

ベイズ理論を用いた迷路におけるロボット行動の学習

菊池 直樹*

千葉大学大学院自然科学研究科
知能情報工学専攻

須鎗 弘樹†

千葉大学工学部情報画像工学科

Abstract: 8個のセンサに反応して左右2つのモータが独立に動くロボットにケーペラロボットがある. このケーペラロボットが迷路上で壁に触れることなく壁に沿って動くときには各センサパターンに対して適した動き方があると考えられる. 本研究では, ベイジアンネットワークを用いて, 各センサパターンに対するモータの正しい動かし方の学習をシミュレーション上で行った. 確率の更新にはベイズ更新を用いてコンピュータ上で学習する.

1 はじめに

ある状況における行動をロボットに教えるのは簡単である. なぜならば最も適していると思われる行動をとるプログラムを組み込んでおけばよいからである. しかし, そこで注意しなければならないことは「最も適した行動」というところに人間的な主観が入ることや, 全てのパターンにおける動作を決定する作業があまりにも困難であるという問題がある.

迷路においてセンサの入力に対する適したモータの動かし方をロボット自身で学習させるシステム構築は, これまではGAを用いて行うことが多かった[1]. ここではベイジアンネットワークを利用することにより構築する. そこでは, それまでの経験からどのような動作をしたらよいかということを経験から学習していくことが可能である. 初期は壁にぶつかってばかりのロボットが, 最終的には目的位置に到達することができるようになる.

2 ベイジアンネットワーク

ケーペラロボットは8個のセンサ入力, 2個のモータ出力となっている. したがって入力における最適な出力を導出するには, 入力 $I = \{i_1, i_2, \dots, i_8\}$, 出力 $O = \{O_{left}, O_{right}\}$ とした時に, 成功したか否かの結果 $R = \{success, failure\}$ を用いて,

$$P(R = success | I, O)$$

を求めればよい. そして, その値が最も大きくなる出力 O を選び出せば成功する可能性が最も高くなる. そのために,

*〒263-8522 千葉市稲毛区弥生町 1-33, 1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan, tel: 043-290-3509, e-mail: kikuchi@icsd3.tj.chiba-u.ac.jp

†住所・telは同上, e-mail: suyari@tj.chiba-u.ac.jp

それらの条件付き確率表 (CPT:Conditional Probability Table) を更新していくことが必要である.

3 ベイズ更新

条件付き確率 $P(I, O | R)$ は, 以下のように書き直すことができる.

$$\begin{aligned} P(I, O | R) &= \frac{P(I, O, R)}{P(R)} \\ &= \frac{P(R | I, O)P(I, O)}{\sum_{I, O} P(R | I, O)P(I, O)} \end{aligned}$$

よって $P(I, O | R)$ は $P(R | I, O)$ と $P(I, O)$ の二つの確率値で表すことができる. ここで $P(R | I, O)$ はある状況における行動の成功確率となる. これはロボットが実体験を通して得ていくものである. また $P(I, O)$ は

$$P(I, O) = P(O | I)P(I)$$

と書くことができる. しかし事前確率として各入力パターンの起こり得る確率 $P(I)$ は持ち合わせていないし, それは現実的でない. したがって全ての $P(I)$ は同じ値とみなすことができる (理由不十分の原則). よって $P(I, O | R)$ は

$$P(I, O | R) = \frac{P(R | I, O)P(O | I)}{\sum_{I, O} P(R | I, O)P(O | I)}$$

と書き直すことができる. $P(O | I)$ は入力パターンを得た時に, どのように動きたいかの指標として

$$\sum_I P(O | I) = 1$$

となるようにしている. すなわち, ある特定の入力パターンに対してのある動作の成功確率が大きくなれば, 今後その動作をより多くとるようになることを示している.

これらより得られる確率 $P(I, O | R)$ は、「成功した」という事実をもとにその動作の正しさを考え直すことができる。つまり、 $P(I, O | R)$ を次の時点での $P(O | I)$ とすることで、次に同じ状況に直面した時の行動を選択する場合の気持ちが更新されるのである。このように前段階の事後分布を現在の事前分布の役割としてみなす更新をベイズ更新という。これにより経験を逐次算入していく学習を行うことができる [2].

4 シミュレーション

まずロボットに迷路を与え、初期位置と目的位置を決めておく。事前情報は何も持たない。またロボットには目的位置や進むべき方向を与えない。その代わりにマップ全体をエリア分けし、正しい方向のエリアに移動できた場合にのみ、それが正しいことを伝える。完全な教師無し学習を行うことは困難なので、最低限の指標は与えることとする。そして壁にぶつかるまたは規定の時間以内に次のエリアに進むことができなければ失敗とし、それ以外は成功とする。壁にぶつかったならば初期位置に戻し、成功や失敗が決定する毎にベイズ更新を行いながら正しい動き方を学習させる。

シミュレーションでの設定を以下に示す。ケーペラは図1のようになっている。センサはそれぞれ直線上に15ピクセル分の距離以内にある壁を感知することができる。左右に付けられたモータは前後に2段階ずつ動かすことができ、また止めることもできるので計5種類の動きをすることができる。よって2つのモータの組み合わせにより全部で25通りの動きとなる。例として、左のモータが止まり、右のモータが前回転(弱め)をした場合の動き方を図2に示す。

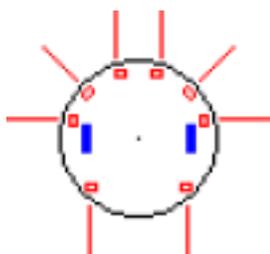


図1: ケーペラにおけるセンサとモータの位置

使用した迷路は全体サイズが204ピクセル×204ピクセルで、初期位置と目的位置は共に行き止まりとなっている。図3にその実験環境を示す。

以上の適用により、シミュレーション上のケーペラは目的位置までたどり着くことができるようになった。しかし、初期位置に関する頑健性の習得を成すことはでき

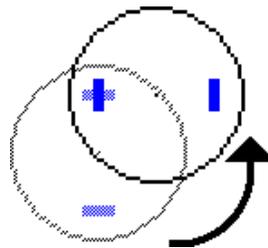


図2: 左モータ0, 右モータ+1の場合の動き
(初めは右方向を向いている)

ず、学習後に初期位置を少しずらしただけでも壁にぶつかったりして、目的位置にたどり着くことはできなかった。



図3: 実行中の様子

5 まとめ

本研究ではベイジアンネットワークとベイズ更新を取り入れることによって学習を行った。その結果、目的位置まで達することができるようにはなれたが、頑健性を習得することはできなかった。これではマップが変更されたときに再度の学習が必要となる。モジュール化やクラス分けを行うことにより頑健性の習得も可能であると思われるので、それらを今後の課題としたい。

参考文献

- [1] 星野 力, 光本 大輔, 長野 徹: ロボット行動の進化とその頑健性, 計測自動制御学会論文集 Vol.33, No.6, 533/540 (1997).
- [2] 松原 望: ベイズ的意思決定, 改訂版 統計の考え方 286/321, 放送大学教育振興会 (2000).