x}_\parallel^* \\ &+ \frac{1}{2} \left(\frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}_\parallel^2} d\mathbf{x}_\parallel^2 + \frac{\partial}{\partial \mathbf{x}_\parallel^*} \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}_\parallel} \right) d\mathbf{x}_\parallel d\mathbf{x}_\parallel^* \right. \\ &+ \left. \frac{\partial}{\partial \mathbf{x}_\parallel} \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}_\parallel^*} \right) d\mathbf{x}_\parallel^* d\mathbf{x}_\parallel + \frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}_\parallel^{*2}} d\mathbf{x}_\parallel^{*2} \right) + O(d\mathbf{x}_\parallel^3) \end{aligned} \quad (19)$$

以上の結果から, 局所平面においては複素関数を四元数関数として用いることが可能である.

3.3 四元数ニューラルネットワーク

本研究で用いるネットワークは入力層, 中間層, 出力層の3層から構成され, 各ニューロンの入出力や結合荷重, 学習係数などのパラメータはすべて四元数により表現される.

入力層, 中間層, 出力層におけるそれぞれのニューロン数を M, N, K とする. 入力層ニューロンへの入力 \mathbf{z} はそのまま中間層に向けて出力される. 入力層の m 番目のニューロンと中間層の n 番目のニューロンの間の結合荷重を \mathbf{v}_{nm} とすると, 中間層の n 番目のニューロン出力 \mathbf{x}_n は次式で表される.

$$\mathbf{x}_n = \mathbf{g} \left(\sum_{m=1}^M \mathbf{v}_{nm} \mathbf{z}_m \right) \quad (20)$$

ここで, \mathbf{g} は四元数関数で表される活性化関数である. 同様に, 出力層の k 番目のニューロン出力 \mathbf{y}_k は, 結合荷重 \mathbf{w} , 活性化関数 \mathbf{h} を用いて次式で表される.

$$\mathbf{y}_k = \mathbf{h} \left(\sum_{n=1}^N \mathbf{w}_{kn} \mathbf{x}_n \right) \quad (21)$$

このネットワークを実際に用いるには問題に応じた学習を行う必要がある. ここではニューラルネットワークの学習アルゴリズムとして広く用いられる誤差逆伝搬法に基づく学習アルゴリズム¹⁴⁾を採用する.

ネットワークに信号 \mathbf{z} が入力された場合における出力層 k 番目のニューロンの教師信号を \mathbf{d}_k とする. この教師信号およびニューロンの出力 \mathbf{y}_k から, t 回目の学習における誤差 $E(t)$ を次式で定義する.

$$E(t) = \sum_{k=1}^K (\mathbf{d}_k - \mathbf{y}_k(t)) (\mathbf{d}_k - \mathbf{y}_k(t))^* \quad (22)$$

中間層-出力層間における結合荷重 \mathbf{w} が学習により次式のように修正されたとする.

$$\mathbf{w}_{kn}(t+1) = \mathbf{w}_{kn}(t) + \Delta \mathbf{w}_{kn} \quad (23)$$

ここで, $\Delta \mathbf{w}_{kn}$ が結合荷重の修正量とする. この場合において, $(t+1)$ 回目の学習における誤差は次のように計算される.

$$\begin{aligned} E(t+1) &= E(\mathbf{w}_{kn}(t), \mathbf{w}_{kn}^*(t)) \\ &+ \sum_{k,n} \frac{\partial E}{\partial \mathbf{w}_{kn\parallel}} \Delta \mathbf{w}_{kn} + \sum_{k,n} \frac{\partial E}{\partial \mathbf{w}_{kn\parallel}^*} \Delta \mathbf{w}_{kn}^* \end{aligned} \quad (24)$$

ここで, 修正量 $\Delta \mathbf{w}_{kn}$ を

$$\Delta \mathbf{w}_{kn} = -\boldsymbol{\mu} \frac{\partial E}{\partial \mathbf{w}_{kn\parallel}^*} \quad (25)$$

とおく. ただし, $\boldsymbol{\mu}$ は四元数の定数である. これにより, 誤差の変化量 $\Delta E = E(t+1) - E(t)$ は次式のように計算される.

$$\Delta E = -2\text{Re}(\boldsymbol{\mu}) \sum_{k,n} \left| \frac{\partial E}{\partial \mathbf{w}_{kn\parallel}^*} \right|^2 \quad (26)$$

式 (25) における修正量を計算するために $\partial E / \partial \mathbf{w}_{kn\parallel}^*$ を展開し, 局所解析条件 $\partial \mathbf{y} / \partial \mathbf{w}_{kn\parallel}^* = 0$ を適用すると次式が得られる.

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{w}_{kn\parallel}^*} &= -(\mathbf{d}_k - \mathbf{y}_k) \mathbf{h}' \left(\sum_{n=1}^N \mathbf{w}_{kn}^* \mathbf{x}_n \right) \mathbf{x}_n^* \\ &= \boldsymbol{\delta}_k \mathbf{x}_n^* \end{aligned} \quad (27)$$

ここで, \mathbf{h}' は活性化関数 \mathbf{h} の一次微分を表す. また, $\boldsymbol{\delta}_k$ は $-(\mathbf{d}_k - \mathbf{y}_k) \mathbf{h}'(\sum_{n=1}^N \mathbf{w}_{kn}^* \mathbf{x}_n)$ である. 以上により, 中間層と出力層の間の結合荷重を誤差 E により修正することが可能である.

同様に, 入力層と中間層の間の結合荷重 \mathbf{v} の修正量を定義する $(t+1)$ 回目の誤差 $E(t+1)$ は次式で表される.

$$\begin{aligned} E(t+1) &= E(\mathbf{v}_{nm}(t), \mathbf{v}_{nm}^*(t)) \\ &+ \sum_{n,m} \frac{\partial E}{\partial \mathbf{v}_{nm\parallel}} \Delta \mathbf{v}_{nm} + \sum_{n,m} \frac{\partial E}{\partial \mathbf{v}_{nm\parallel}^*} \Delta \mathbf{v}_{nm}^* \end{aligned} \quad (28)$$

同様に, $\partial E / \partial \mathbf{v}_{nm\parallel}^*$ を展開し, 局所解析条件 $\partial \mathbf{x} / \partial \mathbf{v}_{nm\parallel}^* = 0$, $\partial \mathbf{y} / \partial \mathbf{x}_\parallel^* = 0$, ならびに $\partial \mathbf{x} / \partial \mathbf{v}_\parallel^* = 0$ を適用する. 最終的な結果として,

$$\frac{\partial E}{\partial \mathbf{v}_{nm\parallel}^*} = \left(\sum_{k=1}^K \boldsymbol{\delta}_k \mathbf{w}_{kn}^* \right) \mathbf{g}' \left(\sum_{m=1}^M \mathbf{v}_{nm}^* \mathbf{z}_m \right) \mathbf{z}_m^* \quad (29)$$

が得られる. これによって, 結合荷重の修正が可能となる.

本研究では, 四元数関数の \tanh を活性化関数 $\mathbf{g}(\cdot)$ および $\mathbf{h}(\cdot)$ に適用する. 複素関数の \tanh を複素 NN に適用する場合と同様に, 四元数 NN においても \tanh 関数に内包する特異点を考慮してニューロンの入出力値を決定する必要がある. また, 局所解析条件についても考慮する必要があるが, これは学習係数 $\boldsymbol{\mu}$ を適切に設定することに対応する.

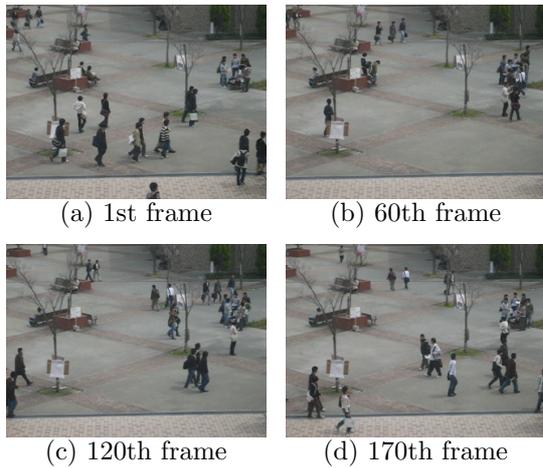


Fig. 3: Examples of image frames used for the experiment

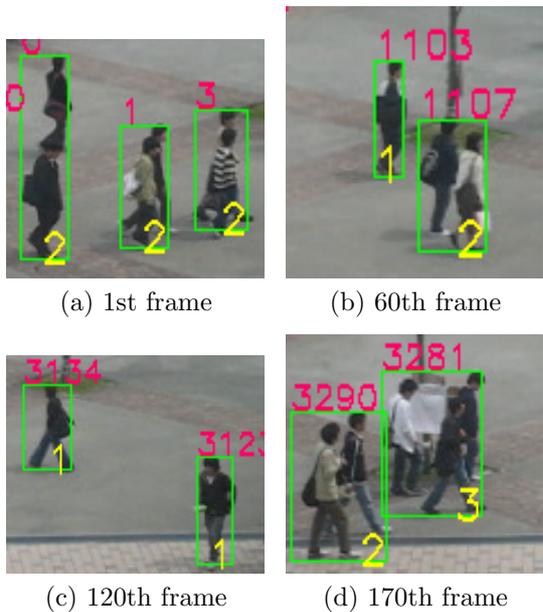


Fig. 4: Examples of extracted and estimated results

4 実動画像を用いた評価

提案システムの有用性を示すために、ビデオカメラによって撮影された動画を用いて評価を行う。Fig. 3に実験に用いる動画のフレーム画像を示す。この動画は49500枚のフレーム画像で構成され、フレームレートは15fpsである。また、フレーム画像の大きさは76800(=320×240)画素であり、各画素は24ビットで表されている。人数推定を行うために必要なネットワークの学習にはこの動画の一部を用いる。今回の実験では動画の開始から100フレームまでのフレーム画像を学習に用いた。ここで、本手法を用いて抽出した歩行者グループの特徴量を入力データ、そのグループに含まれる人の数を手作業によって計数したものの教師データとしている。

Fig. 4に本手法によって抽出された歩行者領域とその領域に含まれる人数を推定した結果を示す。図の緑色の矩形は歩行者領域を示し、領域内右下の黄色の数字は推定された人数を示す。また、領域上部にある赤色の数字は各グループを識別するためのものである。これらの図より歩行者が重なっている場合や多くの歩行

者がいる場合でも正しく人数推定できていることがわかる。

最後に提案手法をフレーム画像全体に適用した結果をFig. 5に示す。図より提案手法によって歩行者を追跡ならびに推定が行われていることがわかる。

5 まとめ

本論文では、四元数ニューラルネットワークを用いた歩行者推定法について提案した。提案手法では、背景差分によって歩行者グループを抽出し、グループ毎の特徴量を求め、それらを用いて四元数ニューラルネットワークで歩行者数推定を行った。実際に撮影された動画画像を用いた実験では、提案手法によって人数計数が可能であることが確認できた。

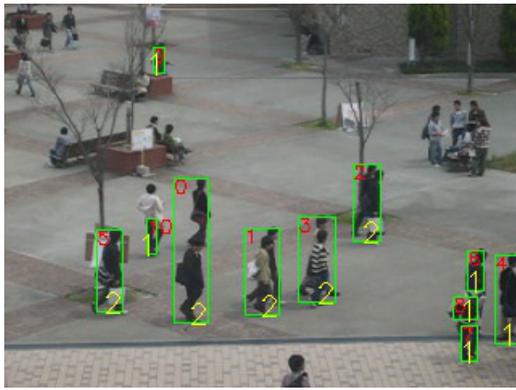
今後の課題として、歩行者グループの抽出処理や追跡処理の改善を行うことによる精度向上、ならびに同一条件下における他手法との性能比較などが挙げられる。

謝辞

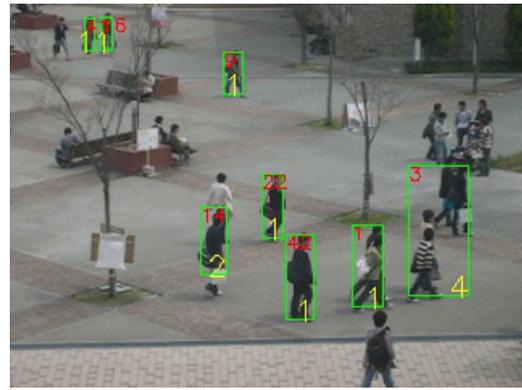
本研究の一部は日本学術振興会の科学研究補助金(若手研究(B)24700227および基盤研究(C)23500286)の支援により行われた。

参考文献

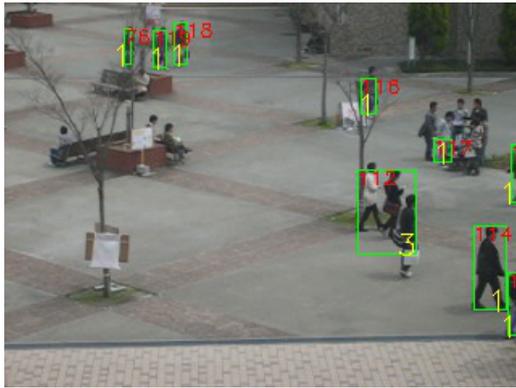
- 1) 小沢慎治: サーベイランスの現状と今後, 映像情報メディア学会誌, Vol. 57, No. 9, 1056/1057, (2003)
- 2) O. Masoud and N. P. Papanikolopoulos: Robust pedestrian tracking using a model-based approach, *IEEE Conf. Intelligent Transportation Systems*, 338/343, (1997)
- 3) C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen: Walking Pedestrian Recognition, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 3, 155/163, (2000)
- 4) 倉爪亮, 由井俊太郎, 辻徳生, 岩下友美, 原健二, 長谷川勉: Fast level set method の提案とビデオ画像の移動物体のリアルタイム追跡, 情報処理学会論文誌, Vol. 44, No. 8, 2244/2254, (2003)
- 5) 庄司武, 池谷崇, 麻生川佳誠: ステレオビジョンを用いた人間検知システムの開発, *OMRON TECHNICS*, Vol. 45, No. 1, 11/15, (2005)
- 6) D. R. P. Gibson, B. Ling, M. Zeifman, S. Dong, and U. Venkataraman: Multipedestrian Tracking, *Public Roads*, Vol. 69, No. 5, (2006)
- 7) L. Zhao and C. Thorpe: Stereo and Neural Network-based Pedestrian Detection, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 3, 148/154, (2000)
- 8) A. Leykin and R. Hammoud: Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos, *2006 Conf. on Computer Vision and Pattern Recognition Workshop*, 136, (2006)



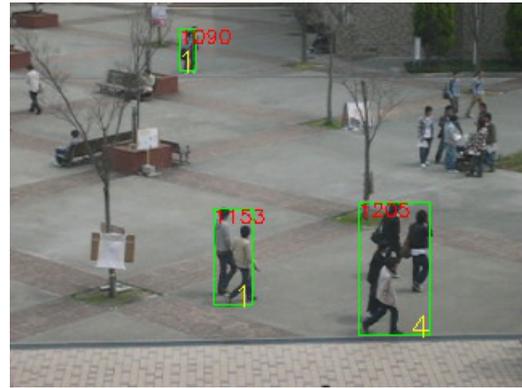
(a) frame #0000



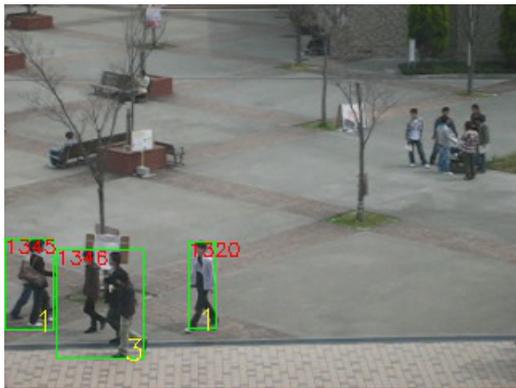
(b) frame #0014



(c) frame #0041



(d) frame #0661



(e) frame #0789



(f) frame #2154

Fig. 5: The output image frames by the presented method

- 9) 池田徹志, 石黒浩, 西村拓一: 床センサと加速度センサの統合による複数人間追跡, 電子情報通信学会技術研究報告, PRMU, Vol. 106, No. 301, 7/12, (2006)
- 10) 寺田賢治, 田所英昭, 山口順一: 2本のななめ計測線を用いた通過人数計測, 電気学会論文誌 C, Vol. 120-C, No. 12, 2105/2106, (2000)
- 11) N. Ikeda, A. Saitoh, T. Isokawa, N. Kamiura, and N. Matsui: A Neural Network Approach for Counting Pedestrians from Video Sequence Images, *Proceedings of SICE Annual Conference*, 2485/2488, (2008)
- 12) N. Muramoto, T. Isokawa, H. Nishimura, and N. Matsui: On Processing Three Dimensional

Data by Quaternionic Neural Networks, *Proceedings of International Joint Conference on Neural Networks*, 3023/3027, (2013)

- 13) S. De Leo and P. P. Rotelli: Quaternionic analyticity, *Applied Mathematics Letters*, Vol. 16, No. 7, 1077/1081, (2003)
- 14) T. Isokawa, H. Nishimura, and N. Matsui: Quaternionic Multilayer Perceptron with Local Analyticity, *Information*, Vol. 3, No. 4, 756/770, (2012)

粒子群最適化における探索の停滞現象の解消法に関する検討

○安田大悟 畠中利治 (大阪大学)

Evaluation on avoidance methods of stagnation in particle swarm optimization

*D. Yasuda and T. Hatanaka (Osaka University)

Abstract— Particle swarm optimization (PSO) has good performance for function optimization problems. However it is well known that it sometimes causes the lack of search ability. Avoiding the phenomenon often called “stagnation” is one of the issues in PSO, in particular for the practical use of PSO. From this point of view, we consider detection indices for stagnation, and then we consider the avoidance methods of stagnation. In this paper, we propose an index based on the controllability of PSO system. The relationship between stagnation and the controllability of the particle is shown by some numerical examples.

Key Words: Particle swarm optimization, Stagnation, Controllability

1 はじめに

最適化問題とは、ある制約のもとで評価関数の値を最小または最大とする設計変数の値を求める問題である。このような問題に対し実用上の最適解を得る方法として、進化計算による手法が多数提案されている。中でも、1995年にJ. Kennedyらによって提案された多点探索による最適化手法であるPSO (Particle Swarm Optimization: 粒子群最適化) は、連続な解空間での単目的最適化において多くの数値実験でその有用性が示されている¹⁾。PSOは、評価関数が非線形で複雑な場合においても高い探索性能を持つことが知られているが、探索粒子が初期に収束することにより、局所解に陥りやすくとされている。このため、初期収束を避け大域解への収束をうながすことが必要である。局所解に収束することなく大域的最適解を得られるようにアルゴリズムのパラメータや実行条件を設定することは、進化計算に共通の課題であり、一般に、探査と探索あるいは、大域探索と局所探索のバランスの取り方の問題とされている。PSOでは、動作モデルの係数や粒子の速度や位置への直接的な操作、粒子が共有する最適解の情報に対する共有範囲の制限などの手法が提案されてきている³⁾。

一方で、PSOでは、粒子数に対して設計変数の次元が大きくなった場合に、実際には最適解ではないところで粒子群が留まって探索が進行しなくなる、stagnation (停滞) と呼ばれる現象が確認されている²⁾。一般に、このクラスの最適化法が適用される問題では、評価関数の形状が不明であり、評価関数の値からstagnationのかどうかを判別することはほぼ不可能である。そのため、探索アルゴリズムとして、stagnationの発生そのものを回避できることが望ましく、また、探索がそのような状態にあるか否かを検知できることが望ましいと考えられるが、これまでに、stagnationの発生のメカニズムに関する理論的研究はあまりなされていない。

本研究では、PSOの探索のstagnationに対して、粒子の移動方向が設計変数空間において、探索できる範囲を十分に持ちえない、すなわち、移動可能な方向が保障されない状況であると考えると、PSOの動作式を離散時間線形動的システムのアナロジーから、状態方程式として表現し、そのシステムの可制御性とstagnationの発生との関連性を調べることを目的に数値実験を行っ

た結果を報告する。

以下、本論文は、2章ではオリジナルのPSOの動作モデルやその改良モデルの一例を紹介し、3章でPSOにおけるstagnationがどのようなものであるかを説明する。また、4章でstagnationの問題解決への新たな切り口として探索粒子の可制御性による手法を提案し、5章では、システムの可制御性とstagnation発生との関連性をベンチマーク問題を用いた数値実験によって示す。最後に、6章で可制御性をstagnation発生の検知のための指標として用いることのできる可能性についてまとめる。

2 Particle Swarm Optimization

2.1 Particle Swarm Optimization

PSOの探索アルゴリズムは自然界における生物の群れの行動をベースに作られており、複数の探索粒子が解空間内を動きまわることによって評価関数の最適解を探索する。各探索粒子の位置は解の値を成分とするベクトルで表される。粒子の移動は、探索世代 (時刻) ごとに粒子の保持する速度ベクトルによって行なわれる。また、各探索粒子は自身の過去の探索履歴の中で、評価値を最適としたときの位置を記憶しており、この位置のことを *pBest* (personal best) という。 *pBest* の中で最適な評価値を持つもの、すなわち粒子群全体の探索履歴の中で最適なものは *gBest* (global best) と呼ばれ、その情報はすべての探索粒子に共有される。 *pBest* および *gBest* の情報は速度ベクトルの更新の際に用いられる。このほか、探索粒子の *gBest* の共有範囲を、あらかじめ定めたトポロジーによって一部に制限するモデルも存在する。このモデルでは、各粒子についての情報共有範囲内での最適なものは *lBest* (local best) と呼ばれる。 *lBest* モデルは粒子の広域な探索を助け、初期収束を抑える働きがあるとされるが、本論文ではこのモデルについては考えないものとする。

PSOの探索粒子は、以下のように各探索粒子の次元ごとに与えられる動作式を元に移動を繰り返す。

$$v_{id}(t+1) = wv_{id}(t) + c_1R_1(p_{id} - x_{id}(t)) + c_2R_2(g_d - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

ここで、添字の i と d はそれぞれ探索粒子の番号と次元を表しており、 $\mathbf{x}_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))^T$ と $\mathbf{v}_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))^T$ はそれぞれ i 番目の探索粒子の位置ベクトルと速度ベクトルである。また、 $\mathbf{p}_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{iD}(t))^T$ は i 番目の探索粒子の $pBest$ を、 $\mathbf{g}(t) = (g_1(t), g_2(t), \dots, g_D(t))^T$ はすべての探索粒子間に共有される $gBest$ のベクトルを表している。なお、 D は、問題の次元、 T は、転置を表す。ここで、 w は慣性定数と呼ばれる正の定数である。また、 c_1, c_2 は正の定数、 R_1, R_2 は $[0, 1]$ の一様乱数である。すべての探索が終了した時点で得られた $gBest$ が最適解として出力される。

2.2 Guaranteed Convergence PSO(GCPSO)

本節においては PSO の改良モデルの一例として、2002 年に van den Bergh によって提案された Guaranteed Convergence PSO (GCPSO) を紹介する⁹⁾¹⁰⁾。このモデルは、探索粒子の極値ではない解への収束を抑制することを意図して作り出されたものである。

GCPSO では、 $gBest$ となる粒子の速度成分にのみ、通常とは異なる更新式が適用される。 $gBest$ 該当粒子の速度成分の更新式は以下のとおりである。GCPSO では、 $gBest$ に対応する $pBest$ を記憶している粒子に対してのみ、(1) 式と異なる速度更新式、

$$v_{id}(t+1) = wv_{id}(t) - x_{id}(t) + y_{id}(t) + \rho(t)r_d \quad (3)$$

が適用される。

ここで、 r_d は一様乱数 $U(-1, 1)$ である。また、 $\rho(t)$ は scaling factor と呼ばれ、

$$\rho(0) = 1.0 \quad (4)$$

$$\rho(t+1) = \begin{cases} 2\rho(t) & \text{if } s_c \geq 5 \\ 0.5\rho(t) & \text{if } f_c \geq 5 \\ \rho(t) & \text{otherwise} \end{cases} \quad (5)$$

によって定義される。ここで、 s_c, f_c はそれぞれ success count, failure count と呼ばれるカウンタである。 s_c や f_c の値は、探索ステップの前後での $gBest$ の評価値の改善または不変が、群のなかでの最良粒子によって何回連続で引き起こされているかを表している。まず、粒子群の位置情報の更新によって群のなかでの最良粒子が変化しなかった場合を考える。このとき、最良粒子が $gBest$ の評価値を改善した場合には、 s_c が 1 加算されると同時に、 f_c は 0 にリセットされる。同様に、最良粒子が評価値を改善しなかった場合には、 f_c が 1 加算されると同時に、 s_c は 0 にリセットされる。次に、位置情報の更新によって最良粒子が前世代と異なる粒子に変わった場合は、 s_c, f_c とも 0 にリセットされる。

GCPSO は評価関数空間上の極値への探索粒子群の収束を促す効果を持つことが示されている。そのため、単峰性関数ではオリジナルの PSO を適用した場合に比べて良い結果をもたらす一方、多峰性関数においては初期収束を引き起こしがちであり、オリジナルの PSO に対して性能が劣る場合があることが確認されている。

3 Stagnation

最適化問題では、解空間内をできるだけ広く探索することによって局所最適解への収束を防ぐことが重要

である。一方で、真の最適解の近傍では解の精度を上げるために局所的な探索を行なう必要がある。そのため PSO は、探索初期は解空間内を大きく動き回り、世代更新が進むにつれて粒子が徐々に $pBest$ や $gBest$ に引きつけられ、最適解に収束していくような探索アルゴリズムとなるよう、探索モデルが設計されることが望ましい。しかし、そのような設計を実現することは、問題にも依存し、PSO においても、動作モデルのパラメータには、推奨の設定があるものの、事前情報なしに、あらかじめ探索モデルのパラメータの与え方などを適切に定めることは困難である。

PSO では、実際には探索が不十分な状態で $pBest$ や $gBest$ が長期間にわたって更新されなかった場合には、新たな領域への探索が進まなくなることがある。このような現象は、評価関数の局所解でない領域でも発生することがあり、これを stagnation という。

Stagnation の発生にはそれぞれの次元での位置成分に対する速度成分の大きさが影響していると考えられる。粒子の速度が慣性項 (第 1 項) に従って減衰し、粒子が新たな位置へ移動、すなわち探索する能力が低下することが原因であるとみられる。通常、粒子の速度成分の大きさは動作式のランダム要素によってごく僅かながらも変化し続けるため、stagnation が発生した場合であってもこれが自然に解消して探索が再開する場合がある。しかしながら、このようなケースは非常にまれであることから、計算コストの有効活用の観点からも stagnation の発生を抑制すること、また早期に発生を検知することが求められる。しかし、最適解が未知である実用の場面においては、速度の急激な低下が局所解への収束か stagnation の発生のどちらによるものかを判別することは非常に困難である。これまでに PSO の改良モデルとして、突然変異の導入¹³⁾、座標軸の回転¹⁴⁾、粒子の再配置等が提案されてきており、これらの操作は上記の影響を打ち消す、あるいは回避する効果をもたらしているものと考えられる。ただし、これらの操作は stagnation の発生そのものを回避あるいは抑制しているとはいえないため、これらとは異なる方法で stagnation の発生を見分ける指標が必要とされる。

また stagnation は、粒子数に比べ問題の次元が大きいつきに発生しやすいことが実験的に確認されている。とはいえ、特に問題の次元が数百～数千を超えるような高次元の問題に対して、それと同数の探索粒子を用意することは計算負荷の面から非現実的であり、Standard PSO2011⁴⁾ では、粒子数の目安として 40 個が推奨されている。一方で、高次元の最適化問題へのニーズは高いことから、進化計算に関する国際会議では高次元関数最適化問題における探索性能を競うコンペティションも開催されている。このような背景から、PSO における stagnation の発生原因を突き止めることの意義は非常に大きいといえる。

4 提案法

前章で確認した通り、stagnation に対処するための PSO の改良モデルはこれまでに数多く生み出されてきたが、一方で stagnation 発生のメカニズムそのものについてはあまり明らかになっていない。そこで、本研究では stagnation がどのような状況下で起こり得るのかを突き止め、それを解消させるための方策を検討した

いと考えている。

ところで、PSO では、世代更新時に粒子の移動できる範囲は、探索粒子の張る (1) 式で示したベクトル空間内に限られる。つまり、1 回の世代交代で 1 つの粒子が探索できる領域はある超平面上に限られる。そのため、粒子の持つベクトルが 1 次独立となる本数が多いほど、より高次元の超平面によって空間内を探索できるようになり、より効率的に解候補の改善が進むことが期待できる。逆に、粒子の持つベクトルが従属である本数が増えるほど、1 回の世代更新で探索できる方向が狭められ、解の探索効率が低下することになる¹²⁾。

我々はこのことに注目し、粒子の探索できる次元の限定によって指標として、粒子の動作モデルに対する可制御性を用いることを検討している。

以下に我々の検討中の PSO の動作式を粒子ごとに粒子の位置と速度を状態変数とする離散時間の状態空間モデルを考える。モデルは以下ようになる。

$$\begin{pmatrix} \mathbf{v}_d(t+1) \\ \mathbf{x}_d(t+1) \end{pmatrix} = \begin{pmatrix} \mathbf{W} & -\varphi \\ \mathbf{W} & 1-\varphi \end{pmatrix} \begin{pmatrix} \mathbf{v}_d(t) \\ \mathbf{x}_d(t) \end{pmatrix} + \begin{pmatrix} \mathbf{b} \\ \mathbf{b} \end{pmatrix} \quad (6)$$

ただし、 \mathbf{W} は、対角成分が w であるような $D \times D$ の正方行列、 φ は、対角成分が $c_1 R_{11} + c_2 R_{21}$ から構成され、各次元でサンプリングされた乱数の値を用いて

$$\varphi = \begin{pmatrix} c_1 R_{11} + c_2 R_{21} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & c_1 R_{1D} + c_2 R_{2D} \end{pmatrix} \quad (7)$$

とかける $D \times D$ の行列である。また、 \mathbf{b} は、 $pBest$ および $gBest$ によって構成される部分であり、同様に各次元においてサンプリングされた乱数の実現値を用いて

$$\mathbf{b} = (c_1 R_{11} p_{i1} + c_2 R_{21} g_{i1} \dots c_1 R_{1D} p_{iD} + c_2 R_{2D} g_{iD})^T \quad (8)$$

とかける。ここで、 $R_{nD}(n = 1, 2, d = 1, 2, \dots, D)$ は、実際にサンプリングされた乱数の値であることに注意されたい。

本研究ではこのシステムの可制御性に着目し、その stagnation との関連を調べる。システムの可制御性は入力が入力が全ての状態変数に影響を及ぼし得ることである。システムを任意の初期状態から有限の時間内に任意の状態に移すことができるような入力が存在するとき、システムは可制御である、と定義されている。可制御性の判定は可制御行列のランクによって行われるが、この行列がフルランクである場合は探索粒子が移動を繰り返すことで、適切な $pBest$ と $gBest$ が選択されていると、任意の位置へ移動しうる状態にあることを意味している。一方で、可制御性を満たさない場合は、移動できる空間が制約され、そのことによって、 $pBest$ や $gBest$ の更新が妨げられている可能性が考えられる。そこで、本研究では、可制御行列のランクが落ちた時点から PSO の探索の性能が悪化するのではないかと仮説のもと、世代ごとの探索粒子の位置や速度、評価値と可制御行列のランクとの関係を数値実験により調べている。

5 数値実験

5.1 実験設定

本論文では、オリジナルの PSO と GCPSO について提案法を適用し、可制御性行列の値にどのような変化が見られるかを調べた。ここではベンチマーク問題として Sphere 関数

$$f(x) = \sum_i^n x_i^2$$

を用いた。関数の次元は 20 と 50 の 2 種類とし、粒子数は 20、探索世代数は 2000 とした。PSO のパラメータは $w = 0.7$ 、 $c_1 = c_2 = 1.4$ とした。

粒子群の初期位置と初期速度は、全ての次元において $[-100, 100]$ で一様サンプリングした乱数を用いた。試行は 5 組の粒子の初期配置にオリジナルの PSO と GCPSO をそれぞれ 5 回ずつ適用して行なった。

5.2 結果

まず、Fig. 1 にある試行での 50 次元の場合の評価値の推移と粒子ごとの可制御性行列のランクの平均値の推移を表したグラフを示す。グラフはオリジナルの PSO 結果と GCPSO の結果とを重ねて表示している。ただし、GCPSO の可制御性行列のランクについては、GCPSO 独自の更新式が適用されるグループ内の最良粒子のランクは常に 2 となることから、これを除いた 19 個の粒子についての平均を載せた。

オリジナルの PSO では 500 世代付近以降から評価値の大幅な改善が見られず、stagnation が発生していることが確認できる一方、GCPSO では探索開始からほぼ一定のペースで評価値が改善し続けていることが読み取れる。ここで、可制御性行列のランクの平均値の変化を見てみると、オリジナルの PSO では評価値の改善ペースの鈍り始めた 300 世代付近以降から徐々に値が低下し始めていることが分かる。stagnation 発生後はほぼ直線的に減少傾向となっている。一方、GCPSO のランクは探索開始から最終世代まで平均的に大きな値の変動はなく、探索後半ではオリジナルの PSO の値を明らかな差をつけて上回っている。

次に、Fig. 2 にある試行での 100 次元の場合の評価値の推移と粒子ごとの可制御性行列のランクの平均値の推移を表したグラフを示す。50 次元の場合と同様に、GCPSO は探索開始から終了まで一定のペースで評価値が改善し続けており、stagnation の発生は見られない。可制御性行列のランクも最後まで低下することはなく、PSO の探索能力が衰えていないことを裏付けているものと思われる。一方、オリジナルの PSO では stagnation の発生が始まった 200 世代目付近から可制御性行列のランクの減少が始まった。探索終了時には開始時点と比べて約 5 程度のランクの低下がみられ、探索能力の低下との関連性が窺える。

これらの実験結果から探索粒子の可制御性は PSO の探索能力の優劣を裏付けているであろうことが推測される。

6 おわりに

本論文では、オリジナルの PSO と極値への収束性能の改善を目的として開発された GCPSO について、stagnation 発生の有無と探索粒子の可制御性との関連性について 50 次元と 100 次元の Sphere 関数の数値実

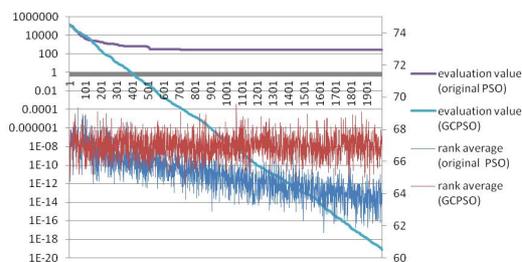


Fig. 1: The transition of the best function value and the rank average in 50 dimension Sphere function.

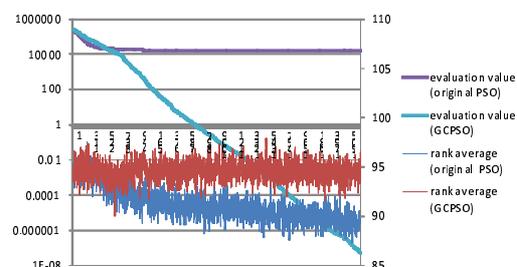


Fig. 2: The transition of the best function value and the rank average in 100 dimension Sphere function.

験によって調べた。そして、stagnation を発生させなかった GCP SO では粒子の可制御性が探索終了まで低下しないことを示した。

この実験結果は粒子の可制御性が PSO の探索能力の優劣を表していることにほかならず、PSO の探索の状態を判定し、stagnation を検出するための指標として用い得る可能性を持つことを示している。今後の展開として、多峰性関数である rastrigin 関数や変数依存性のある rosenbrock 関数などについても同様の傾向が見られるかを調べ、探索粒子の可制御性と stagnation 発生との関連を精査することが挙げられる。

参考文献

- 1) J. Kennedy and R.C. Eberhart : “Particle Swarm Optimization,” *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, pp.1942-1948, 1995
- 2) Y. Shi and R. C. Eberhart : “Parameter selection in particle swarm optimization,” *Proceedings of the 1998 Annual Conference on Evolutionary Programming VII*, pp. 591-600, 1998
- 3) J. Kennedy : “Some Issues and Practices for Particle Swarms,” *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pp. 162-169, 2007
- 4) M. Clerc : “Standard Particle Swarm Optimisation. From 2006 to 2011,” <http://clerc.maurice.free.fr/pso/>
- 5) M. Clerc : “Stagnation Analysis in Particle Swarm Optimisation or What Happens When Nothing Happens,” Technical Report CSM-460, 2006
- 6) M. Jiang, Y. Luo and S. Yang : “Stagnation Analysis in Particle Swarm Optimization,” *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pp. 92-99, 2007
- 7) M. Clerc and J. Kennedy : “The particle swarm: Explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, Vo.6, No.1 pp. 58-73, 2002
- 8) D. Bratton and J. Kennedy : “Defining a Standard for Particle Swarm Optimization,” *Proceedings of*

the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), pp. 120-127, 2007

- 9) F. van den Bergh and A. Engelbrecht : “A new locally convergent particle swarm optimizer,” *Proceedings of the IEEE Conference on Systems, Man, Cybernetics*, Oct.2002
- 10) Peer, E.S., van den Bergh, F. ; Engelbrecht, A.P. : “Using neighbourhoods with the guaranteed convergence PSO” *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS 2003)*, pp. 235 - 242 2003.
- 11) 安田, 島中, 内種 : 粒子群最適化の高次元関数最適化への適用に関する考察, 第 2 回コンピューターショナルインテリジェンス研究会講演論文集, pp. 115-119, 2012
- 12) 安田, 島中: PSO における stagnation の粒子の可制御性に基づく検知, 第 5 回進化計算研究会資料集, pp. 84-87, 2013
- 13) N. Higashi and H. Iba : “Particle swarm optimization with Gaussian mutation,” *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS 2003)*, pp.72-79 , 2003
- 14) T. Korenaga, T.Hatanaka and K. Uosaki : “Performance improvement of particle swarm optimization for high-dimensional function optimization,” *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC2007)*, pp. 3288-3293, 2007

組合せ最適化問題におけるラベルの再割振りに関する研究

○船木亮平 高野浩貴 村田純一 (九州大学)

A Study of Re-assignment of Labels in the Combinatorial Optimization

*R. Funaki, H. Takano, J. Murata (Kyushu University)

Abstract— The authors proposed a new differential evolution technique called Re-labeling differential evolution for combinatorial optimization. In combinatorial optimization problems whose solutions are sets of integer values which are used to identify several different things, landscapes of objective functions depend on these indices. Re-labeling differential evolution re-labels indices so that difference vectors are small between two solutions whose fitness values are close to each other. This idea can be applied to not only differential evolution but also particle swarm optimization. In this paper, the authors discuss the re-labeling method by applying it to particle swarm optimization.

Key Words: Evolutionary Computation, Differential Evolution, Combinatorial Optimization

1 はじめに

Differential Evolution (DE)¹⁾²⁾ は 1995 年に Stone らにより提案された実数値最適化問題向けの進化計算手法である。個体同士の差分ベクトルによって得られる集団の分布情報を用いて効率的に解空間を探索する。また、人間の主観的評価を適合度として与える対話型進化計算 (IEC) において、DE の対比較評価は従来手法の全個体比較による評価と比べ評価時におけるユーザへの負担を軽減する。特に、動画や音声など評価に時間を要する対話型最適化問題では、よりユーザ疲労軽減効果は大きくなる³⁾。

本来、DE は実数値最適化問題を対象に提案された手法であるが、組合せ最適化問題のような離散的な数値を扱う問題に対しても適用可能であり、様々な研究が行われている。例えば、巡回セールスマン問題 (TSP) のような順列最適化問題に DE を拡張した、Relative Position Indexing や Permutation Matrix Approach などが挙げられる⁴⁾。また、筆者らによって提案された Re-labeling Differential Evolution (RLDE)⁵⁾ も組合せ最適化問題に DE を拡張した進化計算手法である。

DE は個体同士の差分ベクトルによって大域的探索、局所探索を実現している。集団が解空間上に広く分布している時は個体間の差分ベクトルは大きく、大域的探索を行う。一方、集団の収束が進むと個体間の差分ベクトルは小さくなり、ベース個体の近辺を局所探索する。このことから、DE は適合度が近い個体間の差分ベクトルが小さくなるような解空間の形状を前提としている事がわかる。組合せ最適化問題では数値を用いて対立遺伝子を区別する場合があります。この時、個体間の差分ベクトルを個体同士の適合度の差の指標にすることができない。RLDE はこのような問題を対象としており、対立遺伝子を区別するために割振られた整数値を割り振り直すことで、個体間差分ベクトルの大きさと個体同士の適合度の差を対応づける。

この対立遺伝子の番号を割振り直す Re-labeling 法は、DE を組合せ最適化問題に拡張するために提案された。しかし、この考えは Particle Swarm Optimization (PSO)⁶⁾ のような他の実数値最適化にも取り入れることができる。本論文では、Re-labeling 法を DE だけではなく PSO にも適用し、シミュレーション実験の結果

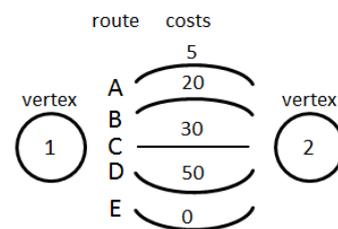


Fig. 1: The problem to minimize a cost between vertex 1 and vertex 2.

を考察することで Re-labeling 法への理解を深め、今後の拡張に役立てることを目的とする。

2 Re-labeling 法

対立遺伝子を区別するための整数値を解とする組合せ最適化問題において、個体間の差分ベクトルの大きさはどのように整数値を割り振るかに依存する。例えば、Fig. 1 の地点 1 と地点 2 の間のいくつかのルートのうち最も疲労せずに移動できるルートを選択する問題を考える。ここではルートは全部で 5 つあり、それぞれ A, B, C, D, E と名前が付いており、移動に要するコストは 5, 20, 30, 50, 0 となっている。DE では数値を用いて探索を行う手法であるため、DE を用いてこの問題を扱うためにはこれらのルートを識別するラベルとして数値を与える必要がある。この時、A のルートを 1, B を 2, C を 3, D を 4, E を 5 と割り振った場合、その解空間は Fig. 2 のように悪特性を持つ形状となり、最適解を見つけるのが難しい問題となる。一方、A のルートを 2, B を 3, C を 4, D を 5, E を 1 と割り振った場合、解空間は Fig. 3 のように緩やかな単峰性となり、最適解を見つけやすい問題となる。このように、DE で組合せ最適化問題を扱う場合、対立遺伝子の識別番号の割振り方に問題難易度が依存してしまう問題がある。探索にベクトルを用いる PSO に関しても同様の事が言える。

Re-labeling 法では、探索を行いながら対立遺伝子の識別番号を割振り直すことで効率的に解を求めることを目指した手法である。集団の収束状況から良い適合度を期待できる遺伝子を予測し、適合度が近いであろう個体同士の差分ベクトルが小さくなるように対立遺伝子の識別番号を振り直す。Re-labeling 法は、対立遺

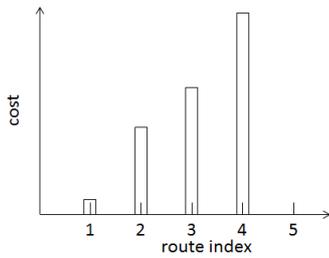


Fig. 2: The global optimal stands alone.

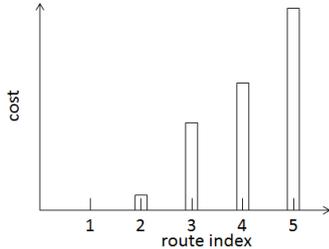


Fig. 3: This problem can be solved easily.

伝子毎により適合度が期待できる度合いである優良度を求める第1段階と、求めた優良度を基に識別番号を割振り直す第2段階に分かれる。

第1段階では、良い適合度になりやすい対立遺伝子を見つける。多くの進化計算手法では、探索序盤で大域的探索を施し大体良い適合度となる解に目星つけた後、その周辺を局所探索する。その時、優良な解周辺に個体が集まっていき、劣悪な解は淘汰される。つまり、多くの個体が集まっている箇所は優良解が存在すると言える (Fig. 4-5)。このことを組合せ最適化問題に置き換えて考えると、多くの個体から保持されている対立遺伝子は優良であり、あまり保持されていない対立遺伝子は淘汰される過程にあると推測することができる。Re-labeling法では、対立遺伝子毎にいくつの個体から保持されているかを優良度として与え、より多くの個体から保持されている対立遺伝子を優良な対立遺伝子とする。

第2段階では、期待される適合度が近い対立遺伝子同士が隣り合うように識別番号を振り直すことで対象問題の解空間の形状を緩やかにする。ここでは、第1段階で求めた優良度が最も高い対立遺伝子を1とし、2番目に高い対立遺伝子を2、続けて3、4と割り振る (Fig. 6)。優良度が等しい対立遺伝子は本来の識別番号の大小関係を保持するように新しい識別番号を割り振られる。

適合度計算時は割り振り直した識別番号を本来の識別番号に戻した後、適合度関数に渡す必要がある。そのため、本来の識別番号と新しい識別番号の対応をリストとして保持しておく。遺伝子のi番目の遺伝子座が再割り振りされた番号は A_i 次配列 R_i に格納される。 A_i は遺伝子のi番目の遺伝子座での対立遺伝子数で、iは1~遺伝子長Dとする。配列 R_i の更新はG世代毎に行う。Gは事前に設定する定数パラメータである。 $R_{i,j}$ には、遺伝子座iの本来の割り振り番号jを再割り振りした番号 r_{ij} を格納する。

この手法は、探索を行う中で優良な対立遺伝子をいくつか予測し、その対立遺伝子を重点的に探索することで収束を早めている。そこで重要なものとなってく

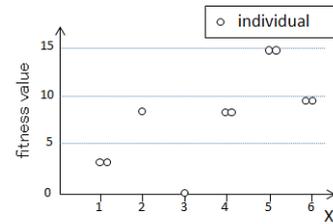


Fig. 4: In the early stages of optimization, the population is widely distributed in the search space.

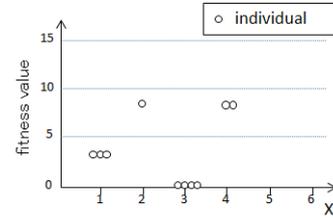


Fig. 5: In the last stages of optimization, individuals are converged at good solutions.

るのが、探索序盤での大域的探索である。十分に大域的探索が施されなかった場合、劣悪な対立遺伝子を優良としてしまったり優良な対立遺伝子を劣悪な対立遺伝子としてしまい、性能低下が懸念される。特に後者に関しては局所解に収束してしまう恐れがある。本論文では、特に個体の多様性がRe-labeling法に与える影響について検証を行う。

2.1 Re-labeling Differential Evolution

DEでは差分ベクトルの大きさが探索する範囲の広さを表す。探索初期の集団が探索領域に広く分布している時、個体間差分ベクトルは大きくなり広い範囲を探索する大域的探索を行う。一方、最適解周辺に集団が収束してくると個体間の差分ベクトルは小さくなり局所探索を行う。このためには、適合度が近い個体ほどその差分ベクトルが小さくある必要がある。前述のように、識別番号としての整数値の差分は必ずしもそのような特徴はなく、Re-labeling法によってそれを実現する。以下にRe-labeling Differential Evolutionの手順を示す。

1. 集団を初期化する。
2. G世代毎に配列Rを更新する。
3. 親個体となるtarget vectorを決定する。
4. 2の個体以外からbase vector(X_{R1})をランダムに選択する。
5. 2, 3以外の個体から2個体 X_{R2} , X_{R3} をランダムに選択する。

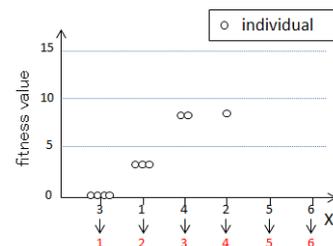


Fig. 6: Re-labeling DE sorts indices by the number of individuals which contain each index.

6. X_{R1} , X_{R2} , X_{R3} を配列 R を用いて識別番号を再割り振りする.
7. 次式より mutant vector, U を生成する.

$$U = X_{R1} + F \times (X_{R2} - X_{R3}) \quad (1)$$

F は 0 よりも大きな定数で, X_{R2} と X_{R3} の差分ベクトルをスケールする.

8. 作成した mutant vector を配列 R を用いて本来の識別番号に戻す.
9. target vector と mutant vector を交差し, trial vector を作成する.
10. target vector と trial vector を比較し, 適合度が優良な方を target vector に上書きする.
11. 3-10 の操作を全ての個体に対して行い, これを 1 世代とる.
12. 2-11 までの操作を指定世代数まで繰り返す.

DE において多様性を調整するパラメータは F である. F が大きな値だと base vector の移動幅が大きくなり広い探索を行う. 一方, F が小さな値だと base vector の変化が少なくあまり大胆な探索を行わない. また, 当然個体数の多さも大きく関わってくる. これらの点に注意してパラメータ設計を行う必要がある.

2.2 Re-labeling Particle Swarm Optimization

Particle Swarm Optimization (PSO) では, 集団全体でこれまで発見した最優良解 gbest 方向のベクトル, その個体 (粒子) 自身がこれまで発見した最優良個体 pbest 方向へのベクトル, その個体が直前まで動いていた方向へのベクトルの 3 つのベクトルを用いて個体の移動方向を更新し, 優良解を見つける. この手法でも個体間の差分ベクトルを用いる点で DE と似ている. PSO を組合せ最適化問題に適用する場合でも, DE と同様に対立遺伝子への識別番号の割り振り方によって対象問題の解空間形状が大きく変化する. そのことから, Re-labeling 法によって解空間形状を緩やかにすることで性能向上が期待出来る. 以下に Re-labeling Particle Swarm Optimization の手順を示す

1. 集団, 各個体の初期速度を初期化する.
2. G 世代毎に配列 R を更新する.
3. 進化の対象となる個体 X を決定する.
4. X , \hat{X}_p , \hat{X}_g を配列 R を用いて識別番号を再割り振りする. ただし, \hat{X}_p は個体自身がこれまで見つけた中で最も良い解, \hat{X}_g は集団全体でこれまで見つけた最優良解である.
5. 以下の式より X の速度 V を更新する.

$$U = wV + c_1r_1(\hat{X}_p - X) + c_2r_2(\hat{X}_g - X) \quad (2)$$

w , c_1 , c_2 はベクトルをスケールする定数, r_1 , r_2 は $[0, 1]$ の乱数である.

6. 以下の式より X を更新する

$$X = X + V \quad (3)$$

7. 新しく得られた X を配列 R を用いて本来の識別番号に戻す.
8. X の適合度を求める.

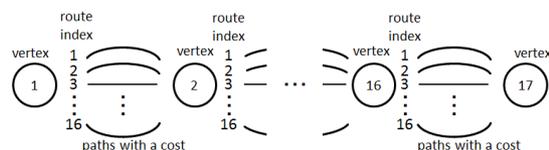


Fig. 7: The problems of finding an optimal route from city 1 to city 17 with the lowest total cost.

9. 3-8 の操作を全ての個体に対して行い, これを 1 世代とする.
10. 2-9 までの操作を指定世代数まで繰り返す.

PSO では定数 w , c_1 , c_2 で多様性の調整を行う. w が大きいと各々の個体は独自に探索空間を移動するだろう. 一方, c_1 が大きいとその個体がこれまで見つけた最良解に引っ張られ, その周辺を重点的に探索し, c_2 が大きいと集団全体での最良解に引き寄せられるように探索を行う. Re-labeling 法を適用する際はこれらのごとについて考慮する必要がある.

Re-labeling 法を適用する際に注意しなければならない点がある. PSO では速度 V によって更新された個体は前世代での適合度とは比較されず必ず次世代に持ち越される. 前世代よりも優良な個体でないと次世代に持ち越されない DE とはこの点で異なる. 集団全体が優良解に集まっていくという前提の上で対立遺伝子の優良度を定義している Re-labeling 法は効果的に働かない可能性がある. しかし, 個体毎に保持している, その個体がこれまで見つけた解で最も優良な解 pbest は悪化することはない, また, より優良な解を見つける度に最適解に近づいていくと考えられる. Re-labeling 法の良い適合度になりやすい対立遺伝子を見つける第 1 段階において, 対立遺伝子毎にいくつの個体の pbest から保持されているかを優良度とすることでこの問題は解決できる. シミュレーション実験では対立遺伝子を保持している個体の数を優良度とする通常の Re-labeling 法と, 対立遺伝子を保持している pbest の数を優良度とする Re-labeling 法の 2 種を適用する.

3 実験条件

3.1 テスト問題

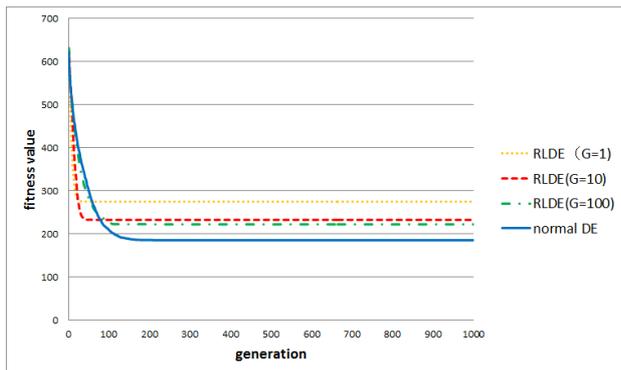
Fig. 7 のような重み付きグラフにおいて頂点 1 から頂点 17 までの移動コストを最小化する問題を取り扱う. 頂点 1 から頂点 17 まで移動するために 16 回重み付きの辺を選択する必要があり, 隣り合う頂点同士は 16 の辺で結ばれている. 辺の重みは Table 1 のようになっており, 頂点 1 から頂点 17 までに選択した辺の重みの総和を移動コストとして最小化を行う. 最適解は 90 である. 個体の解表現は $[1, 17]$ の範囲の実数値を要素とした 16 次元配列とし, 遺伝子の少数点以下を切り捨てた値と一致する識別番号の辺を選択する.

3.2 パラメータ設定

Re-labeling 法の特徴を解析するために DE と PSO に適用する. 両手法とも 16 個体, 100 個体, 最大世代数は 1000 世代, 試行回数は 100 試行とし, 識別番号を最割り振りする間隔を表す定数 G を 1, 10, 100 で探索を行う. さらに, それぞれの手法でいくつかの設定パラメータを調整し, その影響を考察する. Re-labeling DE で

Table 1: The weighted edges in Test problem.

	the array element number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
edge 1	6	96	11	20	51	42	38	68	27	97	38	61	62	32	72	75
edge 2	63	2	43	74	29	65	64	62	95	15	89	17	59	78	20	69
edge 3	74	44	9	67	61	49	42	32	43	64	47	93	11	61	36	55
edge 4	85	52	96	6	22	91	39	47	69	50	92	34	24	45	79	57
edge 5	64	81	80	22	0	72	61	51	83	76	15	38	84	63	83	96
edge 6	87	83	22	58	39	1	13	52	11	11	29	10	43	90	47	30
edge 7	59	57	88	95	65	15	4	31	18	13	72	70	10	61	45	67
edge 8	86	53	88	78	23	97	67	6	66	49	60	85	23	67	70	45
edge 9	48	61	98	56	65	70	98	24	9	35	60	21	71	30	89	48
edge 10	66	45	52	19	30	68	96	40	70	10	87	68	61	90	32	65
edge 11	68	93	81	47	65	70	19	19	18	20	2	74	34	95	62	64
edge 12	44	85	82	52	44	25	68	70	97	42	55	9	73	71	38	52
edge 13	49	16	51	33	94	28	28	69	97	36	57	36	8	50	89	11
edge 14	36	40	56	26	91	42	67	38	89	53	92	33	35	4	96	52
edge 15	90	17	80	79	71	21	24	22	52	20	18	26	96	86	7	37
edge 16	15	68	71	76	50	77	99	80	55	71	92	45	48	98	62	7

Fig. 8: The average convergences graph of Re-labeling DE with 16 individuals and $F=0.3$.

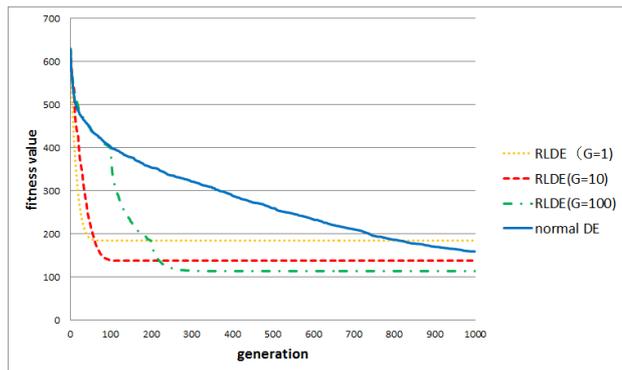
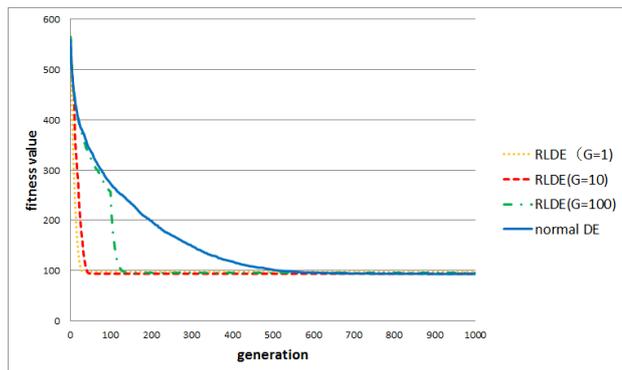
は交叉率を 0.8 として uniform crossover を行い, F を 0.3, 0.8 としてで実験を行う. 比較対象として識別番号の最割振りを行わない通常の DE を同パラメータ設定で実験を行う. Re-labeling PSO では (w, c_1, c_2) の値を $(0.9, 0.1, 0.1)$, $(0.9, 1.0, 1.0)$ とし, 速度の最大値は 2.0 とする. さらに, 対立遺伝子を保持している個体の数を優良度とする通常の Re-labeling 法と, 対立遺伝子を保持している pbest の数を優良度とする Re-labeling 法の 2 種を適用する. また, 比較対象として識別番号の最割振りを行わない通常の PSO を同パラメータ設定で実験を行う.

4 実験結果と考察

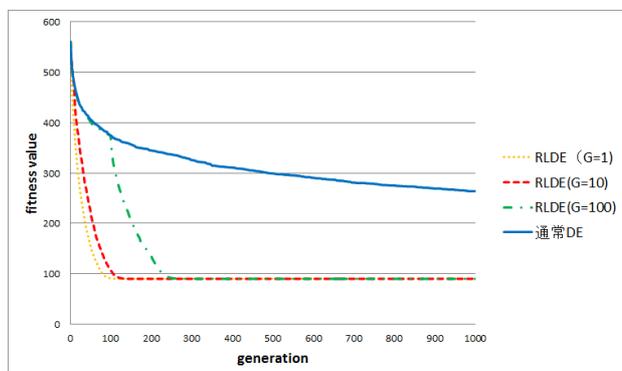
100 試行における最良適合度の平均の推移グラフを Fig. 8-Fig. 15 に示す. 最良適合度の平均の推移グラフは横軸が世代数, 縦軸が適合度となっている. RLDE($G=n$) は n 世代毎に Re-labeling 法を実行している Re-labeling DE, normal DE は Re-labeling 法を用いない基本的な DE である. RLPSO($G=n$) は対立遺伝子を保持している個体の数を優良度とする通常の Re-labeling 法を n 世代毎に行った手法で, RLPSO($G=n$)(pbest) は対立遺伝子を保持している pbest の数を優良度とする Re-labeling 法 n 世代毎に行った手法を示す. normal RLPSO は Re-labeling 法を用いない基本的な PSO である.

4.1 Re-labeling DE の結果と考察

まず, 個体数 16, $F = 0.3$ の Re-labeling DE のグラフ Fig. 8 では, 全ての手法で解を見つけることができず, Re-labeling DE は通常の DE に比べ局所解への収束が多く, Re-labeling を頻繁に行うほど得られる解は

Fig. 9: The average convergences of Re-labeling DE with 16 individuals and $F=0.8$.Fig. 10: The average convergences of Re-labeling DE with 100 individuals and $F=0.3$.

劣っている. 少個体数, $F = 0.3$ という極端に多様性が損なわれる設定において, Re-labeling 法を用いた際

Fig. 11: The average convergences of Re-labeling DE with 100 individuals and $F=0.8$.

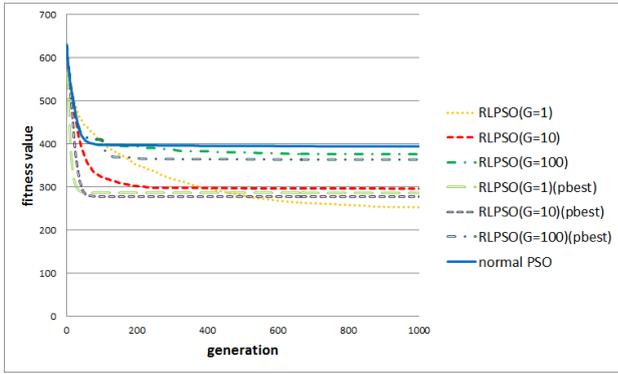


Fig. 12: The average convergences of Re-labeling PSO with 16 individuals and $(c_1, c_2) = (0.1, 0.1)$.

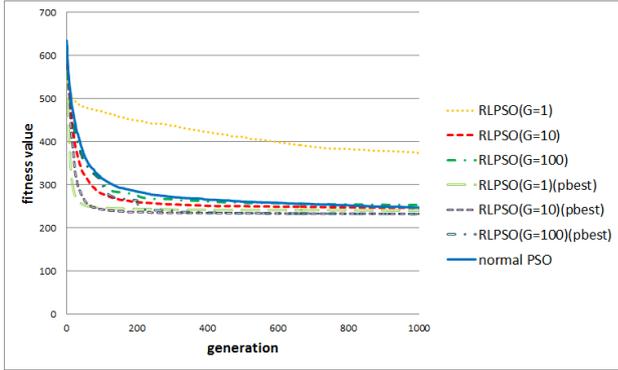


Fig. 13: The average convergences of Re-labeling PSO with 16 individuals and $(c_1, c_2) = (1.0, 1.0)$.

の性能低下が著しい。

一方、個体数 16, $F = 0.8$ の Re-labeling DE のグラフ Fig. 9 では、最適解とは言えないものの比較的優良解に収束している。ここでも $F = 0.3$ の時と同様に頻繁に Re-labeling 法を用いることで個体の収束が早まっている。少個体数でのシミュレーション実験は対話型進化計算を想定しているものである。ユーザの評価時における疲労と人間の認識の弁別閾を考慮すると、対話型最適化問題では解の正確さよりも優良な解への迅速な収束が求められる。通常 DE では、最終的に得られる解は十分優良ではあるが、獲得までの世代数が大きく、ユーザ疲労を考えると現実的ではない。

次に、個体数 100, $F = 0.3$ の Re-labeling DE のグラフ Fig. 10 では、Re-labeling 法を用いることで早期に最適解獲得に成功している。個体数が多く、十分

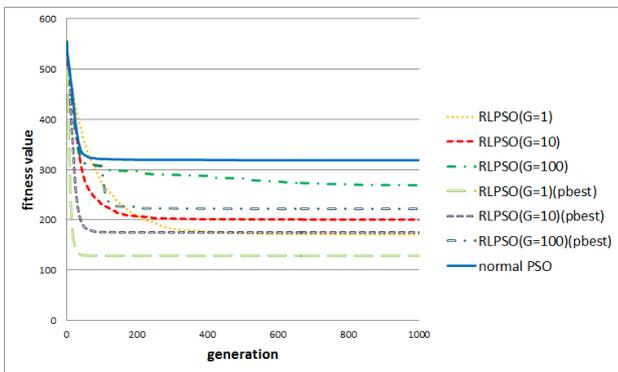


Fig. 14: The average convergences of Re-labeling PSO with 100 individuals and $(c_1, c_2) = (0.1, 0.1)$.

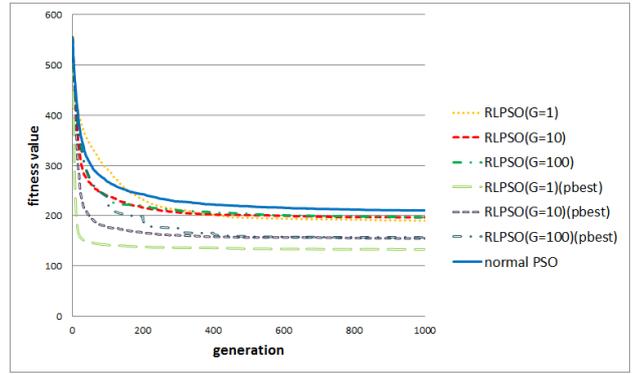


Fig. 15: The average convergences of Re-labeling PSO with 100 individuals and $(c_1, c_2) = (1.0, 1.0)$.

に大域的探索を行うことのできる場合は Re-labeling 法によって最適解を見つけることができる。個体数 100, $F = 0.8$ の場合でも同様のことが言える (Fig. 11)。また、RLDE($G=100$) の 100 世代目から大きく最良適合度が更新されている。これは 100 世代目までの探索の情報から優良対立遺伝子の推測に成功したためだと思われる。

4.2 Re-labeling PSO の結果と考察

個体数 16, $(c_1, c_2) = (0.1, 0.1)$ の Re-labeling PSO のグラフ Fig. 12 を見てみると、全ての手法で優良解を獲得することができなかったが、Re-labeling 法を用いることで性能向上が見られた。RLPSO($G=1$) では解の多様性は確保出来ているがその分収束が遅くなっている。PSO は解が改悪しても次世代に持ち越すため、Re-labeling 法の良い解に個体が集まるという前提に適さない。その点が優良な対立遺伝子を見つけられないことに繋がり、闇雲な探索になっていると思われる。

一方、RLPSO($G=1$)(pbest) では初期収束の早さは向上しているがそのまま局所解に収束しやすい。これは、最も優良な対立遺伝子を劣っていると見なしてしまったことが原因だと考えられる。PSO の演算では優良解方向へのベクトルが使われる。これは、Re-labeling 法で最優良の対立遺伝子番号である 0 の方向へのベクトルとなりやすく、一度劣った対立遺伝子として大きな識別番号を割り当てると再びその対立遺伝子を選ばれる可能性が低くなる。特に本来の識別番号が大きな対立遺伝子で顕著である。今回、同じ優良度を与えられた対立遺伝子は本来の識別番号の大小関係を持続するように新しい識別番号を割り振られる。そのため、本来の識別番号が小さな対立遺伝子は劣った対立遺伝子と判断されてもそれほど大きな識別番号を割り当てられないことが多い。実際に最終的に得られた解を見てみると、最適解が大きな番号になる遺伝子座ではほとんど局所解に収束していた。Re-labeling PSO はこの点を改良することで全体の性能向上が望める。

個体数 16, $(c_1, c_2) = (1.0, 1.0)$ の Re-labeling PSO のグラフ Fig. 13 でも同様のことが言える。優良個体方向へのベクトルが強めであるが、normal PSO に比べ Re-labeling PSO はその恩恵をあまり受けていない。前述のように、Re-labeling PSO は一度良い解だと判断してしまうとその方向へのベクトルが強効いてしまい局所解に収束しやすいためだと思われる。

個体数 100, $(c_1, c_2) = (0.1, 0.1)$ の Re-labeling PSO

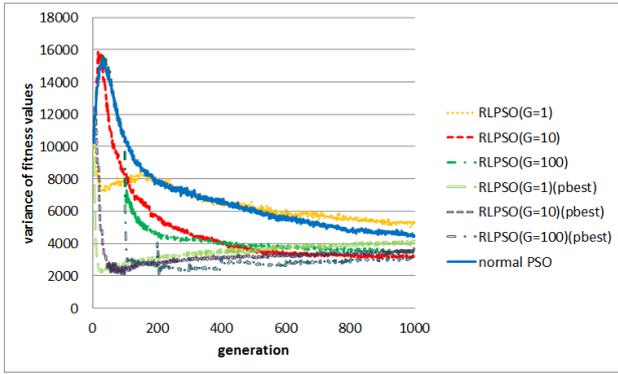


Fig. 16: The change of fitness value variance of Re-labeling PSO with 100 individuals and $(c_1, c_2)=(1.0, 1.0)$.

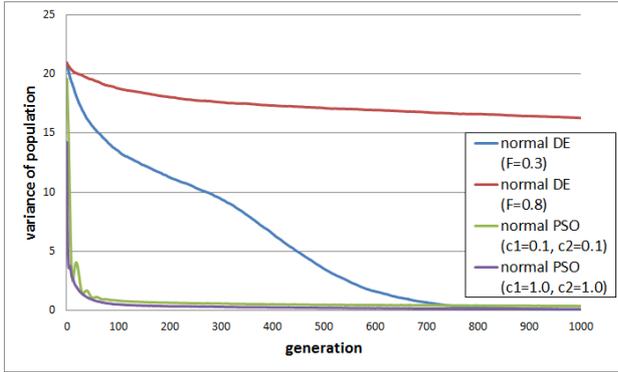


Fig. 17: The change of population variance of normal DE and normal PSO with 100 individuals.

のグラフ Fig. 14, 個体数 100, $(c_1, c_2)=(1.0, 1.0)$ の Re-labeling PSO のグラフ Fig. 15 では, 個体数が多い分局所解への収束は抑えられているが, それでも最適化を獲得するには至っていない。RLPSO($G=100$)(pbest) では 100 世代, 200 世代で識別番号の最割振りが行われているが, Re-labeling DE の場合と比べて最良適合度に大きな変化がない。しかし, 適合度の分散の推移 (Fig. 16) を見てみると 100 世代までは normal PSO とほぼ同じ曲線を描いているが 100 世代目以降で大きく低下している。これも前述のように, Re-labeling 法によって局所的な優良解に集団全体が集まってきているためである。同時に個体の多様性を失い, 探索能力が落ち, 適合度の分散だけが大きく変化したと考えられる。

最後に 100 個体の normal PSO と normal DE の全世代を通じた個体分散の推移グラフを Fig. 17 に示す。横軸が世代数, 縦軸が集団の分散である。また, 集団分散は解配列の各要素毎に分散を求め, 全ての要素で平均を取ったものを集団全体の分散としている。normal DE ($F=0.3$) は F を 0.3 とした通常 DE, normal DE ($F=0.8$) は F を 0.8 とした通常 DE を表す。また, normal PSO ($c_1=0.1, c_2=0.1$) は $(c_1, c_2)=(0.1, 0.1)$ とした通常の PSO, normal PSO ($c_1=1.0, c_2=1.0$) は $(c_1, c_2)=(1.0, 1.0)$ とした通常の PSO を表している。DE は $F = 0.3$ であっても多様性を維持したまま探索を進めているが, PSO では $(c_1, c_2)=(0.1, 0.1)$ と優良個体方向のベクトルを抑制していても早期に多様性が失われている。

5 結論

本論文では, DE を組合せ最適化問題に拡張するために提案した Re-labeling 法を PSO に適用し, シミュレーション実験の結果から考察を行った。Re-labeling 法では, 対立遺伝子毎にいくつの個体から保持されているかを数え, その個数をその対立遺伝子の優良度として取得する。その優良度が高いほど優良な適合度になりやすいものとして, 対象問題の解空間が緩やかになるように対立遺伝子の識別番号を割振り直す。これは, 優良解に個体が収束していくという進化計算の特性を利用したものである。PSO では次の世代の適合度が前の世代よりも良いとは限らないため, 各個体の pbest から優良度を求めた。

また, Re-labeling 法は探索序盤に十分な大域的探索を行わなければ局所解へ収束してしまう。Re-labeling 法では同じ優良度の対立遺伝子は本来の識別番号の大小関係を壊さないように新しい識別番号を再割り振りされる。このことで, 本来の識別番号が大きな対立遺伝子が最適解に含まれる場合に, 一度優良対立遺伝子の可能性から外されてしまうと復帰することが難しい。PSO では優良解方向, つまり割振り番号が小さな方向へのベクトルの影響で特に局所解への収束が見られた。Re-labeling 法は本来の対立遺伝子の割振り番号がどうであっても解を見つけることを目的としているため, 本来の割振り番号に依存する動作は好ましくない。

今後の展開として, まず, 同じ優良度の対立遺伝子の識別番号の割振り方法を変更し, 等しく選択されるように改善する。その後, 優良度が高い対立遺伝子に収束し過ぎないように改良し, 探索範囲を調整するパラメータの影響を大きく受けてしまうという点の改良に努めたい。

参考文献

- 1) R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution", the International Conference on Evolutionary Computation (ICEC1996), pp.842-844 (1996).
- 2) R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization vol.11, pp.341-359 (1997).
- 3) H. Takagi and D. Pallez, "Paired Comparison-based Interactive Differential Evolution," The first World Congress on Nature and Biologically Inspired Computing (NaBIC2009), Coimbatore, India, pp.375-480 (December, 2009).
- 4) D. Lichtblau, "Discrete optimization using Mathematica", 2002 World Multiconference on Systemics, Cybernetics and Informatics (SCI2002), Vol.16, pp.169-174 (2002).
- 5) R. Funaki, H. Takano and J. Murata, "Re-labeling Differential Evolution for Combinatorial Optimization", SICE Annual Conference 2013, Nagoya (September, 2013).
- 6) James Kennedy, "Particle Swarm Optimization", Springer US, pp760-766, (2010).

スマートフォンによるユーザ参加型センシングシステムにおける 観光写真の Levenshtein 距離の免疫システムによる分類手法

○鎌田 真 (県立広島大学大学院総合学術研究科経営情報学専攻)
市村 匠 (県立広島大学経営情報学部経営情報学科)

A Classification Method of Photographs of Sightseeing Spots in Smartphone Based Participatory Sensing System by Clonal Selection Algorithm with Immunological Memory Cells

*Shin Kamada (Graduate School of Comprehensive Scientific Research,
Prefectural University of Hiroshima)

Takumi Ichimura (Faculty of Management and Information Systems,
Prefectural University of Hiroshima)

Abstract— In our research, the tourist subjective data collection system with Android smartphone called ‘Hiroshima Tourist Map’ has been developed. The application can collect subjective data such as GPS, the evaluation, comments, and the photograph in sightseeing spots where a tourist visits. We have already proposed the classification method from the collected subjective data by the interactive GHSOM and the knowledge is extracted from the classification results of the interactive GHSOM by C4.5. However, the image data was not included in the classification tasks, because the extraction of the specified characteristics from images cannot be realized. We propose the effective method of the Levenshtein distance to deduce the spatial proximity of image viewpoints and thus determine the specified pattern in which images should be processed. To verify the proposed method, some experimental results to classify the subjective data with images are compared with by SOM and Clonal Selection Algorithm with Immunological Memory Cells.

Key Words: Mobilephone Based Participatory Sensing System, Levenshtein Distance, Image processing

1 はじめに

近年では、OpenCV¹⁾ と呼ばれるオープンソースの画像処理ライブラリでも知られているように、画像処理技術が発達してきた。人間の顔の輪郭を自動で抽出する方法もあり、デジタルカメラなどで応用されている。また、文献²⁾では、ソフトコンピューティングの手法を用いた顔画像の分類も行われている。文献³⁾では、スマートフォンによって撮影された観光地のランドマークの画像が自己組織化マップにより分類され、顔画像以外の画像に対しても分析が行われている。

近年、モバイルフォンベースユーザ参加型センシング (Mobilephone Based Participatory Sensing; MPPS)⁴⁾ システムによるデータ収集と分析手法が提案されており、本研究では、スマートフォンを用いて広島県の観光情報を収集する Android アプリ (ひろしま観光マップ⁵⁾) を開発している。このアプリでは、GPS による位置情報、5段階評価値、短いテキスト文、写真で構成された旅行者の主観的情報を収集することができる⁶⁾。このような主観的で多次元的なデータには曖昧性が多く含まれるため、分析には厳密な解を求める統計的な手法や機械学習による手法よりも、解に人間の持つ曖昧性を取り入れたソフトコンピューティングによる手法が有効であると考えられている⁷⁾。そのため、我々の研究では、ソフトコンピューティングの手法で教師なし学習であるインタラクティブ階層成長型自己組織化マップ (Interactive Growing Hierarchical SOM; Interactive GHSOM)⁸⁾ を用いて、収集された約 600 件の情報について、写真を除いた、位置情報、評価値、コメント文を分類してきた。さらに、C4.5⁹⁾ を用いて、インタラクティブ GHSOM の分類結果から IF-THEN

ルール形式の知識を獲得した¹⁰⁾。獲得した知識を元に、フィルタリングルールを作成し、観光にとって魅力的な情報や新規性のある情報のみを Twitter や Facebook の SNS に発信するシステムを開発している^{11, 12)}。

インタラクティブ GHSOM は教師なし学習であり、ひろしま観光マップで収集されたデータのように、教師信号のない入力データを人間に近い感覚で分類できるアルゴリズムである。一方、教師あり学習は教師信号のある入力データを学習し、未知の入力データを推定することに用いられている。本研究では、撮影された写真が観光地の代表的なランドマークであるかどうかを判別するために、教師あり学習で分類精度が高いとされている免疫学的記憶細胞を用いたクローン選択アルゴリズム (Clonal Selection Algorithm with Immunological Memory; CSAIM)¹³⁾ を用いて、投稿された写真のランドマークの画像との類似性を測ることを試みた。

画像間の類似性を測る研究としては、パノラマ画像との類似性をレーベンシュタイン距離により測る研究がある¹⁴⁾。本研究では、データ分析における情報量を削減するために特徴点を抽出する方法を提案し、ランドマークとなる画像に対する類似性をレーベンシュタイン距離により測定した。類似性と計算時間の観点から、自己組織化マップ、インタラクティブ GHSOM、CSAIM による計算実験結果を比較し、ここに報告する。

本論文の構成は次のとおりである。2 節ではひろしま観光マップについて述べ、3 節では本論文で提案するレーベンシュタイン距離を用いた特徴点抽出方法について述べ、4 節では CSAIM について述べる。5 節で実験結果を述べ、6 節で結論を述べる。



(a) Initial display (b) Submission of a new sightseeing spot

Fig. 1: Hiroshima Tourist Map

2 MPSSからの情報フィルタリングルールの作成

本研究では、MPSSとして、Androidアプリケーションひろしま観光マップ⁵⁾を開発し、広島県に訪れた旅行者から主観的情報を収集している⁶⁾。Fig. 1(a)は、開発したひろしま観光マップの初期画面である。Fig. 1(b)のように、実際に訪れた観光地において、GPSによる位置情報、5段階評価値、140文字以内のコメント文、写真を収集すると同時に、サーバにデータを投稿することができる。本研究では、収集された旅行者の主観的情報を、次節に示すインタラクティブ GHSOMを用いて分類している。

2.1 インタラクティブ GHSOM⁸⁾

インタラクティブ GHSOMは、A.Rauberらによって提案された階層成長型自己組織化マップ (Growing Hierarchical SOM; GHSOM)¹⁶⁾を拡張したアルゴリズムである。GHSOMは、Fig. 2のように、木構造の概念を取り入れた自己組織化マップ (Self Organizing Maps; SOM)¹⁷⁾であり、ユニットの挿入とマップの階層化を行うことで、入力データに合わせて構造を動的に変化させることができる。しかし、ユニットの挿入とマップの階層化に関わるパラメタの設定が難しく、この値によっては深く階層化し過ぎることがある。例えば、ごく少量のデータしか含まれていないユニットから階層化が起り、かえって特徴を捉えづらいマップ構造が生じる場合がある。この場合、得られた構造からどのような特徴を持ったデータであるのか分かりづらく、知識獲得を行うことが困難になる。この問題点に対応するため、市村らは GHSOMが必要以上に階層化を行うのを抑制する手法を提案し¹⁸⁾、ユーザの要求を満足する分類結果をインタラクティブに得られるインターフェースを開発した。開発したインターフェースでは、特定のノードに対する再分類を行い、人間にとって理解しやすい構造を求めるとともに、マンセルの20色相環¹⁹⁾により、マップ上のユニットが色付けされ、視覚的に分類結果を理解しやすくなっている。

2.2 獲得した知識

インタラクティブ GHSOMを用いて、ひろしま観光マップにより収集された約600件の主観的情報の、位

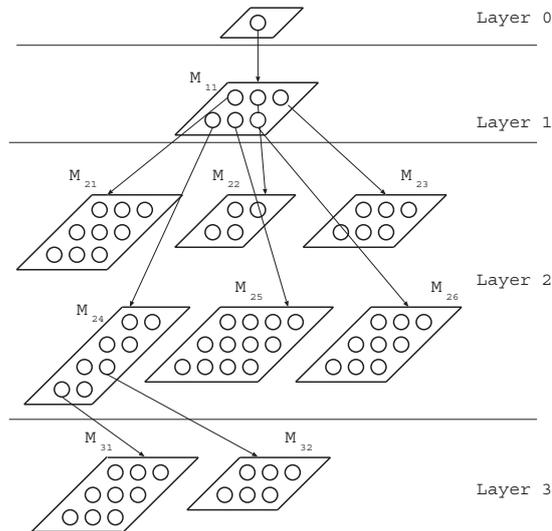


Fig. 2: A Hierarchy Structure in GHSOM

置情報、5段階評価値、コメント文について分類した。また、インタラクティブ GHSOMの分類結果は、IF-THENルール形式で表現されていないため、C4.5を用いて分類結果をIF-THENルール形式の知識に変換した。なお、コメント文は、位置情報や5段階評価値のような数値的情報ではなく、自然言語であるため、ひろしま観光マップで得られた単語と観光Webサイトやブログから得られた単語の出現頻度をもとにTF-IDF値²⁰⁾を求めた。

Fig. 3は、インタラクティブ GHSOMの分類結果である。各マップには、“[R][xx][xx]”のように、ルートマップから各マップまでを特定できるようにマップ番号を割り当てている。Fig. 3(a)では、マップ “[R][11]” から複数のマップが階層化され、構造が複雑になってしまった。このマップを再学習することで、Fig. 3(b)のように、構造を簡略化することができた。

Fig. 4は、Fig. 3の結果の一部をC4.5によりIF-THENルール形式の知識に変換した結果を示している。Fig. 3(a)とFig. 4(a)、Fig. 3(b)とFig. 4(b)がそれぞれ対応している。再学習することにより、A1とA2がA'になり、E1からE4がE'になるなど、複雑だったルールを簡略化することができた。なお、インタラクティブ GHSOMの再学習により構造が複雑になる場合もある。

Fig. 5は、これらの結果を示した図であり、評価値が高い「宮島」と「呉」のサンプルを示している。ここで、Fig. 5のアルファベットは、Fig. 4の各ルールのアルファベットに対応しており、ルールに該当するサンプルを示している。ルールBやD1に含まれるサンプルはTF-IDF値が高く、宮島、かきなどの有名な観光地に関する単語が含まれているが、ルールA'やD2に含まれるサンプルはTF-IDF値が低く、そのような単語は含まれていない。このように、宮島のような有名な観光地で、評価値が高く、観光に関する単語が含まれている情報を抽出するフィルタリングルールを獲得することができた。本研究では、獲得したフィルタリングルールを活用したSNSへの情報発信システムを開発している。しかしながら、獲得したフィルタリングルールには、写真の分析結果は含まれていない。す

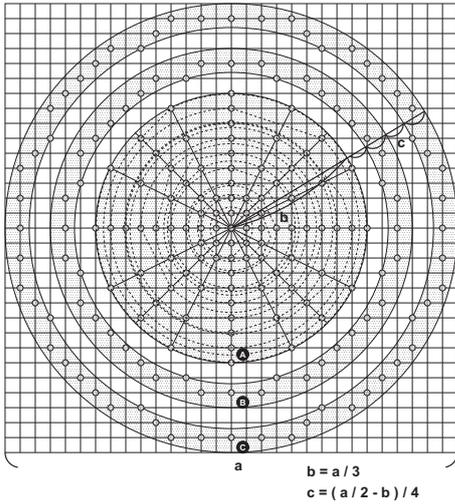


Fig. 6: A sequence of pixel in a photo image

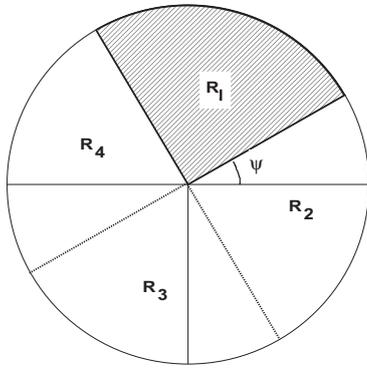


Fig. 7: 4 sub-regions on roulette wheel

3.2 特徴点抽出方法

本研究で提案する、画像の特徴点抽出方法について述べる。提案手法では、全ての領域の画素を使わず、Fig. 6に示す領域の画素を使用する。Fig. 6は、本論文で使用する画像のサイズが 31×31 の例である。Fig. 6に示すように、まず画像を3つ円A, B, Cの領域に分割する。この時、円Aは画像の中心を原点とした時、半径 b の弧の領域、円Bは半径 $b+c$ から半径 $b+2c$ の弧の領域、円Cは半径 $b+3c$ から $b+4c$ の弧の領域である。ここで、画像の縦幅と横幅で、短い方を a とした時、 $b = a/3$, $c = (a/2 - b)/4$ である。

円Aに対しては、Fig. 6のように、中心から傾き ± 1 , ± 2 , $\pm \frac{1}{2}$ の直線上にある点を螺旋状に取得する。円B, Cに対しては、領域に入っている画素を円に沿って取得する。このように、提案手法では画像の中心ほど多くの画素を抽出し、外側ほどあまり画素を抽出しない手法となっている。

さらに、これらの円の領域の中で、対象となるオブジェクトがどこにあるかを発見するために、円A, B, Cを、Fig. 7のように n 個の扇形に分割した。 ψ は、各扇形の角度を決めるパラメータで、写真の特徴に応じて任意の値をとる。Fig. 7は、円をR1からR4の4つに分割している例である。

本論文では、いくつかの ψ に対する全ての扇形対

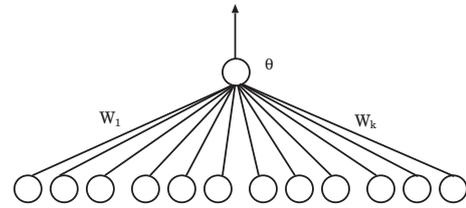


Fig. 8: The antibody structure

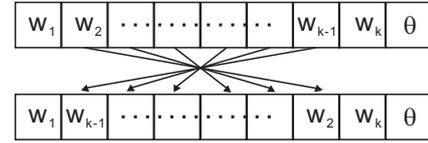


Fig. 9: RE for w_2, w_{k-1}

して分析を行い、最も精度が高くなった扇形が選択される。次節では、抽出した特徴点の分類に用いられる、CSAIMについて述べる。

4 免疫学的記憶細胞を用いたクローン選択アルゴリズム¹³⁾

人工免疫システム (Artificial Immune System; AIS) は、多くの生物に備わっている免疫系を応用したシステムであり、パターン認識や最適化などの様々な分野に適用されている。人工免疫システムには、Burnetが提唱したクローン選択理論 (Clonal Selection Theory)²¹⁾ を元にしたクローン選択アルゴリズムがある。抗体の増殖過程では主に体細胞超変異 (Somatic Hypermutation; HM) とレセプター編集 (Receptor Editing; RE) の2つの手法があるが、GaoはRECSA (Receptor Editing Clonal Selection Algorithm)²²⁾ と呼ばれる、REを取り入れたクローン選択アルゴリズムを提案した。

しかし、GaoのRECSAでは、免疫系本来の機能である免疫学的記憶については実装されていない。そこで、本研究では、RECSAによって生成された親和性が高い抗体を記憶細胞に保存する手法として、免疫学的記憶細胞を取り入れたクローン選択アルゴリズム (Clonal Selection Algorithm with Immunological Memory; CSAIM) を提案した。特定の訓練事例に対して特化した抗体が記憶細胞に保存され、また、正答できない訓練事例が表れた場合、それに適応する新たな抗体が生成される。

4.1 抗体の構造

GaoのRECSAを分類問題に適用するために、抗体をFig. 8に示すように、入力に対する重み $w_i (i = 1, 2, \dots, k)$ と、出力に対するしきい値 θ で定義した¹³⁾。これをパラトープ $P = (w_1, \dots, w_k, \theta)$ とする。

4.1.1 Somatic Hypermutation (HM) と Receptor Editing (RE)

HMは、パラトープ $P = (w_1, \dots, w_k, \theta)$ から w_i または θ をランダムに選択し、式(2)を行う。ここで、 Δw , $\Delta \theta$ は $-\gamma_w \leq \Delta w \leq \gamma_w$, $-1 \leq \Delta \theta \leq \gamma_\theta$ を満たす。

$$w_i = w_i + \Delta w, \theta = \theta + \Delta \theta \quad (2)$$

REは、パラトープ $P = (w_1, \dots, w_k, \theta)$ に対して、Fig. 9のようにランダムに選択された2点間の並びを入れ替える。

4.1.2 親和性

各抗体の親和性は、以下の式により計算される。ここで、 x^p は各訓練事例を表し、 x_{Target}^p は各訓練事例の教師信号であり、 $\{0, 1\}$ の変数である。 tr_num は全ての訓練事例の数である。 E_{sim} は定数である。

$$f(x^p) = \begin{cases} 1 & \text{if } \sum_{i=1}^k |w_i x_i^p - \theta| \geq E_{sim} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$g(x^p) = \begin{cases} 1 & \text{if } f(x^p) = x_{Target}^p \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$Affinity = \sum_{p=1}^{tr_num} g(x^p) \quad (5)$$

4.2 免疫学的記憶細胞

CSAIM における記憶細胞の生成方法について述べる。 Fig. 10 はシステムのフローチャートであり、 Fig. 11 は生成された抗体のクラスタリング結果から記憶細胞を生成する過程を示している。 CSAIM では、クローン選択アルゴリズムにより抗体を生成した後、 Fig. 11 に示すように、生成された抗体をしきい値である θ を基準に $IM(1 \leq q \leq IM)$ 個のクラスタに分類する。次に、各クラスタの中心の抗体が記憶細胞に保存される。ここで、記憶される抗体は、類似性の高い特定の訓練事例に対して、学習が行われる。学習はニューラルネットワークのパーセプトロンに基づいた学習である。記憶される抗体は、しきい値である θ_q を持ち、 θ_q に近づくように、抗体の重みが学習される。これにより、特定の訓練事例に対する類似性が高くなる。学習後に誤答する訓練事例が多い場合は、誤答する訓練事例に適応するために、新しく抗体が HM, RE により生成される。その結果、誤答する訓練事例が減少すれば、生成した抗体が記憶細胞に保存される。

類似性の高い訓練事例は、次のように求められる。訓練事例を $\vec{d} = (d_1, \dots, d_i, \dots, d_k)$ 、抗体を $\vec{h} = (h_1, \dots, h_i, \dots, h_k)$ とした時、 \vec{d} と \vec{h} のユークリッド距離が、あらかじめ決められた距離 μ_θ 以下であれば、一致したと判定するアルゴリズムである。ここで、 \vec{d} の要素 d_i は次の式により求められる。

$$d'_i = d_i \times \frac{h_j}{d_j} (d_i \neq 0 \wedge h_i \neq 0) \quad (6)$$

ここで類似性の高い訓練事例に対して、パーセプトロンに基づいた学習¹³⁾が行われ、学習後、二乗和誤差が期待値よりも小さければ、学習した抗体が記憶細胞に保存される。

5 実験

本節では、提案した特徴点抽出方法を用いて画像から特徴点を抽出し、レーベンシュタイン距離により類似性を測定し、比較のためにソフトコンピューティングの手法を用いて分類精度を測定する実験を行う。実験では、 Fig. 13 に示すような宮島の鳥居 8 枚と原爆ドーム 8 枚の計 16 枚の画像を用いる。各画像のサイズは 320×240 で、画像は Fig. 14 に示すように、事前にエッジ検出が施されている。次に、これらの画像に対

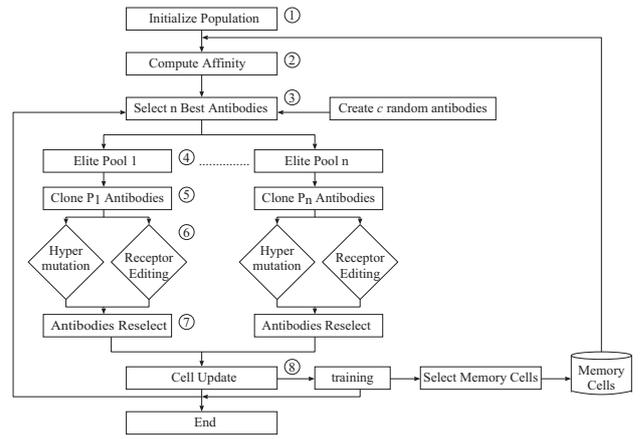


Fig. 10: A flow of CSAIM model

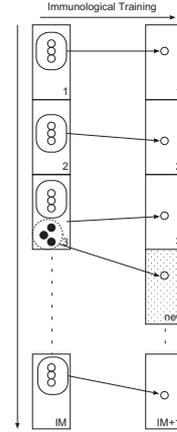


Fig. 11: A clustering method of memory cells

して 3.2 節で述べた提案手法により特徴点を抽出する。提案手法の分割数を決めるパラメータは、 $n = 4$ とし、画像を R1 から R4 の 4 つに分割する。また、分割した各扇形の角度を決めるパラメータ ψ は 0 と $\frac{\pi}{4}$ の 2 通りを試す。つまり、提案手法では、各画像に対して 8 通りのデータを抽出する。ここで、提案手法との比較を行うため、単純に全ての画素を用いる方法として、次の 2 種類の方法を用いた。

Fig. 12 は画像サイズが 5×5 の場合における画素の取り方を示しており、1 つは縦方向 (Fig. 12(a)) に、1 つは横方向 (Fig. 12(b)) に画素を抽出する。ここでは、全ての画素を用いると、データサイズが大きすぎるため、文献²⁾のように、ウェーブレット変換により画像サイズを $\frac{1}{4}$ に圧縮したデータを用いた。

これらの 3 つの手法によって得られた各画像のデータに対して SOM, インタラクティブ GHSOM, CSAIM による分類を行う。16 枚の画像の内 12 枚を訓練用データ (Fig. 13(c) から Fig. 13(h) と Fig. 13(k) から Fig. 13(p)) として使い、4 枚をテスト用データ (Fig. 13(a), Fig. 13(b), Fig. 13(i), Fig. 13(j)) として用いる。10 回の試行を行い、二乗和誤差の平均と標準偏差を求めた。SOM については、SOM.PACK²³⁾ を使い、SOM のマップサイズは 7×6 とした。インタラクティブ GHSOM のパラメータは、 $\tau_1 = 0.7$, $\tau_2 = 0.3$, $\alpha = 0.03$, $\beta = 6$ とし、CSAIM のパラメータは、 $G_{max} = 100$, $m = 150$, $n = 100$, $Q = 50$, $E_{sim} = 0.01$, $t = 10$,

Table 1: Sum of squared error

	SOM			GHSOM			CSAIM		
	提案手法	縦方向	横方向	提案手法	縦方向	横方向	提案手法	縦方向	横方向
平均 (鳥居)	3.23E-05 (1.47E+03)	1.39E+03 (7.18E+03)	1.35E+03 (7.03E+03)	4.32E+00 (1.02E+01)	2.99E+01 (5.00E+01)	2.88E+01 (5.00E+01)	3.08E-05 (5.76E+00)	4.16E+02 (1.20E+03)	3.98E+02 (1.00E+03)
標準偏差 (鳥居)	1.31E-05 (4.37E+02)	2.15E+03 (1.41E+03)	2.08E+03 (1.50E+03)	1.78E-01 (3.77E-02)	3.47E+00 (1.52E+01)	4.43E+00 (1.51E+01)	7.31E-06 (1.72E+00)	2.95E+02 (1.05E+03)	1.65E+02 (3.73E+00)
平均 (原爆ドーム)	5.33E-05 (3.78E+03)	1.73E+03 (9.73E+03)	1.72E+03 (8.87E+03)	2.62E+00 (1.56E+01)	4.95E+00 (7.07E+01)	4.24E+00 (7.04E+01)	8.47E-06 (4.35E+00)	2.82E+02 (7.41E+02)	3.96E+02 (1.00E+03)
標準偏差 (原爆ドーム)	6.60E-06 (1.18E+02)	2.68E+03 (7.32E+01)	2.67E+03 (5.39E+02)	2.41E-01 (9.41E-01)	3.58E+00 (1.47E+01)	3.70E+00 (1.45E+01)	6.31E-06 (6.83E-01)	2.78E+02 (4.05E+02)	1.68E+02 (3.73E+00)

$c = 10$, $\eta = 0.1$, $\mu_\theta = 0.3$, $t_{IM} = 50$, $E_{min} = 0.001$, $c_{max}^{memory} = \frac{1}{2}n$ とした。

Table 1 は、実験の結果、各手法によって得られた二乗和誤差の平均と標準偏差を示している。括弧の値はテスト時の値を示している。全体的に、縦方向、横方向の手法よりも、提案手法の方が二乗和誤差と標準偏差を小さくすることができた。また、縦方向、横方向の手法にあまり大きな違いは見られなかった。

提案手法において、SOM、インタラクティブ GHSOM、CSAIM を比較したところ、SOM では、訓練画像に対する二乗和誤差は小さくできたものの、テスト画像に対する二乗和誤差が大きくなり、3つの手法の中で最も悪い結果となった。これは、SOM における勝者ノードからの距離が大きくなってしまったことが原因である。GHSOM についても同様に、SOM ほど悪くないものの、テスト画像に対する二乗和誤差が大きくなった。これに対して、CSAIM では、訓練画像だけでなく、テスト画像に対する二乗和誤差も小さくでき、3つの手法の中で、最も優れた結果となった。教師なし学習である SOM やインタラクティブ GHSOM は、データを大まかにクラスタリングする能力は高いが、教師あり学習である CSAIM のように、未知なデータに対して推定する能力は低いため、このような結果になったのだと考える。

一方、CSAIM を用いても、ウェーブレット変換した縦方向、横方向のデータに対して二乗和誤差を小さくすることができなかった。これはデータサイズが大きくなりすぎたことが原因だと考える。Table 2 は、提案手法、縦方向、横方向の手法において、各画像から抽出している画素数を示している。括弧の値は、全体の画素数 ($320 \times 240 = 76800$) を 1 とした時の割合を示している。また、CSAIM にかかる計算時間を比較したところ、縦方向では 21894.7 秒、横方向では 22431.4 秒であったが、提案手法では 816.5 秒となり、計算時間を $\frac{1}{27.47}$ に抑えることができた。

Table 3 は、Table 1 の CSAIM における詳細で、各画像における二乗和誤差を示している。提案手法では、各事例に対して最も二乗和誤差を小さくできた位置 (Fig. 7 に示す ψ と R1 から R4 のいずれか) が選択される。鳥居の画像に対しては、R3 や R4 が選択される傾向があった。しかし、Fig. 13(e) や Fig. 13(f) では R2 が選択された。Fig. 13(e) や Fig. 13(f) の鳥居の画像には、R2 に情報量が多いからだと考えられる。また、原爆ドームの画像に対しては、R2 や R3 が選択される傾向があった。原爆ドームの画像は、画像によっては右上や左上

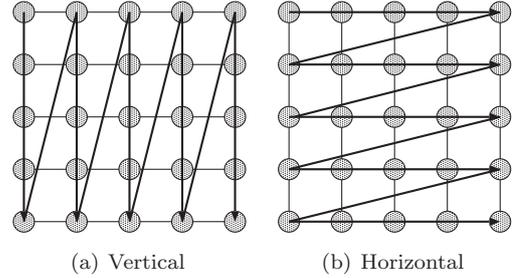


Fig. 12: Vertical and Horizontal methods

Table 2: The number of used pixels

	提案手法	縦方向, 横方向
円 A の画素数	816(0.011)	-
円 B の画素数	1880(0.024)	-
円 C の画素数	2280(0.030)	-
使用する画素数	4976(0.065)	19200(0.250)

に木が写っており、ノイズとなってしまう。そのため、右上や左上には共通した特徴が少なく、共通した特徴が多い R2 や R3 が選択される傾向にあったのだと考えられる。

このように提案手法を用いることで、画像から抽出する情報量を削減し、かつ、各画像において最適な位置から情報を抽出することができた。

6 おわりに

本研究では、これまでに、ひろしま観光マップにより収集された位置情報、5段階評価値、コメント文について、魅力的な情報や新規性のある情報を抽出するフィルタリングルールを獲得していたが、写真の分析は行っていなかったため、写真に関するフィルタリングルールは獲得していなかった。本論文では、写真の分析に着目し、データ分析における情報量を削減するために特徴点を抽出する方法を提案し、ランドマークとなる画像に対する類似性をレーベンシュタイン距離により測定した。

広島県の世界遺産として知られている厳島神社の鳥居と原爆ドームの画像から特徴点を抽出し、SOM、インタラクティブ GHSOM、CSAIM を用いて分類したところ、提案手法はウェーブレット変換した縦方向、横方向の手法よりも分類精度が高く、計算量を小さくすることができた。また、提案手法では、画像をいくつ

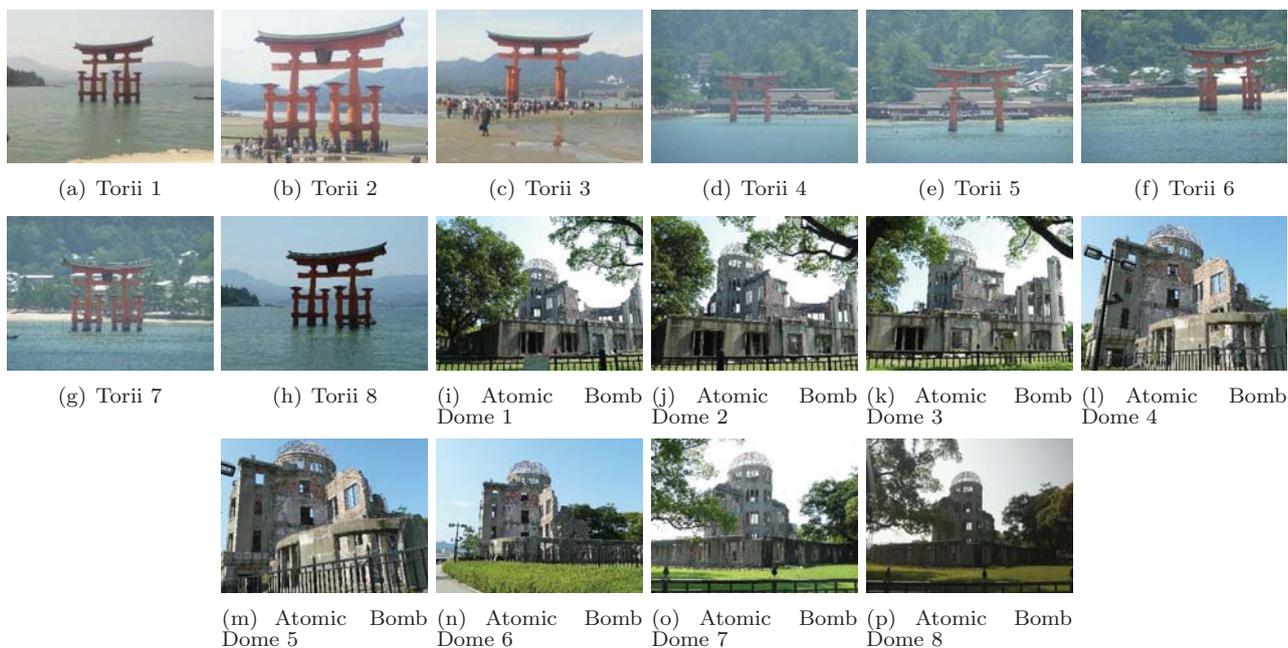


Fig. 13: Experimental images

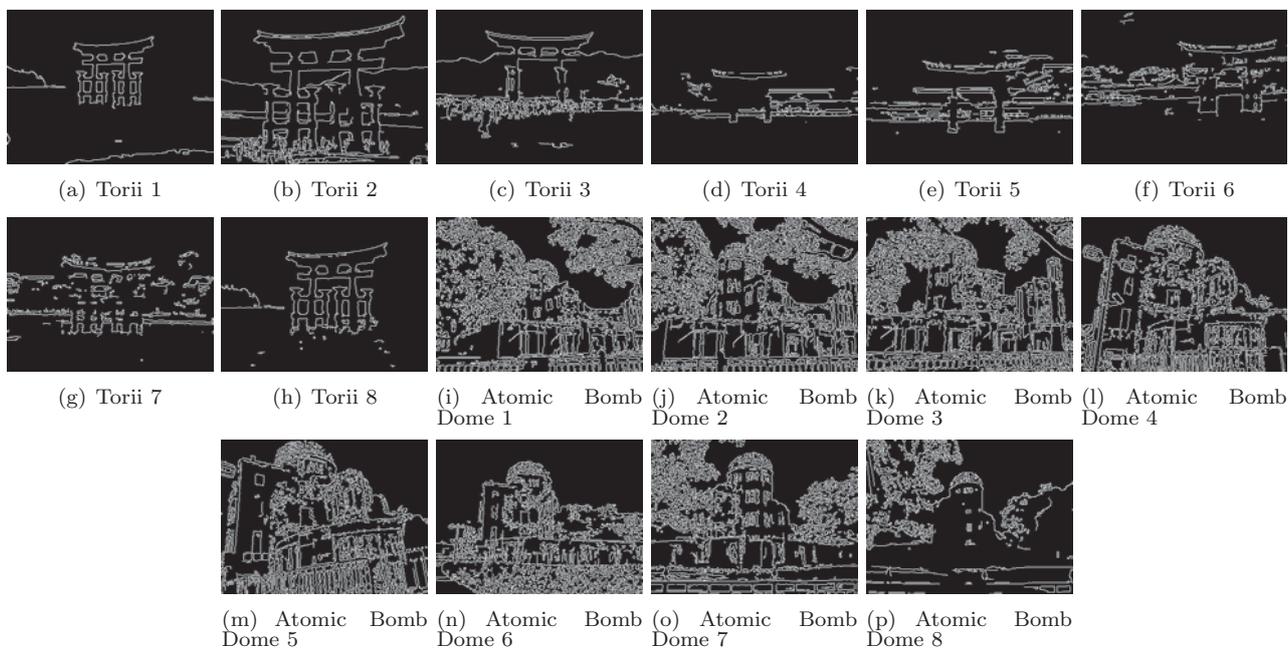


Fig. 14: Experimental images(edge detected)

かの領域に分割し、各画像において最適な領域から特徴点を抽出することができる。抽出された特徴点の領域を調査してみると、画像において情報量が多い領域、また、共通した特徴が多い領域が選択される傾向があることが分かった。また、教師なし学習である SOM やインタラクティブ GHSOM は、クラスタリングには適した手法ではあるが、未知のデータの推定には適していない手法であるため、テスト事例に対して精度の高い分類はできなかった。一方、教師あり学習である CSAIM では、テスト事例に対しても精度の高い分類ができ、比較した 3 つの手法の中で最も優れた手法となった。

今後は、写真の分類結果をフィルタリングルールに

加え、撮影された写真が、代表的な観光地であるかどうかを判別するフィルタリングルールを作成する。また、新規投稿情報から、フィルタリングルールを自動で更新するシステムを開発する。

謝辞

本研究は JSPS 科研費 25330366 の助成を受けたものである。

参考文献

- 1) OpenCV: <http://opencv.org/> (2013/10/31 閲覧)
- 2) 須藤一郎, 湯野智己, 横川昌俊, 工藤健慈, 関根優年: SOM クラスタ中心を用いた画像の低次元表現方法, 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解 Vol.106, No.469, 61/66, (2007)

Table 3: Sum of squared error for each sample (CSAIM)

	CSAIM		
	提案手法	縦方向	横方向
鳥居 1	(4.55E+00) $\psi = 0, R4$ (左上)	(4.55E+02)	(1.00E+03)
鳥居 2	(6.97E+00) $\psi = \frac{\pi}{4}, R4$ (左)	(1.94E+03)	(9.97E+02)
鳥居 3	2.97E-05 $\psi = 0, R3$ (左下)	4.66E+02	3.34E+02
鳥居 4	2.50E-05 $\psi = 0, R4$ (左上)	5.01E+02	3.53E+02
鳥居 5	2.71E-05 $\psi = \frac{\pi}{4}, R2$ (右)	6.26E+02	3.87E+02
鳥居 6	3.77E-05 $\psi = \frac{\pi}{4}, R2$ (右)	7.77E+02	4.53E+02
鳥居 7	4.16E-05 $\psi = \frac{\pi}{4}, R1$ (上)	4.39E+01	6.79E+02
鳥居 8	2.34E-05 $\psi = 0, R3$ (左下)	8.00E+01	1.80E+02
平均	3.08E-05 (5.76E+00)	4.16E+02 (1.20E+03)	3.98E+02 (1.00E+03)
標準偏差	7.31E-06 (1.72E+00)	2.95E+02 (1.05E+03)	1.65E+02 (3.73E+00)
原爆ドーム 1	(4.84E+00) $\psi = \frac{\pi}{4}, R3$ (下)	(4.55E+02)	(1.00E+03)
原爆ドーム 2	(3.87E+00) $\psi = \frac{\pi}{4}, R1$ (上)	(1.03E+03)	(9.97E+02)
原爆ドーム 3	2.38E-06 $\psi = 0, R2$ (右下)	4.66E+02	3.34E+02
原爆ドーム 4	9.01E-07 $\psi = \frac{\pi}{4}, R4$ (左)	5.01E+02	3.53E+02
原爆ドーム 5	9.65E-06 $\psi = 0, R3$ (左下)	6.23E+02	3.87E+02
原爆ドーム 6	6.91E-06 $\psi = 0, R3$ (左下)	2.29E+01	4.53E+02
原爆ドーム 7	1.66E-05 $\psi = 0, R2$ (右下)	3.50E-05	6.79E+02
原爆ドーム 8	1.44E-05 $\psi = \frac{\pi}{4}, R2$ (右)	8.00E+01	1.69E+02
平均	8.47E-06 (4.35E+00)	2.82E+02 (7.41E+02)	3.96E+02 (1.00E+03)
標準偏差	6.31E-06 (6.83E-01)	2.78E+02 (4.05E+02)	1.68E+02 (3.73E+00)

- 3) 岡崎ゆきの, 鎌田真, 市村匠: 自己組織化マップによる分類のための写真のデータ表現について, 2013 IEEE SMC Hiroshima Chapter Young Researchers Workshop, 25/28 (2013)
- 4) N.D.Lane, E.Miluzzo, L.Hong, D.Peebles, T.Choudhury, and A.T.Campbell: *A survey of mobile phone sensing*, IEEE Communications Magazine, Vol.48, No.9, 140/150 (2010)
- 5) ITProducts: ひろしま観光マップ, <https://play.google.com/store/apps/details?id=jp.itproducts.KankouMap> (2013/10/31 閲覧)
- 6) 鎌田真, 市村匠: Android スマートフォンを用いた広島県観光情報収集システムの開発, 2011 IEEE SMC

- Hiroshima Chapter Young Researchers Workshop, 21/24 (2011)
- 7) 市村匠: スマートフォンベースユーザ参加型主観的情報データ分析システム, 日本オペレーションズ・リサーチ学会誌, Vol.58, No.5, 275/280 (2013)
- 8) T.Ichimura and T.Yamaguchi: A Proposal of Interactive Growing Hierarchical SOM, Proc. of 2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011), 3149/3154 (2011)
- 9) J.R.Quinlan: *Improved use of continuous attributes in c4.5*, Journal of Artificial Intelligence Research, No.4, 77/90 (1996)
- 10) T.Ichimura, S.Kamada, and K.Kato: *Knowledge Discovery of Tourist Subjective Data in Smartphone Based Participatory Sensing System by Interactive Growing Hierarchical SOM and C4.5*, Intl. J. Knowledge and Web Intelligence, Vol.3, No.2, 110/129 (2012)
- 11) T.Ichimura and S.Kamada: *A Generation Method of Filtering Rules of Twitter Via Smartphone Based Participatory Sensing System for Tourist by Interactive GHOM and C4.5*, 2012 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2012), 110/115 (2012)
- 12) 市村匠, 鎌田真: スマートフォンベースユーザ参加型主観的情報システムからのソーシャルコミュニティへの情報発信方法について, 2013 IEEE SMC Hiroshima Chapter Young Researchers Workshop, 53/54 (2013)
- 13) T.Ichimura and S.Kamada: *Clustering and Retrieval Method of Immunological Memory Cell in Clonal Selection Algorithm*, Proc. of The 6th International conference on Soft Computing and Intelligent Systems and The 13th International Symposium on Advanced Intelligent Systems(SCIS-ISIS 2012), 1351/1356 (2012)
- 14) D.Michel, A.A.Argyros, and M.I.A.Lourakis: *Horizon matching for localizing unordered panoramic images*, Journal Computer Vision and Image Understanding, Vol.114, No.2, 274/285 (2010)
- 15) V.I.Levenshtein: *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Physics - Doklady, Vol.10, No.8, 707/710 (1996)
- 16) A.Rauber, D.Merkl, and M.Dittenbach: *The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data*, IEEE Transactions on Neural Networks, vol.13, 1331/1341 (2002)
- 17) T.Kohonen: *Self-Organizing Maps*, Springer Series in Information Sciences, Vol.30, Springer (1995)
- 18) 市村匠, 神田優, 原章, 高濱徹行: 階層抑制型 Growing-Hierarchical SOM による分類手法の提案, 第 37 回フアイ・ワークショップ, 1/4, (2010)
- 19) R.G.Kuehni: *The early development of the Munsell system*, Color Research and Application, Vol.27, No.1, 20/27 (2002)
- 20) H.C.Wu, R.W.P.Luk, K.F.Wong, and K.L.Kwok: *Interpreting TF-IDF term weights as making relevance decisions*, ACM Transactions on Information Systems, Vol.26, No.3, Article No.13 (2008)
- 21) F.M.Burnet: *The Clonal Selection Theory of Acquired Immunity*, Cambridge, U.K.: Cambridge Univ. Press (1959)
- 22) S.Gao, H.Dai, G.Yang, and Z.Tang: *A novel clonal selection algorithm and its application to travelling salesman problem*, IEICE Trans. Fundamentals, Vol.E90-A, 2318/2325 (2007)
- 23) SOM_PAK: http://www.cis.hut.fi/research/som_lvq-pak.shtml (2013/11/1 閲覧)

報酬の解釈を導入したマルチエージェント強化学習システムにおける協調行動の獲得

○椿本樹矢 小林邦和 (愛知県立大学)

Cooperative Action Acquisition Introducing Interpretation of Reward in a Multi-agent Reinforcement Learning System

*T. Tsubakimoto and K. Kobayashi (Aichi Prefectural University)

Abstract— In this paper, we propose a method that can acquire cooperative action to reach an appropriate goal without controlling reward to realize cooperative action by designers. We assume every action of other agents as a process to achieve an ultimate goal and then treat it unimportant. In order to confirm the effectiveness of the proposed method, we carried out computer simulations. The simulation results show that the proposed method is superior to a standard Q-learning method and a Q-learning method with cooperation in terms of the number of successful cooperation.

Key Words: Multi-agent, Reinforcement Learning, Cooperation

1 はじめに

現在のコンピュータやロボットを代表とする機械は、人が設計した通りの行動のみを実行している。つまり、機械自らが考えて行動を行うということは困難である。そのため、柔軟性に乏しく、多少の環境の変化で、予期した通りの結果を得られないことが多い。それらを解消するため、自律的な行動を獲得する手法として強化学習が提案されている¹⁾²⁾。強化学習は、学習主体(エージェント、ロボットなど)が自ら環境情報を知覚し、実行可能な行動群から行動を選択・実行し、その結果得られた報酬・罰を通して、自律的に最適な行動系列を獲得する手法である。強化学習の代表的な手法にQ学習がある。Q学習は元々シングルエージェント環境を想定した手法であるが、これまでにマルチエージェント環境へも導入が行われている³⁾⁴⁾⁵⁾。

本研究では、このQ学習を用いたマルチエージェント環境下での協調行動の獲得を目指す。これまで、協調動作は、他エージェントの行動を模倣することで協調を行うものや、エージェント間で通信を行い、協調を行う方法が提案されている。模倣行動では、対応できる環境が少なく、また通信を利用する場合は、雑音が多い、他の通信の妨害をしてはいけないなどの環境では、通信が行えず、協調を行うことが困難となる。

先行研究における通信なしでの協調動作は、ある程度実現されている。通常のQ学習では、設計者がゴール報酬を意図的に変化させることで協調動作の獲得を行うことができる。また、他エージェントの内部モデルを知覚せず、行動のみの観測で協調を行えるモデルとして長行らの政策推定法が提案されている⁶⁾。政策推定法は、他エージェントの行動一つひとつを予測し、それに合わせて自らの行動を決定するといったアイデアに基づいている。しかし、政策推定法は、すべての他エージェントの行動履歴を記憶する必要があり、他エージェントが数多く存在する環境では、大容量の記憶領域が必要となるという問題がある。

本論文では、他エージェントの行動一つひとつは最終的な目標を果たすためのプロセスであると考え、重要視しない。重要であるのは他エージェントの目標で

あり、それを推定することで自らの協調的な行動を獲得していく。そのため、必要なのは少量の記憶容量である。また、設計者が意図的に協調を行えるよう報酬を操作しなくても、自ら考え、適切なゴールへ向かうための協調的な行動を獲得できるような手法を提案する。

次章以降、2章では、本手法で使用する強化学習の代表的な手法であるQ学習を紹介し、3章では、Q学習を用いてマルチエージェント環境下における協調行動の獲得を実現した政策推定法について解説する。また、4章では提案手法である意図推定法を用いた強化学習について説明し、5章において評価シミュレーションを行い、その結果の考察を行う。

2 Q学習

強化学習は学習主体(エージェント、ロボットなど)が行動を起こし、その結果得られた報酬・罰を通して、自律的に最適な行動系列を獲得する手法である。

その代表的な手法であるQ学習は1989年にWatkinsによって提案された⁷⁾。Q学習ではQ値と呼ばれる、状態と行動の対である価値関数 Q (状態, 行動)を用い、環境から与えられる報酬をもとに、現在実行した行動状態に対応するQ値を更新することで、より多くの報酬が期待できる行動系列の学習を進行させる。Q学習における学習の流れは以下である。

0. エージェントはQ値の初期化を行う。
1. 時刻 t において、エージェントは現状態 $s_t \in S$ を観測する。
2. エージェントは、Q値から算出される行動政策 $\pi(s_t, a)$ に基づいて行動を選択する。政策 $\pi(s, a)$ は状態 s で行動群 A の中から行動 a を選択する確率を表している。本論文では、政策として代表的なSoft-Max行動選択法を採用する。Soft-Max行動選択法は式(1)で表される。

$$\pi(s, a) = \frac{\exp(Q(s, a)/T)}{\sum_{b \in A} \exp(Q(s, b)/T)} \quad (1)$$

ここで、 T は温度定数であり、行動選択のランダム性を調整するためのパラメータである。学習初

期は大きな値を取り、ランダム性が高い。学習が進行するにつれて値を減少させ、ランダム性を減らしていくことで学習を収束させていく。

3. エージェントは手続き 2 で選択した行動 a を実行する。それにより、環境状態は s_t から s_{t+1} へと移行する。
4. 手続き 3 で行った行動の結果として、エージェントは環境から報酬 r_{t+1} を得る。
5. エージェントは手続き 4 で得られた報酬 r_{t+1} を用いて、状態 s_t に対応する Q 値を式 (2) によって更新する。

$$Q(s_t, a) \leftarrow Q(s_t, a) + \alpha[r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a)] \quad (2)$$

ここで、 α は学習率、 γ は割引率である。

6. 学習の終了条件を満たしていれば学習終了。そうでなければ、 t に 1 を加え手続き 1 に戻る。

3 政策推定法

政策推定法は他エージェントの内部モデルは知覚せず、実行した行動のみを観測、記憶する。それを元に、他エージェントの行動一つひとつ (政策) を予測し、それに合わせて自らの行動を協調的なものにしていく手法である⁶⁾。

自エージェント k が推定した他エージェント o の政策を $I^k(s, a^o)$ とし、他エージェントが状態 s で行動 a^o を実行する確率を表すものとする。

時刻 t において、他エージェントが状態 s_t で行動 a_t^o を実行したとする。そのとき、状態 s_t で実行可能な、全ての行動 $a^o \in A^o$ に対して、式 (3) を用いて I^k を更新する。

$$I^k(s, a^o) \leftarrow (1 - \phi) I^k(s, a^o) + \begin{cases} \phi & (a^o = a_t^o) \\ 0 & (\text{otherwise}) \end{cases} \quad (3)$$

ここで、 ϕ は観測した行動を将来の行動予測時にどれくらい考慮するかを決定するパラメータである。

他エージェント o の行動予測である $I(s_t, a^o)$ を用いて、自エージェント k は現在時刻 t での行動を式 (4) で示す $\bar{Q}^k(s_t, a^k)$ から算出される政策 $\phi^k(s_t, a^k)$ に従って行動を確率的に選択する。

$$\bar{Q}^k(s_t, a^k) = \sum_{a^o \in A^o} I^k(s_t, a^o) Q^k(s_t, a^k, a^o) \quad (4)$$

次に、自エージェント k は選択した行動 a_t^k を実行する。それと同時に他エージェントも行動を起こし、状態 s_t が次状態 s_{t+1} へと遷移する。それにより、自エージェント k は環境から報酬 r_{t+1}^k を受け取り、Q 値を式 (5) に従って更新し、状態 s_t における関数 I^k を式 (3) によって更新する。

$$Q(s_t, a_t^k, a_t^o) \leftarrow (1 - \alpha) Q(s_t, a_t^k, a_t^o) + \alpha(r_{t+1}^k + \gamma_k \max_{a^k} \bar{Q}^k(s_{t+1}, a^k)) \quad (5)$$

エージェントは時刻 t を進めながら上述の手順を繰り返し、学習条件を満たすまで試行を繰り返すことで協調行動を獲得していく。

4 意図推定法を用いた強化学習

本節では、本論文で提案する意図推定法を示す。この意図推定法で重要な概念が「報酬の解釈」である。前節で説明した Q 学習や政策推定法は、環境から与えられる報酬をそのまま受け取っているが、本手法では環境から与えられる報酬に自らの解釈を加える。この「報酬の解釈」を行うことで協調的な行動を獲得していく。報酬の解釈は各ゴールに対応づけられた目標価値を用いて行われる。ゴールの価値を表す目標価値は、他エージェントがどのゴールに到達しやすいかの到達予想である優先度によって決定される。

4.1 他エージェントの意図の推定

まず、自エージェントは時刻 t において他エージェント k の行動を観測し、図 1 に示すゴール g とのなす角度 θ 、距離 d を用いて k の g への到達しやすさである優先度 GP (Goal Priority) を推定する。 $GP_{t,g,k}$ の計算式を式 (6) に示す。

$$GP_{t,g,k} = \frac{1}{\alpha + \beta + 1} (\alpha \cdot Dir_{t,g,k} + \beta \cdot Dist_{t,g,k} + Dir_{t,g,k} \cdot Dist_{t,g,k}) \quad (6)$$

$$\begin{cases} Dir_{t,g,k} = \frac{\cos\theta + 1}{2}, & 0 \leq Dir_{t,g,k} \leq 1 \\ Dist_{t,g,k} = \frac{1}{d}, & 0 \leq Dist_{t,g,k} \leq 1 \\ 0 \leq GP_{t,g,k} \leq 1 \end{cases} \quad (7)$$

ここで、 α, β は角度・距離をどの程度重要であると考えられるかの割合である。特に、どちらが重要ということがなければ、通常はともに 1.0 でよい。

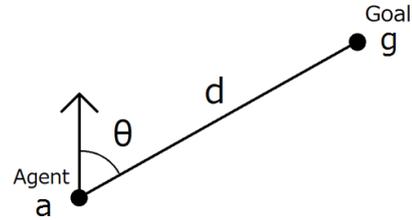


Fig. 1: Positional relation between an agent and a goal

次に、算出した優先度をもとに、学習率 η を用いて自エージェントが考える各ゴールの価値である目標価値 $GV_{t,g}$ (Goal Value) の更新を行う。

$$GV_{t,g} = (1 - \eta) GV_{t-1,g} + \eta \cdot e(\vec{GP}_{t,g}) \quad (8)$$

$$0 \leq GV_{t,g} \leq 1$$

$$\vec{GP}_{t,g} = (GP_{t,g,1}, GP_{t,g,2}, \dots, GP_{t,g,n}) \quad (9)$$

ここで、 n は他エージェントの数、 $e()$ は任意の評価関数であり、エージェントが行うべき協調に合わせた関数を設定する。また、 $\vec{GP}_{t,g}$ は自分以外の全ての $GP_{t,g,k}$ をもつベクトルである。この目標価値 $GV_{t,g}$ は他エージェントの行動から推定した意図を反映したものとなっている。

4.2 報酬の解釈

報酬の解釈とは、環境から与えられる報酬にエージェント自らの主観を交え、受け取る量を調節する手法である。時刻 t に自エージェントがゴール g へ到達した時、環境から報酬 r_g が与えられると同時に、毎ステップ更新を行ってきた目標値を用いて、報酬の解釈を式 (10) で行う。

$$r_t \leftarrow r_t \times GV_{t,g} \quad (10)$$

その後、解釈された報酬が Q 値の更新式に取り入れられ、協調行動を獲得していく。

4.3 提案システムの流れ

提案システムでは、各エージェントが以下に従って自律的に学習を行う。

1. 現在時刻を t とする。自エージェントは他エージェント全員の意図を推定するため、優先度 $GP_{t,g,k}$ を全ての g, k の組み合わせについて計算し、同様に目標値 $GV_{t,g}$ も全ての g について更新する。 G は全てのゴールの集合、 A は他エージェント全体の集合である。
2. 自エージェントは、政策 π によって行動 a を選択し、行動を行う。
3. 行動 a によって状態 s_t が次状態 s_{t+1} へと変化する。この時、自エージェントがゴールに到達していれば報酬の解釈を行ってから、それ以外ではそのまま、手続き 4 へと進む。
4. 得られた報酬、次状態 s_{t+1} における Q 値を用いて、 $Q(s, a)$ を更新する。学習の終了条件を満たしていれば学習終了。そうでなければ、 t に 1 を加え手続き 1 に戻る。

以上の学習法を各エージェントがそれぞれ自律的に行う。

5 評価シミュレーション

5.1 問題設定

- 2次元 (20 × 20) グリッド空間中に、2体のエージェントと2つのゴールが存在する環境を用いる。周囲は壁に囲まれており、エージェントとゴールはランダムに配置される (図2)。
- エージェントはステップごとに隣接する上下左右のグリッドへ移動する。エージェントが実行可能な行動は上下左右の4種類である。
- エージェントは、他エージェントの位置、行動のみを完全に知覚できる。
- 初期配置からすべてのエージェントがゴールに到達するまでを1エピソードとする。
- 300ステップまでに全てのエージェントがゴールに到達できなかった場合、現在の学習を打ち切り、新たなエピソードを開始する。

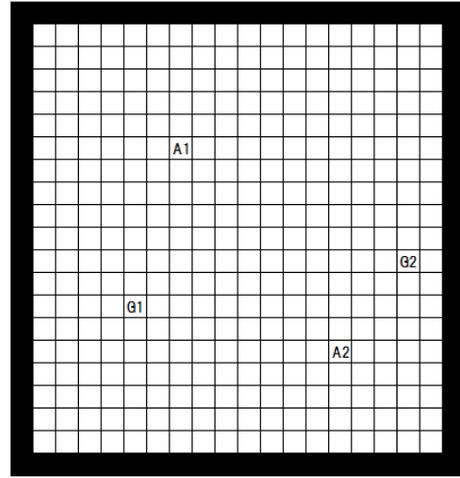


Fig. 2: A simulation field

- ゴールに到達したエージェントは、行動を終了し、その場で完全に停止する。全てのエージェントがゴールに到達するとエピソード終了とし、初期状態に戻る。
- 1000エピソードが終了した時点で、学習終了とし、それを1試行とする。
- 全てのエージェントが同じゴールに到達するような行動系列を獲得することを目的とする。
- エージェントが環境から受け取る報酬は、ゴール到達時 20.0、壁に衝突したとき 0.1、それ以外は 0.01 とする。

5.2 パラメータ設定

Q学習の学習率 0.1, Q学習の割引率 0.9, Soft-Max法の温度定数 T は、シグモイド関数を応用したスケジューリング法を用いる。

$$T = \frac{T_{max}}{1 - e^{a(t+\phi)}}$$

ここで、 T_{max} は最高温度、 a はゲイン、 t は時間 (エピソード)、 ϕ は初期位相である。

シミュレーションでは、これらの定数は $T = 20.0$, $a = 0.01$, $\phi = 500$ と設定した。

また、提案手法では計算式にゴールの位置情報が、必要な部分がある。それは、ゴール位置を観測して得るのではなく、エピソード中、偶然ゴールに到達した時に位置を記憶するという方法で取得する。

提案手法の任意の関数 $e()$ であるが、今回は同じゴールに到達することが目的であるので、平均 $E[]$ とした。優先度 GP の式中の α , β は、距離、角度のどちらが特別重要ということはないため、ともに 1.0 と設定した。目標値 GV の学習率 η は 0.3 とした。

5.3 評価方法

提案手法の評価・比較のために、通常の Q学習と協調を用いた Q学習を用いる。ここで、協調を用いた Q学習とは、協調的な行動を行えた場合のみ報酬を与えるよう設計した Q学習のことである。

本シミュレーションでは、全てのエージェントが同じゴールに到達することを協調の目的とする。同じゴー

ルに到達したエピソードを成功エピソードとし、その回数を10エピソードごとにカウントする。よって成功回数の最大値は10である。成功エピソード数、つまり協調の成功回数を評価基準とし、エピソードと成功回数の推移を比較する。

5.4 シミュレーション結果

協調行動の獲得を目的としたマルチエージェント環境下における、各手法のシミュレーションによる結果の比較を図3に示す。このグラフは1000エピソードの学習をそれぞれ20回、つまり20試行を行った各試行での平均の結果である。

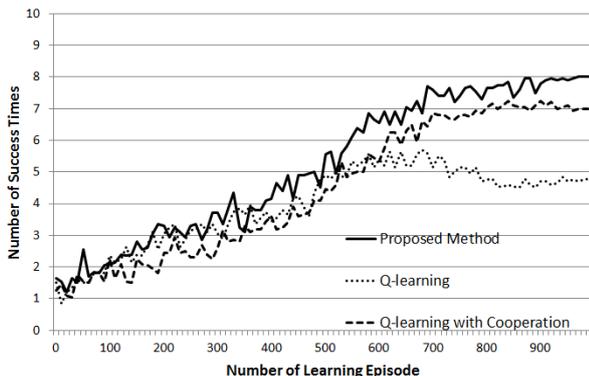


Fig. 3: Simulation results

5.5 考察

図3より、学習初期は提案手法、通常のQ学習、協調を用いたQ学習、ともに差異は見られないが、学習後期では通常のQ学習と比較して、提案手法、協調を用いたQ学習の成功回数が多いことがわかる。これは、通常のQ学習が協調要素をもたないことが原因であり、これにより提案手法と協調を用いたQ学習が協調行動の獲得に成功していることが確認できる。また、提案手法と協調を用いたQ学習では、10エピソード中1回程度ではあるが、成功回数に差が出ており、提案手法の方がより協調行動の獲得に成功していることがわかる。しかし、本シミュレーションでは行動選択のランダム性や初期配置の位置関係によって同手法を用いた場合でも結果に差異が生まれることがある。そのため、一概に協調を用いたQ学習より提案手法の方が協調行動獲得の面で優れているということとはできない。だが、その可能性は十分にあると考えられる。

また、提案手法で、成功回数の少ない試行の初期配置を参照すると、2種類の共通のパターンが確認できた。1つはゴール同士の距離が極めて近い場合(図4)である。この場合、エージェントは2つのゴールの価値をほぼ同等であると判断するため、自分との距離が近いゴールを選択しやすくなる。そのために、協調に失敗する試行が多かったものと考えられる。もう1つは図5のようにエージェント同士が離れており、エージェントと一方のゴールとの距離が極端に離れている場合である。この場合では、ほとんどのエピソードで自己と距離の近いゴールに到達するため、もう一方のゴールにあまり到達できず、Q値の更新が十分に行えなかったものと思われる。図2のように距離がほぼ等間隔な配置では、学習後期のほとんどのエピソードで

成功していた。

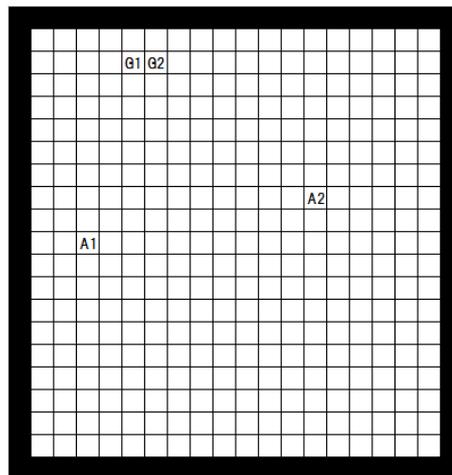


Fig. 4: A placement that two agents are hard to cooperate(case 1)

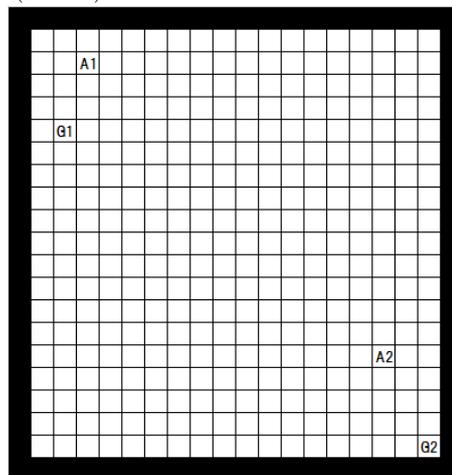


Fig. 5: A placement that two agents are hard to cooperate(case 2)

また、協調を用いたQ学習では、全てのエピソード・ステップで協調が成功したかどうかを判断し、環境から報酬を与えるかどうかの判断を行わなければならない。提案手法では、協調が成功したかどうかを判断する必要はなく、単純にゴール報酬を設定するだけでよい。そのため、設計者が協調を行えるようエージェントを誘導しなくても協調行動系列を獲得することができる。

5.6 まとめと今後の展開

本論文では、マルチエージェントシステムで学習によって自律的に協調行動を獲得する新たな手法を提案した。本手法では、先行研究である政策推定法他エージェントが数多く存在する環境では、大容量の記憶領域が必要となるという問題を解決している。

また、提案法の有効性を確認するため、エージェント、ゴールがともに複数存在する環境を用意し、協調的な行動を獲得することを目的とした計算機シミュレーションを行った。比較対象として、通常のQ学習と協調的な行動を行えた場合のみ報酬を与えるよう設計したQ学習を用いる。協調の成功回数を評価基準とし、エピソードと成功回数の推移を比較することで、提案

手法が設計者の誘導なしで協調行動の獲得に成功していることを示した。また、提案手法が通常の Q 学習と協調を用いた Q 学習より協調の成功回数の面で優れている可能性が高いことを確認した。

今後の展開として、本論文では他エージェントを完全知覚できる環境を想定しているが、知覚できる範囲を限定した場合でも推定が行えるように拡張すること、また複雑な環境であるチェスなどのゲームプレイを学習的に獲得することが考えられる。

謝辞

本研究の一部は日本学術振興会・科学研究費補助金(課題番号 23500181)の助成を受けている。

参考文献

- 1) R. S. Sutton and A. G. Barto (三上 貞芳, 皆川 雅章 共訳): 「強化学習」, 森北出版 (2000)
- 2) L. P. Kaelbling, M. L. Littman, and A. P. Moore: “Reinforcement Learning: A Survey”, *Journal of Artificial Intelligence Research*, Vol.4, pp.237-285 (1996)
- 3) P. Stone and M. Veloso: “Multiagent Systems: A Survey from a Machine Learning Perspective”, *Autonomous Robots*, Vol.8, No.3, pp.345-383 (2000)
- 4) 大内 東, 山本 雅人, 川村 秀憲: 「マルチエージェントシステムの基礎と応用—複雑工学の計算パラダイム—」, コロナ社 (2003)
- 5) 高玉 圭樹: 「マルチエージェント学習—相互作用の謎に迫る—」, コロナ社 (2003)
- 6) 長行 康男, 伊藤 実: 2 体エージェント確率ゲームにおける他エージェントの政策推定を利用した強化学習法, *電子情報通信学会論文誌, D-I Vol. J86-D-I No.11*, pp. 821-829 (2003)
- 7) C. J. C. H. Watkins and P. Dayan: “Q-learning”, *Machine Learning*, Vol.8, No.3-4, pp.279-292 (1992)

複数の餌場がある軍隊アリシミュレーションシステムにおけるエーเจントの挙動からの利他行動の分析

○上本拓也 (県立広島大学経営情報学部経営情報学科)
市村 匠 (県立広島大学経営情報学部経営情報学科)

An Analysis of Altruism Behavior for Agents in Army Ant Simulation System with Multiple Feeding Spots

*Takuya Uemoto (Faculty of Management and Information Systems,
Prefectural University of Hiroshima)
Takumi Ichimura (Faculty of Management and Information Systems,
Prefectural University of Hiroshima)

Abstract— Ant colony optimization (ACO: Ant Colony Optimization) is known as a metaheuristic optimization algorithm for search for an approximate solution in the combination problem such as a traveling salesman problem. In ACO, an ant search for the shortest path from a nest to food via pheromones. Such an algorithm is called Swarm Intelligence, where each agent performs only simple action and their actions lead to the emergence of collective intelligence. In this study, we focused on the army ant with an altruistic behavior instead of general ACO. An altruism behavior is an action which is selected to consider without his loss but to pursue the benefit of a group. When army ants cannot build the shortest path because of obstacle such as a ditch, they act to remove the obstacle by building a bridge with their bodies. We developed the army ant model to simulate the behavior in some environments and report the experimental results of their behaviors related to collective intelligence in multiple feeding spots.

Key Words: Swarm Intelligence, Army Ant Simulation Model, Altruism Behavior

1 はじめに

群知能とは、個々の知能レベルは高くないが、群れを形成し、群全体で1つの目標を達成するための高い能力を実現する現象をコンピュータを用いて模倣する知能技術の1つである¹⁾。その手法にはアリコロニー最適化 (Ant Colony Optimization; ACO) があり²⁾、巡回セールスマン問題 (Traveling Salesman Problem; TSP) のような組み合わせ最適化問題などを解くために用いられる³⁾。ACOはアリが餌までの経路を探索する挙動から考えられたものである。アリは複雑な行動を考えるような脳は持っておらず、至近距離でようやくわずかに認識出来る程度である。彼らは単純な思考のみを繰り返す、目的を達成するために高い能力を実現する。このために重要なものがフェロモンである。フェロモンとはアリのコミュニケーションツールであり、アリを引きつける効果がある。彼らは餌を発見すると揮発性の物質であるフェロモンをまいて巣に戻る。彼らはフェロモンをまくことで巣と餌の間にフェロモンの軌跡を形成し、フェロモンを介して他のアリに餌までの経路を伝える。フェロモンは時間とともに減少するため、時間が経つとアリが知覚できなくなる。巣と餌の間にある経路が短ければ、餌にたどり着く時間が短くなるため、その経路にあるフェロモンは減少するよりも早く補強される。他のアリは、フェロモンが多い方向へ移動することで、より短い経路を選択する。ACOは、このようにしてアリが巣から餌までの最短経路を形成するメカニズムをモデル化したものである。しかし、TSPへの適用に関して、存在している経路の中から最適経路を選択するため、すでに経路が存在している問題にしか適用できず、ACOが持つ群知能の性質を十分に発揮できていない。

一方、利他行動など、一般的なアリとは異なる挙動を行う軍隊アリが知られている。利他行動とは、他の仲間の利益のために自己を犠牲にする行動である。軍隊アリは穴や溝などの障害物を発見すると、自らの身体を用いて橋を形成し、目的を達成するための最短経路を探索する。しかし、軍隊アリが利他行動を行うメカニズムは生物学的にもまだ解明されていない。その軍隊アリの利他行動を、マルチエージェントシステム (Multi-Agent System; MAS) を用いて再現した研究がある⁴⁾⁵⁾。

MASとは、1つの課題を複数のエージェントの相互作用により解決する事を目的としたシステムである。文献⁴⁾では、軍隊アリが利他行動を行う条件として仮説がいくつか立てられ、実験が行われた。ここでは、知覚するフェロモン量と周囲のエージェント数について利他行動を行う条件が設定されていた。餌を発見したエージェントがフェロモンをまくことから、フェロモンを多く知覚できる場所は、現在もしくは将来に通る場所である可能性が高いことから必要であると考えられ、アリが利他行動を行う条件として、知覚するフェロモン量に関する条件が使用するのはではないかと考えられた。文献⁴⁾では、知覚するフェロモン量について条件を設定した結果、餌を巣へ運ぶために求められる場所のみ橋が形成された。しかし、この条件では餌を探索している時に橋が形成されなかった。一方で、周囲のエージェント数に関する条件では、文献⁴⁾によると、餌を探索している時に橋が形成され、餌を巣へ運ぶために求められる場所以外にも橋が形成された。そのため、橋の形成のような利他行動を行う条件として、現実世界のアリは周囲のエージェント数に関する条件を使用していると仮説をたてた。

文献⁵⁾では、軍隊アリ MAS を拡張することで、現実世界の軍隊アリに近い挙動を示している。従来の研究では、エージェントがまいたフェロモンをそのまま知覚するという条件が設定されていたが、文献⁵⁾では、フェロモンが揮発性の物質であることを考慮し、空中に蒸発した後拡散するものとした。エージェントは空中に蒸発したフェロモンを知覚することが可能になった。

また、現実世界の軍隊アリには生殖を行うアリや、狩りを行うアリ、利他行動を行うアリなど様々な役割分担がある⁶⁾。各エージェントは、個別の役割を持つことで、役割を持たない設定と比較して、解の探索効率が向上していた⁵⁾。

従来の研究では、エージェントが利他行動を行っているか観察するために、餌場は1つで実験が行われた。本研究では、複数の餌場がある軍隊アリ MAS を構築した。複数の餌場の大きさは同じではなく、様々な大きさを用いて実験を行うことにより、エージェントがどのように集まるかを観察した。また、エージェント数が異なる場合で、餌を探索する順番などに変化が見られるのではないかと考えた。その結果、エージェントの集団は1つの餌場の探索を終えると、近くにある餌場へ移動することがわかった。また、餌の大きさが違う実験では、餌が大きい餌場にエージェントが集まりやすい傾向が見られ、エージェント数が違う実験では、餌場を探索する順番に変化が見られたのでここに報告する。

2 軍隊アリ MAS

2.1 シミュレーション環境

軍隊アリ MAS を構築するために Swarm ライブラリを用いた⁸⁾。本論文では文献⁵⁾と同じ環境を Java 言語を用いて構築した。シミュレーションは Fig. 1 にあるように、 100×100 マスのフィールドで行う。フィールドの中央に巣 (Nest) を配置し、エージェント (Agent) は巣から出現する。巣に対して左側、右側、下側に幅が3である溝を配置した。エージェントは溝を渡ることができないが、エージェントが利他行動を行い、溝に橋 (Bridge) を形成することで他のエージェントは溝を渡ることが可能となる。1つのマスに存在できるエージェントは1個であるが、利他行動によって形成された橋を渡る時のみ、1つのマスにエージェントが2個存在する。エージェントがすべての餌 (Food) を巣に運ぶとシミュレーションは終了する。

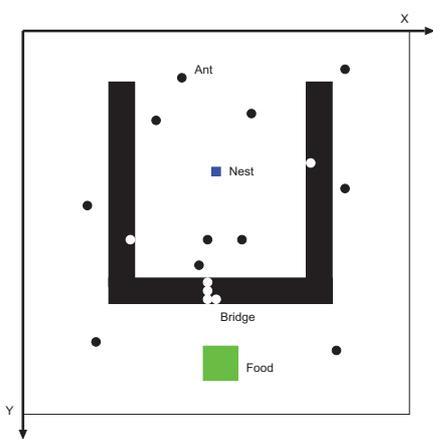


Fig. 1: Image of simulation

2.2 エージェント

軍隊アリは餌の探索活動の際に役割分担を行う。この役割には卵を生む雌アリや、巣を守る兵隊アリなど様々ある。軍隊アリ MAS では、餌の探索における役割分担を再現するために、メジャーエージェントおよびマイナーエージェントの2つのエージェントを定義した。

メジャーエージェントはランダムウォークでフィールド内を動き回り、フェロモンをまくことで他のエージェントを誘導し、広範囲にわたって探索活動を行わせる。餌を発見しても運ぶことはなく、利他行動も行わない。

一方、マイナーエージェントはランダムウォークで探索活動を行い、周囲1マスにフェロモンを感知すると、最もフェロモン量の多いマスに進む。マイナーエージェントには3つの状態があり、条件によって違う状態へ遷移する。その状態と遷移する条件は以下の通りである。

Search) ランダムウォークで餌の探索活動を行う。餌を発見すると Return 状態へ遷移する。後に説明する利他行動を行う条件を満たすと Altruism 状態へ遷移する。

Return) 餌を持っている状態で、動いている間はフェロモンをまき続ける。Return 状態のエージェントはフェロモンの影響を受けない。巣の場所は知っているものとし、通常は巣がある方向へ進む。進む方向に溝があり進めない場合は、溝を迂回する方向へ進む。巣に戻り、餌を離すと Search 状態になる。利他行動を行う条件を満たしても利他行動は行わない。

Altruism) 利他行動を行っている状態である。この状態では動くことはできない。後に説明する利他行動をやめる条件を満たすと Search 状態になる。

2.3 フェロモン

フェロモンは軍隊アリのコミュニケーションツールである。また、揮発性の物質であり、時間とともに薄まっていくが、同じ場所にフェロモンがまかれると、フェロモン量は多くなる。本研究では、まかれたフェロモンは空間に蒸発し、周囲の空間に拡散していくものとした。また、エージェントは空間にあるフェロモンに誘導されるものとした。メジャーエージェントや Return 状態エージェントはフェロモンをまくことで、他のエージェントを引きつけることができる。フェロモンの拡散は、Fig. 2 のように確率分布に基づいて拡散する。Fig. 3 はフェロモンが拡散する方向と、それぞれの方向に対する拡散率を表している。 (x, y) のマスは中央が r_A 、上下左右のマスが r_B 、対角方向のマスが r_C である。フェロモンの拡散は t ステップ目における点 (x, y) の空中にあるフェロモン量 $space_{x,y}(t)$ にそれぞれの拡散率をかけたもので表される。拡散率は体積比により $r_A : r_B : r_C = 0.788 : 0.043 : 0.010$ とした。フェロモンの蒸発は、 t ステップ目における点 (x, y) の地上にあるフェロモン量 $ground_{x,y}(t)$ に蒸発率 r_e をかけることで表される。フェロモンの拡散と蒸発に関する更新式を以下に示す。

$$\begin{aligned}
space_{x,y}(t+1) &= r_A * space_{x,y}(t) \\
&+ r_B (\sum_p space_{(i_p,j_p)}(t) - 4space_{(x,y)}(t)) \\
&+ r_C (\sum_q space_{(i_q,j_q)}(t) - 4space_{(x,y)}(t))
\end{aligned}$$

$$\begin{aligned}
(x_{i_p}, y_{i_p}) &= \{(x, y+1), (x, y-1), (x+1, y), (x-1, y)\} \\
(x_{i_q}, y_{i_q}) &= \{(x+1, y+1), (x+1, y-1), \\
&(x-1, y+1), (x-1, y-1)\}
\end{aligned} \quad (1)$$

$$space_{x,y}(t+1) = space_{x,y}(t) + r_e * ground_{x,y}(t) \quad (2)$$

$$ground_{x,y}(t+1) = ground_{x,y}(t) - r_e * ground_{x,y}(t) \quad (3)$$

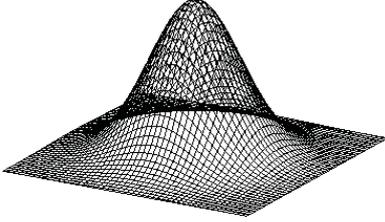


Fig. 2: Image of diffusion of pheromone

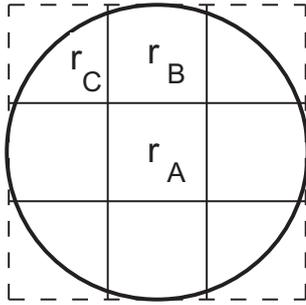


Fig. 3: Diffusion rate of pheromone

3 利他行動

軍隊アリは溝や穴などの障害物に遭遇すると、自らの身体を使って穴を埋め、橋を形成する。この行動は探索を行っている時にも行われる。本研究では、利他行動を溝に入って橋を形成しようとする行動であると定義した。周囲のエージェント数が一定以上になると利他行動を行い、また、利他行動をやめ橋から離れる条件を次のように定義した。

3.1 利他行動を始める条件

利他行動は以下の2つの条件を同時に満たすと行われる。

- 1) エージェント i を中心とした 5×5 マスの中のエージェントの数が一定以上存在する。
- 2) 周囲1マスに溝があることを確認する。

利他行動を始める条件として、周囲のエージェント数を考慮し、フェロモン量を用いなかった。エージェントが餌を探索している時はフェロモンがまかれず、結果として利他行動が行われないためである。

3.2 利他行動をやめる条件

利他行動をやめる条件には、Lioniらによって定義されたアリが橋から抜け出す確率⁹⁾を適用した。この

確率は、形成された橋の大きさといくつかのパラメータを用いて表される。本シミュレーションではこの確率をもとに、エージェントが橋から抜け出す確率 P_i を定義した。エージェントが溝から抜け出す確率 P_i は以下の式で与えられる。

$$P_i = \frac{a}{1 + bX_i^2} \quad (4)$$

ここで、 a は周囲の状況に関係なく溝から抜け出す確率、 b は周囲のフェロモン量に応じた橋への依存度、 X_i は橋にいるエージェント数を表すパラメータである。橋にいるエージェント数だけでなく、フェロモン量に応じたパラメータを取り入れる理由は、Return状態のエージェントが近くにいる場合に溝から抜け出しにくくするためである。そうすることで、Return状態のエージェントがいる場所付近の溝に橋が形成され、巣から餌までのより短い経路を形成できると考えられる。

3.3 利他行動に関する実験

実験は餌場が1つ、メジャーエージェントが3個、マイナーエージェントが100個で行った。3.1節および3.2節で設定した利他行動の条件を設定し、最短経路を構築できることおよび、エージェントの挙動を観察する。

3.3.1 最短経路の構築

Fig. 4はシミュレーション中のスクリーンショットである。フィールドの中央にある巣に対して左側と下側の溝にある程度数のエージェントが利他行動を行うことで橋が形成されていることがわかる。利他行動を行わない場合は、溝を迂回して餌を運ばなければならないが、Fig. 4の下側の溝に見られるように、エージェントが利他行動を行って橋を形成することで、巣と餌の間の最短経路を構築することができた。

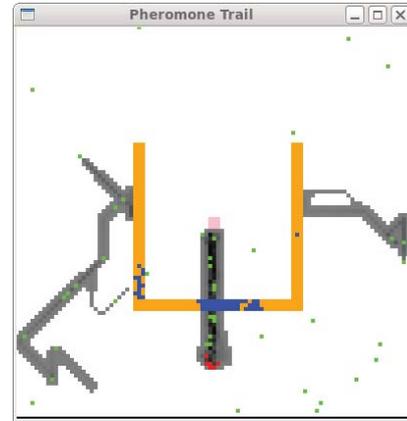


Fig. 4: Construct of the shortest path

3.3.2 利他行動に関する特徴

文献⁷⁾では、エージェントが利他行動を行う時とやめる時、それぞれの特徴を抽出している。まず、利他行動を行うエージェントの特徴は以下の2つである。

利他行動を行う時の特徴

- 1) 周囲の溝に他のエージェントがない場合に利他行動を行う。
- 2) 周囲の溝に他のエージェントがいる場合に利他行動を行う。

1) の特徴を持つエージェントは、集団の先頭を切って溝に入るエージェントのことであり、積極的に溝に入るエージェントがいることで、探索を行っている時にも橋が形成される。2) の特徴を持つエージェントは、他のエージェントが溝に入っていると自分も溝に入るエージェントである。文献⁷⁾より、Return 状態のエージェントが近くにいる場合は多くのエージェントが利他行動を行う。

次に、利他行動をやめるエージェントの特徴について説明する。その特徴は以下の2つである。

利他行動をやめるときの特徴

- 3) 溝を抜け出してもすぐに溝に戻る。
- 4) 溝を抜け出すとそのまま探索を行う。

これらの特徴には、フェロモンを知覚した場合に溝から抜け出すという共通点がある。

3) の特徴を持つエージェントは、橋を現状維持、もしくはより密集させるエージェントである。利他行動をやめる条件は確率であるため、溝から抜け出しにくい状況でも抜け出す可能性がある。その状況で溝から抜け出した場合に、3) の特徴を持つエージェントはすぐに溝に戻って橋を形成する。このエージェントが溝に戻るとき、自分が入っていた場所か、集団の中心に近い側の空いている場所に戻るため、現状維持もしくは橋を密集させる。

4) の特徴を持つエージェントは、周囲に多くのエージェントが存在する状況もしくは、エージェントは少ないがフェロモン量も少ない場合に多く見られた。周囲にエージェントが多く存在する場合は、十分に橋が大きいため、今いる場所の橋にいるよりも探索を行う方が良いと判断し、溝を抜け出す。周囲にエージェントが少なく、フェロモン量も少ない場合は、Return 状態のエージェントに求められていないと判断して、溝を抜け出す。

上記のように、それぞれの特徴を持ったエージェントが生じると考えられた。

3.3.3 橋の再構築

Fig. 4に見られるように、下側の溝にあるような橋が初めからうまく形成されるケースは多くない。Return 状態のエージェントは、溝に橋が形成されていない場合、溝を迂回するように進む。その進んだ先に他のエージェントが集まると、その場所の近くの溝に橋を形成するケースがある。Fig. 5の(a)の下側の溝にあるように、巣と餌の間の最短経路上ではなく、少しずれた部分に橋が形成されている。

Return 状態のエージェントは Fig. 5の(a)の下側の溝に形成された橋を渡って巣に戻る。このときにフェロモンがまかれ、餌を離れた後、フェロモンをたどって溝に到達する。Fig. 5の(b)にあるように、利他行動を行う条件を満たしているため、溝付近にいるエージェントは橋を形成する。その後、Return 状態のエージェントが使用しなかった溝の部分で利他行動を行っているエージェントは溝から抜け出す。そのため、Fig. 5の(c)では、使用されなかった部分の橋が欠けている。また、Fig. 5の(c)では、餌を発見した他のエージェントが餌を巣まで運んでいる。そのエージェントは Fig. 5の(b)と同様に、より最短経路に近い部分の溝で橋

を形成する。以上のことを繰り返すことで、Fig. 5の(d)にあるように、最短経路が構築される。

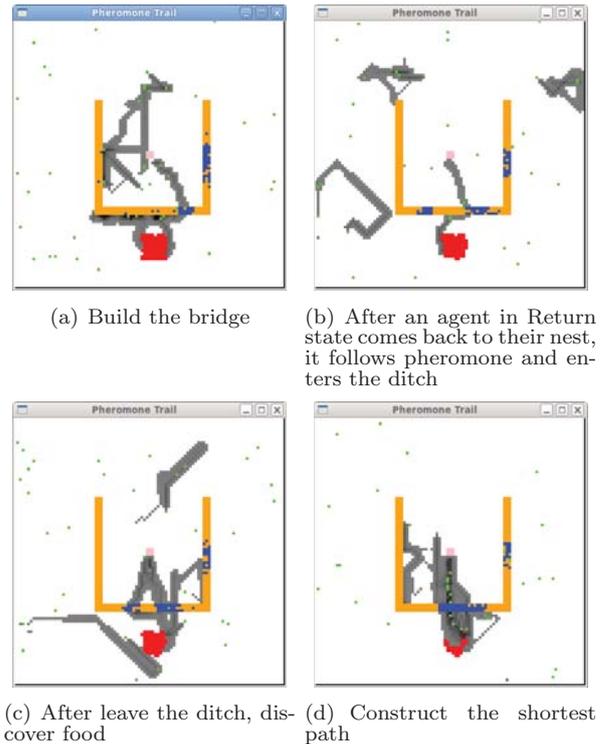


Fig. 5: The flow which construct the shortest path

4 複数の餌場に対する実験

3.3.3節で利他行動による橋の形成と、それによる最短経路の構築が見られた。しかし、今までのシミュレーション環境では餌場が1つであった。現実世界の軍隊アリの特徴として、巣を中心として周囲の餌場からすべての餌を食いつくし、餌がなくなると他の場所に移動するという行動が見られる。本節ではこの特徴を再現するために、Fig. 6にあるように複数の餌場を設置したときのエージェントの行動を観察する。

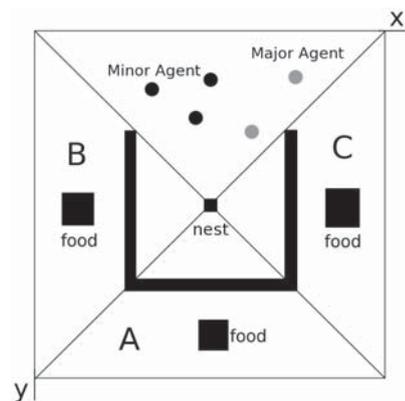


Fig. 6: Three feeding spots and area with each feeding spot

4.1 環境設定

Fig. 6にあるように、下側の餌があるエリアをエリア A、左側の餌があるエリアをエリア B、右側の餌があるエリアをエリア Cとし、それぞれのエリアのエージェント数を検証する実験を行った。餌の大きさの違

いや、エージェント数の違いによって、エージェントの挙動がどのように変化するかを観察するために、以下の8つの環境を用意した。

実験 1-1 餌 A:B:C = 2:1:1, エージェント数:100

実験 1-2 餌 A:B:C = 1:2:1, エージェント数:100

実験 1-3 餌 A:B:C = 1:1:2, エージェント数:100

実験 1-4 餌 A:B:C = 4:2:1, エージェント数:100

実験 2-1 餌 A:B:C = 2:1:1, エージェント数:50

実験 2-2 餌 A:B:C = 1:2:1, エージェント数:50

実験 2-3 餌 A:B:C = 1:1:2, エージェント数:50

実験 2-4 餌 A:B:C = 4:2:1, エージェント数:50

それぞれの実験結果に用いられる図は、横軸にステップ、縦軸にエリアにいるエージェント数であり、3つのエリアに関して同時にプロットしている。

4.2 実験結果 エージェント数:100

4.2.1 A:B:C=2:1:1 (実験 1-1)

Fig. 7はエージェント数の推移を示しており、次のような挙動が見られた。

シミュレーションを開始してから、メジャーエージェントが多くのエージェントをフェロモンによってひきつけた状態でエリア B に移動した。エリア B にはエージェントが多く集まり、溝に大きな橋が形成され、最短経路が容易に構築された。しかし、形成された橋は、エージェントが渡るために必要な大きさをはるかに越えて大きく、エリア B にいるエージェントの半分以上のエージェントが橋を形成していた。そのため、エージェント数が非常に多いにも関わらず、すべての餌を運ぶのにかかった時間は、同じ環境で行われた別の試行でかかった時間と変わらなかった。

エリア B で餌の探索を終えたエージェントはエリア A とエリア C に分かれて移動した。エリア B の橋で巣がある内側から抜け出したエージェントは、巣を通過してエリア C に移動した。エリア B の橋の外側から抜け出したエージェントは、溝を迂回して距離が近い餌場があるエリア A に移動した。エリア A で餌を発見したエージェントは、橋が形成されておらずエリア C の方へ溝を迂回したため、エリア C のエージェント数の方が上昇した。そのため、まずエリア C ですべての餌が探索され、探索を終えたエージェントがエリア A に移動した。その結果、エリア A に多くのエージェントが集まり、最短経路が構築され、餌の運搬が行われた。

4.2.2 A:B:C=1:2:1 (実験 1-2)

Fig. 8はエージェント数の推移を示しており、次のような挙動が見られた。

シミュレーションを開始して間もなく、エリア B とエリア C のエージェント数が大きく上昇した。この2つのエリアは、溝を迂回した場合に巣から一番近い餌場であるため、エージェントが集まりやすい。少し時間が経過した1800ステップあたりでは、エリア B のエージェント数が上昇する部分が見られた。これは、エリア B の餌が大きいいため、Return 状態のエージェント

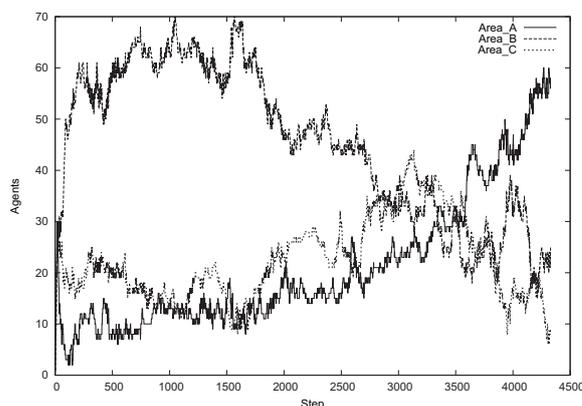


Fig. 7: The number of agents in each area at experiment 1-1

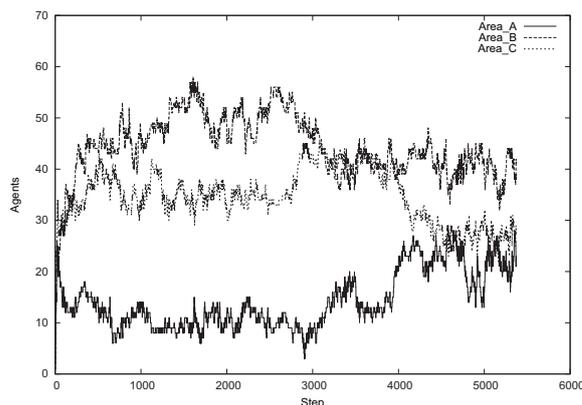


Fig. 8: The number of agents in each area at experiment 1-2

がまくフェロモンが広範囲に渡って影響を与え、多くのエージェントを引きつけたためだと考えられる。エリア B では3000ステップあたり、エリア C では4000ステップあたりで、それぞれのエリアにいるエージェントは、探索を終えてエリア A に移動した。そのため、Fig. 8では、3000ステップと4000ステップあたりで、エリア A のエージェント数が上昇した。しかし、橋を形成して最短経路を構築するにはエージェントが足りないため、Return 状態のエージェントは溝を迂回した。

4.2.3 A:B:C=1:1:2 (実験 1-3)

Fig. 9はエージェント数の推移を示しており、次のような挙動が見られた。

初めにエリア B とエリア C にエージェントが分かれて集まった。その後、餌が大きいエリア C の方が多く集まった。エージェントが多いエリア C では最短経路の構築が早く、すべての餌が運ばれるのに時間がかからなかった。その後、探索を終えたエリア C のエージェントは距離が近いエリア A に移動した。その間、エリア B ではなかなか最短経路が構築されず、餌の運搬がスムーズに行われなかった。そのため、最短経路を構築できたエリア A の方が早く探索を終えた。エリア A にいたエージェントはエリア B に移動し、餌の探索を行った。

4.2.4 A:B:C=4:2:1 (実験 1-4)

Fig. 10はエージェント数の推移を示しており、次のような挙動が見られた。

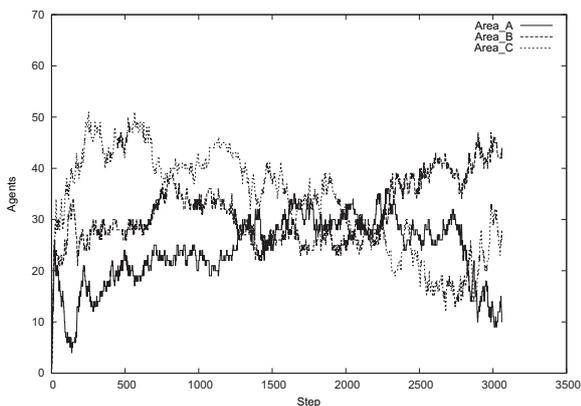


Fig. 9: The number of agents in each area at experiment 1-3

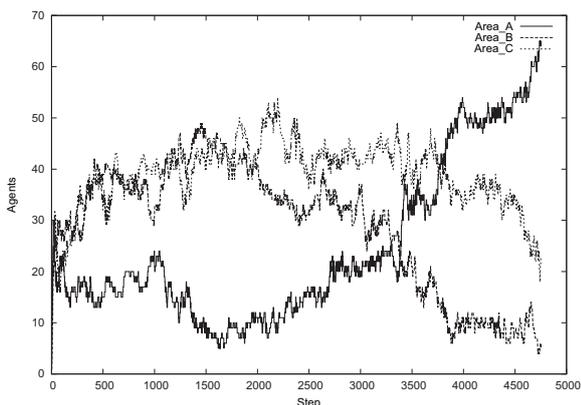


Fig. 10: The number of agents in each area at experiment 1-4

シミュレーション開始後、エリア B とエリア C にエージェントが分かれた。この 2 つのエリアでは、溝に橋が形成され、最短経路が構築されるまで時間がかからなかった。そのため、早い段階ですべての餌を探索を終えた。この 2 つのエリアで探索を終えたエージェントはエリア A に移動した。エリア A で餌を発見した Return 状態のエージェントが溝を迂回して、エリア C の橋を使用したため、エリア C のエージェント数があまり減少しなかった。エージェントが集まったエリア A では、溝に橋が形成され、最短経路が構築された。

4.3 エージェント数が多い環境に対する考察

エージェント数が多い環境でのエージェントの挙動は以下の 2 つが見られた。

- 1) 巣から距離が近い餌場がある左右のエリアに集まり、それぞれのエリアで探索をおえるとエリア A に移動する。
- 2) メジャーエージェントがまくフェロモンの影響を受けて多くのエージェントが左右どちらかのエリアに集まり、探索を終えると残りの 2 つエリアに分かれて移動する。

1) の挙動はエージェント数が多い環境で多く見られた。エージェント数が多い実験 1-1 から実験 1-4 を 10 回の試行の結果、1) の挙動はそれぞれ 7 回見られた。エージェント数が多いため、分かれて探索を行っても、効率的に探索できるためである。また、4.2.2 節で述べたように、左右のエリアに分かれる場合で、餌の大きさ

によって集まるエージェント数に差がある挙動が見られた。すなわち、大きい餌があるエリアにはエージェントが多く集まりやすく、餌を早く運びやすいということが分かる。

2) の挙動はエージェント数が多い環境で、メジャーエージェントが多くのエージェントをひきつけた場合に見られた。メジャーエージェントが他のエージェントを引きつける場合に限られるため、見られる回数が少ない挙動である。10 回の試行の中でそれぞれ 3 回ほど見られた。左右のエリアの餌の大きさの違いによる挙動の変化、すなわち、大きい餌があるエリアにエージェントが集まりやすいということではなく、初めにエージェントが集まるエリアは、メジャーエージェントがランダムウォークで移動したエリアである。そのエリアでは、エージェントが渡るために十分な大きさ以上に大きな橋が形成されるケースがほとんどである。探索を終えると、巣がある側の溝から抜け出すエージェントと、反対側の溝から抜け出すエージェントが存在し、それぞれが別のエリアに移動することが分かる。

4.2.1 節では、エリア B で、エージェントが橋を渡るために必要な大きさより大きな橋が形成された。エージェントが過剰に集まったため、次々とエージェントが溝に入ってしまった。その結果、すべての餌を運ぶのにかかった時間は、同じ実験の別の試行でかかった時間とあまり変わらなかった。

4.4 実験結果 エージェント数:50

エージェント数が 50 の場合は、同時に 2 つ以上の橋を形成することができなかった。

4.4.1 A:B:C=2:1:1(実験 2-1)

Fig. 11 はエージェント数の推移を示しており、次のような挙動が見られた。

初めはエージェントが様々なエリアを移動した。エージェント数が少ないため、エージェントが橋を渡るために十分な大きさの橋を形成することが難しいことが分かる。ステップ数が増すごとに、エリア C にエージェントが集まった。エージェント数が増えたため、エリア C では橋が形成された。この時形成された橋は最短経路上ではなかったが、3.3.3 節の橋の再構築が行われ、最短経路上に橋を形成することができた。

エリア C にエージェントが集まったため、エリア A とエリア B にエージェントが少なく、これらのエリアの溝に橋が形成されなかった。そのため、Return 状態のエージェントは溝を迂回して餌を運んだ。5000 ステップあたりからエリア B のエージェント数が上昇しているのは、溝を迂回した Return 状態のエージェントが巣に帰った後、フェロモンをたどった先がエリア B の溝であり、そこで利他行動を行ったためである。

4.4.2 A:B:C=1:2:1(実験 2-2)

Fig. 12 はエージェント数の推移を示しており、次のような挙動が見られた。

シミュレーション開始からしばらくはランダムに移動を行うが、1000 ステップあたりから一番餌が大きいエリア B に集まった。その結果、エリア B に溝に橋を形成した。しかし、形成された橋の場所は、巣と餌の最短経路上ではなく、少しずれた部分の溝に形成された。そこで 3.3.3 節の橋の再構築が行われ、最短経路上

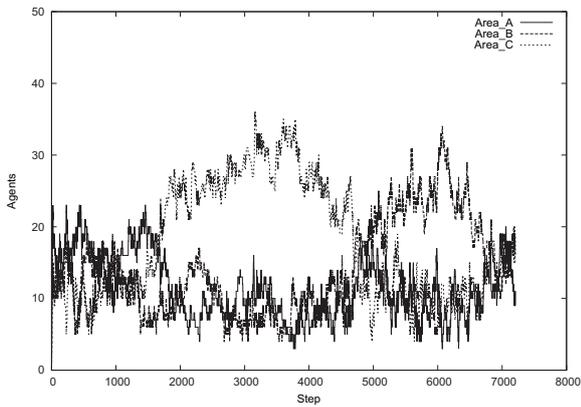


Fig. 11: The number of agents in each area at experiment 2-1

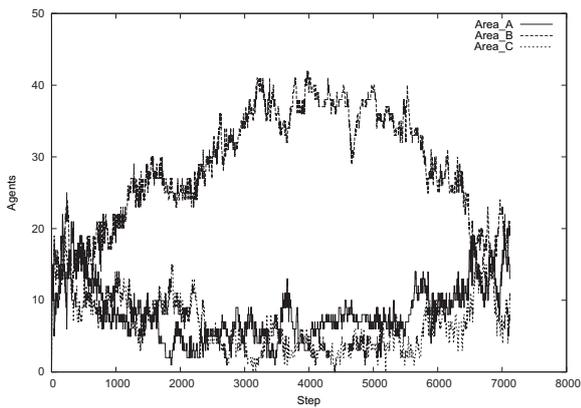


Fig. 12: The number of agents in each area at experiment 2-2

に橋を形成した。エージェントがエリア B に集まったため、エリア A とエリア C にはエージェント数が少なくなった。そのため、橋が形成されず、Return 状態のエージェントは溝を迂回した。

4.4.3 A:B:C=1:1:2(実験 2-3)

Fig. 13 はエージェント数の推移を示しており、次のような挙動が見られた。

前節と同様に、餌が大きいエリア C にエージェントが集まった。そこで橋を形成し、最短経路を構築することができた。しかし、エリア C では 2000 ステップあたりで一度、エージェント数が減少した。この時点でエリア C にいるエージェントのほとんどが利他行動を行い、橋を形成したため、餌を運ぶエージェントがいなくなった。そのため、橋の周辺にフェロモンがまかれず、フェロモンを感知できずにエージェントが溝を抜け出したためエージェント数が減少した。エリア C の溝から抜け出したエージェントはエリア A に移動している。エリア A ではエージェントが多くなり、橋を形成できたが、橋を維持できるほどエージェントがいないため、すぐに橋が消滅した。その後は、餌が多く残っているエリア C にエージェントが集まり、すべての餌を探索した。他のエリアでは、橋を形成することができず、Return 状態のエージェントは溝を迂回して餌を運んだ。

4.4.4 A:B:C=4:2:1(実験 2-4)

Fig. 14 はエージェント数を示しており、次のような挙動が見られた。

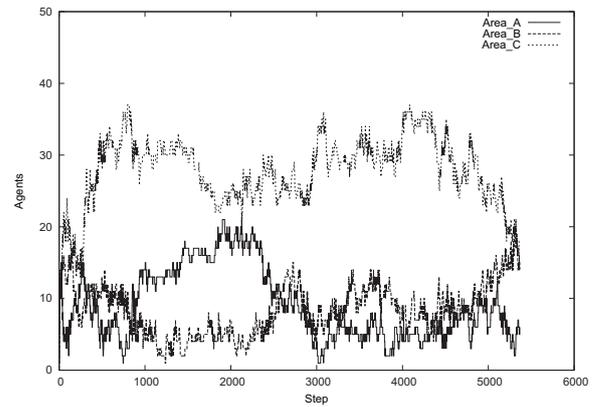


Fig. 13: The number of agents in each area at experiment 2-3

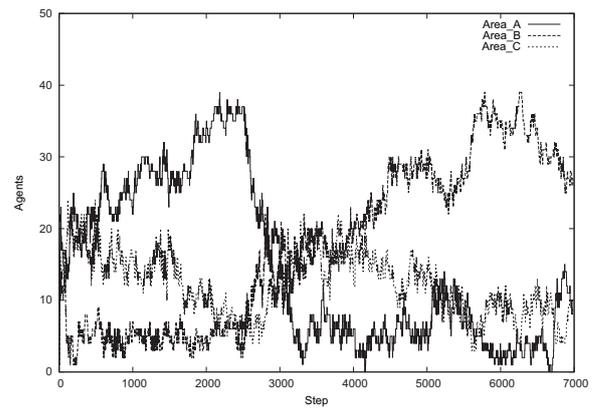


Fig. 14: The number of agents in each area at experiment 2-4

実験 2-4 では、シミュレーションを開始してから初めに、エリア A にエージェントが集まるという現象が見られた。この現象は他の実験では見られなかった。エージェント数が少なく、エリア B とエリア C で餌を発見したエージェントが他のエージェントを引きつけることができなかった。加えて、エリア A の餌が大きく、他のエリアを通過してきたエージェントを引きつけることができたため、エリア A にエージェントが集まった。エリア A で探索を終えたエージェントは、残りのエリアに分かれたが、橋が形成されていないため、Return 状態のエージェントは溝を迂回した。餌の大きさの違いから、エリア C の餌が先にすべて運ばれ、その後エージェントがエリア B に集まった。

4.5 エージェント数が少ない環境に対する考察

エージェント数が少ない環境では以下の 2 つのエージェントの挙動が観察された。

- 3) 巣から距離が近い餌場がある左右どちらかのエリアに集まり、探索をおえるとエリア A に移動し、最後に残りのエリアに移動する。
- 4) 巣から距離が遠いエリアに集まり、探索を終えると残りのエリアに分かれて移動する。

ほとんどのシミュレーションで 3) の挙動が見られた。エージェント数が少ない環境では、2 つ以上のエリアを同時に探索するほどエージェントがいないため、1 つのエリアに集まって探索を行う。初めに探索されるエリアは、巣から距離が近い餌場が探索されるが、その

中で、大きい餌があるエリアが初めに探索される傾向が10回の試行のうち8回観察された。そのエリアの餌の探索を終えると、次に探索するエリアとして一番近い餌があるエリアAが選択され、その後に残りのエリアに移動するという挙動が見られた。

4)の挙動は4.4.4節のようにエージェントが少ない環境で、巣から遠い場所にある餌が他の餌に比べて非常に大きい場合にのみ見られた。エリアAは餌が非常に大きく、広い範囲でエージェントを引きつけることができる。そのため、エージェントが巣から距離が近い餌場があるエリアに集まらず、餌を通過してエリアAに集まった場合にのみ、初めに探索される。左右のエリアをエージェントが通過するという条件はなかなか見られないため、4)の挙動が見られる回数は少ない。

上の2つの挙動以外に、エリアに集まったエージェントのほとんどが利他行動で溝に入ってしまう、餌を運ぶエージェントがいなくなり、フェロモンを知覚できなくなったエージェントが出現し、溝から抜け出していく現象が勘さるされた。

5 おわりに

文献⁷⁾では、利他行動に関するエージェントの特徴について取り上げている。様々な特徴を持つエージェントが集まることで集団が形成され、その集団が本研究で述べた挙動を示すことで、餌を探索するという目的を達成していることが分かる。

シミュレーション実験の結果をもとに、エージェントの挙動の特徴を述べた。エージェント数が多い環境では、エージェントが分かれて探索を行う挙動が見られ、餌の大きさによってそれぞれのエリアに集まるエージェント数に差が生じることが分かる。エージェントが少ない環境では、群全体で餌場を1つずつ探索していく挙動が見られる。また、これらの環境で見られる挙動の共通点として、エージェントの集団が現在の位置から近い場所にある餌場に移動しやすい傾向がある。1つのエリアの餌をすべて探索すると、近い餌場があるエリアに移動し、餌がなくなった後、遠い餌場まで探索を行うという挙動である。一方、エージェントが少ない場合で、餌場が大きい場合には、巣から遠かったとしても、大きい餌場から探索を始めることが分かった。シミュレーションのこれらの挙動は、現実の軍隊アリにも見られるものである。

利他行動に関するエージェントの特徴や、集団の挙動などから、集合知が形成されるプロセスを表したアルゴリズムを提案する事を今後の課題とする。

参考文献

- 1) Beni, G., Wang, J.: Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on robots and Biological Systems, Tuscany, Italy (1989)
- 2) Dorigo, M.: Optimization, Learning and Natural Algorithms, Ph.D. Thesis, Politecnico di Milano (1992)
- 3) Dorigo, M. and Gambardella, L.M.: Ant Colonies for the Traveling Salesman Problem, Biosystems, Vol.43, 73/81 (1997)
- 4) Hiroyuki Ishiwata, N. Noman, and Hitoshi Iba,: Emergence of Cooperation in a Bio-inspired Multi-agent System, The Japanese Society for Artificial Intelligence 2010, 364/374 (2010)
- 5) Takumi Ichimura and Yoshiaki Douzono: Altruism Simulation based on Pheromone Evaporation and Its Diffusion in Army Ant Inspired Social Evolutionary System, Proc. of The 6th International conference on Soft Computing and Intelligent Systems and The 13th International Symposium on Advanced Intelligent Systems(SCIS-ISIS 2012), 1357/1362 (2012)
- 6) S.Powell. How a few help all: living pothole plugs speed prey delivery in the army ant *Eciton burchellii*. Animal Behavior, Vol.73, No.6, 1067/1076 (2007)
- 7) 上本拓也, 市村匠: 軍隊アリのシミュレーションにおける利他行動形成のための状態と行動ルールについて, 2013 IEEE SMC Hiroshima Chapter Young Researchers WorkShop, 59/62 (2013)
- 8) Swarm インストール URL http://web.archive.org/web/20120208110755/http://swarm.org/index.php/Main_Page (2013/10/30 現在)
- 9) J.L.Deneuborg, A.Lioni and C.Detrain.: Dynamics of Aggregation and Emergence of Cooperation, The Biological Bulletin ,Vol.202, No.3, 262/267 (2002)

マルチエージェント環境における協調度を用いた行動選択法

○河村将範 小林邦和 (愛知県立大学)

Action Selection Method Using Degree of Cooperation in a Multi-agent System

*M.Kawamura and K.Kobayashi (Aichi Prefectural University)

Abstract— In recent years, a concept of a dividual is proposed to interact properly with another person. To construct a model of the dividual, the degree of cooperation is assigned to the corresponding dividual. By introducing the degree of cooperation into multi-agent systems, we evaluate what kind of changes appears in the agent behavior. In addition, we propose an action selection method by introducing the degree of cooperation into the soft-max method in multi-agent systems. Using the proposed method, we confirm whether the cooperative action is promoted or suppressed through computer simulations.

Key Words: Dividual, Degree of cooperation, Multi-agent system, Reinforcement learning

1 はじめに

本研究に至った経緯として、まずヒューマンロボットインタラクションという観点に着目した。近い将来、ロボットが人間と同じようにコミュニケーションをとることで、ロボットが人の助けとなったり、また、話し相手となってくれることで、人々が過ごしやすい社会となることが想定される。人間同士のコミュニケーションをロボットに模倣させることで、ロボットと人間の円滑なコミュニケーションが実現できるのではないかと考えている。人間社会では、人と人が言語、ジェスチャー、アイコンタクトといったコミュニケーションをとりながら、協調行動を行っている。一方、人が相手に対して接し方を使い分けるといった分人という概念が平野により提唱されている¹⁾。私はこの分人という概念をマルチエージェントシステム²⁾³⁾⁴⁾に導入し、協調行動の実現を目標とし、分人モデルの構築について研究を行うことにした。

本研究では、主に分人モデルの特徴である他者に応じた接し方の違いを取り扱う。その方法として、分人が構成されると同時に、協調度という尺度を導入する。この協調度をマルチエージェントシステムに導入することにより、エージェントがとりうる行動にどのような変化が現れるかを評価する。また、提案法を用いることにより、エージェントの協調行動が促進または抑制されることを計算機シミュレーションにより、確認する。

2章では、本研究で用いる強化学習法として、Q学習について説明を行う。3章では、本研究で提案する行動決定法で用いる soft-max 法について説明する。4章では、本研究の提案法について説明する。5章では、計算機シミュレーションとその結果について説明する。6章では、まとめと今後の課題について述べる。

2 Q学習

本研究では、代表的な強化学習法⁵⁾⁶⁾として、Q学習⁷⁾を用いている。Q学習とは、現在の状態とその時の行動を対にし、その時の価値を更新することで学習を行うものである。現在の状態における行動の価値は、現在の状態によって得られる報酬と、その行動を行った後の状態の価値を用いることで学習を行う。現在の状態 s における行動 a の価値を Q 値と呼び、 $Q(s, a)$ と

表す。また、 Q 値の更新は、現在の状態で得られる報酬を r 、次の状態を s' として式 (1) で表す。

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \cdot \max_{a' \in A} Q(s', a') - Q(s, a)] \quad (1)$$

ここで、 A は行動全体の集合である。 α は学習率と呼び、学習速度を調節する値である。 γ は割引率と呼び、将来の報酬が現在において、どれだけの価値があるかを決定づける役割を持つ。

3 soft-max 行動選択法

本研究で提案する行動決定法では、行動方策の手法として、soft-max 行動選択法というものをを用いる。具体的な式を式 (2) に示す。

$$\pi(a|s) = \frac{\exp(Q(s, a)/T)}{\sum_{b \in A} \exp(Q(s, b)/T)} \quad (2)$$

ここで、 A は行動の集合である。また、 T は温度定数と呼ぶものであり、ランダム性を調節する値である。 T が大きいほどランダム性が強くなり、 T が小さいほどランダム性が弱くなる。この方法では、指数関数を用いているため、 Q 値が大きくなるほど、その時の行動確率はより大きい値となる。よって、学習が進むにつれて、結果は収束する。

4 提案法

ここでは、4.1 節において分人の概念を説明し、4.2 節では、分人と同時に定義される協調度という尺度を説明し、4.3 節では、本研究で用いる行動決定法について説明する。

4.1 分人

人は、他人との接し方を使い分けことによってコミュニケーションを行っている。例として、自分にとって目上の方に対しては敬語を用い、友達に対しては普段使っているような言葉で話をする。目上、友達という簡単な分類を例に取り上げたが、人は性別や国籍、自分との関係などの要因により、話し方や接し方を使い分けている。このような概念を分人といい、平野啓一郎によって提唱されている¹⁾。

分人は、他人とコミュニケーションを行い友人、知人となった相手の数だけ存在する。自分がある人 A さ

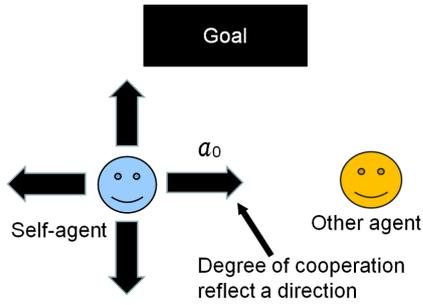


Fig. 1: Relationship between two agents

んとコミュニケーションを行い、知人となった場合には、分人 A が自分の中に作成される。また、B さんとコミュニケーションを行い、知人となった場合には、A さんと同様に分人 B が自分の中に作成される。人間の個性や性格といったものは、この分人の構成要素によって決定されるものであると考えられている。今回の例では、自分の中に A さんと B さんの分人が存在するため、その A さんと B さんという分人から情報を得ることで、自分の個性が表現されると考えられる。よって、分人モデルを構成する際には、個性や性格の設定は、分人の構成要素を基に構成する。また、初対面の人と接する際には、自分の中に標準的な接し方を行う分人が存在するものとし、これを社会的分人とする。本研究では、この分人というものが作成されると同時に、協調度という尺度を定義するため、A さんと B さんの分人が作成されると同時に A さんに対する協調度と B さんに対する協調度を分人と対にして定義する。

4.2 協調度

本研究では、分人が作成されると同時に、協調度 c という尺度を定義する。 c は $0 \leq c \leq 1$ の範囲で定義されるものとし、スカラー値を取る。分人が作成される際には、社会的分人をデフォルトの分人とし、その分人で学習することによって新しい分人が作成されていく。この社会的分人の協調度は 0.5 とする。協調度 c が $c > 0.5$ である場合には協調度が高く、 $c < 0.5$ である場合には協調度は低いといった形で評価する。この協調度における高低によって、協調の促進と抑制を行う。また、協調度が低いことによって、協調行動に反発がおこる可能性も考慮している。ここで言う反発とは、協調行動をできるだけ避けるということである。

4.3 協調度を用いた行動決定法

提案法では、相手が存在する方向を取得し、その方向を a_0 とする(図 3 参照)。この a_0 は、自エージェントが行う事ができる行動から選択する。そして、 a_0 の方向へ行動する時の確率に、協調度を反映させることで、協調行動の促進と抑制を行う。

これを Fig. 1 を用いて説明する。図 1 では、ゴールが上方向に存在し、右方向に協調相手が存在する。そのため、自エージェントが行動可能な行動が上、右、下、左であったとすると、この 4 つから確率的に行動を選ぶ際、自エージェントが右に行動する確率に協調度を反映させるといった手法をとる。

本研究の提案法は、soft-max 行動選択法を基に、Q

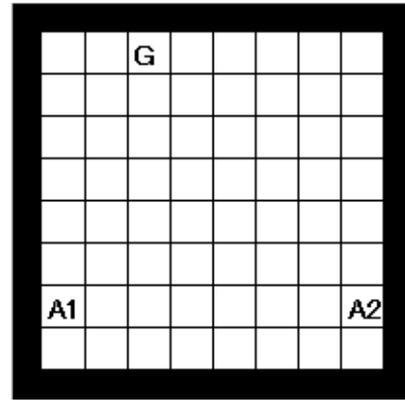


Fig. 2: A Simulation field

学習を用いることで、式 (3) で定義する。

$$\pi(a|s) = \frac{\exp(Q(s, a) * g(c))}{\sum_{b \in A} \exp(Q(s, b) * g(c))} \quad (3)$$

ここで、 A はエージェントのとりうる行動の集合である。協調度を $g(c)$ という関数を用いることで行動選択確率に反映させる。提案法では、 $g(c)$ を式 (4) のように定義する。

$$g(c) = \begin{cases} 2c & (a = a_0) \\ 1 & (otherwise) \end{cases} \quad (4)$$

社会的分人の協調度を 0.5 と設定しているため、その時の行動選択確率は、協調度を反映させない形とする必要がある。これは、街中において、全く知らない人とは協調を考慮しないということに相当する。そのため、協調度が 0.5 の時は $g(0.5) = 1$ とし、通常の行動選択とする必要がある。一方、協調度が 0.5 より大きい時は $g(c) > 1$ となり協調が促進され、0.5 より小さい時は $g(c) < 1$ となり協調が抑制される。この制約を実現させるために $g(c)$ は c の 2 倍の値と設定した。

5 計算機シミュレーション

10 × 10 の格子状フィールドを用意し、2 体のエージェントと 1 つの目的地を配置する。2 体のエージェントには、エージェント A_1 、エージェント A_2 として区別できるようにしておく。このフィールドは壁で囲まれているものとし、ステップ数の上限は 100、それを 10000 エピソード繰り返して学習を行う。性能評価は、ある程度の学習が終了した 10000 エピソード目のステップ毎のエージェント間の距離によって行う。協調度は、相手と協調しやすい時は 0.8、協調しにくい時は 0.2 の値で固定する。なお、今回のシミュレーションにおいて、エージェントが可能な行動は上、右、下、左の 4 つの方向へ行動できるものとする。また、 A_1 から A_2 に対する協調度を $C(A_1, A_2)$ のように表現する。

今回のシミュレーションで用いたフィールドを Fig. 2 に示す。

外側の黒い部分が壁であり、エージェントとゴールは図のように配置した。エージェントはマス目を遷移することで、行動を行う。今回のシミュレーション内容における最小ステップは、エージェント A_1 が 8 ステップ、エージェント A_2 が 11 ステップとなっている。

両エージェントの協調度がともに高い時のステップ毎におけるエージェント間の距離の変化を Fig. 3 に示し、

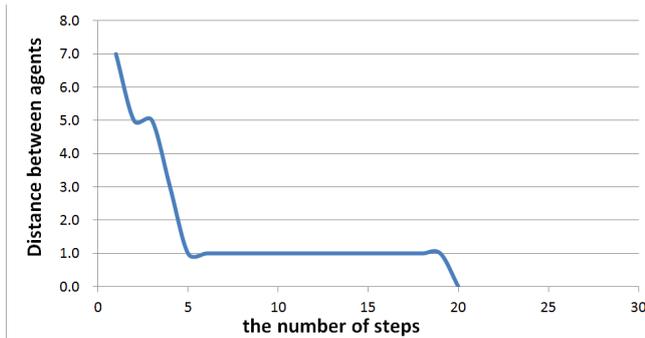


Fig. 3: Transition of distance between agents ($C(A_1, A_2)=0.8$, $C(A_2, A_1)=0.8$)

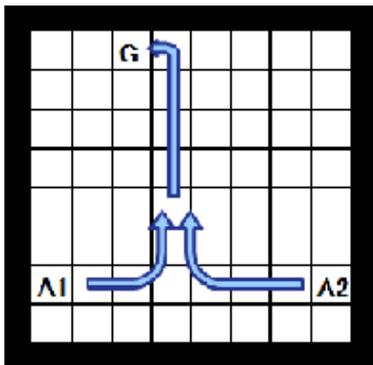


Fig. 4: An action image when both $C(A_1, A_2)$ and $C(A_2, A_1)$ are high

その時のフィールドにおける行動のイメージを Fig. 4 に示す。

この結果より、初めの数ステップでエージェント間の距離が縮まり、その後、近い距離を保ちながらゴールへ辿り着いたことが分かる。よって、お互いの協調度が高い場合は、お互いが協調しているという結果となった。また、エージェント A_2 の最小ステップが 11 ステップというところから考えると、協調しようとすることを優先し、少し寄り道をしてからゴールに辿り着いたことが考えられる。

次にエージェント A_1 の協調度が高く、エージェント A_2 の協調度が低い時のステップ毎におけるエージェント間の距離の変化を Fig. 5 に示し、その時のフィールドにおける行動のイメージを Fig. 6 に示す。

この結果より、エージェント間の距離に急激な変化はなく、ゴールに辿り着くまでに、徐々に距離が短くなっていくということがわかる。よって、片方のエージェントの協調度が低い場合には、お互いの協調度が高い場合に比べて、エージェント間の距離の縮まる速度が遅いため、協調行動が抑制されていることが考えられる。また、エージェント A_2 の最小ステップ数から考えると、ゴールまでに 24 ステップかかってしまっている。これは、エージェント A_1 がゴールに辿り着くよりも協調を行うことを優先し、エージェント A_2 がゴールに辿り着くよりも協調しないことを優先してしまったため、フィールドにおける右上の部分で少し寄り道をしてしまったと考えられる。

また、エージェント A_1 の協調度が低く、エージェント A_2 の協調度が高い時のステップ毎におけるエージェント間の距離の変化を Fig. 7 に示し、この時のフィールドにおける行動のイメージを Fig. 8 に示す。

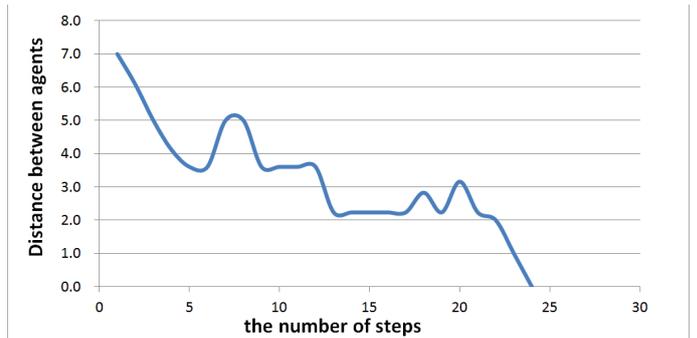


Fig. 5: Transition of distance between agents ($C(A_1, A_2)=0.8$, $C(A_2, A_1)=0.2$)

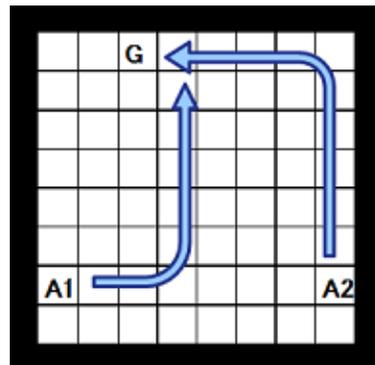


Fig. 6: An action image when $C(A_1, A_2)$ is high and $C(A_2, A_1)$ is low

先ほどの結果と同様に、エージェント間の距離に急激な変化はなく、ゴールに辿り着くまでに、徐々に距離が短くなっていくということがわかる。しかし、ゴールまでのステップ数は、先ほどの結果よりも少ない結果となった。これは、ゴールが少しだけエージェント A_1 に近い位置に存在しているからであると考えられる。エージェント A_2 の最小ステップ数から考えると、最短経路に近いようなステップ数で辿りついていることが分かる。これは、ゴールがエージェント A_1 によっていることも考えられるが、協調度のバランスがよかったのではないとも考えられる。よって、ゴールの位置と協調度の高さにステップ数が依存するということがわかる。

最後に、両エージェントの協調度がともに低い時のステップ毎におけるエージェント間の距離の変化を Fig. 9 に示し、その時のフィールドにおける行動のイメージを Fig. 10 に示す。

この結果より、全てのステップにおいて、エージェントの距離が短くなることはなく、最終的にゴールにも辿り着かないということがわかる。よって、お互いの協調度が低い場合には、お互いが協調しなかったという結果が得られた。これは協調行動が反発したと考えられる。また、ゴールに辿り着くよりも協調しないことを優先するという結果となった。これは、エージェント A_1 に関して、協調相手であるエージェント A_2 が右方向に存在し、今回は協調度が低い値として定義したため、右に行動する確率が低下してしまったからである。そして、右の行動確率が低下してしまうことにより、右に移動した時の状態の Q 値が更新されにくくなり、エピソードが増えるごとに右に行動する確率が

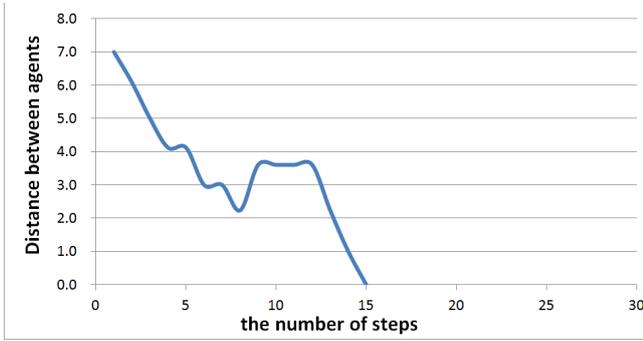


Fig. 7: Transition of distance between agents ($C(A_1, A_2)=0.2$, $C(A_2, A_1)=0.8$)

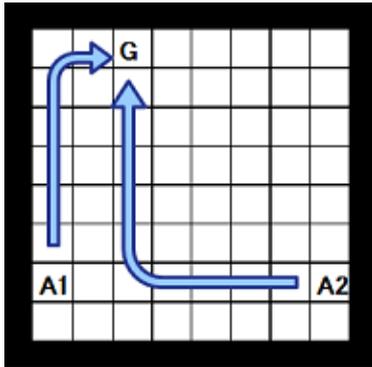


Fig. 8: An action image when $C(A_1, A_2)$ is low and $C(A_2, A_1)$ is high

低下してしまったからである。よって上と下を往復してしまう結果となった。

6 まとめ

本研究では、分人が作成されると同時に協調度という尺度を定義し、この協調度を用いた行動決定法を提案した。お互いのエージェントの協調度が高い時には、他のエージェントに近づきながらゴールへ辿り着いた。計算機シミュレーションにより、お互いのエージェントの協調度が低い時には、他のエージェントに近づこうとせず、ゴールにも辿り着かなかった。そのため、ゴールに辿り着くよりも協調度を優先した結果となった。また、片方のエージェントの協調度が高く、もう片方のエージェントの協調度が低い場合には、エージェント間の距離が徐々に近くなり、最終的にゴールに辿り着いた結果となった。この結果より、提案法によって、協調行動の促進または抑制が行われることが示された。

本研究では、協調度を固定しているが、将来的には協調度の更新をすることを視野に入れている。協調度が更新されることによって、より実環境に近いシミュレーションを行うことができると予想される。また、人間同士の協調に関しては、顔や背丈、服装といった外見的情報や、個性や性別、国籍などの内面的情報、まわりの風景などの自分の状態、自分を主観とした相手の印象や好感度といった要因に左右されるので、協調度の更新にどのような方法で反映させるかが問題となる。

分人モデルの構成という点においては、最終的にコミュニケーションが取れるようにすることを目標としている。よって文脈の生成ができるようにする必要がある。相手の言葉に対して既存の言語を話すことは容易であるが、返答として文脈生成を行うことは難しい

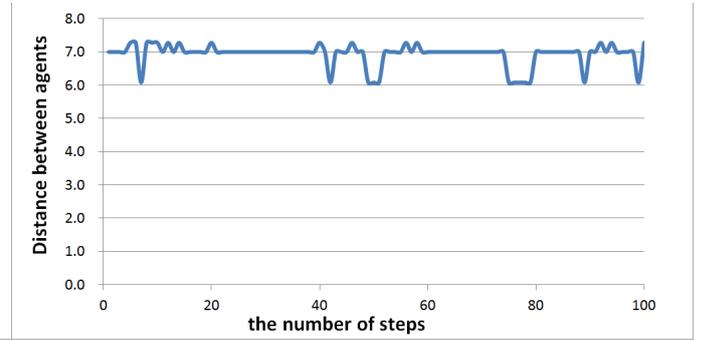


Fig. 9: Transition of distance between agents ($C(A_1, A_2)=0.2$, $C(A_2, A_1)=0.2$)

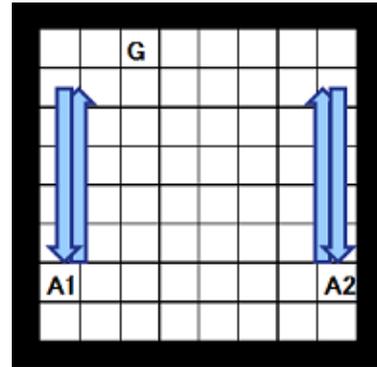


Fig. 10: An action image when both $C(A_1, A_2)$ and $C(A_2, A_1)$ are low

問題であると言える。この文脈生成が可能となった場合、言語によるコミュニケーションを行うことができるので、そのコミュニケーションによって得られる相手の印象や好感度を特定しやすくなる。これらを協調度の更新に用いることで、より実環境に近い協調を表現できるようになる。

さらに、社会的分人をどのように定義するかという問題がある。人間の場合は、この社会的分人も変化していき、子供の時の社会的分人と成長して大人になった社会的分人とはかなりの違いがある。よって、社会的分人も学習を行う必要があると考えられる。自分の中に存在する分人の構成要素によって社会的分人も学習を行う方法が考えられるが、それは自分の個性と似たものとなり、社会的分人が人間の個性を表すものとして、適切であるかを判断する必要がある。

また、分人モデルが完成した際に、学習が困難であることも考えられる。分人は学習によって更新されていくものであり、その学習をおこなった分人を用いることで自分の個性を表す。そのため、様々な人や分人モデルを導入したロボットとコミュニケーションを行う必要がある。ある程度の長い期間の学習を重ねなければ、信頼できる結果が得られないことが考えられる。家庭で稼働するサービスロボットを導入するためには、その家庭において分人を学習させる必要がある。実用的な分人モデルとするためには、学習効率がより高いものである必要があると考えられる。

分人モデルが構成されることで社会学における人間関係をシミュレーションすることができ、人間関係によって予想される問題を特定し、対策が早急に行えるようになると考えている。また、家庭用ロボットに分

人モデルを導入することで、お年寄りの話し相手となることや、家政婦として用いる、子供の遊び相手になることで、人間の生活に役立ってくれることを想定している。また、人には話せないことを相談できるようにすることで、引きこもりなどの問題の解決に貢献してくれることを想定している。

謝辞

本研究の一部は日本学術振興会・科学研究費補助金（課題番号 23500181）の助成を受けている。

参考文献

- 1) 平野啓一郎: 「私とは何か「個人」から「分人」へ」, 講談社新書 (2012)
- 2) P. Stone and M. Veloso: “Multiagent Systems: A Survey from a Machine Learning Perspective”, *Autonomous Robots*, Vol.8, No.3, pp.345-383 (2000)
- 3) 大内 東, 山本 雅人, 川村 秀憲: 「マルチエージェントシステムの基礎と応用—複雑工学の計算パラダイム—」, コロナ社 (2003)
- 4) 高玉 圭樹: 「マルチエージェント学習—相互作用の謎に迫る—」, コロナ社 (2003)
- 5) R. S. Sutton and A. G. Barto (三上 貞芳, 皆川 雅章 共訳): 「強化学習」, 森北出版 (2000)
- 6) L. P. Kaelbling, M. L. Littman, and A. P. Moore: “Reinforcement Learning: A Survey”, *Journal of Artificial Intelligence Research*, Vol.4, pp.237-285 (1996)
- 7) C. J. C. H. Watkins and P. Dayan: “Q-learning”, *Machine Learning*, Vol.8, No.3-4, pp.279-292 (1992)

四元数ネットワークインバージョンのための正則化

○井浦翼 小川毅彦 (拓殖大学)

Regularization for Quaternion Network Inversion

* T. Iura and T. Ogawa (Takushoku University)

Abstract—Quaternion network inversion has been proposed as a solution for inverse problems using a multi-layered quaternion neural network. Regularization is often used as a method to solve ill-posed inverse problems. In this study, we propose a regularization method for applying to the quaternion network inversion. We examine the effectiveness of the quaternion network inversion with regularization, by setting a simple ill-posed inverse problem about the uniqueness of the solution.

Key Words: Quaternion, Network inversion, Regularization.

1 はじめに

近年、複素数や四元数などの高次元領域に拡張されたニューラルネットモデルが盛んに研究されている¹⁾²⁾。高次元数の中でも特に四元数は、3次元空間における代数表現を容易に行うことができるため、コンピュータグラフィックスや姿勢制御などの分野で注目されている³⁾⁴⁾。

観測された結果から原因を求める問題は逆問題と呼ばれ、さまざまな分野で研究が行われている⁵⁾⁶⁾。そのニューラルネット解法としてネットワークインバージョンが提案され⁷⁾、さらに複素数や四元数などの高次元数への拡張が提案されている⁸⁾⁹⁾。

一方、逆問題の解法においては、不良設定性に起因する問題により、一般に解の存在性や一意性、安定性が保証されない。その解決法として、解の条件付けを行う正則化法がしばしば用いられる¹⁰⁾¹¹⁾。ネットワークインバージョンによる解法においても、正則化法によって解に条件付けを行い、不良設定性を緩和する方法が検討されている¹²⁾¹³⁾。

本研究では、四元数に拡張されたネットワークインバージョンについて、正則化法により不良設定性を緩和する方法を提案するとともに、3次元空間における不良設定性を有する逆写像問題の計算機実験によってその効果を示す。

2 逆問題とニューラルネット

逆問題とは観測された現象から原因やその現象を起こす内部機構を推定する問題である。順問題が原因から結果を導くものであるのに対し、逆問題は結果から原因を求める、または出力から入力を求めるものを指す。一般に逆問題には、解の存在性・一意性・安定性が保証されない問題があり、これを不適切性または不良設定性などと呼ぶ。不良設定逆問題の概念を Fig.1 に示す。本研究では特に一意性に関する不良設定性に着目し、この解決法について考える。

2.1 ネットワークインバージョン

通常、多層型ニューラルネットでは、入力から出力への方で学習を行い、学習した関係を用いて入力から出力を推定する。すなわち、学習した関係を順方向に用いて順問題を解いている。これに対しネットワークインバージョンでは、誤差逆伝搬法等で通常通り学

習した入出力関係を固定し、勾配法に基づいて与えた初期ランダム入力と、出力の誤差を、

$$x(n+1) = x(n) - \varepsilon_e \frac{\partial E}{\partial x} \quad (1)$$

として繰り返し修正することで、与えられた出力に対応する入力を求めることができる。ここで E , x および ε_e はそれぞれ出力誤差、入力および入力修正のための微小係数を表す。この繰り返し修正の手順により、学習によって得た入出力関係を用いて、逆問題を解くことができる⁷⁾。

2.2 正則化

不良設定逆問題の適切化の方法として、解空間に条件付けを行う正則化法が知られている。これは繰り返し法によって誤差汎関数最小となる準解を求める際に、制約条件を加えるものである¹⁰⁾。具体的には、解の拘束条件を表現する正則化関数 K を誤差関数に加え、

$$E = \frac{1}{2} \sum_r (d_r - y_r)^2 + \lambda K \quad (2)$$

として E 全体を同時に最小化することで拘束条件を満たしつつ誤差の最小化を図る。ここで d_r , y_r および λ は、それぞれ r 番目出力素子に与えられた出力、ネットワーク出力および拘束条件と誤差のバランスをとるための微小係数である。ネットワークインバージョンによる逆問題解法の場合も、式(1)による出力誤差最小化に正則化項を加えた式(2)を用いることで、正則化を適用することができる¹²⁾。すなわち、正則化項を加えた誤差汎関数を最小化することで、拘束条件を満たす入力を求める。

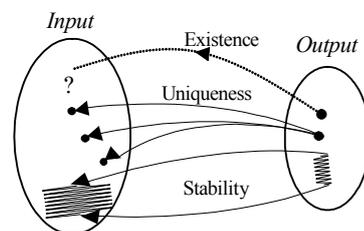


Fig. 1: Ill-posedness of inverse problems.

3 四元数とニューラルネット

四元数は高次元数の一つで 3 つの虚数単位を i, j, k とすると, $\mathbf{x} = x_1 + ix_2 + jx_3 + kx_4$ と表現される. x_1, x_2, x_3, x_4 は実数であり四元数の各成分を表す. また, $i^2 = j^2 = k^2 = ijk = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j$ であり, 四元数間の積は結合法則を, 和は分配法則を満たす. 特に, 実数部分が 0 の四元数を純虚四元数と呼び, $\mathbf{x} = ix_2 + jx_3 + kx_4$ と表現する¹⁴⁾¹⁵⁾.

四元数に拡張されたニューラルネットモデルとしては, 単一四元数ニューロンの特性をはじめ階層型やリカレント型などさまざまなものが研究されている. 多層型の四元数ニューラルネットでは, 四元数の荷重と四元数ニューロンを用いて, 四元数に拡張された入出力間の関係を学習できる¹⁶⁾¹⁷⁾. 本研究では階層型の四元数ニューラルネットを対象とし, その上での逆問題解法を検討する.

逆問題における原因と結果を表す変数を複素数や四元数に拡張することで, 複素関数間の逆問題や四元数関数間の逆問題などが考えられる. 例えば画像復元の問題を周波数領域で考える場合や, コンピュータグラフィックスにおける 3 次元空間内の移動オブジェクトの逆推定などの場合において, 複素数や四元数に拡張された逆問題が考えられる. 高次元空間での逆問題においても, 不良設定性の問題は生じる. 本研究では, 四元数に拡張された逆問題を解くためのニューラルネット手法を考え, さらに不良設定性の解決のための方法について検討を行う.

3.1 四元数ネットワークインバージョン

本研究では, 入力, 荷重および出力を四元数に拡張した多層型ニューラルネットを考え, 順方向の学習済みネットワークを用いて逆問題を解くための四元数ネットワークインバージョンを考える⁹⁾. この方法は, 入力修正の原理を四元数領域に拡張し, 学習済み多層型四元数ニューラルネットを用いて, 与えられた四元数出力に対応する四元数入力を推定するものである. なお, 本研究では四元数入力の各部それぞれ独立にシグモイド関数を用いる各部独立型のニューロンを考える. すなわち, $\mathbf{s} = s_1 + is_2 + js_3 + ks_4$ に対して

$$\begin{aligned} f(\mathbf{s}) &= f(s_1) + if(s_2) + jf(s_3) + kf(s_4) \\ f(u) &= \frac{1 - e^{-u}}{1 + e^{-u}} \end{aligned} \quad (3)$$

とし, これを中間層・出力層に用いる 3 層型のネットワークを考える.

学習時は, 四元数に拡張された誤差逆伝搬法等により, 四元数学習入出力データを用いて四元数荷重の更新を行う. 逆推定時は, 学習時に得られた関係を固定したまま仮の四元数入力を与え, 得られた四元数出力から出力誤差関数を求める. ここで出力誤差関数は四元数出力の各部の 2 乗誤差とし,

$$E = \frac{1}{2} \sum_r (d_r - y_r) \cdot \overline{(d_r - y_r)} \quad (4)$$

の式に基づき更新する. ここで d_r および y_r は, それぞれ r 番目素子に与えられた出力とネットワーク出力を

表し, ともに四元数である. さらに r 番目出力素子からの誤差信号 δ_r および, q 番目中間素子出力からの誤差信号 δ_q はそれぞれ,

$$\begin{aligned} \delta_r &= (d_r - y_r) \cdot (1 - y_r) \cdot (1 + y_r), \\ \delta_q &= (1 - v_q) \cdot (1 + v_q) \cdot \sum_r \delta_r \cdot \overline{w_{qr}}, \end{aligned} \quad (5)$$

と計算できる. ここでは, “.” の記号によって四元数の各要素ごとの乗算を表すことにする. また, v_q および w_{qr} は, それぞれ q 番目中間素子からの出力および, q 番目中間素子から r 番目出力素子への荷重を表す. さらに, $\mathbf{1}$ は各要素が 1 の四元数とする. これらの誤差信号による p 番目入力素子の入力の修正量 δ_p は,

$$\delta_p = \sum_q \delta_q \cdot \overline{w_{pq}} \quad (6)$$

と書ける. ここで w_{pq} は, それぞれ p 番目入力素子から q 番目中間素子への荷重を表す. この修正量に基づいて,

$$\mathbf{x}_p^{\text{new}} = \mathbf{x}_p^{\text{old}} - \varepsilon_e \delta_p, \quad (7)$$

の式に基づき入力を更新する. ここで ε_e は入力更新の係数を意味する. この更新手順を繰り返すことで, 結果として, 学習によって得られた四元数荷重分布を使用して, 四元数出力から四元数入力を繰り返し法によって逆推定できる.

3.2 四元数ネットワークインバージョンの正則化

解空間に拘束条件を与える正則化の方法を, 四元数ネットワークインバージョンに適用することを提案する. エネルギー関数を, 出力誤差関数に正則化項を加え,

$$E = \frac{1}{2} \sum_r (d_r - y_r) \cdot \overline{(d_r - y_r)} + \lambda K \quad (8)$$

と定義する. ここで K を正則化汎関数とする.

逆問題の正則化汎関数として, 解の最大化や最小化がしばしば用いられる. しかし, ニューラルネット入力は一般に $[0, 1]$ や $[-1, 1]$ などの正規化された範囲の値を考慮することが多く, 入力を無限に最大・最小化する必要はない. そこで本研究では, 解を指定した値に近づける正則化を考える. 具体的な四元数入力の正則化汎関数としては, p 番目入力の目標値を t_p とすると,

$$K = \frac{1}{2} \sum_p (\mathbf{x}_p - t_p) \cdot \overline{(\mathbf{x}_p - t_p)} \quad (9)$$

として, 四元数入力を目標に近づけることを考える. 解の最大化および最小化について, それぞれ目標値を 0.5 および -0.5 として正則化を行う.

正則化係数は, $\lambda(n) = \lambda(0)(1 - n/N)$ のように繰り返し回数に従って減少させるものとする. $\lambda(0)$, n および N は, それぞれ正則化係数の初期値, 修正繰り返し回数およびその最大数を表す.

4 3次元逆写像問題

本研究では、四元数ネットワークインバージョンにおける正則化の効果を明らかにすることを目的とする。高次元領域にわたる逆問題として、3次元写像の逆推定問題を考える。この問題では、3次元空間内で与えられる写像関係を学習し、学習した関係を用いて、与えられた出力に対応する入力 \mathbf{x} の推定を行う。正則化の効果を示すために、一意性に関する不良設定性をもつ問題を考える。

一意性に関する不良設定性をもつ写像としては、3次元空間における回転移動を考える。具体的には、入力として3次元空間内に与えた点の、各部の絶対値をとった値をもつ点を出力とする変換を考える。すなわち、入力を $\mathbf{x} = ix_2 + jx_3 + kx_4$ 、出力を $\mathbf{y} = iy_2 + jy_3 + ky_4$ とするとき、

$$\mathbf{y} = i|x_2| + j|x_3| + k|x_4| \quad (10)$$

の関係を学習させ、学習済みのネットワークを用いて与えた出力 \mathbf{y} に対応する入力 \mathbf{x} を推定する。実際には $x_2, x_3, x_4 = \{-0.6, -0.3, 0.3, 0.6\}$ として64点の学習データを用意した。この問題は、Fig.2に示すように1つの出力に対し8つの入力に対応する不良設定問題である。

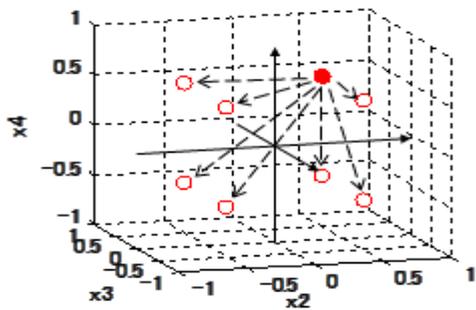


Fig. 2 Ill-posedness of the problem.

5 実験

上記の3次元データを用いて、3次元写像の逆推定問題を解く。ネットワークのパラメータはTable 1の通りである。学習データは式(10)に対して $x_2, x_3, x_4 = \{-0.6, -0.3, 0.3, 0.6\}$ として作成した64点の入出力データとし、学習は四元数に拡張された誤差逆伝搬法で行った。その結果、Table 1の最大学習回数20000回以内で学習誤差0.0001を下回り、十分に誤差が減少して入出力関係を学習できた。

続いて逆推定を行った。推定対象データは $y_2, y_3, y_4 = \{0.3, 0.6\}$ の8点の出力点とし、これを8回繰り返すことで64点の推定対象データとした。最初に、正則化の効果を示すために、正則化を適用する四元数の要素を、①なし、② x_2 のみ、③ x_2 および x_3 、④ x_2, x_3, x_4 の全て、として逆推定を行った。式(9)における正則化の目標値は $t = 0.5i + 0.5j + 0.5k$ とし、各要素を正の領域に向けるようにした。正則化係数は $\lambda = 0.01$ とした。推定結果の点の分布をFig.3に示す。結果の評価のために、推定された四元数入力の各要素のうち、正の値の解の割合を求めた。また、推定された値の平均誤差も求めた。これらの結果をTable 2に示す。

この結果によると、正則化を行わなかった場合は、正の値の解と負の値の解が推定されているが、正則化を適用することによって、適用した要素が100%正の値の解を推定している。また、いずれの場合も推定誤差はほぼ同じく小さな値となり、拘束条件を満たしつつ推定誤差を最小化していることがわかる。

さらに、正則化の係数による効果を確認するために、 λ の値を、①0.001、②0.003、③0.005、④0.008、⑤0.01、⑥0.1、⑦1.0、として逆推定を行った。式(9)における正則化の目標値は $t = 0.5i + 0.5j + 0.5k$ とし、 x_2 のみに正則化を適用した。結果の評価のために、推定された四元数入力の各要素のうち、正の値の解の割合を求めた。また、推定された値の平均誤差も求めた。これらの結果をTable 3に示す。

この結果によると、 $\lambda = 0.001 \sim 0.01$ においては、係数値の増加に従って x_2 の正の推定値の割合が増大している。また、推定誤差はいずれの場合も0.1の近くで、いずれの場合も最小化されている。それに対して $\lambda = 0.1, 1.0$ においては、 x_2 に関しては全て正の値であるが、推定誤差が大きくなっている。すなわち、正則化の効果が誤差の最小化を上回っている。

以上より、四元数ネットワークインバージョンの正則化法の効果と、正則化係数の変化による推定への影響を確認した。

Table 1 Network parameters

Parameters	Values
Number of input neurons	1
Number of hidden neurons	15
Number of output neurons	1
Training rate ε_t	0.0005
Input correcting rate ε_e	0.0005
Max. number of training epoch	20000
Max. number of estimating epoch	5000
Training error to be attained	0.0001
Estimation error to be attained	0.00001

Table 2 Rate of estimated positive components for each regularization, and estimated error.

Regularization	Rate of positive components (%)			Estimated error
	x_2	x_3	x_4	
N/A	35.9	70.3	43.8	0.0864
for x_2	100.0	54.7	50.0	0.0990
for x_2, x_3	100.0	100.0	56.3	0.0997
for x_2, x_3, x_4	100.0	100.0	100.0	0.0808

Table 3 Rate of estimated positive components for each regularization coefficient, and estimated error.

Regularization coefficient λ	Rate of positive components (%)			Estimated error
	x_2	x_3	x_4	
0.001	48.4	56.3	45.3	0.0885
0.003	67.2	46.9	45.3	0.1171
0.005	75.0	53.1	43.8	0.1168
0.008	93.8	46.9	43.8	0.1002
0.01	100.0	54.7	50.0	0.0990
0.1	100.0	53.1	46.9	0.1653
1.0	100.0	43.8	45.3	0.1628

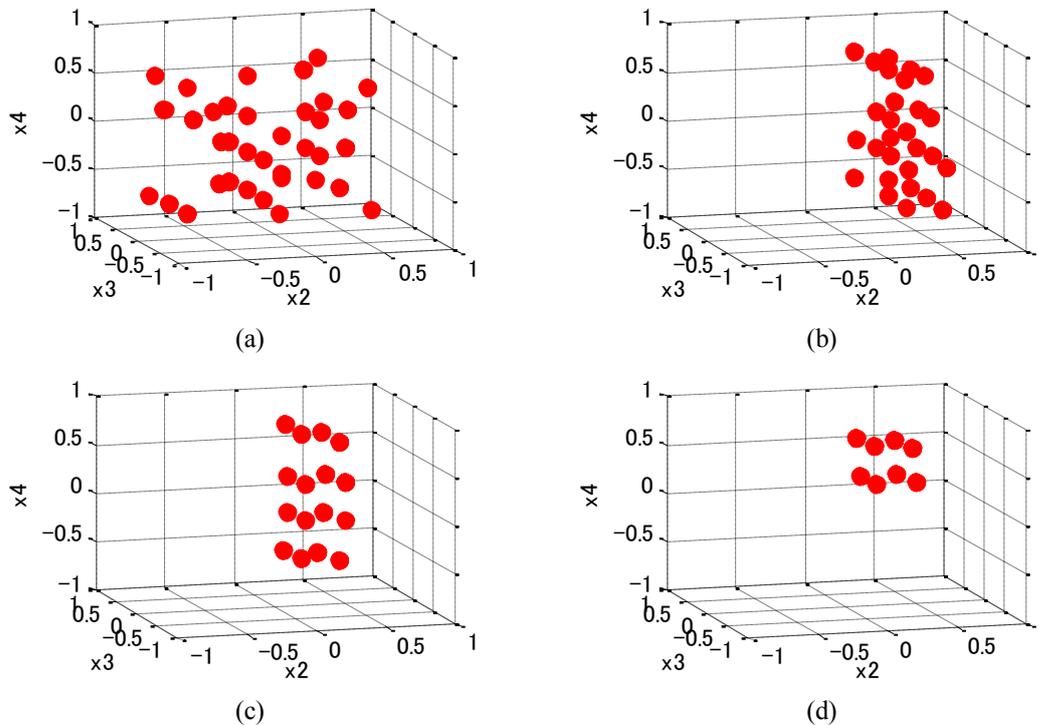


Fig. 3 Results of inverse estimation, (a) without regularization, (b) with regularizations for x_2 , (c) for x_2, x_3 , and (d) for x_2, x_3, x_4 .

6 まとめ

本研究では、四元数ニューラルネット上で逆問題を解くための四元数ネットワークインバージョン法について、解の一意性に関する不良設定性に対処するための正則化法の導入を提案した。さらに、3次元空間内での絶対値に基づく変換の不良設定逆問題によって動作を示すとともに、正則化係数の設定について検討を行った。結果として、正則化法を導入した四元数ネットワークインバージョン法の、一意性に関する不良設定逆問題への有効性を示すとともに、正則化係数の設定とその効果を明らかにした。

今後の課題として、解の一意性に加えて存在性、安定性について検討を行うことを考えている。さらに、コンピュータグラフィクス等実際の問題への適用を検討している。

参考文献

- 1) A. Hirose, "Complex-valued neural networks", Springer (2006)
- 2) T. Nitta (ed), "Complex-valued neural networks: Utilizing high-dimensional parameters", IGI-Global (2009)
- 3) J. Vince, "Geometric algebra for computer graphics", Springer, (2010)
- 4) 金谷, 「3D-CG プログラマーのための実践クォータニオン」, 工学社 (2004)
- 5) C. W. Grötsch, Inverse problems in the mathematical sciences, Informatica International (1993)
- 6) F. D. M. Neto and A. J. Neto, "An introduction to inverse problems with applications", Springer (2011)

- 7) A. Linden and J. Kindermann, "Inversion of multilayer nets," in Proc. Int. Joint Conf. on Neural Networks, 425/430 (1989)
- 8) 小川, 金田, 「複素逆問題の解法のための複素ネットワークインバージョン」, 信学論 D-II, Vol.J88-D-II, No.9, 1954/1962 (2005)
- 9) T. Iura and T. Ogawa, "Quaternion network inversion for solving inverse problems", Proc. of the SICE Annual Conf. 2012, 1802/1805 (2012)
- 10) A. N. Tikhonov and V. Y. Arsenin, "Solutions of ill-posed problems", Winston and Sons (1977)
- 11) T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks", Science, Vol.247, 978/982 (1990)
- 12) 小杉, 植本, 小川, 「ネットワークインバージョンにおける動的正則化」信学論 D-II, vol.J81-D-II, No.7, 1639/1646 (1998)
- 13) 小川, 高根, 「複素ネットワークインバージョンのための目標位相角に基づく正則化法」, 計測自動制御学会論文集 Vol.49, No.6, 658/660 (2013)
- 14) コンウェイ, スミス, 山田(訳), 「四元数と八元数」培風館 (2006)
- 15) D. J. H. Garling, "Clifford algebras: An introduction", Cambridge University Press (2011)
- 16) T. Nitta, "An extension of the back-propagation algorithm to quaternions", Proc. of Int'l Conf. on Neural Information Processing, Vol.1, 247/250 (1996)
- 17) N. Matsui, T. Isokawa, H. Kusamichi, F. Peper and H. Nishimura, "Quaternion neural network with geometrical operators", Journal of Intelligent and Fuzzy Systems, Vol.15, No.3-4, 149/164 (2004)

大規模 fan-in を有するニューロンを実現する多数決回路

○秋間 学尚¹ 片山 康弘² 佐藤 茂雄¹ 櫻庭 政夫¹ 中島 康治¹ (¹東北大学, ²東芝)

Majority Circuit for Artificial Neurons with Large Fan-in

* H. Akima¹, Y. Katayama², S. Sato¹, M. Sakuraba¹ and K. Nakajima¹ (¹Tohoku University, ²Toshiba)

Abstract— The function of an artificial neuron is described as majority logic if all synaptic weights are 1. Though a majority circuit implemented by logic gates is suitable for stable operation, it is not easy to achieve large fan-in because the number of transistors is proportional to the combination of the number of inputs N . In this paper, we propose a new majority circuit with large fan-in, in which the total number of transistors is proportional to N . The proposed circuit is robust because fluctuation of device parameters is automatically canceled out.

Key Words: Majority circuit, Large fan-in, Artificial neuron

1 はじめに

生体の脳は膨大な数のニューロンの並列動作で情報を処理している。ニューロンは、入力の重み付き総和がある閾値を超えたときに 0 または 1 を出力する素子としてモデル化される。ここで重みをすべて 1 と置き閾値を総入力数の半分とすれば、ニューロンを多数決素子とみなすことができる。多数決論理の入出力関係は以下の式で表される。

$$n = \sum_{i=1}^N x_i \quad x_i \in \{0, 1\}, \quad (1)$$
$$Y = \begin{cases} 1 & (n > N/2) \\ 0 & (n < N/2) \end{cases}$$

ここで、 N は総入力数で奇数とする。入力 x_i は 0 (Low) または 1 (High) をとり、その総和 n は High 入力数となる。 n が $N/2$ より大きいとき、すなわち High 入力数が過半数を超えると出力 Y は 1、それ以外は 0 となる。多数決論理を AND ゲートと OR ゲートで表現すると入力数の増加に伴いゲート数が増えるため[†]、これをデジタル回路で実装すると大きな回路面積が必要となる。

そこで我々は、MOS のアナログ特性を利用した多数決回路を開発した。本回路は、MOS をスイッチとしてだけでなく抵抗として使うことで(1)式に示した入力の加算を実装しており、デジタル回路による実装と比較して高速かつ小面積である。これまでに同様の原理で動作する多数決回路が提案されているが、アナログ回路一般において問題となるように回路を構成するデバイスのパラメータばらつきの影響を受けやすく、また動作マージンが小さいという課題があった。これらの課題を解決することで、人間のニューロンに匹敵する $10^3 \sim 10^4$ 程度の大きなファンインを持った多数決ニューロン回路の実現を目指す。

2 多数決回路の構成

まず、従来提案されている Output Wired Inverterを用いた多数決回路 (以下、OWI回路)¹⁾の構成をFig. 1に示す。これは、High (V_{DD}) または Low (GND) の二

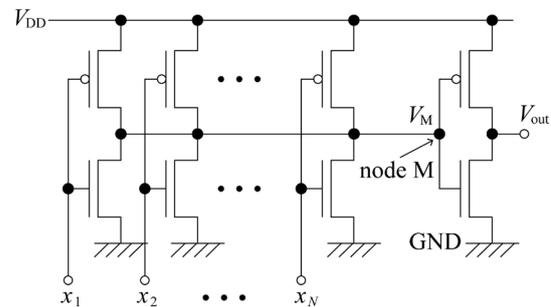


Fig. 1: Majority circuit composed of output-wired inverters (OWI majority circuit)

値をとる電圧 x_i を入力とする CMOS インバータを、多数決入力の総数 N だけ並列接続した構成になっている。入力 x_i が High のときには V_{DD} に接続された上段の pMOS が OFF、GND に接続された下段の nMOS は ON となり、 x_i が Low のときには pMOS が ON で nMOS が OFF となる。ON となった MOS は有限の値を持つ抵抗とみなせ、また全ての CMOS インバータの出力は中間ノード M に接続されている。よって、High 入力数が n のとき、ON となった $(N-n)$ 個の pMOS の並列合成抵抗と ON となった n 個の nMOS の並列合成抵抗によって抵抗分割された電圧が中間ノード M の電圧 V_M となる。そして、 V_M を入力とする最終段 CMOS インバータの出力 V_{out} が OWI 回路の出力となる。OWI 回路が多数決回路として働くためには、High 入力数 n が総入力数 N の半分となったときの V_M が最終段 CMOS インバータの反転閾値 V_{inv} と等しくなる必要がある[‡]。そのためには、MOS のゲート寸法や閾値電圧といったデバイスパラメータを正確に制御する必要があるが、製造プロセスのばらつきに起因するデバイスパラメータの設計値からのずれは避けられないため、大規模回路の実現は困難である。

Fig. 2 に我々の提案した多数決回路 (以下、CCI 回路) を示す。これは、CMOS インバータの代わりに、我々が CCI (Controlled-Current Inverter)^{2), 3), 4)} と名付けた回路要素から構成される。CCI は、ゲートに入力電圧 x_i を受け取る nMOS と pMOS (スイッチ用 MOS と呼ぶ)、

[‡] N が奇数ならば、 $n = (N-1)/2$ のときの V_M と $n = (N+1)/2$ のときの V_M の間に V_{inv} があればよい。

[†] 初段のゲート数は $C^N_{(N+1)/2}$ となる。

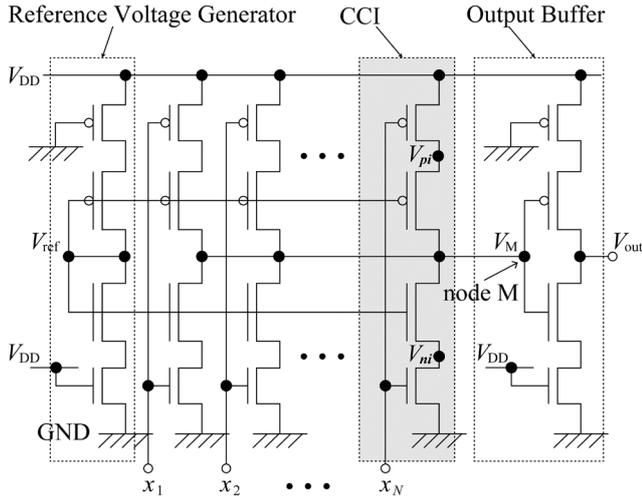


Fig. 2: Majority circuit composed of controlled-current inverters (CCI majority circuit)

及び基準電圧 V_{ref} でゲートがバイアスされた nMOS と pMOS (電流制御用 MOS と呼ぶ) の縦続接続となっている。全ての CCI の電流制御用 MOS が共通の V_{ref} でバイアスされることで、High 入力数 n が総入力数 N の半分となる極限で V_M は V_{ref} に一致する。すなわち、 $N \gg n$ に対して

$$V_{MM} \equiv \lim_{n \rightarrow N/2} V_M(n) \quad (2)$$

と定義すると、 $V_{MM} = V_{ref}$ となる。また、基準電圧発生回路 (Reference Voltage Generator) と出力バッファ (Output Buffer) はスイッチ用 MOS が同一のバイアス条件となっており、 V_{ref} は出力バッファの反転閾値 V_{inv} と一致する。すなわち $V_{MM} = V_{ref} = V_{inv}$ であり、High 入力数 n が過半数を超えたときに $V_M < V_{MM} = V_{inv}$ となり多数決出力 V_{out} は High (V_{DD}) となる。この V_{inv} と V_{MM} の一致条件は、nMOS 同士、pMOS 同士の特性が一致する限り保たれる。すなわち、デバイスパラメータの設計値からのずれが一樣、例えば、nMOS の閾値が一樣に設計値から数%ずれるような場合には多数決動作に影響はない。実際の LSI レイアウトにおいても、Fig. 2 に示した回路図のように nMOS と pMOS を規則正しく対称的に配置すれば、設計値からのずれが nMOS、pMOS それぞれに対して一樣に生じるという仮定は妥当である。CCI 回路のロバスト性の高さは、大きな fan-in、すなわち多くの入力を受け付けることのできる大規模な多数決回路を実現する上で重要である。

CCI 回路のもう一つの特徴は、動作マージンが大きいことである。 V_{M0} と V_{M1} を

$$\begin{aligned} V_{M0} &\equiv V_M \left(n = \frac{N-1}{2} \right), \\ V_{M1} &\equiv V_M \left(n = \frac{N+1}{2} \right) \end{aligned} \quad (3)$$

と定義すると、動作マージンは以下の式で定義される。

$$\begin{aligned} \text{margin} &= \min(\text{margin}^+, \text{margin}^-), \\ \text{margin}^- &\equiv V_{M0} - V_{ref}, \quad \text{margin}^+ \equiv V_{ref} - V_{M1}. \end{aligned} \quad (4)$$

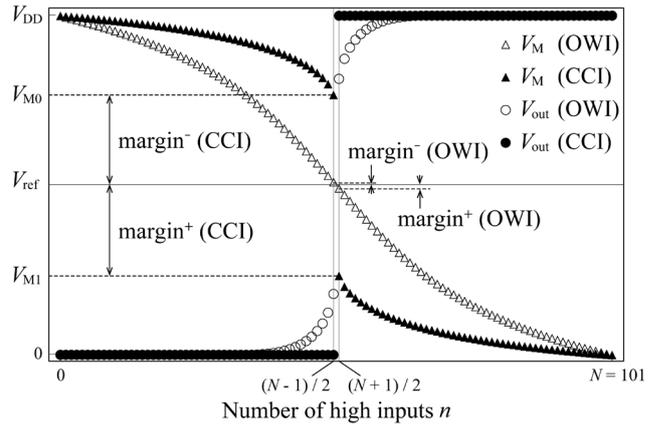


Fig. 3: Simulation result of OWI and CCI circuit ($N = 101$)

ここで、 $\min(A, B)$ は A と B のうち小さい方を表す。もしノイズによる V_M の変動が動作マージンよりも大きければ、 $n = N/2$ 近傍で V_{out} が不安定になり多数決回路として安定に動作しない。

総入力数 $N = 101$ の OWI 回路と CCI 回路について、High 入力数 n に対する中間ノード電圧 V_M と出力電圧 V_{out} の関係をシミュレーションした結果を Fig. 3 に示す。デバイスパラメータは OWI 回路の最終段 CMOS インバータの反転閾値が V_{ref} と等しくなるように選んだ。 $n \cong N/2$ で OWI 回路の V_M が僅かしか変化していないのに対して、CCI 回路では V_M が大きく変化しているため、CCI 回路は OWI 回路よりも大きな動作マージンを持つ。この大きな動作マージンは、CCI を構成する MOS の非線形特性に起因する。 $n \cong N/2$ では電流制御用 MOS が飽和領域で動作し、高いドレイン抵抗を持つ。Fig. 3 に示したような $n \cong N/2$ での V_M の急峻な変化は、この高いドレイン抵抗によるものである。一方、OWI 回路を構成する MOS は $n \cong N/2$ では線形領域で動作する。線形領域におけるドレイン抵抗は飽和領域のそれよりも小さいので、 V_M の変化は小さい。

2.1 CCI 回路の動作マージン解析

以下では、CCI 回路の動作マージンを総入力数 N の関数として導出する。CCI 回路中の一つの CCI について、入力 x_i が V_{DD} のときの nMOS 側の部分回路 (Fig. 4) について考察する。 $n = N/2$ 近傍で V_M は V_{ref} に近づくので、 V_{ref} と V_M の差、すなわち Fig. 4 上側の電流制御用 nMOS (c-nMOS) のゲート電位とドレイン電位の差は nMOS の閾値以下となり、c-nMOS は飽和領域で動作する。

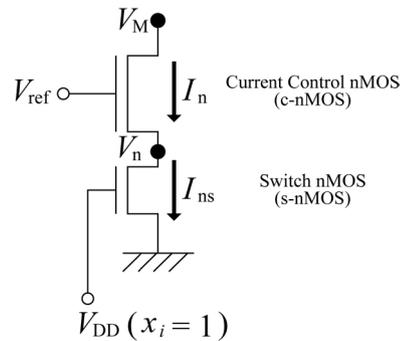


Fig. 4: Sub-circuit of a CCI

Fig. 4下側のスイッチ用nMOS (s-nMOS) のゲートには V_{DD} が印可されており, s-nMOSのオン抵抗が小さければs-nMOSのドレイン電位 V_n は V_{DD} より十分小さいと考えられるので, s-nMOSは線形領域で動作する. c-nMOSのソース (電位 V_n) が接地されておらず基板バイアス効果を考慮する必要があることに注意して, 各々のnMOSを流れるドレイン電流 I_n , I_{ns} を書き下すと,

$$I_n = \frac{1}{2} \beta_n (V_{ref} - V_n - V_{t0} - \delta_n) \{1 + \lambda_n (V_M - V_n)\}, \quad (5)$$

$$I_{ns} = \frac{1}{2} \beta_{ns} \{2(V_{DD} - V_{t0})V_n - V_n^2\} (1 + \lambda_{ns} V_n) \quad (6)$$

となる⁹⁾. V_{t0} は基板バイアスがないときのnMOSの閾値電圧, λ はチャネル長変調係数, $\beta_n = \mu_n C_{ox} W/L$ で μ_n は電子の移動度, C_{ox} はゲート容量, W はチャネル幅, L はチャネル長である. 基板バイアス効果による閾値電圧の変化分 δ_n は次式で与えられる.

$$\delta_n = \gamma (\sqrt{\phi + V_n} - \sqrt{\phi}). \quad (7)$$

ここで, γ は基板バイアス効果係数, ϕ は基板の表面ポテンシャルである.

今, $n \equiv N/2$ で $|V_M - V_{ref}| \ll 1$ であるから, I_n を次式で近似できる.

$$I_n \approx I_{n0} + \left. \frac{\partial I_n}{\partial V_M} \right|_{V_M=V_{ref}} (V_M - V_{ref}). \quad (8)$$

I_{n0} は $V_M = V_{ref}$ のときの I_n である. $\partial I_n / \partial V_M$ を求めるために(5)式の両辺を V_M で偏微分して,

$$\begin{aligned} \frac{\partial I_n}{\partial V_M} &\approx \frac{1}{2} \beta_n \{-2(V_{ref} - V_n - V_{t0} - \delta_n) \frac{\partial V_n}{\partial V_M} (1 + \frac{\partial \delta_n}{\partial V_n}) \\ &\quad + \lambda_n \left(1 - \frac{\partial V_n}{\partial V_M}\right) (V_{ref} - V_n - V_{t0} - \delta_n)^2\} \\ &\approx I_n \left\{ \lambda_n - \frac{2 \left(1 + \frac{\partial \delta_n}{\partial V_n}\right)}{(V_{ref} - V_n - V_{t0} - \delta_n)} \frac{\partial V_n}{\partial V_M} \right\} \end{aligned} \quad (9)$$

を得る. (9)式の導出にあたって, チャネル長変調係数が十分小さいこと, すなわち,

$$|\lambda_n| \ll 1, \quad |\lambda_{ns}| \ll 1 \quad (10)$$

及び,

$$\left| \frac{\partial V_n}{\partial V_M} \right| \ll 1 \quad (11)$$

を仮定した. (9)式中の $\partial \delta_n / \partial V_n$ は(7)式の両辺を V_n で偏微分して

$$\frac{\partial \delta_n}{\partial V_n} = \frac{\gamma}{2} (\phi + V_n)^{-1/2} \quad (12)$$

となる. 次に, (9)式中の $\partial V_n / \partial V_M$ を求める. まず, キルヒホッフの電流則から

$$I_n = I_{ns} \quad (13)$$

が得られる. (13)式に(5)式と(6)式を代入して両辺を V_n で微分し, (10)式の近似を用いて整理すると

$$-\beta_n (V_{ref} - V_n - V_{t0} - \delta_n) \left(1 + \frac{\partial \delta_n}{\partial V_n}\right) + I_n \lambda_n \left(\frac{\partial V_M}{\partial V_n} - 1\right)$$

$$\approx \beta_{ns} (V_{DD} - V_n - V_{t0}) + I_n \lambda_{ns},$$

$$\begin{aligned} \frac{\partial V_M}{\partial V_n} - 1 &\approx \frac{1}{\lambda_n} \left\{ \frac{\beta_n}{I_n} (V_{ref} - V_n - V_{t0} - \delta_n) \left(1 + \frac{\partial \delta_n}{\partial V_n}\right) \right. \\ &\quad \left. + \frac{\beta_{ns}}{I_n} (V_{DD} - V_n - V_{t0}) + I_n \lambda_{ns} \right\}, \end{aligned}$$

$$\begin{aligned} \therefore \frac{\partial V_n}{\partial V_M} &\approx \lambda_n \left\{ \frac{\beta_n}{I_n} (V_{ref} - V_n - V_{t0} - \delta_n) \left(1 + \frac{\partial \delta_n}{\partial V_n}\right) \right. \\ &\quad \left. + \frac{\beta_{ns}}{I_n} (V_{DD} - V_n - V_{t0}) + \lambda_n + \lambda_{ns} \right\}^{-1} \end{aligned} \quad (14)$$

となる. (8)式に(9), (12), (14)式を代入することで, $n \equiv N/2$ でCCIのnMOS側を流れる電流を求めることができる. 同様の手法でpMOS側を流れる電流 I_p を求めることができ, 結果のみを書き下すと以下ようになる.

$$I_p \approx I_{p0} + \left. \frac{\partial I_p}{\partial V_M} \right|_{V_M=V_{ref}} (V_M - V_{ref}), \quad (15)$$

$$\frac{\partial I_p}{\partial V_M} \approx -I_p \left\{ \lambda_p - \frac{2 \left(1 + \frac{\partial \delta_p}{\partial V_p}\right)}{(V_{ref} - V_p - V_{tp0} - \delta_p)} \frac{\partial V_p}{\partial V_M} \right\}, \quad (16)$$

$$\begin{aligned} \frac{\partial V_p}{\partial V_M} &\approx -\lambda_p \left\{ \frac{\beta_p}{I_p} (V_{ref} - V_p - V_{tp0} - \delta_p) \left(1 + \frac{\partial \delta_p}{\partial V_p}\right) \right. \\ &\quad \left. - \frac{\beta_{ps}}{I_p} (V_p + V_{tp0}) - \lambda_p - \lambda_{ps} \right\}^{-1}, \end{aligned} \quad (17)$$

$$\frac{\partial \delta_p}{\partial V_p} = \frac{\gamma}{2} (\phi + V_p - V_{DD})^{-1/2}. \quad (18)$$

CCI回路全体を考えて中間ノードMについてキルヒホッフの電流則を適用すると,

$$nI_n = -(N-n)I_p \quad (19)$$

が得られる. ここで, 電流の向きはドレインからソー

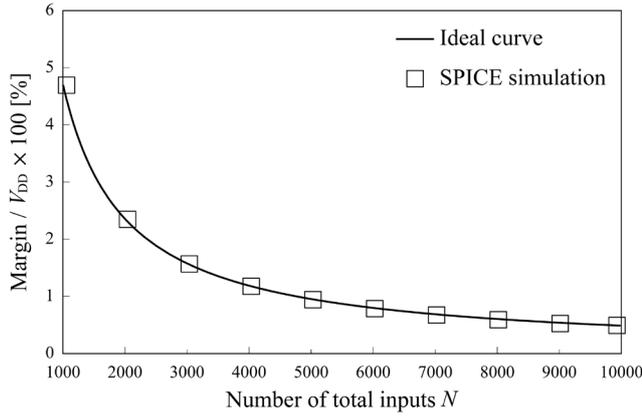


Fig.5: Margin of a CCI majority circuit.

スに流れる方向，すなわち中間ノードMからnMOSを通してGNDに流れる方向と，中間ノードMからpMOSを通して V_{DD} に流れる方向を正と定義した．(19)式に(8)式と(15)式を代入して整理すると

$$V_M - V_{ref} = \frac{N - 2n}{n\Delta_n - (N - n)\Delta_p} \quad (20)$$

となる．ここで，

$$\begin{aligned} I_{n0} &\equiv -I_{p0} \equiv I_0 \\ \Delta_n &\equiv \left. \frac{1}{I_0} \frac{\partial I_n}{\partial V_M} \right|_{V_M=V_{ref}}, \\ \Delta_p &\equiv - \left. \frac{1}{I_0} \frac{\partial I_p}{\partial V_M} \right|_{V_M=V_{ref}} \end{aligned} \quad (21)$$

と定義した．(20)式の n にそれぞれ $(N-1)/2$ と $(N+1)/2$ を代入して

$$\begin{aligned} \text{margin}^- &= \frac{-2}{\Delta_n(1-N) + \Delta_p(1+N)}, \\ \text{margin}^+ &= \frac{2}{\Delta_n(1+N) + \Delta_p(1-N)} \end{aligned} \quad (22)$$

を得る．もし $\Delta_n = -\Delta_p \equiv \Delta$ であれば

$$\text{margin}^- = \text{margin}^+ = \frac{1}{N \cdot \Delta} \quad (23)$$

となる．デバイスパラメータとして $V_{tn0} = -V_{tp0} = 0.7$ [V], $\gamma = 0.5$ [V^{1/2}], $\lambda_n = \lambda_{ns} = \lambda_p = \lambda_{ps} = 0.01$ [V⁻¹]を仮定し， $\Delta_n = -\Delta_p = 7.1 \times 10^{-3}$ [V⁻¹]として計算した総入力数 N に対する動作マージンの変化をFig. 5に示す．(23)式を用いて計算した結果 (Ideal curve) は，シミュレーション結果 (SPICE simulation) と良く一致している．これらのデバイスパラメータの場合，例えば V_{DD} の4%以上の動作マージンを要求すると， $N=1000$ 程度までの入力を受け付けることができる．

3 まとめ

CCI (Controlled-Current Inverter) を用いた多数決回路 (CCI回路) を提案し，その動作マージンの解析を

行った．CCI回路はデバイスパラメータのずれに対してロバストであり，MOSの基本的なモデルを用いた場合には総入力数 $N=1000$ で電源電圧の4%程度の動作マージンを持つことを明らかにした．

今後の課題は，精密なMOSのモデルを用いた動作マージンの再評価と，シナプス荷重値の実現である．本報告の解析ではMOSの基本的なモデル式を用いたが，最先端のプロセスで作成したゲート長が数十～数百nmのMOSの解析にはより精密なモデルが必要になる．シナプス荷重値の実現方法として，スイッチ用MOSと V_{DD} 及びGNDの間に可変抵抗を挿入し，抵抗分割のバランスを制御することが考えられる．可変抵抗としては，例えばMTJ (Magnetic Tunnel Junction) 素子⁶⁾などの不揮発性素子の利用を検討する．

参考文献

- 1) Chang Long Lee, Chein-Wei Jen: "A novel design of binary majority gate and its application to median filtering", in *Proc. 1990 IEEE Int. Symp. on Circuits and Systems (ISCAS' 90)*, vol.1, pp.570-573 (1990).
- 2) 片山 康弘, 鈴木 康介, 佐藤 茂雄, 中島 康治: 「大きな fan-in を実現する高速 CMOS 多数決回路」, 電子情報通信学会エレクトロニクスソサイエティ大会, 講演論文集, C-12-28 (1999).
- 3) Y. Katayama, K. Suzuki, S. Sato, K. Nakajima, "Implementation of a large fan-in majority circuit", in *Proc. 2000 Int. Symp. on Nonlinear Theory and its Applications (NOLTA2000)*, pp.413-416 (2000).
- 4) 中島 康治, 佐藤 茂雄: 「CMOS 多数決回路」, 特許第 3297738 号 (2002).
- 5) R. Jacob Baker, "CMOS Circuit Design, Layout, and Simulation, 3rd Edition", IEEE Press Series on Microelectronic Systems (2010).
- 6) S. Fukami, N. Ishiwata, N. Kasai, M. Yamanouchi, H. Sato, S. Ikeda, H. Ohono, "Scalability Prospect of Three-Terminal Magnetic Domain-Wall Motion Device", *IEEE Trans. Magn.* vol.48, pp.2152-2157 (2012).

リカレントクリフォードニューラルネットワークのダイナミクス

— エネルギー関数とその存在条件 —

○黒江康明 Vagif Hasanov 飯間等 (京都工芸繊維大学)

Dynamics of Recurrent Clifford Neural Networks - Energy Functions and Their Existence Conditions -

Y. Kuroe, V. Hasanov and H. Iima (Kyoto Institute of Technology)

Abstract— Recently, models of neural networks in the real domain have been extended into the high dimensional domain such as the complex number domain and quaternion number domain, and several high-dimensional models have been proposed. These extensions are generalized by introducing Clifford algebra (geometric algebra). We have already proposed models of recurrent Clifford neural networks, which are extensions of the real-valued Hopfield type neural networks to the domain defined by Clifford algebra. We have studied dynamics of the models from the point view of existence conditions of energy functions for some classes of the Hopfield type Clifford neural networks. In this paper we derive existence conditions of energy functions for a general class of the Hopfield-type Clifford neural networks.

Key Words: Clifford algebra, Recurrent neural network, Hopfield neural network, Dynamics, Energy function

1 はじめに

近年、複素数(複素信号)を扱う事ができる複素数値化されたニューラルネットワーク(以後 NN)のモデルがいくつか提案され、その情報処理能力、学習法や応用などに関する研究が盛んに行われている^{1,2)}。また最近では、さらに高次元化された四元数、八元数などを用いた NN のモデルも提案され、活発に研究が行われている。これら高次元の表現は、クリフォード代数(Clifford algebra)表現として一般化される。近年、クリフォード代数は様々な幾何学的問題の強力かつ実用的な表現や解法を与えるものとして非常に注目されており、様々なサイエンスや工学の分野への応用の研究が盛んになってきている³⁾。ニューロコンピューティングにおいても、クリフォード代数表現を導入することにより豊かな表現能力を持ち高度な計算知能を実現できるものとして非常に期待されている⁴⁾。

このような背景の下、著者らは、リカレントクリフォードニューラルネットワークのモデルを提案し、そのダイナミクス、特にエネルギー関数の存在性について議論してきた。特にいくつかのサブクラス、すなわち、hyperbolic 数、複素数、dual 数および四元数表現の NN に対し具体的にエネルギー関数の存在条件を導いてきた^{5,6,7,8,9)}。本稿では、一般的なリカレントクリフォード NN に対し、そのエネルギー関数の存在条件を導き、実際にエネルギー関数を構成する¹⁰⁾。またこの条件より、これまで導いてきたサブクラスの NN に対する条件が導けることを示す。

2 クリフォード代数の概要

本稿では、実数体 \mathbb{R} 上で定義される有限次元のクリフォード代数を対象とする。本章では本稿で必要とするクリフォード代数の概要を、主として文献 11) に基づいて説明する。

2.1 定義

$\mathbb{R}^{p,q,r}$ を実数体 \mathbb{R} 上の $(p+q+r)$ 次元のベクトル空間とし、 $\mathbb{R}^{p,q,r}$ に二次形式(可換なスカラー積) :

$\mathbb{R}^{p,q,r} \times \mathbb{R}^{p,q,r} \rightarrow \mathbb{R}$ が定義されているとする。また $\mathbb{R}^{p,q,r}$ を以下のような $\mathbb{R}^{p,q,r}$ の基底の集合とする。

$$\overline{\mathbb{R}^{p,q,r}} := \{e_1, \dots, e_p, e_{p+1}, \dots, e_{p+q}, e_{p+q+1}, \dots, e_{p+q+r}\} \subset \mathbb{R}^{p,q,r} \quad (1)$$

ここで $\{e_i\}$ は次の性質を持つとする。

$$e_i \cdot e_j = \begin{cases} +1, & 1 \leq i = j \leq p, \\ 1, & p < i = j \leq p+q, \\ 0, & p+q < i = j \leq p+q+r, \\ 0, & i \neq j \end{cases} \quad (2)$$

二次形式(スカラー積)を持つベクトル空間は二次空間(quadratic space)と呼ばれる。ここで、基底の自分自身のスカラー積の値が 1 だけでなく、-1 および 0 をとることを許していることに注意されたい。

$\mathbb{R}^{p,q,r}$ 上のクリフォード代数を $\mathbb{G}_{p,q,r}$ と表すことにする。 $\mathbb{G}_{p,q,r}$ は、クリフォード積(Clifford product)と呼ばれる積 \circ を導入することにより、以下のように定義される。

[クリフォード代数 $\mathbb{G}_{p,q,r}$ の定義]

$\mathbb{G}_{p,q,r}$ は、次の性質を満たすとき、 $\mathbb{R}^{p,q,r}$ 上のクリフォード代数と呼ばれる。

- $\mathbb{G}_{p,q,r}$ は、実数体 \mathbb{R} とベクトル空間 $\mathbb{R}^{p,q,r}$ を、その部分空間として含む。
- $\mathbb{G}_{p,q,r}$ は、加算 $+$ およびスカラー $(\alpha \in \mathbb{R})$ との積に関してベクトル空間をなす。
- $\mathbb{G}_{p,q,r}$ に次の性質を満たす積 \circ が存在する。

1. $\mathbb{G}_{p,q,r}$ は、積 \circ に対し閉じている。

$$\mathbf{a} \circ \mathbf{b} \in \mathbb{G}_{p,q,r}, \quad \forall \mathbf{a}, \mathbf{b} \in \mathbb{G}_{p,q,r}$$

2. 結合則:

$$(\mathbf{a} \circ \mathbf{b}) \circ \mathbf{c} = \mathbf{a} \circ (\mathbf{b} \circ \mathbf{c}), \quad \forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{G}_{p,q,r}$$

3. 分配則：

$$\mathbf{a} \circ (\mathbf{b} + \mathbf{c}) = \mathbf{a} \circ \mathbf{b} + \mathbf{a} \circ \mathbf{c}, \quad \forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{G}_{p,q,r}$$

4. スカラーとの積：

$$\alpha \circ \mathbf{a} = \mathbf{a} \circ \alpha = \alpha \mathbf{a}, \quad \forall \mathbf{a} \in \mathbb{G}_{p,q,r}, \forall \alpha \in \mathbb{R}$$

5. もし $\mathbf{a} \in \mathbb{R}^{p,q,r} \subset \mathbb{G}_{p,q,r}$ なら次式を満たす.

$$\mathbf{a} \circ \mathbf{a} = \mathbf{a} \quad \mathbf{a} \in \mathbb{R} \quad (3)$$

最後の5は、もし \mathbf{a} がベクトル空間 $\mathbb{R}^{p,q,r}$ の要素ならそれ自身のクリフォード積はそれ自身のスカラー積に一致することを意味しており、ベクトル空間 $\mathbb{R}^{p,q,r}$ とクリフォード代数 $\mathbb{G}_{p,q,r}$ を繋ぐ重要な性質である。また定義から分かるように、クリフォード代数 $\mathbb{G}_{p,q,r}$ においては、クリフォード積 \circ に可換であることは要請していない。

2.2 基本的な性質と基底

クリフォード代数 $\mathbb{G}_{p,q,r}$ の要素は、 $\mathbb{R}^{p,q,r}$ の要素がベクトルと呼ばれるのに対し、マルチベクトル (multivector) と呼ばれる。任意のマルチベクトル $\mathbf{a}, \mathbf{b} \in \mathbb{G}_{p,q,r}$ に対し、クリフォード積 $\mathbf{a} \circ \mathbf{b}$ は、その対称な部分と反対称な部分の和として

$$\mathbf{a} \circ \mathbf{b} = \frac{1}{2}(\mathbf{a} \circ \mathbf{b} + \mathbf{b} \circ \mathbf{a}) + \frac{1}{2}(\mathbf{a} \circ \mathbf{b} - \mathbf{b} \circ \mathbf{a})$$

と表される。特に \mathbf{a} と \mathbf{b} がベクトル、すなわち $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{p,q,r}$ ならば、(3) 式より次の関係が成立する。

$$(\mathbf{a} + \mathbf{b}) \circ (\mathbf{a} + \mathbf{b}) = (\mathbf{a} + \mathbf{b}) \circ (\mathbf{a} + \mathbf{b})$$

$$\Leftrightarrow \mathbf{a} \circ \mathbf{a} + \mathbf{a} \circ \mathbf{b} + \mathbf{b} \circ \mathbf{a} + \mathbf{b} \circ \mathbf{b} = \mathbf{a} \circ \mathbf{a} + 2\mathbf{a} \circ \mathbf{b} + \mathbf{b} \circ \mathbf{b}$$

$$\Leftrightarrow \frac{1}{2}(\mathbf{a} \circ \mathbf{b} + \mathbf{b} \circ \mathbf{a}) = \mathbf{a} \circ \mathbf{b}$$

したがって、この場合、対称な成分はスカラー積に一致する。ここで、反対称な成分を

$$\mathbf{a} \wedge \mathbf{b} := \frac{1}{2}(\mathbf{a} \circ \mathbf{b} - \mathbf{b} \circ \mathbf{a})$$

と表すことにすると

$$\mathbf{a} \circ \mathbf{b} = \mathbf{a} \circ \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$$

と表せる。積 \wedge は、外積 (outer product) あるいはくさび積 (wedge product) と呼ばれる。特に、 $\mathbb{R}^{p,q,r}$ の基底ベクトル $\mathbf{e}_i, \mathbf{e}_j$ に対しては、(2) 式より $\mathbf{e}_i \circ \mathbf{e}_j = 0$ ($i \neq j$) が成立するので、 $\mathbf{e}_i \circ \mathbf{e}_j$ は、

$$\mathbf{e}_i \circ \mathbf{e}_j = \mathbf{e}_i \wedge \mathbf{e}_j$$

となり、

$$\mathbf{e}_i \circ \mathbf{e}_j = \mathbf{e}_j \circ \mathbf{e}_i \quad (4)$$

が成立する。

ここでクリフォード代数 $\mathbb{G}_{p,q,r}$ の基底の生成について説明する。以下では簡単のためクリフォード積である $\mathbf{a} \circ \mathbf{b}$ を、 \circ を省略して $\mathbf{a}\mathbf{b}$ と表す。また結合則を満たすので、 $(\mathbf{a} \circ \mathbf{b}) \circ \mathbf{c}$ あるいは $\mathbf{a} \circ (\mathbf{b} \circ \mathbf{c})$ を単に $\mathbf{a}\mathbf{b}\mathbf{c}$

と表す。また \prod で、複数の要素のクリフォード積を表す。たとえば、 $\prod_{i=1}^3 \mathbf{a}_i = \mathbf{a}_1 \circ \mathbf{a}_2 \circ \mathbf{a}_3$ である。 $\mathbb{G}_{p,q,r}$ の基底の生成には、基底ブレード (basis blade) と呼ばれるベクトル空間 $\mathbb{R}^{p,q,r}$ の相異なる任意の数の基底のクリフォード積が重要な役割を果たす。まず基底ブレードの定義を与える。 \mathbb{A} を順序集合とし、 $\mathbb{A}[i]$ を \mathbb{A} の第 i 番目の要素とする。たとえば $\mathbb{A} = \{2, 3, 1\}$ とすると、 $\mathbb{A}[2] = 3$ である。このとき、 $\mathbb{G}_{p,q,r}$ の基底ブレードは、 $\mathbb{R}^{p,q,r}$ の基底の集合 $\overline{\mathbb{R}^{p,q,r}}$ の相異なるいくつかの要素のクリフォード積として定義される。したがって \mathbb{A} を $\mathbb{A} \subset \{1, 2, \dots, p+q+r\}$ として、基底ブレード $\mathbf{e}_{\mathbb{A}}$ は次のように定義される。

$$\mathbf{e}_{\mathbb{A}} = \prod_{i=1}^{|\mathbb{A}|} \overline{\mathbb{R}^{p,q,r}}[\mathbb{A}[i]] \quad (5)$$

ここで、 $\mathbb{A}[i]$ の定義と同様、 $\overline{\mathbb{R}^{p,q,r}}[\mathbb{A}[i]]$ は $\overline{\mathbb{R}^{p,q,r}}$ の第 $\mathbb{A}[i]$ 番目の要素である。また、 $|\mathbb{A}|$ は集合 \mathbb{A} の要素の数を表す。各基底ブレード $\mathbf{e}_{\mathbb{A}}$ におけるクリフォード積により演算される要素の数、すなわち $|\mathbb{A}|$ は、基底ブレードの grade と呼ばれる。たとえば $\mathbb{A} = \{2, 3, 1\}$ ならば、 $\mathbf{e}_{\mathbb{A}} = \mathbf{e}_2 \mathbf{e}_3 \mathbf{e}_1$ で、この grade は 3 である。

(1) 式で表される $\mathbb{R}^{p,q,r}$ の基底は $p+q+r$ 個あるので、これらの全ての組み合わせのクリフォード積のうち独立な組み合わせのクリフォード積の数は 2^{p+q+r} 個である。 $\mathbb{G}_{p,q,r}$ の基底は、これら独立な 2^{p+q+r} 個の基底ブレードから構成される。ここで、 $\mathbb{I} = \{1, 2, \dots, p+q+r\}$ とし、 $\mathcal{P}[\mathbb{I}]$ を \mathbb{I} のべき集合、また $\mathcal{P}_0[\mathbb{I}]$ を順序づけられた \mathbb{I} のべき集合とする。たとえば、 $\mathbb{I} = \{1, 2, 3\}$ とすると、

$$\mathcal{P}_0[\mathbb{I}] = \{\{\emptyset\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

である。ただし、 \emptyset は空集合である。これより、 $\mathbb{G}_{p,q,r}$ の基底 $\overline{\mathbb{G}}_{p,q,r}$ は、次のように定義される

$$\overline{\mathbb{G}}_{p,q,r} := \{\mathbf{e}_{\mathbb{A}} : \mathbb{A} \in \mathcal{P}_0[\mathbb{I}]\}$$

ただし、 $\mathbf{e}_{\emptyset} = 1 \in \mathbb{R}$ としている。たとえば、 $p+q+r = 3$ として $\overline{\mathbb{G}}_3 := \overline{\mathbb{G}}_{p,q,r}$ と表すと、この正準基底 $\overline{\mathbb{G}}_3$ は次のようになる。

$$\overline{\mathbb{G}}_3 = \{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_1 \mathbf{e}_2, \mathbf{e}_1 \mathbf{e}_3, \mathbf{e}_2 \mathbf{e}_3, \mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3\}$$

クリフォード代数 $\mathbb{G}_{p,q,r}$ の要素、すなわちマルチベクトルは、以上のように定義された基底の線形結合として表される。すなわち、 $\mathbf{a} \in \mathbb{G}_{p,q,r}$ は、 $a^{(i)} \in \mathbb{R}$ として次のように表される。

$$\mathbf{a} = \sum_{i=1}^{2^{p+q+r}} a^{(i)} \overline{\mathbb{G}}_{p,q,r}[i] \quad (6)$$

また、 $\mathbf{a} \in \mathbb{G}_{p,q,r}$ の絶対値 (modulus) $|\mathbf{a}|$ は次のように定義される。

$$|\mathbf{a}| = \left(\sum_{i=1}^{2^{p+q+r}} a^{(i)2} \right)^{1/2}$$

3 リカレントクリフォードニューラルネットワークのモデル

本章では、クリフォード代数で表現されたリカレント NN のモデルを示す。特に、実数の連続時間 Hopfield 型 NN のモデルをクリフォード代数表現に拡張したものを 3 種類示す。1 つ目モデルは、以下のような微分方程式で表されるモデルであり、Hopfield が実数のモデルとして提案した NN モデルの自然な拡張となっている。

$$\begin{cases} \tau_i \frac{du_i}{dt} = u_i + \sum_{j=1}^n w_{ij} v_j + b_i \\ v_i = f(u_i) \quad (i = 1, 2, \dots, n) \end{cases} \quad (7)$$

ここで、 n はネットワークを構成するニューロンの個数、 u_i, v_i は、時刻 t における i 番目のニューロンの状態と出力、 b_i は i 番目のニューロンのしきい値、 w_{ij} は j 番目のニューロンから i 番目のニューロンへの結合荷重、 τ_i は i 番目のニューロンの時定数である。このモデルにおいて、 u_i, v_i, b_i, w_{ij} はすべてクリフォード代数 $\mathbb{G}_{p,q,r}$ の要素 (マルチベクトル)、すなわち $u_i \in \mathbb{G}_{p,q,r}, v_i \in \mathbb{G}_{p,q,r}, b_i \in \mathbb{G}_{p,q,r}, w_{ij} \in \mathbb{G}_{p,q,r}$ である。時定数 τ_i のみ正の実数、すなわち $\tau_i \in \mathbb{R}, \tau_i > 0$ とする。また $w_{ij} v_j$ は、 $\mathbb{G}_{p,q,r}$ における w_{ij} と v_j のクリフォード積である。 $f(\cdot)$ はニューロンの活性化関数であり、 $\mathbb{G}_{p,q,r} \rightarrow \mathbb{G}_{p,q,r}$ なる非線形関数である。また時間微分 du_i/dt は、 u_i の各要素ごとの時間微分、すなわち

$$\frac{d}{dt} u_i(t) := \sum_{i=1}^{2^{p+q}} \frac{d}{dt} u^{(i)}(t) \overline{\mathbb{G}}_{p,q}[i]$$

と定義している。なお、前章では $\mathbb{G}_{p,q,r}$ の要素 (マルチベクトル) をすべて太活字 (bold face) で表していたが、このモデルにおいては普通字体で表しており、本稿の以下ではすべてこのように表す。

$\mathbb{G}_{p,q,r}$ の積は一般には可換でないので、結合荷重係数とニューロンの出力との積 $w_{ij} v_j$ の順序を逆、すなわち $v_j w_{ij}$ としたモデルは (7) と異なるモデルとなる。したがって、二つ目のモデルとして、次の微分方程式で表されるモデルを考える。

$$\begin{cases} \tau_i \frac{du_i}{dt} = u_i + \sum_{j=1}^n v_j w_{ij} + b_i \\ v_i = f(u_i) \quad (i = 1, 2, \dots, n) \end{cases} \quad (8)$$

またニューロンの出力に結合荷重を両側から乗じたもの、すなわち $w_{ij}^* v_j w_{ij}$ も¹、(7) とは異なるモデルとなるので次のようなモデルも考えられる。

$$\begin{cases} \tau_i \frac{du_i}{dt} = u_i + \sum_{j=1}^n w_{ij}^* v_j w_{ij} + b_i \\ v_i = f(u_i) \quad (i = 1, 2, \dots, n) \end{cases} \quad (9)$$

ここで、結合加重 w_{ij}^* は、一般には結合加重 w_{ij} と異なる任意のクリフォード代数 $\mathbb{G}_{p,q,r}$ の要素でよいが、 w_{ij}^* を w_{ij} に対合 (involution) をとったものとするのがクリ

¹クリフォード代数においては、回転など幾何学的変換はこのようないちいち表現となる。

フォード代数においては特に意味を持つ。対合とは二度作用させると元に戻る、すなわち $(w^*)^* = w$ となる作用素で、クリフォード代数においては反転 (inversion)、逆転 (reversion)、共役 (conjugation) などがある。複素数や四元数における共役もその一つである。

4 エネルギー関数の存在条件

4.1 エネルギー関数の定義と仮定

リカレント NN のダイナミクスに関する重要な問題に、エネルギー関数の存在性の問題がある。Hopfield は (7) 式を実数の NN とした場合、すなわち u_i, v_i, b_i, w_{ij} をすべて $u_i \in \mathbb{R}, v_i \in \mathbb{R}, b_i \in \mathbb{R}, w_{ij} \in \mathbb{R}$ とし、活性化関数 f を $f: \mathbb{R} \rightarrow \mathbb{R}$ とした場合のモデルを対象として、エネルギー関数が存在するための条件とエネルギー関数を示した¹²⁾。すなわち、 $E(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$ なるある関数をエネルギー関数として定義し、次のような性質を持つことを示した。

実数の (7) 式のモデルにおいて、重み行列 $W = \{w_{ij}\}$ が対称行列 $W^T = W$ でかつ活性化関数 $f(\cdot)$ が連続微分可能で有界かつ単調増加ならば、NN はエネルギー関数を持つ。またエネルギー関数の NN の解に沿っての時間微分は常に非正となる。

このことより、エネルギー関数が存在すると NN の解は必ずエネルギー関数を減少させるように動作し、解は $t \rightarrow \infty$ である平衡点へ収束することが示せる。また彼らはさらに、エネルギー関数を利用して NN により最適化問題を解く方法を提案し、実際に巡回セールスマン問題が解けることを示した¹³⁾。この研究は脳を模倣した NN で実際に最適化問題を解けることを示したものととして、各方面に大きなインパクトを与えた。その後、エネルギー関数を用いた NN の定性的な性質の解析やその結果を利用したさまざまな応用、たとえば連想記憶の設計への応用、各種の最適化問題の応用など非常に活発に研究されてきた。

(7)、(8) および (9) 式で表される 3 種類のクリフォード代数で表された NN に対しても同様の性質を持つエネルギー関数が定義できれば、これを用いて、実数の NN でなされてきた種々の研究がクリフォード代数表現された NN にも可能となり、NN の応用範囲も一層広がると考えられる。従ってエネルギー関数が存在する条件を明らかにしておくことは重要である。

さて、リカレント NN のダイナミクスを論ずるためにはまず、活性化関数 $f(\cdot)$ として、どのような性質を持つ関数を用いるかが問題となる。実数のニューラルネットワークでは、シグモイド関数のような滑らかで有界な非線形関数が通常用いられる。また、複素 NN においては活性化関数に有界性を課すと、滑らかな関数として正則な関数 (複素微分可能な関数) は選ぶことはできない。これは有界で複素平面全体で正則な関数は定数に限るというリュービルの定理による。このため文献 5) では、滑らかさとして正則性の代わりに、複素非線形関数の実数部および虚数部が、それぞれ実数部、虚数部に関して連続偏微分可能な関数を活性化関数として選び、エネルギー関数が存在する条件を導き、具体的にエネルギー関数を構成している。

リュービルの定理のような定理がクリフォード代数 $\mathbb{G}_{p,q,r}$ に対して常に成立するとは限らないが、(7)、(8) および (9) 式のモデルに対し、活性化関数に複素 NN に

対する条件と同様な条件を考えてエネルギー関数の存在条件を議論する。

クリフォード代数上の非線形関数を 2^{p+q+r} 個の実数値関数 $f^{(i)} : \mathbb{R}^{2^{p+q+r}} \rightarrow \mathbb{R}$ を用いて次のように表現する。

$$f(u) = \sum_{i=1}^{2^{p+q+r}} f^{(i)}(u^{(1)}, u^{(2)}, \dots, u^{(2^{p+q+r})}) \overline{\mathbb{G}}_{p,q,r}[i] \quad (10)$$

ここで u は、(6) 式に従って以下としている。

$$u = \sum_{i=1}^{2^{p+q+r}} u^{(i)} \overline{\mathbb{G}}_{p,q,r}[i] \quad (11)$$

たとえば、 $p+q+r=2$ とした $\mathbb{G}_2 := \mathbb{G}_{p,q,r}$ においては $\mathbb{G}_2 = \{1, e_1, e_2, e_1 e_2\}$ となるので、 $f(u)$ は次のように表わされる。

$$\begin{aligned} f(u) = & f^{(0)}(u^{(0)}, u^{(1)}, u^{(2)}, u^{(3)}) \\ & + f^{(1)}(u^{(0)}, u^{(1)}, u^{(2)}, u^{(3)}) e_1 \\ & + f^{(2)}(u^{(0)}, u^{(1)}, u^{(2)}, u^{(3)}) e_2 \\ & + f^{(3)}(u^{(0)}, u^{(1)}, u^{(2)}, u^{(3)}) e_1 e_2 \end{aligned} \quad (12)$$

活性化関数 $f(\cdot)$ に対し、次を仮定する。

- (i) $f^{(l)}(\cdot)$ は $u^{(m)}$ ($l, m = 0, 1, \dots, 2^{p+q+r}$) に関して連続偏微分可能である。
- (ii) $f(\cdot)$ は有界な関数である、すなわち $|f(\cdot)| \leq M$ を満たす $M > 0$ が存在する。

ここで、活性化関数 $f(\cdot)$ の点 u におけるヤコビ行列を $\mathbf{J}_f(u) = \{\alpha_{lm}(u)\} \in \mathbb{R}^{2^{p+q+r} \times 2^{p+q+r}}$ と表すことにする。ただし、 α_{lm} は次のように定義される。

$$\alpha_{lm}(u) = \left. \frac{\partial f^{(l)}}{\partial u^{(m)}} \right|_u \quad (13)$$

実数の Hopfield 型 NN のエネルギー関数にならって、(7), (8) 式および (9) 式の NN モデルに対しエネルギー関数を次のように定義する。

定義 1 次の条件を満足する関数 $E(\cdot)$ を、クリフォード代数表現されたニューラルネットワーク (\mathcal{N}) のエネルギー関数と呼ぶ。ただし、(\mathcal{N}) は NN のモデルの式番号、すなわち (\mathcal{N}) = (7), (\mathcal{N}) = (8), あるいは (\mathcal{N}) = (9) とする。

- (i) 関数 $E(\cdot)$ は、 $\mathbb{G}_{p,q,r}$ から \mathbb{R} への下に有界な写像である。
- (ii) $E(\cdot)$ の NN の解軌道に沿った時間微分を $\left. \frac{dE}{dt} \right|_{(\mathcal{N})}$ と表すと、 $\left. \frac{dE}{dt} \right|_{(\mathcal{N})} \leq 0$ を満たす。また、その時間微分が 0 となるのはネットワークの平衡点においてのみである、すなわち $\left. \frac{dE}{dt} \right|_{(\mathcal{N})} = 0$ となるのは、 $\frac{dv_i}{dt} = 0$ ($i = 1, 2, \dots, n$) が成り立つときであり、またそのときに限る。

なお、(7) 式のモデルが実数のモデルである場合、次の関数がエネルギー関数として提案されている。

$$\begin{aligned} E(\mathbf{v}) = & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} v_i v_j - \sum_{i=1}^n b_i v_i \\ & + \sum_{i=1}^n \int_0^{v_i} f^{-1}(\cdot) d\cdot \end{aligned} \quad (14)$$

ここで $\mathbf{v} = [v_1, v_2, \dots, v_n]^T \in \mathbb{R}^n$ であり、 f^{-1} は f の逆関数である。

4.2 エネルギー関数の構成と存在条件の導出

本節では、クリフォード代数表現された NN に対しエネルギー関数の存在条件を導き、具体的にエネルギー関数を構成する。ただし、本節ではクリフォード代数 $\mathbb{G}_{p,q,r}$ として $r=0$ としたクリフォード代数 $\mathbb{G}_{p,q,0}$ を対象とし、これを慣例に従い $\mathbb{G}_{p,q} := \mathbb{G}_{p,q,0}$ と書くことにする。すなわちクリフォード代数 $\mathbb{G}_{p,q}$ で表現された (7), (8) および (9) 式の NN を対象とする。また、(9) 式の NN において、対合 (involution)* は共役 (conjugation) する。(6) 式で表される $a \in \mathbb{G}_{p,q}$ に対し、その共役 a^* は次のように定義される。

$$a^* = \sum_{i=1}^{2^{p+q+r}} (\overline{\mathbb{G}}_{p,q}[i])^2 a^{(i)} \overline{\mathbb{G}}_{p,q}[i] \quad (15)$$

(2) 式の定義より、 $1 \leq i \leq p$ に対し $(\overline{\mathbb{G}}_{p,q}[i])^2 = 1$ であり、 $p < i \leq p+q$ に対し $(\overline{\mathbb{G}}_{p,q}[i])^2 = -1$ である。

ここで、次のような仮定を置く。

仮定 1 クリフォード代数 $\mathbb{G}_{p,q}$ で表現された (7), (8) あるいは (9) 式の NN の結合重み係数は次式を満足する。

$$w_{ji} = w_{ij}^* \quad (i, j = 1, 2, \dots, n) \quad (16)$$

仮定 2 クリフォード代数 $\mathbb{G}_{p,q}$ で表現された (7), (8) あるいは (9) 式の NN の活性化関数 $f(\cdot)$ は次の条件を満足する。

- (i) $f(\cdot)$ は 1 対 1 の関数である。
- (ii) 任意の $u \in \mathbb{G}_{p,q}$ に対して、 $f(\cdot)$ のヤコビ行列 $\mathbf{J}_f(u)$ は対称である。
- (iii) 任意の $u \in \mathbb{G}_{p,q}$ に対して、 $f(\cdot)$ のヤコビ行列 $\mathbf{J}_f(u)$ は正定値である。

仮定 2 の条件 (i) と活性化関数の有界性から、関数 $f(\cdot)$ の逆関数 $f^{-1}(\cdot) : \mathbb{G}_{p,q} \rightarrow \mathbb{G}_{p,q}$ が存在する。この逆関数を $g = f^{-1}$ と定義すると、 $u = g(v)$ は 2^{p+q} 個の実数値関数 $g^{(i)} : \mathbb{R}^{2^{p+q}} \rightarrow \mathbb{R}$ を用いて次のように表される。

$$g(v) = \sum_{i=1}^{2^{p+q}} g^{(i)}(v^{(1)}, v^{(2)}, \dots, v^{(2^{p+q})}) \overline{\mathbb{G}}_{p,q}[i] \quad (17)$$

この逆関数 g に対し次のような補題が成り立つ。

補題 1 仮定 2 を満足する非線形関数 f の逆関数 g に対し、すべて $v \in \mathbb{G}_{p,q}$ に対し次式を満足する関数 $G(\cdot) : \mathbb{R}^{2^{p+q}} \rightarrow R$ が存在する。

$$\frac{\partial G}{\partial v^{(i)}} = g^{(i)}(v^{(1)}, v^{(2)}, \dots, v^{(2^{p+q})}) \quad (18)$$

$$(i = 1, 2, \dots, 2^{p+q})$$

この補題の証明は省略する。

(18) 式を満たす関数 $G(\cdot)$ を用いて、(7), (8) あるいは (9) 式のクリフォード代数 $\mathbb{G}_{p,q}$ 表現された NN に対し、エネルギー関数が次のように構成できる。(7) 式の NN に対しては、エネルギー関数を次のように構成できる。

$$E_{(7)}(\mathbf{v}) = \sum_{i=1}^n \sum_{q=j}^n \left\{ \frac{1}{2} \mathcal{S}c(v_i^* w_{ij} v_j + 2b_i^* v_i) G(v_i) \right\} \quad (19)$$

ここで $\mathbf{v} = [v_1, v_2, \dots, v_n]^T \in \mathbb{G}_{p,q}^n$ であり、 $\mathcal{S}c(\cdot)$ は、 $a \in \mathbb{G}_{p,q}$ に対し、 $\mathcal{S}c(a) = a^{(0)}$ と定義される。すなわち $\mathcal{S}c(\cdot)$ は、 $\mathbb{G}_{p,q}$ の要素のスカラ部を取り出す作用素である。

また、(8) 式の NN に対しては、

$$E_{(8)}(\mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{1}{2} \mathcal{S}c(v_i^* v_j w_{ij} + 2b_i^* v_i) G(v_i) \right\} \quad (20)$$

(9) 式の NN に対しては

$$E_{(9)}(\mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{1}{2} \mathcal{S}c(v_i^* w_{ij}^* v_j w_{ij} + 2b_i^* v_i) G(v_i) \right\} \quad (21)$$

のようにエネルギー関数が構成できる。ここで、 $\mathbf{v} = [v_1, v_2, \dots, v_n]^T \in \mathbb{G}_{p,q}^n$ としている。

エネルギー関数の存在条件として次の定理が成り立つ。

定理 1 クリフォード代数 $\mathbb{G}_{p,q}$ で表現された (7), (8) あるいは (9) 式の NN に対して、結合重み係数と活性化関数が仮定 1, 2 を満足するとき、定義 1 を満たすエネルギー関数がそれぞれ存在する。

この定理の証明は、構成した (19), (20) および (21) 式の関数が定理の条件を満足するとき、定義 1 で定義されるエネルギー関数の条件を満たすことを示すことにより行える。紙面の都合上、証明は省略する。

4.3 いくつかのサブクラスの NN に対する存在条件

前節では、クリフォード代数 $\mathbb{G}_{p,q}$ で表現されたりカレント NN に対してエネルギー関数の存在条件を示し、また具体的にエネルギー関数を構成した。前節の定理 1 の仮定 1 と仮定 2 がエネルギー関数の存在条件であり、結合重みと活性化関数に関する条件となっている。本節では、前節で得られた結果が、これまでいくつかのサブクラスのリカレント NN に対して得られている結果の一般化になっていることを示す。すなわち、これ

までに筆者ら導いてきた hyperbolic 数、複素数、および四元数表現はそれぞれクリフォード代数 $\mathbb{G}_{p,q}$ のあるひとつのクラスと同型になっており、ここでは前節で導いた結果より、これらのクラスのエネルギー関数の存在条件とエネルギー関数が導けることを示す。

まず、Hopfield により得られた実数の NN の存在条件に対しては、 $\mathbb{G}_{p,q}$ において $p = 0, q = 0$ とおいた $\mathbb{G}_{0,0}$ の場合に相当している。この場合、仮定 1 の条件は明らかに $w_{ji} = w_{ij}$ 、 $(i, j = 1, 2, \dots, n)$ となり、これは結合加重行列が対称行列であることを意味している。また仮定 2 の条件は $\frac{\partial f^{(0)}}{\partial u^{(0)}} > 0$ となりこれは活性化関数が単調増加であることを意味している。したがって、これらは実数の NN の場合の存在条件と一致している。

以下では複素数、hyperbolic 数および四元数表現された NN に対するエネルギー関数の存在条件、およびエネルギー関数が、前節の結果から求められることを示す。

4.3.1 複素数表現の NN の場合

複素数は、 $\mathbb{G}_{p,q}$ において $p = 0, q = 1$ とおいた $\mathbb{G}_{0,1}$ のクリフォード代数と同型である。この場合クリフォード積は可換であるので、(7), (8) および (9) 式で表される NN のモデルは本質的に等価である。したがって (7) 式で表現されるモデルのみを対象とすればよい。クリフォード代数 $\mathbb{G}_{0,1}$ の要素、すなわち $a \in \mathbb{G}_{0,1}$ は、

$$a = a^{(0)} + a^{(1)} e_1 \quad (22)$$

と表される。クリフォード代数 $\mathbb{G}_{0,1}$ においては $(e_1)^2 = -1$ なので、 e_1 を虚数単位 i に対応させることができ、また、(15) 式で定義される共役は、 $a = 1^2 - a^{(0)} + (e_1)^2 - a^{(1)} e_1 = a^{(0)} - a^{(1)} e_1$ となるので、複素数における共役と等価であることがわかる。また $\mathbb{G}_{0,1}$ における活性化関数は次のように表される。

$$f(u) = f^{(0)}(u^{(0)}, u^{(1)}) + f^{(1)}(u^{(0)}, u^{(1)}) e_1 \quad (23)$$

ここで、 $u = u^{(0)} + u^{(1)} e_1$ である。複素数表現の場合、 $f^{(0)}$ および $f^{(1)}$ はそれぞれ、非線形関数 f の実数部および虚数部に対応している。仮定 1 と仮定 2 の条件より複素数で表現された NN の場合のエネルギーの存在条件は、以下のようになる。

[エネルギー関数の存在条件]

- NN の結合重み係数に関する条件

$$w_{ji} = w_{ij}^* \quad (i, j = 1, 2, \dots, n) \quad (24)$$

- 活性化関数 f に関する条件

$$\left\{ \begin{array}{l} (i) \frac{\partial f^{(0)}}{\partial u^{(0)}} > 0, \\ (ii) \frac{\partial f^{(0)}}{\partial u^{(1)}} = \frac{\partial f^{(1)}}{\partial u^{(0)}}, \\ (iii) \frac{\partial f^{(0)}}{\partial u^{(0)}} \frac{\partial f^{(1)}}{\partial u^{(1)}} - \frac{\partial f^{(0)}}{\partial u^{(1)}} \frac{\partial f^{(1)}}{\partial u^{(0)}} > 0 \end{array} \right. \quad (25)$$

結合重み係数に関する存在条件の (24) 式は、複素数表現の場合、結合重み行列がエルミート行列であることである。

(19) のエネルギー関数は複素数で表現された NN の場合、以下ようになる。

$$E(\mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{1}{2} \mathcal{S}c(v_i^* w_{ij} v_j + 2b_i^* v_i) \quad G(v_i) \right\} \quad (26)$$

ここで $\mathbf{v} = [v_1, v_2, \dots, v_n]^T \in \mathbb{G}_{0,1}^n$ である。複素数表現の場合、作用素 $\mathcal{S}c()$ は実数部を取り出す作用素 $Re()$ となっている。

これらのエネルギー関数の存在条件および構成されたエネルギー関数は文献 5) で導かれたそれらと一致している。

4.3.2 hyperbolic 数表現の NN の場合

hyperbolic 数は、二乗すると 1 となる単位 $\mathbf{h}, \mathbf{h}^2 = 1$ を用いて $a = a^{(0)} + \mathbf{h}a^{(1)}$ ($a^{(0)}$ および $a^{(1)}$ は実数) と表される数で、 $\mathbb{G}_{p,q}$ において $p = 1, q = 0$ とおいたクリフォード代数 $\mathbb{G}_{1,0}$ と同型である。この場合もクリフォード積は可換であるので、(7), (8) および (9) 式で表される NN のモデルは本質的に等価であり、(7) 式で表現されるモデルのみを対象とすればよい。クリフォード代数 $\mathbb{G}_{1,0}$ の要素、すなわち $a \in \mathbb{G}_{1,0}$ は複素数と同様 (22) 式で表され、非線形関数は (23) 式で表される。ただし、 $(e_1)^2 = 1$ である。

したがって、仮定 1 と仮定 2 の条件より hyperbolic 数で表現された NN の場合のエネルギーの存在条件は、以下ようになる。

[エネルギー関数の存在条件]

- NN の結合重み係数に関する条件。

$$w_{ji} = w_{ij} \quad (i, j = 1, 2, \dots, n) \quad (27)$$

- 活性化関数 f に関する条件

$$\left\{ \begin{array}{l} (i) \quad \frac{\partial f^{(0)}}{\partial u^{(0)}} > 0, \\ (ii) \quad \frac{\partial f^{(0)}}{\partial u^{(1)}} = \frac{\partial f^{(1)}}{\partial u^{(0)}}, \\ (iii) \quad \frac{\partial f^{(0)}}{\partial u^{(0)}} \frac{\partial f^{(1)}}{\partial u^{(1)}} - \frac{\partial f^{(0)}}{\partial u^{(1)}} \frac{\partial f^{(1)}}{\partial u^{(0)}} > 0 \end{array} \right. \quad (28)$$

hyperbolic 数表現の場合、結合重み係数に関する存在条件は、結合重み行列が対称行列であることであり、また活性化関数に関する条件は複素数表現の場合と同じとなっている。

また、(19) のエネルギー関数は hyperbolic 数で表現された NN の場合、以下ようになる。

$$E(\mathbf{v}) = \sum_{i=1}^n \sum_{j=1}^n \left\{ \frac{1}{2} \mathcal{S}c(v_i w_{ij} v_j + 2b_i v_i) \quad G(v_i) \right\} \quad (29)$$

これらのエネルギー関数の存在条件および構成されたエネルギー関数は文献 9) で導かれたそれらと一致している。

4.3.3 四元数表現の NN の場合

文献 7) では、四元数表現された Hopfield 型リカレントの NN に対するエネルギー存在条件を導出し、また実際にエネルギー関数を構成している。四元数は次のように定義される超複素数の一種である。

$$a = a^{(0)} + i a^{(1)} + j a^{(2)} + k a^{(3)} \quad (30)$$

ここで、 $a^{(0)}, a^{(1)}, a^{(2)}, a^{(3)}$ は実数、すなわち \mathbb{R} の要素であり、 $\{i, j, k\}$ は次の関係を満たす虚数単位である。

$$\begin{aligned} i^2 &= -1, & j^2 &= -1, & k^2 &= -1, \\ ij &= jk = ki = ijk, & ji &= kj = ki = -ijk \end{aligned} \quad (31)$$

四元数の積演算はこれらの関係に従って定義され、非可換演算となる。また、四元数 a の共役 a^* は

$$a^* = a^{(0)} - i a^{(1)} - j a^{(2)} - k a^{(3)} \quad (32)$$

と定義される。四元数はクリフォード代数におけるクリフォード代数 $\mathbb{G}_{0,2}$ と同型である。まず、このことを見てみよう。 $\mathbb{G}_{0,2}$ の基底の集合 $\overline{\mathbb{G}}_{0,2}$ は、

$$\overline{\mathbb{G}}_{0,2} = \{1, e_1, e_2, e_1 e_2\}$$

であり、 $a \in \mathbb{G}_{0,2}$ は次のように表される。

$$a = a^{(0)} + a^{(1)} e_1 + a^{(2)} e_2 + a^{(3)} e_1 e_2 \quad (33)$$

基底どうしの積を表にして表 1 に示す。以上により、 e_1

Table 1: Multiplication Table for Clifford Algebra $\mathbb{G}_{0,2}$

	1	e_1	e_2	$e_1 e_2$
1	1	e_1	e_2	$e_1 e_2$
e_1	e_1	1	$e_1 e_2$	e_2
e_2	e_2	$e_1 e_2$	1	e_1
$e_1 e_2$	$e_1 e_2$	e_2	e_1	1

を四元数の虚数単位 i 、 e_2 を四元数の虚数単位 j 、 $e_1 e_2$ を四元数の虚数単位 k に対応させるとクリフォード代数 $\mathbb{G}_{0,2}$ と四元数代数は同型であることがわかる。また (15) 式で定義されるクリフォード代数 $\mathbb{G}_{p,q}$ における共役は、クリフォード代数 $\mathbb{G}_{0,2}$ においては (32) 式の四元数の共役と等しい。したがって、仮定 1 と仮定 2 の条件より四元数表現された (7), (8) および (9) 式で表される NN の場合のエネルギーの存在条件は、以下ようになる。

[エネルギー関数の存在条件]

- NN の結合重み係数に関する条件

$$w_{ji} = w_{ij}^* \quad (i, j = 1, 2, \dots, n) \quad (34)$$

- 活性化関数 f に関する条件

- $f(\cdot)$ は 1 対 1 の関数である。
- 任意の $u \in \mathbb{G}_{0,2}$ に対して、 $f(\cdot)$ のヤコビ行列 $\mathbf{J}_f(u)$ は対称である。
- 任意の $u \in \mathbb{G}_{0,2}$ に対して、 $f(\cdot)$ のヤコビ行列 $\mathbf{J}_f(u)$ は正定値である。

この存在条件は、文献 7) で導出されている四元数表現されたリカレントの NN に対するエネルギー存在条件と一致している。

また、四元数表現された (7), (8) および (9) 式で表される NN に対し、それぞれエネルギー関数は (19), (20) および (21) 式の関数として構成できる。これらの関数は、文献 7) で構成されているエネルギー関数と一致している。

5 おわりに

近年、ニューラルネットワークモデルを複素数値化、四元数値化など高次元化し、その情報処理能力、学習法や応用などの研究が盛んに行われている。これら高次元の表現は Clifford algebra (geometric algebra) 表現として一般化される。本稿では、クリフォード代数 $\mathbb{G}_{p,q}$ で表現されたリカレント NN のモデルを 3 種類示し、それらのエネルギー関数の存在条件を導いた。また実際に 3 種類のモデルに対しエネルギー関数を構成した。さらに、筆者らがこれまで導いてきたサブクラスの NN に対する存在条件、およびエネルギー関数がそれぞれ、本稿で導いた存在条件および構成したエネルギー関数から導けることを示した。具体的には複素数、hyperbolic 数、複素数および四元数で表現された NN に対するエネルギー関数の存在条件とエネルギー関数である。

今後の課題としては、リュービルの定理が成立しない Clifford algebra で表現される Hopfield 型 NN のモデルの構築とそのエネルギー関数の存在条件の導出が挙げられる。また、さらに一般のクラスのクリフォード代数 $\mathbb{G}_{p,q,r}$ で表現された Hopfield 型 NN に対しても、エネルギー関数の存在条件の導出とエネルギー関数の構成も今後の課題である。

エネルギー関数は、最適化問題への応用など種々の工学的問題への応用が期待でき、豊かな問題表現能力および情報表現能力を有しているクリフォード代数表現を導入することにより、NN の応用の新たなパラダイムが拓けると期待される。

参考文献

- 1) A. Hirose (ed.): Complex-Valued Neural Networks Theoris and Applications, World Scientific (2003)
- 2) T. Nitta (ed.): Complex-Valued Neural Networks Utilizing High-Dimentionoanal Parameters, IGI Global (2009)
- 3) 黒江, 新田, E. Hitzer: クリフォード代数の応用 - サーパーベイ, 計測自動制御学会誌 計測と制御, Vol.51, No.4, pp.328/338 (2012)
- 4) 特集 コンピューテーショナル・インテリジェンスの展開 - クリフォード代数表現など高次元表現を中心として -, 計測自動制御学会学会誌 計測と制御, Vol.51 (2012)
- 5) 橋本, 黒江, 森: 複素ニューラルネットワークのエネルギー関数について, システム制御情報学会論文誌, Vol.15, No.10, 559/565 (2002)
- 6) Y. Kuroe, M. Yoshida and T. Mori: On Activation Functions for Complex-Valued Neural Networks - Existence of Energy Functions -, Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, Okyay Kaynak et. al.(Eds.), Lecture Notes in Computer Science, 2714, 985/992, Springer (2003)
- 7) M. Yoshida, Y. Kuroe and T. Mori: Models of Hopfield-Type Quaternion Neural Networks and Their Energy Functions, International Journal of Neural Systems, Vol.15, Nos.1 & 2, 129/135 (2005)
- 8) Y. Kuroe: Models of Clifford Recurrent Neural Networks and Their Dynamics, Proceedings of 2011 International Joint Conference on Neural Networks, 1035/1041 (2011)

- 9) Y. Kuroe, S. Tanigawa and H. Iima: Models of Hopfield-type Clifford Neural Networks and Their Energy Functions - Hyperbolic and Dual Valued Networks -, Proceedings of ICONIP 2011, Lecture Notes in Computer Science 7062, 560/569, Springer (2011)
- 10) V. Hasanov, Y. Kuroe and H. Iima: Energy Functions of General Clifford-Valued Hopfield-Type Neural Networks and Their Existence Conditions: 計測自動制御学会第 40 回知能システムシンポジウム資料, 251/256 (2013)
- 11) Christian Perwass: Geometric Algebra with Applications in Engineering, Springer-Verlag (2009)
- 12) J. J. Hopfield: Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. Natl. Acad. Sci. USA*, Vol.81, 3088/3092 (1984)
- 13) J. J. Hopfield and D. W. Tank: "Neural" computation of decisions in optimization problems, *Biol. Cybern.*, Vol.52, 141/152 (1985)

ユーザの性格特性に基づく 学習適応的感情指向型インタフェース

○田邊 幸祐 (県立広島大学大学院総合学術研究科経営情報学専攻)
市村 匠 (県立広島大学経営情報学部)

Personality Characteristics Adaptive Learning Emotion Orientated Interface

*Kousuke Tanabe

(Graduate School of Comprehensive Scientific Research, Prefectural University of Hiroshima)

Takumi Ichimura

(Faculty of Management and Information Systems, Prefectural University of Hiroshima)

Abstract— A communication tool such as a chat becomes the popular method in the Internet conversation. However, there are some troubles related to the misunderstanding of the intentions in the conversation. In order to avoid such situation, the communication tool, which enables to measure the user's emotion is required. Emotion orientated interface with the EGC(Emotion Generating Calculation) and MSTN (Mental State Transition Network) is a useful method to estimate the emotion value and the kinds of emotion and keep their values and its transition situation of emotions. In this paper, we developed the emotion orientated interface with adaptive learning method of personality characteristics to change the transition probability by using recurrent neural network.

Key Words: Emotion Generating Calculations, Mental State Transition Network, Adaptive Learning

1 はじめに

高度情報化社会において、コンピュータと人間のインタラクションは複雑かつ高度なものになっている。インターネット上のコミュニケーションにおいては、チャットツールなどのツールを用いた会話が一般的となりインターネットを通じて様々な人と会話を行えるようになった¹⁾²⁾。

しかし、インターネット上の会話では必ずしも発話者の意図がそのまま伝わらないこともあり、会話中のトラブルが発生することがある。このようなトラブルを回避するためには、会話中の相手の感情を理解するだけでなく、自分の感情をコントロールし、お互いが誤解のないように会話を進めることが重要である。しかしながら、対面的な人間同士の会話でさえ、相手の感情を常に理解できるとは限らず、コンピュータを介した会話ではさらにその問題は顕著に表れるだろう³⁾。これを解決するためには、発話者の意図を理解することが必要であるが、今日、会話中に円滑な対話を進めるコミュニケーション能力も求められており、このために、会話中で感情の変化を推定することも必要となっている。特に、対面的コミュニケーションよりもインターネットを利用して会話を楽しむ人が増えているなか、コミュニケーションツールの機能として、発話者の感情を評価する手法が必要になってきている。

我々は、このようなコミュニケーションツールを実現するために、感情の概念を情緒、気分、表情の3つに分けて感情処理を実現する感情指向型インタフェースを開発している⁴⁾⁵⁾。このインタフェースにおいて、情緒は何らかの出来事に遭遇することが原因で生起する比較的持続時間の短い強い感情を表し、気分は生起した情緒やエージェントの内部状態などに起因して生起する弱いながらもエージェントの思考や行動に影響を与える持続的な感情を表している。

人間の情緒と気分を推定する手法として情緒生起手法(EGC)⁶⁾⁷⁾と心的状態遷移ネットワーク(MSTN)⁸⁾⁹⁾が提案されており、本研究ではこれら2つの手法を用いて、インタフェースに入力された文章から情緒と気分を推定することによって人間の複雑な感情の遷移を表現する。

EGCは、人間が持つ最も単純な情緒である快/不快の情緒の値を入力された文章から計算する手法である⁶⁾⁷⁾。EGCでは入力された文章に対して構文解析を行い、構文解析によって得られた文章の格フレーム表現によって決定される文章の事象タイプと、文章中の単語に対する好き/嫌いの度合いを[-1.0, 1.0]の値で示した好感度(Favorite Value; FV)から、文章が表している事象に対する快/不快の印象を計算し、[-1.0, 1.0]の値で表現する。さらに、計算された快/不快の情緒値と格フレーム表現を用いて、Elliottの感情誘発条件による情緒の分類を行うことにより20種類の情緒を表現することができる。

EGCは会話文から話者の情緒の推定するために有効な手法と考えられるが、EGCで用いられるFV値は予め与えられた一般的な値を用いている。しかし、ユーザが生起する情緒は個人差が非常に大きいものであり、ユーザの経験によって変化する可能性もある。そのため、事前に与えられたFV値のみでユーザの感情を推定することには限界があり、感情指向型インタフェースのFV値はユーザの情緒や気分に対して適応的に変化させる必要がある。

文献¹⁰⁾ではユーザが持つFV値の集合をユーザの嗜好情報と定義し、好感度学習手法を用いたFV値の学習によってユーザの嗜好情報に適応することを提案した。好感度学習手法では、好き/嫌いの直接表明、好感度変化条件、不快事象の連想、感情表現からの逆算という4つの学習手法が提案されており、4つの手法の学

習結果としてユーザの持つ FV 値に適応することが可能である。しかし、好感度変化条件の手法で提案されている知識構造にもとづいた推論では、知識構造の具体的な方法は未定義であったため、本研究においてはファジィベトリネット (FPN) によって表現されたファジィ IF-THEN ルールを知識構造として定義し、FPN を用いた推論から FV 値を学習する。

MSTN は、人間の複雑な気分の遷移を表現するための手法である⁸⁾⁹⁾。MSTN では人間の気分を7種類の状態として表現することにより心的状態遷移ネットワークを構成し、心的状態遷移確率とネットワークに入力された情緒にもとづいて気分の変化を表現する。

MSTN の状態遷移確率は統計的に得られた人間の一般的な気分の変化の傾向を表したものであり、ユーザ個人の気分の変化の傾向とは必ずしも一致しない。そのため、感情指向型インタフェースにおいてユーザの気分の変化を適切に表現するためには、MSTN で発生する状態遷移をユーザの気分の変化に適応することが必要である。

本論文では、発話に応じて生起する気分の変化がユーザの性格特性によって異なることに着目し、ユーザの性格特性を表現した MSTN を適応的学習によって実現する手法を提案する。提案手法では、ある1つの出来事において発せられた一連の発話を1つの発話エピソードと定義し、Profit Sharing の手法を用いることによって発話エピソード内で生起した MSTN の状態遷移系列における心理的最短パスの発見と、リカレントニューラルネットワークで表現された MSTN における通時的逆伝播によって MSTN の状態遷移確率を変化させる。ここで、心理的最短パスとは、人間の複雑な心理は同じ状態を繰り返すように変化することがあるため、これらの遷移を除いた心的状態遷移を示す系列をさす。本論文で、提案した手法の有効性を小説の会話をを用いて検証したので、その結果をここに報告する。

以降の2節と3節では、EGCの手法と提案したEGCの個人嗜好情報の学習について説明する。その後の4節ではMSTNの手法について説明し、5節と6節では、本論文で提案するユーザの性格特性に応じたMSTNを適応的学習の手法の説明及び、実験について述べる。

2 情緒生起手法

情緒生起手法 (EGC) は、入力された文章の格フレーム表現と情緒計算式によって、人間の持つ最も基本的な情緒である快/不快の値を計算する手法である⁶⁾⁷⁾。この手法と Elliott の感情誘発条件を組み合わせることによって、文章から20種類の情緒を表現することができる。

情緒生起手法では、まず初めに文章の構文分析を行い、文章の主体や客体といった格フレーム表現と文章の事象タイプを求める。次に、事象タイプと好き/嫌いの程度を [-1.0 1.0] で表した単語の好感度 (FV 値) にもとづいて快/不快の強度を計算し、計算された快/不快の情緒値を Elliott の感情誘発条件によって分類することで20種類の情緒を表現する。

2.1 格フレーム表現と事象タイプ

情緒生起手法では、文章の事象タイプを決定する為に必要となる文章の格要素の種類を以下のように定義している。

S(Subject)	: 主体
O(Object)	: 客体
OF(Object-From)	: 出発点または源
OT(Object-to)	: 目標
OM(Object-Mutual)	: 相互作用の相手
OS(Object-Source)	: 投げ所
OC(Object-Content)	: 属性の補足
I(Implement)	: 道具または手段
L(Location)	: 場所
T(Time)	: 時間
R(Reason)	: 原因または理由
D(Degree)	: 程度

さらに、これら各要素の組み合わせによって文章の事象タイプを18種類に分類する。これらの事象タイプは、文章中で発生した事象を共通の形式で表現するためのものであり、例えば V(S) の事象タイプならば、“主体 (S) がそれ自身に影響を及ぼすような事象 (V) をする”ことを表現する。

Table 1: The Event type and The Axis

事象タイプ	f_1	f_2	f_3
V(S)	f_S		f_P
A(S, C)			
A(S, OF, C)			
A(S, OT, C)			
A(S, OM, C)			
A(S, OS, C)			
N(S)			
V(S, OF)	f_S	$f_{OT} - f_{OF}$	f_P
V(S, OT)			
V(S, OM)	f_S	f_{OM}	f_P
V(S, OS)	$f_S - f_{OS}$		f_P
V(S, O)	f_S	f_O	f_P
	f_O		f_P
V(S, O, OF)	f_O	$f_{OT} - f_{OF}$	f_P
V(S, O, OT)		f_{OM}	
V(S, O, OM)	f_O	f_{OM}	f_O
V(S, O, OI)	f_O	$ f_I $	f_P
V(S, O, OC)	f_O		f_{OC}
V(S, O, C)	f_O		f_P

2.2 情緒計算式

情緒値を計算する際には、まず Table 1 に基づいて、各要素の好感度を対応した f_1, f_2, f_3 を取得し、 f_1, f_2, f_3 を直行ベクトルとする感情空間を定義する。次に、この感情空間における Table 2 の関係から快/不快を判断し、 f_1, f_2, f_3 からなる合成ベクトルの大きさから情緒の強さを判断する。

f_S	: 主体の好感度
f_O	: 客体の好感度
f_{OF}	: 出発点または源の好感度
f_{OT}	: 目標の好感度
f_{OM}	: 相互作用の相手の好感度
f_{OS}	: 投げ所の好感度
f_{OC}	: 属性の補足の好感度
f_P	: 事象そのもの (Predicate) の好感度

Table 2: Pleasure/Displeasure in Emotional Space

Area	f_1	f_2	f_3	Emotion
I	+	+	+	Pleasure
II	-	+	+	Displeasure
III	-	-	+	Pleasure
IV	+	-	+	Displeasure
V	+	+	-	Displeasure
VI	-	+	-	Pleasure
VII	-	-	-	Displeasure
VIII	+	-	-	Pleasure

3 嗜好情報の学習

単語のFV値はその単語が示す事象に対するユーザの好き/嫌いの度合いを示すものであり、本研究ではFV値の集合によってユーザの嗜好情報を表現する。

EGCにおいて情緒を計算する際、FV値の設定はユーザの情緒を正しく推定するために非常に重要であるが、ユーザの嗜好に最適化されたFV値を全て事前に定義することは困難である。そのため本研究では、ユーザからの入力をもとに好感度学習手法⁷⁾¹¹⁾の手法による嗜好情報の学習を行い、学習結果によってユーザが持つFV値を獲得する。

好感度学習手法は、文法知識と既知のFV値を用いて文章から個人の嗜好情報を学習する手法であり、次の4つの学習手法が提案されている。

1) 好き/嫌いの直接表明

好き/嫌いの直接表明では、文章の述語が好き/嫌いである時に、文章の客体に正/負の好感度与える。また、この手法では単純な好き/嫌いという表現だけでなく、良い/悪いのイメージを表す形容詞についても考える。

例えば、“私はりんごが好きだ”という文章が与えられた場合、りんごの好感度を上昇させ、“私の部屋は汚い”という文章が与えられた場合には、部屋に対する好感度を減少させる。

しかし、この手法は、文章の主体が発話者でない時には適用できない。

2) 好感度変化条件

好感度変化条件では、対話履歴から好感度変化に影響する知識を獲得し、知識構造を基にして好感度の学習を行う。この手法は、条件:知識が表す条件、状況:発生した事象、好感度変化:好感度の増加/減少、という3つの規則によって定義される。

3) 不快事象の連想

人がいくつかの不快事象に遭遇したとき、人はその不快事象に関わっている客体を憎むと考えられる。従ってこの手法では、不快であると判断された文章が与えられた場合、その文章に含まれている客体の好感度を減少させる。

例えば、“私は階段で転んだ”という文章が与えられ、この文章が不快事象であると判断された場合、この手法によって“階段”の好感度が減少する。なお、この手法による学習は、不快事象が繰り返

し発生する事によって行われるので、1度の学習での好感度の変化量は、好感度学習手法の中で最も少なくなる。

4) 感情表現からの逆算

この手法では、何らかの感情を生起する文章が与えられ、その文章に好感度が未定義の単語が1つだけ存在する場合、EGCによる情緒計算の逆算を行う。この逆算によって、文章全体の情緒から未定義の単語の好感度を推定することができる。以下にこの手法の例を示す。

$$\begin{aligned} \text{EGC} &= f_S(\text{私}) * f_O(\text{カルタ}) * f_P(\text{遊ぶ}) \\ \text{正值(快)} &= \text{正值} * \text{未定義} * \text{正值} \\ &\downarrow \\ \text{カルタの好感度} &= \text{正值(好き)} \end{aligned}$$

この例では、“カルタ”の好感度が未定義で、文章全体から快の情緒が与えられている。この時、EGCの逆算を適用する事によって“カルタ”の好感度が正值であることを推定する事ができる。

3.1 ファジィペトリネットによる推論

好感度学習手法の4つの手法うち、“好き/嫌いの直接表明”、“不快事象の連想”、“感情表現からの逆算”手法は、理論的に有効な手法であると思われる。しかし、文献⁷⁾¹¹⁾では“好感度変化条件”の手法には推論モデルが含まれていない。そのため本研究ではファジィペトリネット(FPN)¹²⁾を用いたファジィIF-THENルールによる推論を行う¹⁰⁾。

FPNを用いた推論ではEGCの計算が単純なルールで表現できることに着目し、Table 3に示すEGCの事象タイプごとのFPNルールにもとづいた推論からFV値の学習が行われる。CFは各ルールの確信度であり、 μ の値はMSTNの現在の気分から他の気分への状態遷移確率から与えられる。

Table 3: FPN Rules in Cace Frame representation

R_1	IF S and V THEN LIKE, CF = μ_1
R_2	IF S and V and OF THEN LIKE, CF = μ_2
R_3	IF S and V and OT THEN LIKE, CF = μ_3
R_4	IF S and V and OM THEN LIKE, CF = μ_4
R_5	IF S and V and OS THEN LIKE, CF = μ_5
R_6	IF S and V and O THEN LIKE, CF = μ_6
R_{71}	IF S and V and O and OF THEN LIKE, CF = μ_{71}
R_{72}	IF S and V and O and OT THEN LIKE, CF = μ_{72}
R_{81}	IF S and V and O and OF THEN LIKE, CF = μ_{81}
R_{82}	IF S and V and O and OT THEN LIKE, CF = μ_{82}
R_9	IF S and V and O and OM THEN LIKE, CF = μ_9
R_{10}	IF S and V and O and I THEN LIKE, CF = μ_{10}
R_{11}	IF S and V and O and OC THEN LIKE, CF = μ_{11}

FPNを用いたFV値学習の主な役割は、未知のFV値の適切な値を導き出すことである。さらに、既知のFV値についても、環境や状況によって生起する情緒に応じて変更する。

FPNのトークン y の値は[0.0, 1.0]であるが、FV値の範囲は[-1.0, 1.0]である。そのため、トークン y はFV値に対応する値をとる。この時、FV値とトークン y の間には1対1対応の関係があるとする。単語のFV値が未知である場合、FPNで用いられるトークンの y は0.5に設定した。

FV 値の学習は、発話文に FV 値不明の単語が含まれている場合と、含まれていない場合とで学習の手順が異なる。

まず、発話文に FV 値が未知の単語 ($y = 0.5$ と仮に定義されている単語) が含まれている場合の学習について説明する。ここでは、入力プレースを P_j とし、 P_j に含まれる FV 値が不明な単語に対応するプレースを P_u と表す。

P_j のうち y が最小の値である P_{\min} ($P_{\min} \in P_j$) を発見する。その後、式 (1) に示すように、 ΔFV_{\min} に従って P_{\min} の FV 値を修正する。

$$\begin{aligned} FV_{\min} &= FV_{\min} + \eta \Delta FV_{\min} \\ &= FV_{\min} + \eta \left(\frac{EV - y_k}{\mu_i} \right) \end{aligned} \quad (1)$$

ここで、 η は学習率であり、 EV は $[0.0, 1.0]$ の値をとるユーザの情緒値であり、 y_k は出力プレース P_k のトークンの値である。

さらに、 $P_u \neq P_{\min}$ であるならば、 P_u の FV 値を式 (2) に従って修正する。

$$\begin{aligned} FV_u &= FV_u + \eta \Delta FV_u \\ &= FV_u + \eta \left(\frac{EV - y_u \cdot \mu_i}{\mu_i} \right) \end{aligned} \quad (2)$$

以上の手順によって、発話文に含まれている未知の FV 値を学習する。

次に、発話文に FV 値が未知の単語が含まれていない場合の学習について説明する。発話文に FV 値が未知の単語が含まれていない場合には、 P_j のうち y が最小の値である P_{\min} ($P_{\min} \in P_j$) を発見し、 P_{\min} の FV 値を式 (1) に従って修正することで学習を行う。

以上の手順によって発話文に含まれている単語の FV 値を学習する。

好感度学習手法の有効性を確かめるために学習実験を行う。Table 4 の 10 種類の文章には FV 値が未知である単語が含まれており、これらの文章に対して好感度学習手法を用いた学習を行うことによって FV 値がどのように変化するかを確認する。

10 種類の文章に対するユーザの快/不快の値を用いて好感度変化条件によって FV 値を学習を行った結果、Table 5 に示す結果が得られた。Table 5 では、既知の FV 値が変化しているほか、未知の FV 値についても学習が行われ適した FV 値が設定された。学習後の FV 値を用いた EGC の情緒計算の結果は、ユーザが持つ快/不快の区別と値と比較して同等であった。

Table 4: Example sentences

Sentence 1	私は広島で渋滞に巻き込まれた
Sentence 2	待ち合わせに遅刻した
Sentence 3	私の知っている食堂は閉店した
Sentence 4	美術館に行った
Sentence 5	私は美術館を楽しんだ
Sentence 6	原爆資料館に行った
Sentence 7	もうすぐ雨が降るようだ
Sentence 8	友人が私に傘を貸さない
Sentence 9	私は知らない道に迷い込んだ
Sentence 10	私はお土産を買った

Table 5: Favorable Value Learning

Sentence	Cace Frame	Initial FV	Reinforced FV
S1	私	0.500	0.500
	渋滞	-0.700	-0.730
	巻き込む	unknown	0.604
S2	待ち合わせ	0.100	0.100
	遅刻	-0.500	-0.522
S3	食堂	0.400	0.357
	閉店	-0.100	-0.322
S4	行く	0.600	0.600
	美術館	0.500	0.400
S5	私	0.500	0.600
	楽しい	0.500	0.600
S6	行く	0.600	0.600
	原爆資料館	unknown	0.299
S7	降る	0.000	0.000
	雨	-0.500	-0.413
S8	貸す	0.500	0.250
	傘	0.500	0.250
	私	1.000	1.000
	迷い込む	unknown	0.420
S9	私	1.000	1.000
	道	0.100	-0.113
S10	買う	0.600	0.600
	お土産	unknown	0.299

4 心的状態遷移ネットワーク

MSTN は人間の複雑な気分の遷移を推定するための手法である⁸⁾⁹⁾。

MSTN では人間の持つ恒常的な気分を「驚き」、「幸福」、「悲しみ」、「怒り」、「嫌悪」、「恐れ」の 6 種類の気分と「平常」を加えた合計 7 種類に分類し、この 7 種類からなる心的状態遷移ネットワークを構築する。心的状態遷移ネットワークは条件に従って別の状態へと遷移することができ、この状態遷移によってユーザの気分の変化を推定することができる。

状態遷移を発生させる条件には、EGC による 20 種類の情緒に対話において生じるであろう 8 種類の情緒を加えた 28 種類の情緒を 9 つに分類した情緒グループと状態間の状態遷移確率が用いられ、これらの条件の組み合わせによって状態遷移が発生する。Table 6 は 28 種類の情緒グループへの分類を示してており、Fig. 1 は情緒グループに基づいた MSTN の状態遷移を示している。

本研究においては、MSTN で生起する情緒は EGC によって計算された値であると定義し、EGC の情緒を分類した各情緒グループと状態遷移確率の値に応じた状態遷移を発生させる。

また本研究では、ある一つの出来事において発せられた一連の発話文を発話エピソードと定義し、発話エピソードにおいて発生した状態遷移によって状態遷移系列を生成する。例えば、Table 4 の 10 の発話からなる発話エピソードによって生成される MSTN の状態遷移系列は Fig. 2 のようになる。

5 性格特性の学習

本研究では、個人によって異なる、落ち込みやすい、怒りやすいといった気分の遷移の傾向に対して MSTN を

Table 6: Classification of generated Emotion

No.	Emotion
①	gloating, hope, satisfaction, relief, pride, admiration, liking, gratitude, gratification, love, shy
②	joy, happy_for
③	sorry-for, shame, remorse
④	fear-confirmed, disappointment, sadness
⑤	distress, perplexity
⑥	disliking, hate
⑦	resentment, reproach, anger
⑧	fear
⑨	surprise

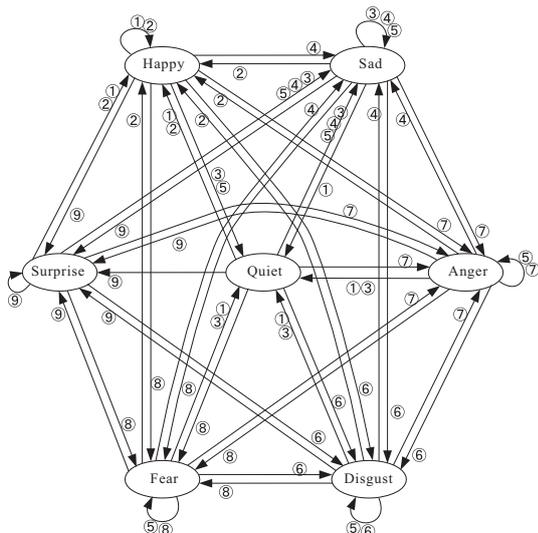


Fig. 1: MSTN with EGC

適応させることを考える。ここでは、こうした個人の気分の傾向を性格特性と呼ぶこととし、主要5因子性格検査¹³⁾においては、外向性、協調性、情緒安定性に関係する特性であると考えられる。以下では、本論文で新たに提案する性格特性の学習について説明する。

性格特性の学習は、MSTNにおいて生じた状態遷移とユーザの実際の気分の遷移が一致するようなMSTNの状態遷移確率を発見するために実行される¹⁴⁾。本研究では、発話によってその気分を変化させていると考えて、EGCによって計算された情緒にもとづいて状態遷移確率を変化させる。学習は発話エピソードごとに行われ、1つのエピソードは、EGCから入力を受けたMSTNが実行した一連の状態遷移系列として定義される。

MSTNの学習は、Profit Sharingによる発話エピソードの最短パスの獲得と、リカレントニューラルネットワークによる状態遷移確率の学習の2つの手順によ

Table 7: Transition probability in MSTN

	current mental state	next mental state						
		surprise	happy	sad	angry	disgust	fear	quiet
	surprise	0.2430	0.1900	0.0910	0.0860	0.0480	0.0760	0.2640
	happy	0.0600	0.4210	0.0610	0.0270	0.0320	0.0340	0.3640
	sad	0.0580	0.0840	0.3200	0.1080	0.0680	0.0640	0.2960
	angry	0.0750	0.0560	0.1230	0.2930	0.1210	0.0690	0.2640
	disgust	0.0560	0.0470	0.0920	0.1640	0.3130	0.0750	0.2520
	fear	0.1010	0.0500	0.1370	0.0960	0.0920	0.2790	0.2440
	quiet	0.0550	0.2130	0.0900	0.0390	0.0420	0.0510	0.5090

quiet → sad → sad → sad → happy → happy → happy → quiet → sad → sad → quiet

Fig. 2: The episode of MSTN

て実行され、以下の節ではそれぞれの手順の内容について説明する。ここで、発話エピソードの最短パスとは、発話エピソードにおいてMSTNが同じ状態を繰り返すような心理的迂回系列を除いた状態遷移系列を意味し、心理的迂回系列はシステムの観点から不要であるため削除される。

5.1 Profit Sharing

Profit Sharingは与えられた環境に適応するための強化学習として有用であり、エージェントが目標状態に到達したときに、それまでに経由した行動系列を一括して強化する¹⁵⁾。Profit Sharingにおけるルール r_i は (s, a) と記述され、 s に与えられた状態 s に対する実行可能な行動 a を表現する。

1つのエピソードは、エージェントが開始状態から時間 i で目標を達成し、報酬 R が提供される終端状態になることによって決定される。その後、エピソード($i < W$)内の発火条件 (s_i, a_i) から部分報酬 R を得る。部分報酬 R は価値関数 $f(i, R, W)$ によって決定される。

各ルールの強化は、現在の重みと式(3)の傾斜報酬の合計によって行われる。ここで、 S_{ri} は、エピソードの i 番目のルールの重みを意味し、 f_i は強化関数であり、得られた R から $-i$ 番目の時の強化値を意味する。

$$S_{ri} = S_{ri} + f_i, i = 0, 1, \dots, W - 1 \quad (3)$$

同じ感覚入力に対して違うルールが選択される時、その間に存在するルールを迂回系列と呼ぶ。迂回系列上のルールにおいて常にエピソードの迂回路上に存在するルールを無効ルールと呼び、その他のルールは有効ルールと呼ばれる。無効ルールと有効ルール間の競合が存在する場合、無効ルールは強化されない。強化関数が式(4)の無効ルール抑制定理を満たす場合、強化関数は無効ルールより効果的なルールに報酬を分配できる。式(4)式では、 W はエピソードの最大長であり、 L は有効ルールの最大数である。

$$L \sum_{j=1}^w f_j < f_{i-1}, \forall i = 1, 2, \dots, W \quad (4)$$

$i = 1$ から $i = W$ のエピソードをもとにルールの強化は式(5)に従って行われる。ここで、 $M(\geq L + 1)$ は割引率である。

$$f_i = \frac{1}{M} f_{i-1}, i = 1, 2, \dots, W - 1 \quad (5)$$

5.2 発話エピソードの最短パスの獲得

発話エピソードには、MSTNが同じ状態を繰り返すような心理的迂回系列が存在している可能性があり、システムの観点からこのような心理的迂回系列は不要である。そのため本研究では、Profit Sharingの手法を用いてエピソード内の迂回系列を発見し、エピソードの最短パスに対して学習を行う。

最短パス発見の例について説明する．ここでは，Fig. 2 で示した状態遷移系列の迂回系列を発見する．Fig. 2 に対して Profit Sharing を実行すると Fig. 3 の太字で示している部分が迂回系列として発見され，その結果，Fig. 2 から Fig. 4 に示すような最短パスが発見される．

quiet → (sad → sad → sad → **happy → happy** → **happy** → quiet → sad →) sad → quiet

Fig. 3: The episode and the detour

quiet → sad → quiet

Fig. 4: Shortest path of The episode

5.3 リカレントニューラルネットワークを用いた状態遷移確率の学習

エピソードの最短パスに対して，MSTN の状態遷移確率がユーザの状態遷移確率と一致するような誤差学習を行うことによって，MSTN の状態遷移はユーザの性格特性に適応できるようになる．

本研究では，対話における MSTN とユーザの気分の誤差を学習するために，リカレントニューラルネットワークの通時的逆伝播の学習手法を MSTN を RNN で表現した Fig. 5 に適用する．四角のユニットは入力ユニットを表し，円形のユニットは出力ユニットを表す．Fig. 5 において，7つの出力ユニットは MSTN の7つの状態のいずれかを表すものである．

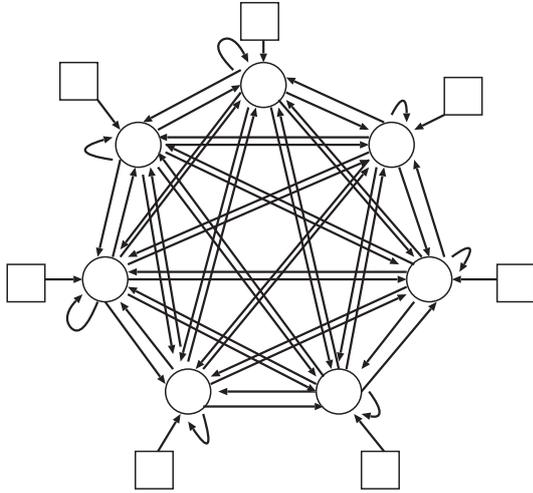


Fig. 5: Recurrent Neural Network for MSTN

リカレントニューラルネットワークの各ユニットの結合荷重は，その結合荷重に対応する MSTN の現在の状態遷移確率を用いる．

ユニット i の閾値 θ_i には初期値として 0 を与える．

また，MSTN をリカレントニューラルネットワークで行われる前向き計算では，時刻 t_0 における入力，時刻 t_0 での EGC の出力と時刻 t_1 での MSTN の状態遷移によって決定される．例えば，エピソード中の時刻 t_1 の時点で MSTN が幸福に遷移している場合，時刻 t_0 における幸福を表現する出力ユニットの入力ユニットに対して，時刻 t_0 で EGC が生じた情緒値の大き

さが入力として与えられる．

通時的逆伝播は，フィードバックを含む階層型ネットワークを離散時刻に対応する階層型ネットワークとみなして，誤差逆伝播法を適用する手法である．エピソードごと通時的逆伝播では，Fig. 6 に示すように， $t_0+1 < t_1-1$ である離散時刻 t_0, t_0+1, \dots, t_1-1 において，外部からの入力とフィードバック信号による入力 $x_i(t_0), x_i(t_0+1), \dots, x_i(t_1-1)$ を与え，時刻 t_0+1, \dots, t_1-1, t_1 において，それぞれ教師信号 $d_i(t_0+1), \dots, d_i(t_1-1), d_i(t_1)$ を与えられたときに，時刻 t_0 から t_1 までのエピソードと呼ばれる時刻区間内のネットワークの出力と教師信号との誤差の総和を極小化するような結合荷重を求める学習法である．

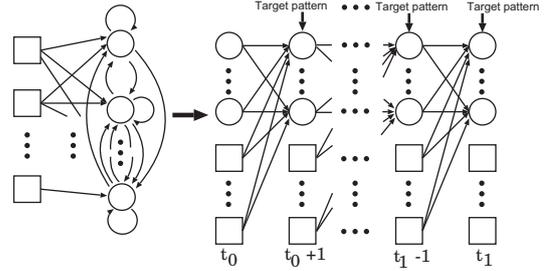


Fig. 6: Epochwise back propagation through time

MSTN ではエピソードの最短パス上の各状態遷移をエピソードとして考え，それぞれのエピソードに対して，その時点における EGC からの出力の大きさを教師信号として用いる事によって学習を行う．

この学習法での結合荷重の更新式について説明する¹⁶⁾．以下の式において，時刻 t におけるユニット i の出力を $z_i(t)$ ，入力ユニットの出力を $x_i(t)$ ，隠れユニットと出力ユニットの出力を $y_i(t)$ とする．

$$z_i(t) = \begin{cases} x_i(t) & , (i \in I \text{ のとき}) \\ y_i(t) & , (i \in H \cup O \text{ のとき}) \end{cases} \quad (6)$$

$$y_i(t) = f_i(s_i(t+1)), \quad i \in H \cup O \quad (7)$$

$$\begin{aligned} s_i(t+1) &= \sum_{j \in I} w_{ij} x_j(t) + \sum_{j \in H \cup O} w_{ij} y_j(t) + \theta_i \\ &= \sum_{j \in H \cup O} w_{ij} z_j(t) + \theta_i \end{aligned} \quad (8)$$

ここで， w_{ij} はユニット j からユニット i への結合荷重， θ_i はユニット i の閾値， $s_i(t+1)$ はユニット i の内部状態，入出力関数 $f_i(\cdot)$ は微分可能な単調増加関数であるとする．また， I, H, O はそれぞれ入力ユニット，隠れユニット，出力ユニットの添字の集合を表す．

さらに，ネットワーク全体の誤差関数は，各時刻における誤差の総和で与えられる．

$$\begin{aligned} J(t_0, t_1) &= \sum_{\tau=t_0+1}^{t_1} E(\tau) \\ &= \frac{1}{2} \sum_{\tau=t_0+1}^{t_1} \sum_{i \in O} e_i(\tau)^2 \end{aligned} \quad (9)$$

この時， $e_i(t)$ は教師信号 $d_i(t)$ と出力ユニット $i \in O$ の出力 $y_i(t)$ との誤差として表される．

$$e_i(t) = \begin{cases} d_i(t) - y_i(t) & , (i \in O \text{ のとき}) \\ 0 & , (\text{その他のとき}) \end{cases} \quad (10)$$

結合荷重と閾値の更新は、それぞれ式 (11) と式 (12) にもとづいて行われる。ここで η_w , η_θ は学習率を表している。

$$\begin{aligned} w_{ij}(t) &= w_{ij}(t) + \Delta w_{ij}(t) \\ &= w_{ij}(t) + \left(-\eta_w \sum_{\tau=t_0}^{t_1-1} \delta_i(\tau+1) z_j(\tau) \right) \end{aligned} \quad (11)$$

$$\begin{aligned} \theta_i(t) &= \theta_i(t) + \Delta \theta_i(t) \\ &= \theta_i(t) + \left(-\eta_\theta \sum_{\tau=t_0}^{t_1-1} \delta_i(\tau+1) \right) \end{aligned} \quad (12)$$

$\delta_i(\tau)$ は、 $\tau = t_1$, $t_0 \leq \tau \leq t_1 - 1$ に依存して式 (13), 式 (14) のように求められる。

$$\delta_i(t_1) = (y_i(t_1) - d_i(t_1)) f'_i(s_i(t_1)) \quad (13)$$

$$\delta_i(\tau) = f'_i(s_i(\tau)) \left((y_i(\tau) - d_i(\tau)) + \sum_{l \in H \cup O} w_{li} \frac{\partial J(t_0, t_1)}{\partial s_l(\tau)} \right) \quad (14)$$

最後に、学習された結合荷重の値から MSTN の新しい状態遷移確率を獲得する。この時、学習後の結合荷重は負の値を取る場合があるため、結合荷重の値をそのまま状態遷移確率として用いることは不適切である。そのため本研究では、Fig. 7 に示したファジィメンバーシップ関数を用いて結合荷重の値を変換し、その後、各状態における状態遷移確率の和が 1 となるように正規化することによって新しい状態遷移確率を獲得する。

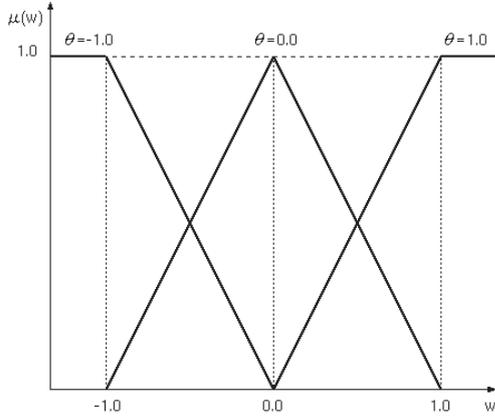


Fig. 7: Fuzzy Membership Function

6 実験

小説“恋愛寫真”¹⁷⁾において行われている会話文について MSTN の実験を行った。この小説は、誠人と静流が主人公で、2 人の恋愛が形成されていく過程が書かれており、2 人の感情の変化を測定することで、本手法の有効性を検証する試みのために用いられる。

実験で使用する会話文を Table 8 に示す。実験では、Table 8 を 1 つの発話エピソードとして、誠人が生起する気分の状態遷移系列について学習を行った。この学

Table 8: Utterance of the novel

Speaker	Utterance
誠人	ごめん
静流	うん
誠人	口の悪い連中ばかりでさ。いつもあんな調子なんだ
静流	別に気にしてないわ
誠人	そう?
誠人	ああ、なら、良かった
静流	本気でそう思う?
誠人	え?
静流	だとしたら、あなたもそうとう無神経バカ男よね
誠人	え?何で?
静流	私がつらかったのは、『変人』って言われたからじゃないわ
静流	私がつらかったのはあなたよ
誠人	ぼく?
静流	あなただけは私の味方だと思ってたのに、
静流	私のことをきちんと分かってくれと思ってたのに
静流	でも
静流	私たちは友達でしょ?あなた何度もそう言ってたわよね
誠人	うん
静流	だったら、あの子は変人なんかじゃない、
静流	ただ人よりも少しオリジナルなだけなんだって、
誠人	そのぐらい言ってくれてもよかったじゃない
誠人	ごめん
誠人	きみの言うとおりだ。ぼくは黙ってちゃいけないんだ。友達なら
静流	もういいよ。もういい。あなた、あの綺麗な女の人が好きなのね
静流	彼女に向かって一生懸命言い訳してるみたいだった
誠人	それは……
静流	いいの
静流	私には関係のないことだし
誠人	だって
静流	いいよ、ほんとに
誠人	これ
静流	どうして?
誠人	どうしてって、静流が喜ぶと思ってさ。電車に乗って買いに行ったんだ
静流	わざわざ?
誠人	わざわざってほどのことじゃないよ。
静流	誰かがさ、おいしいって言っているのを聞いたんだ。ちゃんと穴もあいてるよ
静流	子供じゃないんだから……
誠人	食べてみなよ。おいしいよ
静流	いつも静流が食べてるのはまたちょっと違うんだ。香ばしくてさ
誠人	ねえ、私を泣かせたいの?
静流	何で泣くの?
誠人	わからない。わからないけど、このビスケットを食べたら私はきっと泣くわ
静流	ものすごい泣き方するの?
静流	そうじゃない
静流	あなたよ
誠人	ぼく?
静流	あなたがきつとつらくなるの
誠人	ほんとかな
静流	そうよ

Table 9: Initial Transition probability

		next mental state						
		surprise	happy	sad	angry	disgust	fear	quiet
current	surprise	0.2430	0.1900	0.0910	0.0860	0.0480	0.0760	0.2640
	happy	0.0600	0.4210	0.0610	0.0270	0.0320	0.0340	0.3640
	sad	0.0580	0.0840	0.3200	0.1080	0.0680	0.0640	0.2960
mental	angry	0.0750	0.0560	0.1230	0.2930	0.1210	0.0690	0.2640
	disgust	0.0560	0.0470	0.0920	0.1640	0.3130	0.0750	0.2520
state	fear	0.1010	0.0500	0.1370	0.0960	0.0920	0.2790	0.2440
	quiet	0.0550	0.2130	0.0900	0.0390	0.0420	0.0510	0.0590

Table 10: Reinforced Transition probability

		next mental state						
		surprise	happy	sad	angry	disgust	fear	quiet
current	surprise	0.2499	0.2058	0.0924	0.0753	0.0328	0.0638	0.2801
	happy	0.0457	0.4641	0.0586	0.0090	0.0143	0.0164	0.3919
	sad	0.0437	0.0882	0.3476	0.0996	0.0547	0.0501	0.3162
mental	angry	0.0626	0.0567	0.1275	0.3048	0.1135	0.0556	0.2794
	disgust	0.0417	0.0467	0.0934	0.1619	0.3273	0.0625	0.2665
state	fear	0.0918	0.0500	0.1435	0.0863	0.0816	0.2893	0.2575
	quiet	0.0389	0.2344	0.0902	0.0211	0.0242	0.0341	0.5572

Table 11: Initial Threshold θ

surprise	happy	sad	angry	disgust	fear	quiet
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 12: Reinforced Threshold θ

surprise	happy	sad	angry	disgust	fear	quiet
-0.0312	-0.0115	-0.0105	-0.0311	-0.0315	-0.0317	-0.0212

習においては、出力される EGC の情緒値は誠人の嗜好にもとづいた適切な値を推定しているものとする。

実験では、一連の発話エピソードに対して Profit Sharing を実行した結果、(quiet→sad→happy) というエピソードの最短パスが発見され、その後、最短パスによって構成されたりカレントニューラルネットワークによって学習が行われた。学習を行う前の初期状態

の状態遷移確率を Table 9 に示しており、学習を行った結果、Table 10 に示す状態遷移確率を獲得した。Table 11 と Table 12 はそれぞれ、学習前の閾値 θ と、学習後の閾値 θ を示している。

まず、Table 9 に示した学習前の状態遷移確率と、Table 10 に示した学習後の状態遷移確率について比較してみる。すると、最短パス上の状態遷移に対応する (quiet→sad) と (sad→happy) において状態遷移確率が増加している事が確認できるほか、最短パス上に存在しない状態遷移確率についても、最短パス上の状態遷移に影響されるように値が変化している。例えば、surprise や angry のような最短パス上に存在しない気分からの状態遷移確率について、最短パス上で行われた sad と happy への状態遷移に影響されるように (surprise→sad) と (angry→happy) の状態遷移確率が増加している。sad と happy への状態遷移確率が増加する傾向は他の気分についても見ることができ、MSTN 全体が sad と happy へ遷移しやすくなっていると言える。このことから、提案手法では直接的にエピソードで表現されていないユーザの性格特性についても学習していると考えられる。

次に、Table 11 と Table 12 について考察する。学習後の θ は全て負の値になっている。これは結合荷重が正の値であったため、大きすぎる内部状態の値を小さくしようとした結果である。また、 θ の値は結合荷重 w の学習に影響を与える値であるため、実験結果に対する学習では、 θ の値に応じて w の変化量が増減すると考えられる。

最後に、学習結果が誠人 (ユーザ) の性格特性に一致しているかどうかを確認する。この実験では MSTN 全体で sad と happy の気分への状態遷移確率が増加しており、その他の状態への遷移確率が弱められている。この結果は、この発話エピソードにおける誠人が sad と happy の気分へ遷移しやすい性格であることを示しており、このエピソードにおける誠人の性格特性に感覚的に一致しているように思える。このことから、提案手法によって発話エピソードを学習した結果、誠人 (ユーザ) の性格特性に適応した状態遷移確率を学習できたと考えられる。

7 おわりに

本論文では、感情指向型インタフェースをユーザの性格特性に対して適応的に変化させるために、MSTN の状態遷移確率を学習する手法について提案した。提案した手法では、Profit Sharing を用いて発話エピソードの最短パスを発見し、最短パスにもとづいてリカレントニューラルネットワークのエポックごと通時的逆伝播の学習手法を行うことで MSTN の状態遷移確率を学習した。さらに、提案手法を用いて小説“恋愛写真”における発話エピソードに対する学習実験を行い、本手法によって発話エピソードからユーザの性格特性を学習できることを検証した。

今後の課題として、Fig. 7 に示した状態遷移確率を獲得するメンバーシップ関数を改良する事を考えている。リカレントニューラルネットワークにおける閾値 θ は、学習における結合荷重 w の値の変化量に影響する値であり、現在の θ の値によって学習における w の変化量が増減する。このことから、 θ の値がユーザの感

情の起伏の激しさを表していると考え、感情の起伏の激しさを MSTN の学習に適用することによってより適切な性格特性を学習するために、現在の θ の値に応じて使用するメンバーシップ関数を変更することを考えている。今後は、 θ の値に対応するメンバーシップ関数について具体的な検討を行い、実験によって状態遷移確率の学習への影響について調査を行い、個人の性格特性を表現するメンバーシップ関数の開発が急がれる。

謝辞

本研究は JSPS 科研費 25330366 の助成を受けたものである。

参考文献

- 1) 三浦麻子：ネットコミュニティでの自己表現と他者との交流，電子情報通信学会誌，Vol.91,No.2 137/141 (2008)
- 2) 三浦麻子，篠原一光：チャット・コミュニケーションに関する心理学的研究—ログ記録の解析にもとづく探索的検討，対人社会心理学研究，25/34 (2002)
- 3) 西村洋一：対人不安、インターネット利用、およびインターネットにおける人間関係，社会心理学研究，Vol.19, No.2, 124/134, (2003)
- 4) 田邊幸祐，橋一聖，市村匠：音声認識を用いた Android スマートフォン感情指向型インターフェイス，2012 IEEE SMC Hiroshima Chapter Young Researchers Workshop, 25/26 (2012)
- 5) 田邊幸祐，橋一聖，市村匠：Android スマートフォンによる感情指向型インターフェイスの開発，第 28 回ファジィシステムシンポジウム講演論文集，261/264 (2012)
- 6) 目良和也，市村匠，相沢輝昭，山下利之：語の好感度に基づく自然言語発話からの情緒生起手法，人工知能学会誌，17 巻 3 号 A, 186/195(2002)
- 7) 目良和也：語の好感度に基づく情緒生起手法の応用：電子情報通信学会技術研究報告。TL, 思考と言語 99(76), 9/16 (1999)
- 8) 任福継：言語・表情など外観情報と心的状態遷移に基づく人間感情の認知について (SLP), 情報処理学会研究報告。SLP, 音声言語情報処理, Vol.2006, No.73, 43/48 (2006)
- 9) 目良和也，市村匠，黒澤義明，竹澤寿幸：情緒計算手法と心的状態遷移ネットワークを用いた音声対話エージェントの気分変化手法，知能と情報 (日本知能情報ファジィ学会誌)，Vol.22, No.1, 10/24 (2010)
- 10) T.Ichimura and K.Tanabe: An Estimation of Favorite Value in Emotion Generating Calculation by Fuzzy Petri Net, Proc. of IEEE 6th International Workshop on Computational Intelligence and Applications (IW-CIA2013), 21/26 (2013)
- 11) 小野賢治，目良和也，市村匠，山下利之，相沢輝昭：感情表現発話からの嗜好情報学習手法，第 17 回ファジィシステムシンポジウム講演論文集，372/376 (2001)
- 12) 椎塚久雄：ファジィペトリネット，日本ファジィ学会誌，63/79 (1992)
- 13) 村上宣寛，村上千恵子：主要 5 因子性格検査の尺度構成，性格心理学研究，Vol.6, No.1, 29/39, (1997)
- 14) T.Ichimura and K.Mera: Emotion Oriented Agent in Mental State Transition Learning Network, Intl. J. Computational Intelligence Studies, Vol.2, No.1, 26/51 (2013)
- 15) 宮崎和光，木村元，小林重信：Profit Sharing に基づく強化学習の理論と応用，人工知能学会誌，vol.14, No.5, 800/807 (1999)
- 16) Jin L, Gupta MM, Stable dynamic backpropagation learning in recurrent neural networks, IEEE Transactions on Neural Networks, Vol.10, No.6, 1321/1334 (1999)
- 17) 市川拓司：恋愛写真 もうひとつの物語，42/47, 小学館出版，(2003)

サポートベクターマシンによる 導関数データを利用した関数近似法とその評価

○三木翔登 黒江康明 飯間等 (京都工芸繊維大学)

Function Approximation Methods Utilizing Derivatives Data by Support Vector Machine and Their valuation

* . Miki, Y. Kuroe and H. Iima (Kyoto Institute of Technology)

Abstract— Generally, only function value data are used for function approximation. However, not only the function value data but also derivatives data can be used in restoring surfaces of a three-dimensional object and in simulating the motion of atoms in molecular dynamics. As function approximation methods in which the derivatives data are used, we proposed methods by support vector machine (SVM) in which the cost function is based on an ε -insensitive function or the square error function. However, the performance of these methods is not evaluated sufficiently. In this paper, we evaluate the performance of the methods and their extensions through numerical experiments.

Key Words: Support vector machine, Function approximation

1 はじめに

関数近似は運動物の軌跡を扱うときなど、様々な場面で用いられる基本技術である。通常は関数値データのみが用いられるが、3次元物体表面を復元したり、分子動力学において原子の運動をシミュレーション¹⁾したりする場合には、導関数値データも使う事ができる。ところが、導関数値データは、ノイズに影響されやすく真値からかけはなれた値が出る傾向がある。

一方、コスト関数として ε -insensitive 関数を用いたサポートベクターマシン (SVM) による関数近似法が関数値データの場合に対して提案されており、ノイズに対してロバストであることが示されている^{2, 3)}。また、パターン認識の分野では、認識の精度を高めるために最小二乗 SVM が提案されている⁴⁾。そこで、著者らは、 ε -insensitive 関数をコスト関数に使用した SVM および最小二乗 SVM により、関数値データ、導関数値データを用いて未知関数を高精度に近似する方法を提案した^{5, 6)}。ところが、これらの方法の性能を十分には評価できていなかった。また、著者らは以前の研究⁷⁾で、滑らかな未知の関数から得られるノイズを含んだ関数値データ、導関数値データおよび事前知識としての滑らかさから、SVM とは異なる方法で関数を近似できることを示した。この近似関数はグリーン関数とその導関数で構成されており、この考え方を文献5, 6)の方法に導入することにより、近似精度が高められることが期待できる。そこで本研究では、データにノイズを含む場合の数値実験を通して、文献5, 6)の方法および、これらの方法で用いる近似関数にグリーン関数の導関数を追加する方法の性能を評価する。

2 問題設定

関数値データだけでなく導関数値データが与えられた場合に未知関数を近似する問題を説明する。

データの様々な獲得状況に応じるため、関数値だけが得られるサンプル位置、関数値と導関数値の両方が得られるサンプル位置、導関数値だけが得られるサンプル位置を考える。スカラー関数である滑らかな n 次元未知関数 $f(\mathbf{x})(\mathbf{x} = [x_1, \dots, x_n]^T)$ を疎なサンプル位置集合

$$D_k = \{\mathbf{x}_i \in \mathbf{R}^n \mid i = 1, \dots, N_k, \dots, N_k + N_{kd}\} \quad (1)$$

$$D_d = \{\mathbf{x}_i \in \mathbf{R}^n \mid i = N_k + 1, \dots, N_k + N_{kd}, \dots, N_k + N_{kd} + N_d\} \quad (2)$$

で観測してデータを得る。ここで、 N_k , N_{kd} , N_d はそれぞれ関数値データだけが得られるサンプル位置の数、関数値データと導関数値データの両方が得られるサンプル位置の数、導関数値データだけが得られるサンプル位置の数であり、 D_k , D_d はそれぞれ関数値データが得られる位置集合、導関数値データが得られる位置集合である。ここで、 \mathbf{x}_i における関数値データ d_i 及び $1 \sim M$ 階導関数値データ $d_i^{(m)}$ を

$$d_i = f(\mathbf{x}_i) + v_i \quad \mathbf{x}_i \in D_k \quad (3)$$

$$d_i^{(m)} = \frac{\partial^m f(\mathbf{x})}{\partial x_q^m} \Big|_{\mathbf{x}=\mathbf{x}_i} + v_i^{(m)} \quad \mathbf{x}_i \in D_d \quad (4)$$
$$(q = 1, \dots, n \quad m = 1, \dots, M)$$

とする。ここで、 v_i , $v_i^{(m)}$ はノイズを表す。このとき、以上のように与えるサンプル位置集合、関数値データ、及び導関数値データから $f(\mathbf{x})$ への近似関数 $\hat{f}(\mathbf{x})$ を構成することを目的とする。

3 関数近似法

2章の問題設定のもとで、従来法である関数値データのみを用いる SVM (以降では、通常の SVM と呼ぶ) を拡張し、SVM で定式化される最適化問題の目的関数において導関数値データと近似関数の導関数とを考慮して近似関数を構成する方法を用いる。このような方法として著者らは以前に、 ε -insensitive 関数を用いる SVM と二乗誤差関数を用いる最小二乗 SVM⁴⁾ の 2 種類を提案している^{5, 6)}。 ε -insensitive 関数とは、近似関数値から ε の距離内にある関数値データの誤差を 0 とみなす関数である。この関数を導関数値データにも導入することにより、データに含まれるノイズに対してロバストとなることが期待できる。一方、最小二乗 SVM はもともとパターン認識の分野で用いられている手法で、二乗誤差を考慮することで認識精度を高めている。文献 6) では関数の近似の精度が向上することを期待して最小二乗 SVM を用いている。この方法では、二乗誤差を用いることにより、最終的には連立方程式を解くだけで近似関数を求めることができる。また、近似精度を高めるためには近似関数の基底系もグリーン関数とその導関数の線形和で表わせれば良いことを導出している⁷⁾。そこで、2 種類の SVM の近似関数の基底系にグリーン関数の導関数を追加する手法を提案し、2 種類の SVM とともにそれらの性能を評価する。

3.1 節に ε -insensitive 関数を用いる SVM で導関数値データを用いる関数近似法を、3.2 節に二乗誤差関数を用いる最小二乗 SVM で導関数値データを用いる関数近似法を示す。3.3 節では、3.1 節、3.2 節で示した手法を基に、近似関数の基底にグリーン関数の導関数を追加して近似を行う手法を示す。

3.1 ε -insensitive 関数を用いる SVM

未知関数を近似する関数 $\hat{f}_1(\mathbf{x})$ は、通常の SVM と同様に以下の関数とする。

$$\hat{f}_1(\mathbf{x}) = \mathbf{C}^T \Phi(\mathbf{x}) \quad (5)$$

$$\Phi(\mathbf{x}) = [G(\mathbf{x}, \mathbf{x}_1), \dots, G(\mathbf{x}, \mathbf{x}_{N_k+N_{kd}+N_d})]^T \quad (6)$$

ここで、 $G(\mathbf{x}, \mathbf{x}_i)$ はグリーン関数である。 $\Phi(\mathbf{x})$ は、事前知識として滑らかさを考慮にいたした場合に、近似関数の基底系が各サンプル点を中心にもつグリーン関数となること⁷⁾に基づいて導入している。また \mathbf{C} は、 $N_k + N_{kd} + N_d$ 次元の未知ベクトルである。

\mathbf{C} を決定するにあたり、通常の SVM のマージン最大化に加えて、導関数値データと近似関数の導関数値との ε -insensitive 関数による損失を加えた最適化問題

を次のように定義する。

$$\begin{aligned} \mathbf{C}, \varepsilon, \xi^*, \xi_{m,q}, \xi_{m,q}^* \quad \min \quad J_1 = & \frac{1}{2} \mathbf{C}^T \mathbf{C} + U \sum_{i=1}^{N_k+N_{kd}} (\xi_i + \xi_i^*) \\ & + \sum_{m=1}^M \sum_{q=1}^n U_{m,q} \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\xi_{m,q,i} + \xi_{m,q,i}^*) \end{aligned} \quad (7)$$

subject to

$$d_i - \hat{f}_1(\mathbf{x}_i) \leq \varepsilon + \xi_i \quad (i = 1, \dots, N_k + N_{kd}) \quad (8)$$

$$\hat{f}_1(\mathbf{x}_i) - d_i \leq \varepsilon + \xi_i^* \quad (i = 1, \dots, N_k + N_{kd}) \quad (9)$$

$$\xi_i \geq 0 \quad (i = 1, \dots, N_k + N_{kd}) \quad (10)$$

$$\xi_i^* \geq 0 \quad (i = 1, \dots, N_k + N_{kd}) \quad (11)$$

$$d_i^{(m,q)} - \hat{f}_1^{(m,q)}(\mathbf{x}_i) \leq \varepsilon_{m,q} + \xi_{m,q,i} \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m) \quad (12)$$

$$\hat{f}_1^{(m,q)}(\mathbf{x}_i) - d_i^{(m,q)} \leq \varepsilon_{m,q} + \xi_{m,q,i}^* \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m) \quad (13)$$

$$\xi_{m,q,i} \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m) \quad (14)$$

$$\xi_{m,q,i}^* \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m) \quad (15)$$

ここで、 $\hat{f}_1^{(m,q)}(\mathbf{x}_i) = \left. \frac{\partial^m \hat{f}_1(\mathbf{x})}{\partial \mathbf{x}_q^m} \right|_{\mathbf{x}=\mathbf{x}_i}$ である。また、 $\xi_i, \xi_i^*, \xi_{m,q,i}, \xi_{m,q,i}^*$ はスラック変数、 $U, U_{m,q}$ は重みである。 $\varepsilon, \varepsilon_{m,q}$ はそれぞれ近似関数値、近似導関数値と関数値データ、導関数値データとの誤差を無視する範囲を表す非負の定数である。この制約つき最適化問題を解くために、次のラグランジュ関数を導入する。

$$\begin{aligned} L_1 = & J_1 + \sum_{i=1}^{N_k+N_{kd}} \alpha_i [d_i - \hat{f}_1(\mathbf{x}_i) - \varepsilon - \xi_i] \\ & + \sum_{i=1}^{N_k+N_{kd}} \alpha_i^* [\hat{f}_1(\mathbf{x}_i) - d_i - \varepsilon - \xi_i^*] \\ & - \sum_{i=1}^{N_k+N_{kd}} (\gamma_i \xi_i + \gamma_i^* \xi_i^*) \\ & + \sum_{m=1}^M \sum_{q=1}^n \left[\sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \alpha_{m,q,i} \times \right. \\ & \quad [d_i^{(m,q)} - \hat{f}_1^{(m,q)}(\mathbf{x}_i) - \varepsilon_{m,q} - \xi_{m,q,i}] \\ & \quad + \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \alpha_{m,q,i}^* \times \\ & \quad \left. [\hat{f}_1^{(m,q)}(\mathbf{x}_i) - d_i^{(m,q)} - \varepsilon_{m,q} - \xi_{m,q,i}^*] \right. \\ & \quad \left. - \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\gamma_{m,q,i} \xi_{m,q,i} + \gamma_{m,q,i}^* \xi_{m,q,i}^*) \right] \end{aligned} \quad (16)$$

ただし, $\alpha_i, \alpha_i^*, \alpha_{m,q,i}, \alpha_{m,q,i}^*, \gamma_i, \gamma_i^*, \gamma_{m,q,i}, \gamma_{m,q,i}^*$ は, 次式を満たすラグランジュ乗数である.

$$\alpha_i \geq 0 \quad (i = 1, \dots, N_k + N_{kd}) \quad (17)$$

$$\alpha_i^* \geq 0 \quad (i = 1, \dots, N_k + N_{kd}) \quad (18)$$

$$\alpha_{m,q,i} \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (19)$$

$$\alpha_{m,q,i}^* \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (20)$$

$$\gamma_i \geq 0 \quad (i = 1, \dots, N) \quad (21)$$

$$\gamma_i^* \geq 0 \quad (i = 1, \dots, N) \quad (22)$$

$$\gamma_{m,q,i} \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (23)$$

$$\gamma_{m,q,i}^* \geq 0 \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (24)$$

最適解である鞍点において, $\frac{\partial L_1}{\partial \mathbf{C}} = \mathbf{0}$, $\frac{\partial L_1}{\partial \xi} = \frac{\partial L_1}{\partial \xi^*} = 0$ ($i = 1, 2, \dots, N_k + N_{kd}$), $\frac{\partial L_1}{\partial \xi_{m,q}} = \frac{\partial L_1}{\partial \xi_{m,q}^*} = 0$ ($i = N_k + 1, N_k + 2, \dots, N_k + N_{kd} + N_d, \forall q, m$) が成立し, 以下の双対問題を得る.

$$\begin{aligned} \max_{\alpha, \alpha^*, \alpha_{m,q}, \alpha_{m,q}^*} L_1 = & -\frac{1}{2} \mathbf{C}^T \mathbf{C} \\ & + \sum_{i=1}^{N_k + N_{kd}} (\alpha_i - \alpha_i^*) (d_i - \hat{f}_1(\mathbf{x}_i)) - \varepsilon \sum_{i=1}^{N_k + N_{kd}} (\alpha_i + \alpha_i^*) \\ & + \sum_{m=1}^M \sum_{q=1}^n \left[\sum_{i=N_k+1}^{N_k + N_{kd} + N_d} (\alpha_{m,q,i} - \alpha_{m,q,i}^*) \right. \\ & \times (d_i^{(m,q)} - \hat{f}_1^{(m,q)}(\mathbf{x}_i)) \\ & \left. - \varepsilon_{m,q} \sum_{i=N_k+1}^{N_k + N_{kd} + N_d} (\alpha_{m,q,i} + \alpha_{m,q,i}^*) \right] \end{aligned} \quad (25)$$

subject to

$$0 \leq \alpha_i \leq U \quad (i = 1, \dots, N_k + N_{kd}) \quad (26)$$

$$0 \leq \alpha_i^* \leq U \quad (i = 1, \dots, N_k + N_{kd}) \quad (27)$$

$$0 \leq \alpha_{m,q,i} \leq U_{m,q} \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (28)$$

$$0 \leq \alpha_{m,q,i}^* \leq U_{m,q} \quad (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \quad (29)$$

以上より, 導関数値データを利用して近似関数を求めることは, 双対問題 (25)~(29) を解くことに帰着される.

双対問題の解を $\tilde{\alpha}_i, \tilde{\alpha}_i^*, \tilde{\alpha}_{m,q,i}, \tilde{\alpha}_{m,q,i}^*$ とすると, (5)

式と $\frac{\partial L_1}{\partial \mathbf{C}} = \mathbf{0}$ より近似関数は次のように求まる.

$$\begin{aligned} \hat{f}_1(\mathbf{x}) = & \sum_{i=1}^{N_k + N_{kd}} (\tilde{\alpha}_i - \tilde{\alpha}_i^*) \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) \\ & + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k + N_{kd} + N_d} (\tilde{\alpha}_{m,q,i} - \tilde{\alpha}_{m,q,i}^*) \times \\ & \Phi^{(m,q)}(\mathbf{x}_i)^T \Phi(\mathbf{x}) \end{aligned} \quad (30)$$

ここで, $\Phi^{(m,q)}(\mathbf{x}) = \frac{\partial^m}{\partial \mathbf{x}_q^m} \Phi(\mathbf{x})$ である.

3.2 二乗誤差関数を用いる SVM

未知関数を近似する関数 $\hat{f}_1(\mathbf{x})$ は, 通常の SVM と同様に (5), (6) 式の関数とする. \mathbf{C} を決定するにあたり通常の SVM のマージン最大化に加えて, サンプル位置の関数値と導関数値の二乗誤差を最小化する最適化問題を次式で定義する.

$$\begin{aligned} \min_{\mathbf{C}, e, e^{(m,q)}} J_2 = & \frac{1}{2} \mathbf{C}^T \mathbf{C} + \frac{1}{2} \sum_{i=1}^{N_k + N_{kd}} \gamma e_i^2 \\ & + \frac{1}{2} \sum_{m=1}^M \sum_{q=1}^n \delta^{(m,q)} \sum_{i=N_k+1}^{N_k + N_{kd} + N_d} e_i^{(m,q)2} \end{aligned} \quad (31)$$

subject to

$$e_i = d_i - \hat{f}_1(\mathbf{x}_i) \quad (i = 1, \dots, N_k + N_{kd}) \quad (32)$$

$$\begin{aligned} e_i^{(m,q)} = & d_i^{(m,q)} - \hat{f}_1^{(m,q)}(\mathbf{x}_i) \\ & (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m) \end{aligned} \quad (33)$$

ここで, γ と $\delta^{(m,q)}$ は重みである.

この制約条件付き最適化問題を解くために, 次のラグランジュ関数を導入する.

$$\begin{aligned} L_2 = J_2 + & \sum_{i=1}^{N_k + N_{kd}} \alpha_i (e_i - (d_i - \hat{f}_1(\mathbf{x}_i))) \\ & + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k + N_{kd} + N_d} \beta_i^{(m,q)} \\ & \times (e_i^{(m,q)} - (d_i^{(m,q)} - \hat{f}_1^{(m,q)}(\mathbf{x}_i))) \end{aligned} \quad (34)$$

ただし, $\alpha_i, \beta_i^{(m,q)}$ はラグランジュ乗数である. 最適解において $\frac{\partial L_2}{\partial \mathbf{C}} = \mathbf{0}$, $\frac{\partial L_2}{\partial e} = 0$, $\frac{\partial L_2}{\partial e^{(m,q)}} = 0$ が成立することから, 問題 (31)~(33) は次の双対問題に置き換えら

れる。

$$\begin{aligned}
\max_{\alpha, \beta^{(m^q)}} L_2 = & -\frac{1}{2} \sum_{i,j=1}^{N_k+N_{kd}} \alpha_i \alpha_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \\
& - \sum_{i,j=N_k+1}^{N_k+N_{kd}} \sum_{m=1}^M \sum_{q=1}^n \alpha_i \beta_j^{(m^q)} \Phi^{(m^q)}(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \\
& - \frac{1}{2} \sum_{i,j=N_k+1}^{N_k+N_{kd}+N_d} \sum_{m,l=1}^M \sum_{p,q=1}^n \beta_i^{(m^q)} \beta_j^{(l^p)} \\
& \quad \times \Phi^{(m^q)}(\mathbf{x}_i)^T \Phi^{(l^p)}(\mathbf{x}_j) \\
& - \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \left(\frac{\beta_i^{(m^q)^2}}{2\delta^{(m^q)}} + \beta_i^{(m^q)} d_i^{(m^q)} \right) \\
& - \sum_{i=1}^{N_k+N_{kd}} \frac{\alpha_i^2}{2\gamma} + \alpha_i d_i \tag{35}
\end{aligned}$$

この問題は無制約の2次計画問題であり、容易に最適解が求められる。最適解を求めるために、 $\frac{\partial L_2}{\partial \alpha} = 0$, $\frac{\partial L_2}{\partial \beta^{(m^q)}} = 0$ とすることにより次式が得られる。

$$\begin{aligned}
& \sum_{j=1}^{N_k+N_{kd}} \alpha_j \Phi(\mathbf{x}_j)^T \Phi(\mathbf{x}_i) \\
& + \sum_{m=1}^M \sum_{q=1}^n \sum_{j=N_k+1}^{N_k+N_{kd}+N_d} \beta_j^{(m^q)} \Phi^{(m^q)}(\mathbf{x}_j)^T \Phi(\mathbf{x}_i) \\
& + \frac{\alpha_i}{\gamma} - d_i = 0 \tag{36} \\
& (i = 1, \dots, N_k + N_{kd})
\end{aligned}$$

$$\begin{aligned}
& \sum_{j=N_k+1}^{N_k+N_{kd}} \alpha_j \Phi(\mathbf{x}_j)^T \Phi^{(m^q)}(\mathbf{x}_i) \\
& + \sum_{l=1}^M \sum_{p=1}^n \sum_{j=N_k+1}^{N_k+N_{kd}+N_d} \beta_j^{(l^p)} \Phi^{(l^p)}(\mathbf{x}_j)^T \Phi^{(m^q)}(\mathbf{x}_i) \\
& + \frac{\beta_i^{(m^q)}}{\delta^{(m^q)}} - d_i^{(m^q)} = 0 \tag{37} \\
& (i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m)
\end{aligned}$$

以上より、連立方程式(37)(38)を解けばよい。これらの解 $\alpha_i^*, \beta_i^{*(m^q)}$ を用いて、(5)式と $\frac{\partial L_2}{\partial \mathbf{C}} = \mathbf{0}$ より近似関数は次式で求まる。

$$\begin{aligned}
\hat{f}_1(\mathbf{x}) = & \sum_{i=1}^{N_k+N_{kd}} \alpha_i^* \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) \\
& + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \beta_i^{*(m^q)} \Phi^{(m^q)}(\mathbf{x}_i)^T \Phi(\mathbf{x}) \tag{38}
\end{aligned}$$

3.3 近似関数の基底にグリーン関数の導関数を追加する手法

未知関数を近似する関数 $\hat{f}_2(\mathbf{x})$ は、以下の関数とする。

$$\hat{f}_2(\mathbf{x}) = \hat{f}_1(\mathbf{x}) + \sum_{m=1}^M \sum_{q=1}^n \mathbf{C}_{m,q}^T \Phi^{(m^q)}(\mathbf{x}) \tag{39}$$

この関数は、近似関数の基底系がグリーン関数とその導関数の線形和で表されることに基づいている⁷⁾。また、 $\mathbf{C}_{m,q}$ は $N_k + N_{kd} + N_d$ 次元のベクトルである。

3.3.1 ε -insensitive 関数を用いる SVM

3.1項と同様に、最適化問題を次のように定義する。

$$\begin{aligned}
\min_{\mathbf{C}, \mathbf{C}_{m,q}, \xi, \xi^*, \xi_{m,q}, \xi_{m,q}^*} J_3 = & J_1 \\
& + \frac{1}{2} \sum_{m=1}^M \sum_{q=1}^n \mathbf{C}_{m,q}^T \mathbf{C}_{m,q} \tag{40}
\end{aligned}$$

subject to (10), (11), (14), (15) and

$$d_i - \hat{f}_2(\mathbf{x}_i) \leq \varepsilon + \xi_i \quad (i = 1, 2, \dots, N_k + N_{kd}) \tag{41}$$

$$\hat{f}_2(\mathbf{x}_i) - d_i \leq \varepsilon + \xi_i^* \quad (i = 1, 2, \dots, N_k + N_{kd}) \tag{42}$$

$$d_i^{(m^q)} - \hat{f}_2^{(m^q)}(\mathbf{x}_i) \leq \varepsilon_{m,q} + \xi_{m,q,i} \tag{43}$$

$$(i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m)$$

$$\hat{f}_2^{(m^q)}(\mathbf{x}_i) - d_i^{(m^q)} \leq \varepsilon_{m,q} + \xi_{m,q,i}^* \tag{44}$$

$$(i = N_k + 1, \dots, N_k + N_{kd} + N_d, \forall q, m)$$

ここで $\hat{f}_2^{(m^q)}(\mathbf{x}_i) = \frac{\partial^m \hat{f}_2(\mathbf{x})}{\partial x_i^m} \Big|_{\mathbf{x}=\mathbf{x}_i}$ である。双対問題は次式となる。

$$\begin{aligned}
\max_{\alpha, \alpha^*, \alpha_{m,q}, \alpha_{m,q}^*} L_3 = & -\frac{1}{2} \mathbf{C}^T \mathbf{C} \\
& + \sum_{i=1}^{N_k+N_{kd}} (\alpha_i - \alpha_i^*) (d_i - \hat{f}_2(\mathbf{x}_i)) - \varepsilon \sum_{i=1}^{N_k+N_{kd}} (\alpha_i + \alpha_i^*) \\
& + \sum_{m=1}^M \sum_{q=1}^n \left[\sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\alpha_{m,q,i} - \alpha_{m,q,i}^*) \right. \\
& \quad \times (d_i^{(m^q)} - \hat{f}_2^{(m^q)}(\mathbf{x}_i)) \\
& \quad \left. - \varepsilon_{m,q} \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\alpha_{m,q,i} + \alpha_{m,q,i}^*) \right] \tag{45}
\end{aligned}$$

$$+ \sum_{m=1}^M \sum_{q=1}^n \frac{1}{2} \mathbf{C}_{m,q}^T \mathbf{C}_{m,q} \tag{46}$$

subject to (26), (27), (28), (29)

これを解くと $\hat{f}_2(\mathbf{x})$ は次式で求まる.

$$\begin{aligned} \hat{f}_2(\mathbf{x}) = & \sum_{i=1}^{N_k+N_{kd}} (\tilde{\alpha}_i - \tilde{\alpha}_i^*) \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) \\ & + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\tilde{\alpha}_{m,q,i} - \tilde{\alpha}_{m,q,i}^*) \\ & \quad \times \Phi^{(m^q)^T}(\mathbf{x}_i) \Phi(\mathbf{x}) \\ & + \sum_{l=1}^M \sum_{p=1}^n \left[\sum_{i=1}^{N_k+N_{kd}} (\tilde{\alpha}_i - \tilde{\alpha}_i^*) \Phi^{(lp)}(\mathbf{x}_i)^T \Phi^{(lp)}(\mathbf{x}) \right. \\ & \quad + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} (\tilde{\alpha}_{m,q,i} - \tilde{\alpha}_{m,q,i}^*) \\ & \quad \left. \times \Phi^{(m^q)(lp)}(\mathbf{x}_i)^T \Phi^{(lp)}(\mathbf{x}) \right] \end{aligned} \quad (47)$$

ここで,

$$\Phi^{(m^q)(lp)}(\mathbf{x}_i) = \begin{cases} \frac{\partial^{m+} \Phi(\mathbf{x})}{\partial x_q^m \partial x_p} \Big|_{\mathbf{x}=\mathbf{x}} & (q \neq p) \\ \frac{\partial^{m+} \Phi(\mathbf{x})}{\partial x_q^{m+}} \Big|_{\mathbf{x}=\mathbf{x}} & (q = p) \end{cases} \quad (48)$$

である.

3.3.2 二乗誤差関数を用いる SVM

3.2 項と同様に, 最適化問題を次のように定義する.

$$\mathcal{C}, \mathcal{C}_{m,q,e}, e^{(m^q)} \min J_4 = J_2 + \frac{1}{2} \sum_{m=1}^M \sum_{q=1}^n \mathcal{C}_{m,q}^T \mathcal{C}_{m,q} \quad (49)$$

subject to

$$e_i = d_i - \hat{f}_2(\mathbf{x}_i) \quad (i = 1, \dots, N_k + N_{kd}) \quad (50)$$

$$e_i^{(m^q)} = d_i^{(m^q)} - \hat{f}_2^{(m^q)}(\mathbf{x}_i) \quad (51)$$

$$(i = N_k + 1, \dots, N_k + N_{kd} + N_d \quad \forall q, m)$$

双対問題は次式となる.

$$\begin{aligned} L_4 = & J_4 + \sum_{i=1}^{N_k+N_{kd}} \alpha_i (e_i - (d_i - \hat{f}_2(\mathbf{x}_i))) \\ & + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \beta_i^{(m^q)} \\ & \times (e_i^{(m^q)} - (d_i^{(m^q)} - \hat{f}_2^{(m^q)}(\mathbf{x}_i))) \end{aligned} \quad (52)$$

これを解くと $\hat{f}_2(\mathbf{x})$ は次式で求まる.

$$\begin{aligned} \hat{f}_2(\mathbf{x}) = & \sum_{i=1}^{N_k+N_{kd}} \alpha_i^* \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) \\ & + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \beta_i^{*(m^q)} \Phi^{(m^q)}(\mathbf{x}_i)^T \Phi(\mathbf{x}) \\ & + \sum_{l=1}^M \sum_{p=1}^n \left[\sum_{i=1}^{N_k+N_{kd}} \alpha_i^* \Phi^{(lp)}(\mathbf{x}_i)^T \Phi^{(lp)}(\mathbf{x}) \right. \\ & \quad \left. + \sum_{m=1}^M \sum_{q=1}^n \sum_{i=N_k+1}^{N_k+N_{kd}+N_d} \beta_i^{*(m^q)} \right. \\ & \quad \left. \times \Phi^{(m^q)(lp)}(\mathbf{x}_i)^T \Phi^{(lp)}(\mathbf{x}) \right] \end{aligned} \quad (53)$$

4 数値実験

数値実験により導関数値データを利用した 4 種類の SVM の有効性を検証する.

4.1 実験方法

変数 \mathbf{x} の次元 n を 2, すなわち, $\mathbf{x} = [x_1, x_2]^T$ とする. 未知関数 f は次式のガウス関数とする.

$$\begin{aligned} f(\mathbf{x}) = & \exp\left(-\frac{\|\mathbf{x} - \mathbf{o}_1\|^2}{2}\right) + \exp\left(-\frac{\|\mathbf{x} - \mathbf{o}_2\|^2}{2}\right) \\ & + \exp\left(-\frac{\|\mathbf{x} - \mathbf{o}_3\|^2}{2}\right) \end{aligned} \quad (54)$$

$$\mathbf{o}_1 = [0.0, 0.0]^T, \quad \mathbf{o}_2 = [-5.0, -2.5]^T, \quad \mathbf{o}_3 = [2.5, 5.0]^T$$

関数値データ, 導関数値データには観測に伴うノイズを加え, ノイズの値は次のように与えた.

$$v_i = \pi \Pi \quad (55)$$

$$v_i^{(m^q)} = \pi_q^{(m)} \Pi_q^{(m)} \quad (56)$$

ここで Π と $\Pi_q^{(m)}$ は平均値が 0, 分散が 1 である独立な正規乱数である. また, $\pi, \pi_q^{(m)}$ はノイズの大きさをデータ値によって正規化するためのものであり, これらをノイズ強度と呼ぶ. 導関数値データに加わる各ノイズ強度 $\pi_q^{(m)}$ ($q = 1, 2, m = 1, 2, \dots, M$) は, 真の関数値と導関数値群のパワー

$$P = \left(\frac{\sum_{\mathbf{x} \in D_k} |f(\mathbf{x}_i)|^2}{N_k + N_{kd}} \right)^{\frac{1}{2}} \quad (57)$$

$$P_q^{(m)} = \left(\frac{\sum_{\mathbf{x} \in D_d} \left| \frac{\partial^m}{\partial x_q^m} f(\mathbf{x}_i) \right|^2}{N_{kd} + N_d} \right)^{\frac{1}{2}} \quad (58)$$

のそれぞれに対する各ノイズのパワーの割合が全て等しくなるように, π を用いて次式で正規化する.

$$\pi_q^{(m)} = \frac{P_q^{(m)}}{P} \pi \quad (59)$$

サンプル位置の数に関して、 $N_{kd} = 25$ となる例題と $N_{kd} = 64$ となる例題とをそれぞれ複数個ずつ作成し、いずれも $N_k = N_d = 0$ とする。サンプル位置集合 D_k, D_d は $(-5 \leq x_1 \leq 5, -5 \leq x_2 \leq 5)$ の範囲の中で選択された点の集合とする。導関数値データは $M = 2$ 、すなわち 1 階と 2 階のみとする。ノイズ強度は $\pi = 0.0$ から 0.1 まで変化させて、各 N_{kd} 、 π に対して 5 例ずつ例題を作成した。

以上のように作成した例題に対して下記の 5 つの SVM を適用し、結果を比較する。

- SVM: 導関数値データを用いない通常の SVM.
- ε SVM: 導関数値データを利用し、 ε -insensitive 関数を用いる SVM.
- LSSVM: 導関数値データを利用し、二乗誤差関数を用いる SVM.
- ε SVM-dG: ε SVM の、近似関数の基底にグリーン関数の導関数を追加する SVM.
- ε SVM-dG: LSSVM の、近似関数の基底にグリーン関数の導関数を追加する SVM.

導関数値データを利用する提案法の (6) 式で用いるグリーン関数は、下記のガウス関数とする。

$$G(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2}\right) \quad (60)$$

また、目的関数 (7) における重み $U, U_{m,q}$ はそれぞれ 100, 1、目的関数 (31) における重み $\gamma, \delta^{(m,q)}$ はそれぞれ 50, 1、(8)(9)(12)(13) 式における定数 $\varepsilon, \varepsilon_{m,q}$ はいずれも 0.01 とした。これらは何通りかの値の組み合わせを変えて実験し、結果を比べた時に近似精度が良かったものを利用した。 ε -insensitive 関数を用いる SVM における最適化問題の解法は逐次勾配登り法⁸⁾を用いた。

4.2 評価基準

近似関数 \hat{f} の精度を定量的に評価するための基準として、次式で定義される近似関数 \hat{f} と未知関数 f の平均二乗誤差 (RMSE) を用いる。

$$\text{RMSE}[\hat{f}, f] \equiv \sqrt{\frac{1}{|D_E|} \sum_{\mathbf{x} \in D_E} |\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)|^2} \quad (61)$$

$$D_E = \{x_1 = -5 + 0.2k\} \otimes \{x_2 = -5 + 0.2k\} \quad (62)$$

$$(k = 0, 1, \dots, 50), |D_E| = 51 \times 51$$

各 N_{kd} 、 π ごとに各 SVM の RMSE の 5 例分に対する平均値を比較する。

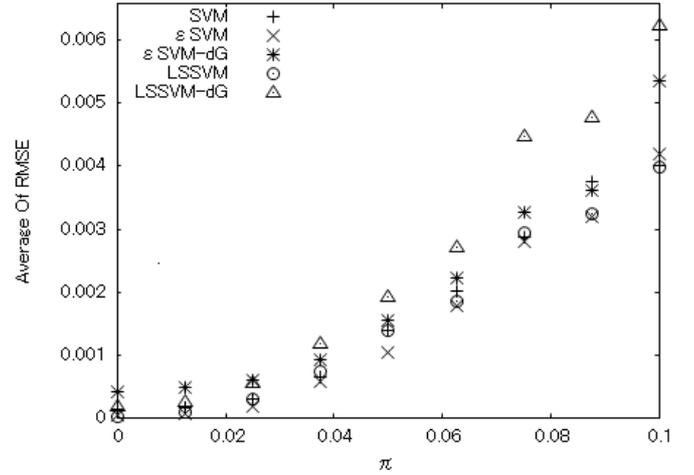


Fig. 1: Comparison of the RMSE among the fire approximation function methods ($N_{kd} = 25$)

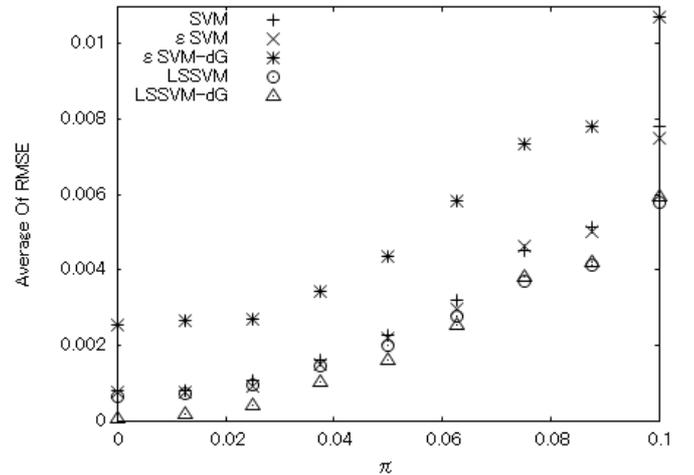


Fig. 2: Comparison of the RMSE among the fire approximation function methods ($N_{kd} = 64$)

4.3 結果と考察

結果を Fig.1,2 として示す。

Fig.1 の $N_{kd} = 25$ の場合をみると、ほぼすべての場合において ε SVM の RMSE が最も小さい。これにより、 ε SVM は導関数値データに含まれるノイズに対する一定のロバスト性を有していると言える。一方で LSSVM-dG は、他に比べると近似精度が劣り、特に $\pi > 0.05$ では通常の SVM の 1.2 から 1.5 倍の RMSE を示している。

Fig.2 の $N_{kd} = 64$ の場合をみると、多くの場合において LSSVM-dG の RMSE が最も小さい。ノイズの強度が大きくなると、前述した $N_{kd} = 25$ の場合と同様に近似精度が下がっている。一方で ε SVM-dG は、総じて他の手法の 1.5~2 倍の RMSE を示している。サ

サンプル数 N_{kd} が 25 から 64 に多くなることで、ロバスト性が十分に発揮されなくなっている。

5 まとめ

本研究は、滑らかな未知関数からの観測値がノイズを含む場合に、関数値データに加えて導関数値データも利用して近似関数を構成する 4 種類の方法を提案し、それらの性能を評価した。数値実験の結果より、各提案手法の間ではノイズの強度やサンプル点の数に依存して近似精度の良い手法は異なっているが、これらの方法の近似精度は導関数値データを用いない方法のそれより優れていることが分かった。今回は対象とした未知関数と近似関数の両方にガウス関数を用いたが、今後は他の複数の未知関数に関して提案手法の近似精度を評価していきたい。

参考文献

- 1) A. Pukrittayakamee, M. Hagan, L. Raff, S.T.S. Bukkapatnam, and R. Komanduri : "Practical Training Framework for Fitting a Function and Its Derivatives", IEEE Transactions on Neural Networks, Volume 22, Issue 6, pp.936-947 (2011)
- 2) S. Haykin: Neural Networks, pp.318-350, Prentice Hall (1999)
- 3) V. Vapnik, S. Golowich, and A. Smola : "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing", Advances in Neural Information Processing Systems, Vol.9, pp.281-289, MIT Press (1996)
- 4) J.A.K. Suykens, L. Lukas, P. Van Dooren, B. De Moor, and J. Vandewalle : "Least Squares Support Vector Machine Classifiers: a Large Scale Algorithm", Statistical Analysis and Data Mining, pp.272-286 (2010)
- 5) 木村, 黒江, 飯間 : "サポートベクターマシンによる導関数データを利用した関数近似", 第 32 回知能システムシンポジウム資料, pp.59-64 (2005)
- 6) 三木, 黒江, 飯間 : "最小二乗サポートベクターマシンによる導関数データを利用した関数近似", 第 40 回知能システムシンポジウム資料, pp.257-260 (2013)
- 7) 川上, 黒江, 森 : "導関数値群の融合に基づく未知関数の近似", システム制御情報学会論文誌, Vol. 9, No. 5, pp.210-218 (1996)
- 8) N.Cristianini, J.S.Taylor (訳, 大北 剛): サポートベクターマシン入門, pp.169-176, 共立出版 (2010)

遺伝子ネットワークにおけるアイソクロンの簡単な計算法

○森 禎弘 黒江康明 (京都工芸繊維大学)

A Simple Computational Method of Isochrons in Gene Regulatory Networks

*Y. Mori and Y. Kuroe (Kyoto Institute of Technology)

Abstract— There exist periodic phenomena in organisms such as circadian rhythm and those are considered to be generated by gene regulatory networks. Some periodic phenomena in organisms are affected by some stimuli and timings of the oscillations change. For example, clocks of circadian rhythm in organisms are considered to be affected by light stimuli and phases of the clocks shift, that is timings of the oscillations change. In order to understand such mechanisms, phase shifts caused by perturbations are investigated. Isochron and asymptotic phase are often used for analysis of phase shift of periodic phenomena. However, it is almost impossible to obtain asymptotic phases and isochrons analytically. Therefore, they are computed approximately by using numerical methods. In this paper, we show a simple computational method of isochrons in gene regulatory networks. In the method, we compute isochrons by using maps obtained analytically. Therefore we can obtain isochrons without using numerical method solving nonlinear differential equations.

Key Words: gene regulatory network, isochron, phase

1 はじめに

生体には概日リズムなど様々な周期現象が存在し、それらは遺伝子ネットワークによって生成されていると考えられている。周りの環境や相互作用などの摂動を受けたとき解軌道は周期軌道から外れるが、いずれ元の周期解軌道に戻る。このとき、一般にその位相は変化する。すなわち、摂動を受けなかったときとは異なるタイミングで振動する。例えば、何らかの理由で時差ぼけ状態になったとき、概日リズムを実現する体内時計は環境の光刺激を受けてその位相が調整され、周りの環境の昼夜のサイクルに合わせたリズムを刻むようになると考えられている。

このような現象を理解するために、周期現象において摂動により位相がどのように変化するかに興味を持たれ解析されている^{1, 2)}。このとき、摂動によってどの位置に移ったかという情報から元の周期軌道に戻ったときの位相の変化量が直ちに分かるならば有用である。そこで、以下に説明するアイソクロンや位相を用いて摂動による位相の変化について解析が行われている。安定な周期解軌道の吸引領域内の異なる複数の点から始まった解軌道が周期軌道に収束したとき、それらが同じタイミングで振動するようになる場合がある。そのような点の集合をアイソクロンという^{3, 4)}。同じアイソクロン上の点は同じタイミングで振動する周期解軌道に収束することからもともと同じ位相をもつと考え、これを基にして吸引領域内の位相を定義する。このように定義すると、摂動によって周期軌道から外れたときの解軌道の位置と位相の変化量を対応付けることができる。

吸引領域内の位相やアイソクロンを求めることができれば、摂動を受けた直後の状態量から位相の変化が分かり、摂動に対する位相の変化を解析するときにも有用である。しかし、安定な周期解軌道をもつネットワークのダイナミクスは非線形であるため、一般にアイソクロンや位相を解析的に求めることは困難である。そのため、通常はそれらを数値的に求めることになる。本稿では、文献⁵⁾で提案している遺伝子ネットワークに

おけるアイソクロンの計算法を示す。遺伝子ネットワークのモデルとして、区分線形ネットワークモデルを用いる。この計算法は、区分線形ネットワークモデルであることを利用する簡単な計算法である。すなわち、モデルの非線形微分方程式を数値的に解くことなくアイソクロンを近似的に求めることができる簡単な計算法で、モデルのダイナミクスから解析的に求めたポアンカレ写像とポアンカレ断面から始まる解軌道が再びポアンカレ断面と交差するまでの時間間隔を表す写像を用いた繰り返し計算により、ポアンカレ断面上の十分な密度の点集合の点の位相を近似的に求める。そして、それらの点の位相をもとにして、モデルのダイナミクスから解析的に求めた写像を用いてアイソクロンを求める。提案法は、このような方法であるため、遺伝子ネットワークモデルの非線形微分方程式を数値的に解くことなく簡単にアイソクロンを求めることができる。

2 問題設定

本稿で対象とする遺伝子ネットワークのモデルは、次式の区分線形ネットワークである⁶⁾。

$$\begin{aligned} \dot{x}_i(t) &= -d_i x_i(t) + f_i(w_{i1}, w_{i2}, \dots, w_{im}, \\ & \quad y_1(t), y_2(t), \dots, y_n(t)), \end{aligned}$$

$$x_i(0) = x_{(0,i)} \quad (1)$$

$$y_i(t) = h(x_i(t)), \quad i = 1, 2, \dots, n \quad (2)$$

ここで、 h はしきい値関数で、次式で定義される。

$$h(x_i) = \begin{cases} 0 & \text{if } x_i < 0 \\ 1 & \text{if } x_i \geq 0 \end{cases} \quad (3)$$

以下ではこのモデルをベクトル形式で次式のように表す。

$$\dot{x}(t) = -Dx(t) + f(w, y(t)), \quad x(0) = x_0 \quad (4)$$

$$y(t) = H(x(t)) \quad (5)$$

ここで、 $x(t) = [x_1(t) \ x_2(t) \ \dots \ x_n(t)]^T$ であり、 $x_i(t)$ は i 番目の遺伝子の生成物の正規化された濃度を表す。 $x_0 = [x_{(0,1)} \ x_{(0,2)} \ \dots \ x_{(0,n)}]^T$ は初期状態である。 n は

遺伝子の数である。\$y\$ は \$y(t) = [y_1(t) \ y_2(t) \ \dots \ y_n(t)]^T\$ であり、各遺伝子の発現レベルを表すベクトルである。ここではこの \$y\$ を遺伝子ネットワークの発現パターンと呼ぶ。\$f = [f_1 \ f_2 \ \dots \ f_n]^T\$ は遺伝子間の相互作用を表す関数である。\$w\$ は \$w^T = [w_1^T \ w_2^T \ \dots \ w_n^T]\$ で、\$w_i = [w_{i1} \ w_{i2} \ \dots \ w_{im}]^T\$ は \$f_i\$ のパラメータを表すベクトル、\$m_i\$ は \$f_i\$ のパラメータの個数を表す。\$D\$ は対角行列で、\$D = \text{diag}\{d_1, d_2, \dots, d_n\}\$ であり、\$d_i > 0\$ は \$i\$ 番目の遺伝子の生成物の分解速度に関する正実数パラメータである。\$H(x)\$ は、\$H(x) = [h(x_1) \ h(x_2) \ \dots \ h(x_n)]^T\$ である。

ここで、式 (4), (5) の遺伝子ネットワークが発現パターン \$y\$ をとる \$x\$ の \$n\$ 次元実数空間の領域を \$\Omega_y\$ と定義する。すなわち、

$$\Omega_y = \{x \mid y = H(x)\}$$

と定義する。さらに、発現パターン \$y\$ に対して \$e(y)\$ を

$$e(y) = D^{-1}f(w, y)$$

と定義する。

本稿では、式 (4), (5) の遺伝子ネットワークが周期 \$T\$ の安定な周期解軌道 \$\Gamma\$ をもつとし、その吸引領域を \$B\$ で表す。また、周期解軌道 \$\Gamma\$ のときに現れる周期発現パターン遷移列を、\$p\$ を遷移回数として

$$y^{*(0)} \rightarrow y^{*(1)} \rightarrow \dots \rightarrow y^{*(p)}, y^{*(0)} = y^{*(p)} \quad (6)$$

とし、\$y^{*(r)}\$ と \$y^{*(r+1)}\$、\$r = 0, 1, \dots, p-1\$ で異なる要素は一つのみとする。さらに、

$$e(y^{*(r)}) \in \Omega_{y^{*(r+1)}}, r = 0, 1, \dots, p-1 \quad (7)$$

が成り立っているとする。

以下では、\$y^{*(r)}\$ と \$y^{*(r+1)}\$ で異なる要素の添字を \$i_r\$ と表記する。すなわち、\$y_{i_r}^{*(r)} \neq y_{i_r}^{*(r+1)}\$、\$y_i^{*(r)} = y_i^{*(r+1)}\$、\$i \neq i_r\$ である。また、\$\Omega_{y^{*(r-1)}}\$ と \$\Omega_{y^{*(r)}}\$ の境界を \$S_r\$、\$r = 1, 2, \dots, p\$ と定義する。ただし、周期発現パターンの場合 \$y^{*(0)} = y^{*(p)}\$ であるので、\$\Omega_{y^{*(p-1)}}\$ と \$\Omega_{y^{*(0)}}\$ の境界であることを強調するときには、\$\Omega_{y^{*(p-1)}}\$ と \$\Omega_{y^{*(p)}} (= \Omega_{y^{*(0)}})\$ の境界 \$S_p\$ を \$S_0\$ と表記する。

本稿では、周期解軌道 \$\Gamma\$ の \$S_0\$ との交点 \$\gamma_0\$ を基準として周期軌道上の位相を定義する。すなわち、\$\gamma_0\$ の位相を \$0\$ とし、周期軌道上の点 \$x\$ の位相 \$\theta\$ は、\$\gamma_0\$ からその点に到達するまでにかかる時間を \$\Delta t\$ として

$$\theta = \omega \Delta t$$

で定義する。ここで、\$\omega = \frac{2\pi}{T}\$ である。また、定義からわかるように位相 \$\theta\$ は \$0\$ から \$2\pi\$ の値をとる。

次に、アイソクロンとそれに基づく吸引領域内の点に対する位相を定義する。安定な周期解軌道 \$\Gamma\$ の吸引領域内の点を初期値として解軌道を考えて、最初は異なるが周期解軌道 \$\Gamma\$ に収束した時同じタイミングでその解軌道が振動するようになる初期点の集合が存在する。そのような点の集合をアイソクロンという³⁾。アイソクロン \$\mathcal{I}\$ の任意の点 \$x_0 \in \mathcal{I}\$ と周期軌道の位相 \$\theta\$ の点 \$\gamma_\theta\$ を初期値とする解軌道をそれぞれ \$x(t, x_0)\$、\$\gamma(t, \gamma_\theta)\$ とし、

$$\lim_{t \rightarrow \infty} \|x(t, x_0) - \gamma(t, \gamma_\theta)\| = 0$$

となったとする。ここで、\$\|x\|\$ は \$x \in R^n\$ の適当なノルムである。\$x(t, x_0)\$ は位相 \$\theta\$ の点から始まった周期解軌道 \$\gamma(t, \gamma_\theta)\$ と同じタイミングで振動する周期解軌道に収束するので、そのアイソクロン上の点から始まる解軌道は \$\gamma(t, \gamma_\theta)\$ と同じ位相をもともともっていると考えられる。そこで、そのアイソクロンの位相を \$\theta\$ とし、そのアイソクロンに属する点の位相は \$\theta\$ であると定義する。すなわち、以下のように定義する。【定義 (アイソクロン)】 \$x_0\$ から始まる解軌道を \$x(t, x_0)\$、周期軌道上の位相 \$\theta\$ の点 \$\gamma_\theta\$ から始まる解軌道を \$\gamma(t, \gamma_\theta)\$ とする。このとき、位相 \$\theta\$ のアイソクロン \$\mathcal{I}_\theta\$ は、

$$\mathcal{I}_\theta = \left\{ x_0 \in \mathcal{B} \mid \lim_{t \rightarrow 0} \|x(t, x_0) - \gamma(t, \gamma_\theta)\| = 0 \right\}$$

である。

【定義 (吸引領域内の点の位相)】 吸引領域内の点 \$x_0\$ から始まる解軌道を \$x(t, x_0)\$ とする。また、同じ時刻に周期軌道上の位相 \$\theta\$ から始まる解軌道を \$\gamma(t, \gamma_\theta)\$ とする。

$$\lim_{t \rightarrow \infty} \|x(t, x_0) - \gamma(t, \gamma_\theta)\| = 0$$

となるならば、\$x_0\$ の位相は \$\theta\$ である。

このようにアイソクロンや位相を定義すると、摂動により周期軌道から外れた解軌道が周期軌道に収束したとき、どのようなタイミングで振動するかが摂動を受けた後の解軌道の場所から直ちに分かるようになり、摂動による位相の変化の解析を行うときに有用である。そこで、本稿では式 (4), (5) の遺伝子ネットワークを対象にして、文献 5) のアイソクロンの計算法を示し、それがどのような点で簡単な計算法かを説明する。

3 アイソクロンの計算法

一般には、モデルの非線形微分方程式を解いてアイソクロンを求めるが、本稿で示すアイソクロンの計算法では、モデルの非線形微分方程式を解くことなく吸引領域内のポアンカレ断面上の点の位相を求め、それらの情報をもとにアイソクロンを求める。本節では、まず位相の計算法を示し、次に、ポアンカレ断面上の位相の計算結果を利用したアイソクロンの計算法を示す。

3.1 吸引領域内の位相の計算法

安定な周期解軌道の吸引領域内の点 \$x_0\$ の位相は、位相の定義に従って次のような考え方に基づいて近似的に求めることができる。\$x_0\$ と \$\gamma_0\$ から同時に始まった解軌道の時間発展をそれぞれ考える。\$x_0\$ から始まる解軌道 \$x(t, x_0)\$ が \$S_0\$ 上で周期解軌道 \$\Gamma\$ に十分近づいたときの時刻を求める。この時刻に \$x(t, x_0)\$ は周期解軌道上で位相 \$0\$ の場所に存在することになる。また、その時刻における \$\gamma_0\$ から始まった解軌道 \$\gamma(t, \gamma_0)\$ が存在する場所の位相は、その時刻と周期解軌道の周期より求めることができる。\$x(t, x_0)\$ は周期解軌道上における位相 \$0\$ のところに存在することから、\$\gamma(t, \gamma_0)\$ が存在する場所の位相は周期解軌道上における \$x_0\$ が存在する場所と \$\gamma_0\$ が存在する場所の位相差に対応する。よって、その位相差から直ちに \$x_0\$ の位相の近似値を得ることができる。

このような考え方で、式 (4), (5) の微分方程式を数値的に解くことなく位相を求める方法を示すために、\$S_0\$ をポアンカレ断面としたときのポアンカレ写像と解軌

道がポアンカレ断面と交差してから次に交差するまでの時間間隔を表す写像を示す。ポアンカレ写像は、 x_0 から始まる解軌道が S_0 と交差したときに周期解軌道 Γ に十分に近づいているかどうかを調べるために利用される。時間間隔の写像は、解軌道が周期解軌道に十分近づいたときの時刻を求めるときに利用される。

先にポアンカレ写像を示す。そのために、 $\Omega_{y^*(r)}$, $r = 0, 1, \dots, p-1$ 内の点 x に対してつぎの写像を導入する。

$$\begin{aligned} \psi_i^{(r+1)}(x) &= e_i(y^*(r)) - \left(e_i(y^*(r)) - x_i \right) \\ &\quad \times \left(\frac{e_{i_r}(y^*(r))}{e_{i_r}(y^*(r)) - x_{i_r}} \right)^{\frac{d}{d_r}}, \\ i &= 1, 2, \dots, n, \quad r = 0, 1, \dots, p-1 \end{aligned}$$

この写像は、 $\Omega_{y^*(r)}$ 内の点 x から始まる解軌道が境界 S_{r+1} と交差する点 $\psi^{(r+1)}(x)$, $r = 0, 1, \dots, p-1$ を与える⁸⁾。

さて、式 (7) の条件が成り立つとしているので、 $\Omega_{y^*(0)}$ 内の点 x_0 から出発する解軌道は、式 (6) の周期発現パターン遷移列に対応して境界 $S_1, S_2, \dots, S_{p-1}, S_p$ と順に交差する⁷⁾。このとき、 S_r と交差する点は、 $\psi^{(k)}$, $k = 1, 2, \dots, r$ を用いて x_0 に対して

$$\Psi^{(r)}(x_0) = \psi^{(r)} \circ \psi^{(r-1)} \circ \dots \circ \psi^{(1)}(x_0) \quad (8)$$

で定義される写像により与えられる。ここで、 $g \circ f(x) = g(f(x))$ である。 $\Psi^{(p)}(x_0)$ は S_0 上の x_0 から始まる解軌道が S_0 と交差する点を与える。よって、 S_0 をポアンカレ断面としたときのポアンカレ写像は、 $\Psi^{(p)}(x)$ となる⁸⁾。以下では、表記の簡単のため、このポアンカレ写像を $\Psi(x)$ と表す。

次に、解軌道がポアンカレ断面と交差してから次にポアンカレ断面と交差するまでにかかる時間を表す写像を求めるため、解軌道が領域 $\Omega_{y^*(r)}$ 内を通過するのにかかる時間を表す写像を示す。 S_0 上の点 x_0 から始まった解軌道が S_r と交差した時刻を $t = 0$ とする。このときの S_r との交点は $\Psi^{(r)}(x_0)$ である。時刻 $t = 0$ に $\Psi^{(r)}(x_0)$ で S_r と交差した解軌道は領域 $\Omega_{y^*(r)}$ 内を通過して $e(y^*(r)) \in \Omega_{y^*(r+1)}$ へ向かい、 S_{r+1} と $\Psi^{(r+1)}(x_0)$ で交差し、遺伝子ネットワークの発現パターンは $y^*(r)$ から $y^*(r+1)$ へと変化する⁷⁾。このときの、領域 $\Omega_{y^*(r)}$ 内の解軌道は、 $\Psi^{(r)}(x_0)$ を用いて

$$\begin{aligned} x_i(t) &= \Psi_i^{(r)}(x_0) \exp(-d_i t) \\ &\quad + e_i(y^*(r))(1 - \exp(-d_i t)), \\ i &= 1, 2, \dots, n \end{aligned} \quad (9)$$

と表される。この軌道が S_{r+1} と交差するとき $x_{i_r} = 0$ となることから、 S_r 上の点 $\Psi^{(r)}(x_0)$ から始まる解軌道が境界 S_{r+1} と交差するまでにかかる時間 $T^{(r+1)}(\Psi^{(r)}(x_0))$ は

$$\begin{aligned} T^{(r+1)}(\Psi^{(r)}(x_0)) \\ = -\frac{1}{d_{i_r}} \left\{ \log \frac{e_{i_r}(y^*(r))}{e_{i_r}(y^*(r)) - \Psi_{i_r}^{(r)}(x_0)} \right\} \end{aligned} \quad (10)$$

で得られる。以上のことは、任意の $r \in \{0, 1, \dots, p-1\}$ に対して成り立つので、解軌道がポアンカレ断面 S_0 と x_0 で交差してから、次にポアンカレ断面と交差するまでの時間間隔 $T_p(x_0)$ は

$$T_p(x_0) = \sum_{r=1}^p T^{(r)}(\Psi^{(r-1)}(x_0)) \quad (11)$$

となる⁹⁾。ただし、 $\Psi^{(0)}(x) = x$ である。

x_0 が S_0 上ではない場合、 x_0 から始まった解軌道がはじめて S_0 と交差するまでの経過時間とそのときの交点を求める必要がある。それらはつぎのようにして求めることができる。 x_0 が $\Omega_{y^*(v)}$ 内の点であるとする。二つの写像 $\Psi^{(r)}$ と $T^{(r)}$ を求めたときの議論は、 $\Omega_{y^*(r)}$ 内の任意の点に対して成り立つので、 $\Omega_{y^*(v)}$ 内の点 x_0 から始まる解軌道が S_0 と交差する点 \bar{x} と交差するまでにかかる時間 \bar{T} は $\psi^{(r)}$ と $T^{(r)}$ を用いて次式で表すことができる。

$$\bar{x} = \psi^{(p)} \circ \psi^{(p-1)} \circ \dots \circ \psi^{(v+1)}(x_0) \quad (12)$$

$$\bar{T} = \sum_{r=v+1}^p T^{(r)}(\bar{\Psi}^{(r-1)}(x_0)) \quad (13)$$

$$\begin{aligned} \bar{\Psi}^{(k)}(x_0) &= \psi^{(k)} \circ \psi^{(k-1)} \circ \dots \circ \psi^{(v+1)}(x_0), \\ k &= v+1, v+2, \dots, p-1 \end{aligned} \quad (14)$$

ただし、 $\bar{\Psi}^{(v)}(x) = x$ である。

以上で示した写像を使って吸引領域内の点 x_0 の位相を求めるための考え方を説明する。 $x_0 \in \Omega_{y^*(v)}$ なる x_0 から始まる解軌道 $x(t, x_0)$ が S_0 と初めて交差する時刻 \bar{T} と交点の座標 \bar{x} は、式 (13) と式 (12) よりそれぞれ求められる。 S_0 と交差してからつぎに S_0 と交差する時刻と交点の座標は、式 (11) を用いて $\bar{T} + T_p(\bar{x})$ 、ポアンカレ写像を用いて $\Psi(\bar{x})$ とそれぞれ求めることができる。さらに式 (11) とポアンカレ写像を繰り返し用いて、 S_0 上で周期軌道 Γ に十分近づいたときの時刻 t^k がわかる。また、周期軌道 Γ 上の位相 0 の点 γ_0 から $x(t, x_0)$ と同時に始まった解軌道 $\gamma(t, \gamma_0)$ が時刻 t^k に周期軌道上のどの位相の場所に存在するかは、周期が分かっているので簡単に求めることができ、その場所の位相が $x(t, x_0)$ と $\gamma(t, \gamma_0)$ の位相差であることから、 x_0 の位相はその位相差から簡単に求めることができる。

このような考え方から次のアルゴリズムが得られる。 ϵ_T と ϵ_x は適切に設定する。

【位相を求めるアルゴリズム】

step 1 吸引領域内の点 $x_0 \in \Omega_{y^*(v)}$ から始まる解軌道が S_0 平面と交差したときの時刻 $t^{(0)}$ と交点の座標 $x^{(0)}$ を式 (12), (13) を用いて求める。 $k = 0$ とする。

step 2 $x^{(k)}$ を通過した解軌道が、次に S_0 と交差する時刻 $t^{(k+1)}$ と交点 $x^{(k+1)}$ を式 (11) と式 (8) を用いて

$$t^{(k+1)} = t^{(k)} + T_p(x^{(k)}),$$

$$x^{(k+1)} = \Psi(x^{(k)})$$

と求める。また、 $\bar{T}^{(k+1)} = T_p(x^{(k)})$ とする。

step 3 $k = k + 1$ とする。 $|T - \bar{T}^{(k)}| > \epsilon_T$ あるいは $\|\gamma_0 - x^{(k)}\| > \epsilon_x$ ならば step 2 へ、そうでなければ step 4 へ。

step 4 $t^{(k)}$ を T で割ったときの余り \hat{T} を求め、 $\Delta t = T - \hat{T}$ とする。

step 5 x_0 の位相の近似値 θ は $\theta = \omega \Delta t$ である。

解析的に求めたポアンカレ写像などの繰り返し計算により位相を求めており、簡単な計算で位相が得られる。

3.2 アイソクロンの計算

ここではアイソクロンを求める方法を説明する。提案法では、ポアンカレ断面上の位相を求め、ポアンカレ断面上の点から指定した位相差の場所にある点を解析的に求めることでアイソクロンを求める。この方法の説明のために、指定した位相差の点を解析的に求める方法を示す。

位相 θ_A の点 x_A から、 x_A を通る解軌道上の位相 θ_B の点 x_B を求めるとし、 $x_A \in \Omega_{y^*(v)}$ であるとする。同じ解軌道上の二つの点の位相差は、片方の点からもう一方の点へ解軌道が到達するまでにかかる時間に対応する。 $\theta_B > \theta_A$ の場合、 x_A と x_B の位相差から、その時間は

$$\Delta t = \frac{\theta_B - \theta_A}{\omega} \quad (15)$$

と求められる。 $\theta_B \leq \theta_A$ の場合は、

$$\Delta t = \frac{2\pi - (\theta_A - \theta_B)}{\omega} \quad (16)$$

と求められる。 x_A から Δt だけ進んだ場所を求めるべき x_B である。領域 $\Omega_{y^*(v)}$ 内の x_A から始まる解軌道の式は、式 (9) において $\Psi^{(r)}(x_0)$ を x_A に、 $e(y^{*(r)})$ を $e(y^{*(v)})$ にそれぞれ置き換えることで

$$\begin{aligned} x_i(t) &= x_A \exp(-d_i t) \\ &\quad + e_i(y^{*(v)})(1 - \exp(-d_i t)), \\ i &= 1, 2, \dots, n \end{aligned} \quad (17)$$

と得られる。その軌道が S_{v+k} と交差する時刻 $t^{(k)}$ と交差する点の座標は、

$$t^{(k)} = \sum_{j=v+1}^{v+k} T^{(j)}(\bar{\Psi}^{(j-1)}(x_A)) \quad (18)$$

と $\bar{\Psi}^{(k+1)}(x_A)$ で与えられる。ただし、 $j \geq p$ なら $j = j - p$ として計算する。このとき、 $t^{(k)} < \Delta t \leq t^{(k+1)}$ となる k を求めると、 x_A から Δt だけ進んだ解軌道は吸引領域内の領域 $\Omega_{y^*(r)}$ に存在することがわかる。ただし、 $r = v + k$ とし、 $r \geq p$ なら $r = r - p$ とする。そこで、次式で得られる $\hat{T}^{(k+1)}$ を求め、 $\hat{T}^{(k+1)} < 0$ となる最小の k を求める。

$$\begin{aligned} \hat{T}^{(k+1)} &= \Delta t - t^{(k+1)} \\ &= \Delta t - (t^{(k)} + T^{(v+k+1)}(\bar{\Psi}^{(v+k)}(x_A))) \\ &= \hat{T}^{(k)} - T^{(v+k+1)}(\bar{\Psi}^{(v+k)}(x_A)) \end{aligned} \quad (19)$$

S_{v+k} と交差するときの交点は $\bar{\Psi}^{(v+k)}(x_A)$ 、その交点から x_B に到達するまでの経過時間は $\hat{T}^{(k)}$ として求められているので、 x_B の座標を

$$\begin{aligned} x_{(B,i)} &= \bar{\Psi}^{(v+k)}(x_A) \exp(-d_i \hat{T}^{(k)}) \\ &\quad + e_i(y^{*(v+k)})(1 - \exp(-d_i \hat{T}^{(k)})), \\ i &= 1, 2, \dots, n \end{aligned} \quad (20)$$

と求めることができる。以上の考え方より x_B を求めるアルゴリズムはつぎのようになる。

【指定した位相差の座標を求めるアルゴリズム】

step 1 $\theta_B > \theta_A$ のときは式 (15)、そうでないときは式 (16) により Δt を求める。

step 2 $\hat{T}^{(0)} = \Delta t$, $x^{(0)} = x_A$, $k = 0$ とする。

step 3 $\hat{T}^{(k+1)} = \hat{T}^{(k)} - T^{(r+1)}(x^{(k)})$ と $x^{(k+1)} = \psi^{(r+1)}(x^{(k)})$ を求める。

step 4 $\hat{T}^{(k+1)} < 0$ ならば step 6 へ。

step 5 $r = r + 1$ とする。 $r = p$ なら $r = 0$ とする。

step 6 $k = k + 1$ として step 2 へ。

step 7 $x_0 = x^{(k)}$, $\hat{T} = \hat{T}^{(k)}$ として x_B を求める。

$$\begin{aligned} x_{(B,i)} &= x_{(0,i)} \exp(-d_i \hat{T}) \\ &\quad + e_i(y^{*(r)})(1 - \exp(-d_i \hat{T})), \\ i &= 1, 2, \dots, n \end{aligned} \quad (21)$$

また、解軌道上で位相 θ_B の場所を探すときに、時刻を遡って探すことも同様の方法で可能である。この計算においても先の【位相を求めるアルゴリズム】で利用したポアンカレ写像などの繰り返し計算のみで簡単に所望の座標を求めることができる。

ポアンカレ断面上の十分な密度の点集合に対して位相を求め、このアルゴリズムによりそれらの点を通る解軌道上の同位相の点の集合を求めることでアイソクロンを求めることができる。位相 θ_B のアイソクロンを求めるときは、次のようにすればよい。

【アイソクロンを求めるアルゴリズム】

step 1 ポアンカレ断面上の十分な密度の点に対して、【位相を求めるアルゴリズム】によりそれぞれの位相を求める。

step 2 位相の値を求めたポアンカレ断面上の点に対して、【指定した位相差の場所を求めるアルゴリズム】によりその点を通る軌道上を時間について順方向に位相 θ_B の場所を求める。

step 3 位相の値を求めたポアンカレ断面上の点に対して、その点を通る軌道上を時間について逆方向に位相 θ_B の場所を求める。

これにより、ポアンカレ断面上の位相を求めた点に対応して位相 θ_B のアイソクロンに属する点の集合が求められる。このアルゴリズムで必要な計算は、先に述べたように解析的に求められたポアンカレ写像などの計算の繰り返しであるため、簡単にアイソクロンを求めることができる。

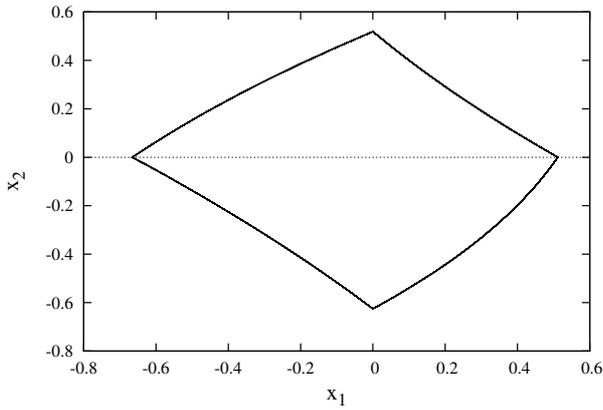


Fig. 1: Periodic Orbit

4 数値例

ここでは、本稿で示した計算法によりアイソクロンを求めた結果を示す。相互作用関数は、

$$f_i(a^{(i)}, y) = a_0^{(i)} + \sum_{j=1}^n a_j^{(i)} y_j + \sum_{j=1}^{n-1} \sum_{k=j+1}^n a_{jk}^{(i)} y_j y_k + \cdots + a_{12 \dots n}^{(i)} y_1 \cdots y_n, \quad i = 1, 2, \dots, n$$

がよく用いられる。ここでもこれを用いることにする。

遺伝子の数は 2 個で、次の遺伝子ネットワークのアイソクロンを求めた結果を示す。

$$\dot{x}_1(t) = -x_1(t) + (3 - y_1(t) - 7y_2(t) + 3y_1(t)y_2(t)) \quad (22)$$

$$\dot{x}_2(t) = -3x_2(t) + (-4 + 5y_1(t) + 2y_2(t)) \quad (23)$$

この遺伝子ネットワークは、Fig. 1 に示す安定な周期解軌道をもつ。周期解軌道の周期は 1.0314 で、このときの発現パターンは、

$$[1 \ 1]^T \rightarrow [0 \ 1]^T \rightarrow [0 \ 0]^T \rightarrow [1 \ 0]^T \rightarrow [1 \ 1]^T \quad (24)$$

と変化する。

まず、本稿で示した方法により、ポアンカレ断面 S_0 上の位相を求める。ポアンカレ断面 S_0 は、 $S_0 = \{x | x_1 \geq 0, x_2 = 0\}$ であり、周期解軌道は S_0 と $[0.5982 \ 0.0]^T$ で交差する。ポアンカレ断面上で、 x_1 の値を 0 から 5 まで 0.0001 刻みで変化させ、50000 点の位相を求めた。その結果の一部を Table 1 に示す。これらのポアンカレ断面上の点を通る解軌道上において位相が $0.0, 0.2\pi, 0.4\pi, 0.6\pi, 0.8\pi, 1.0\pi, 1.2\pi, 1.4\pi, 1.6\pi, 1.8\pi$ の点を求めることで、それぞれの位相に対応するアイソクロンを求めた。その結果を Fig. 2 に示す。Fig. 2 において、 $[0.5982 \ 0.0]^T$ を通る赤い曲線が位相 0 のアイソクロンである。この位相 0 のアイソクロンから反時計回りに $0.2\pi, 0.4\pi, 0.6\pi, 0.8\pi, 1.0\pi, 1.2\pi, 1.4\pi, 1.6\pi, 1.8\pi$ のアイソクロンが得られている。

Fig. 2 において、アイソクロンが得られていない領域がある。これは、位相を求めた S_0 上の点を通る解軌道において、 S_0 と交差するときの前後で 1 カ所ずつのみ位相 θ_B となる点を求めているためである。

Table 1: Phases of points on Poincaré map

x_1	0.25	0.5	0.75	1.0
位相	2.682	0.560	5.564	4.635
x_1	1.25	1.5	1.75	2.0
位相	3.887	3.266	2.719	2.229
x_1	2.25	2.5	2.75	3.0
位相	1.796	1.394	1.017	0.672
x_1	3.25	3.5	3.75	4.0
位相	0.349	0.0425	6.035	5.759
x_1	4.25	4.5	4.75	5.0
位相	5.495	5.244	5.01	4.779

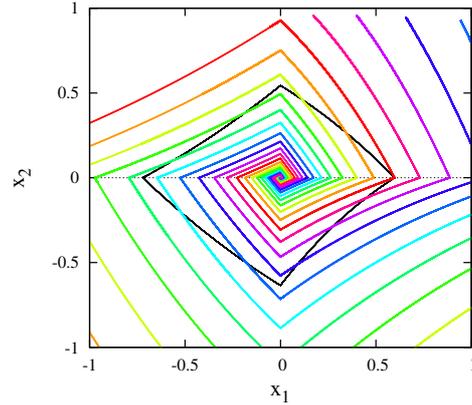


Fig. 2: Isochrons of phases of $0.0, 0.2\pi, 0.4\pi, 0.6\pi, 0.8\pi, 1.0\pi, 1.2\pi, 1.4\pi, 1.6\pi, 1.8\pi$

5 おわりに

生体には概日リズムなど様々な安定な周期現象が存在し、それらは遺伝子ネットワークによって生成されていると考えられている。また、それらは環境などからの影響により位相が調整されることも多い。その仕組みを調べるために、摂動により位相がどのように変化するかに興味を持たれ調べられている。このとき、安定な周期軌道の吸引領域内の位相やアイソクロンを求めることができれば、解析のときに有用である。しかし、位相やアイソクロンを解析的に求めることは困難で数値的に求める。そこで、本稿では区分線形ネットワークモデルを対象にして文献 5) で提案されている遺伝子ネットワークのアイソクロンの計算法を示し、それが簡単な計算法であることを示した。一般には遺伝子ネットワークのモデルの非線形微分方程式を数値的に解いて位相を求めるが、提案法では非線形微分方程式を解くことなく、区分線形ネットワークモデルであることを利用し、解析的に求めた写像を用いた計算によりアイソクロンを簡単に求めることができる。

参考文献

- 1) 蔵本 由紀:リズム現象の世界, 東京大学出版会 (2005)
- 2) 郡 宏, 森田善久:生物リズムと力学系, 共立出版 (2011)
- 3) A. T. Winfree: Patterns of phase compromise in biological cycles, *Journal of Mathematical Biology*, **1**, 73-95 (1974)
- 4) K. Josic, E. T. Shea-Brown, and J. Moehlis: Isochron, *Scholarpedia*, **1**-8, 1361 (2006)
- 5) 森 禎弘, 黒江康明: 遺伝子ネットワークにおけるアイソクロンの計算法, 第 2-3 回インテリジェント・システ

ム・インポジウム FAN2013 講演論文集, 63-68 (2013)

- 6) L. Glass: Classification of biological networks by their qualitative dynamics, *Journal of Theoretical Biology*, **54-1**, 85-107 (1975)
- 7) 森 禎弘, 黒江 康明, 森 武宏: 発現パターンに基づく遺伝子ネットワークの設計法, *計測自動制御学会論文集*, **44-11**, 936-945 (2008)
- 8) . Mori and . Kuroe: Synthesis method of gene regulatory networks having cyclic expression pattern sequences –realization by introducing Poincarè map–, *Proc. of SICE Annual Conference 2011*, 544-549 (2011)
- 9) . Mori and . Kuroe: Synthesis method of gene regulatory networks having desired periodic expression pattern sequences, *Proc. of 2012 International Conference on Systems, Man, and Cybernetics*, 1159-1164 (2012)