平成１０年度知能システム科学専攻修士論文

ベイジアンネットワーク上の強化学習の
ケペラロボットシミュレータへの応用

小野塚 卓

An application of Reinforcement Learning on a Bayesian Network to the Khepera robot simulator

Takashi Onozuka

提出年月日 平成１１年２月１６日

主査教官 小林 重信 教授
審査教官 伊藤 宏司 教授
審査教官 山田 誠二 助教授

ベイジアンネットワーク上の強化学習のケペラロボットシミュレータへの応用

小野塚 卓

An application of Reinforcement Learning on a Bayesian Network to the Khepera robot simulator

Takashi Onozuka

Reinforcement Learning is a framework of learning in which a learner will learn a policy based on uncertain and delayed information. In most previous Reinforcement Learning researches, a learner has no prior knowledge on the surrounding environment. This caused difficulty to implement knowledge in Reinforcement Learning. To overcome this difficulty, Yamamura introduced Reinforcement Learning on a Bayesian Network as a way to use prior knowledge. Bayesian Network is a model of stochastic induction in which stochastic knowledge is expressed as a directed graph with conditional probability tables. This novel method made it possible to use Bayesian Network, to which is implemented uncertain knowledge as well as physical constraints, on the framework of Reinforcement Learning. In this paper, we overcome some problems brought about while implementing the theory by proposing a general method to update probability values using gradient method. The experimental results of its application to the Khepera robot simulator was examined.

1 はじめに

強化学習とは，学習者（エージェント）が環境とのインターラクションを通して，遅れや不確実性を含んだ弱い情報源を頼りに学習する枠組みである．強化学習では，エージェントは環境へ行動の信号を送り，環境から感覚入力と報酬信号のみを受け取る．よって，エージェントは対象に関する明示的なモデルや教師付き学習にあったような正解についての知識などは必要としない．ゆえに，強化学習は，厳密な正解は分からないが出力を評価することが可能な問題領域で有望な枠組みである．

一方で，物理的な制約などの断片的な知識さえも強化学習の枠組みに入れずに，エージェントに試行錯誤をさせて学習させることが前提とされてきた．同時に，実際に強化学習を応用するにあたって，エージェントが学習する環境についての知識をエージェントにあらかじめ組み込む方法が模索されてきた．

そこで，強化学習と知識処理との融合の一つの形としてベイジアンネットワーク上の強化学習が山村によって提案された [山村 97]．知識を強化学習に取り込む一つの手段としてベイジアンネットワークを用い，そのベイジアンネットワークを確率的な政策として用いることが提案された．

しかし，理論を実際に応用するにあたりいくつかの乗り越えなければならない問題が生じた．その一つが，それぞれの行動を行う確率を変更するにあたり，変更後の値が確率の定義である０以上１以下の範囲を越えてしまうという問題である．この問題は，一般的な微分を足すような傾斜法を確率空間で用いる時の本質的な問題と考えられる．もう一つは，実際にどのような知識をどのようにベイジアンネットワークに埋め込んでロボット制御に用いるかという課題である．

本論文では，ベイジアンネットワーク上の強化学習の応用における前述の２点の課題を克服する手法を提案し，その上で本手法の挙動を観察する．また，センサ情報が不安定かつ乏しく知識なしでは学習が大変困難な状況として，ケペラロボットシミュレータに応用し，実験結果を通して本手法の有効性を確認する．

以下，第２章では，本研究の背景をベイジアンネットワーク上の強化学習とその基にある確率的傾斜法とベイジアンネットワークを含めて概説する．第３章では，本理論を実装するにあたっての問題点を克服するための提案手法を説明する．第４章では，予備実験としてグリッドワールドにおける実験を行い，本手法の有効性を確認する．第５章では，より現実的な問題としてケペラロボットシミュレータに応用し，結果を考察する．第７章では，本論文の成果をまとめ，今後の課題について述べる．

2 ベイジアンネットワーク上の強化学習

2.1 強化学習

強化学習とは，学習者／学習システムがそれを取り巻く外界に対して何らかの行動をし，外界からセンサ情報

と報酬を受け取ることを繰り返しながら学習する枠組のことである．強化学習の特徴として，外界から与えられる情報に遅れと不確実性を含むということが挙げられる．ここで，情報の遅れとは学習者がいくつかの行動をした後にそれら全体に対して報酬が与えられたりと，必ずしも現在行った行動に対しての報酬をもとに学習するとは限らないということである．また，不確実性とは，行った行動それぞれについて教師付き学習のように正しい答え，すなわち，するべきであった行動が教えられるのではなく，報酬という数値が学習者に与えられるだけで，学習者はその報酬を基に自分のするべき行動を学習する必要があるということである．

2.2 確率的傾斜法

代表的な強化学習の手法の Q-learning[Watkins and Dayan 92] においては，エージェントが環境の状態を完全に観測することができるという前提で，環境をマルコフ決定過程（MDP）でモデル化していた．しかし実際には，実世界のセンサを用いることで，同じ状態としてエージェントに認識されるべきであるのに違う状態と認識されてしまうという不完全知覚（perceptual aliasing）の問題が生じる．また，エージェントが環境から情報を受け取るセンサの能力が不十分なために環境を完全に認識できなかったり，ノイズの多い情報しか得られない場合には隠れ状態（hidden state）問題が生じる．

確率的傾斜法は，不完全知覚問題や隠れ状態問題をも含めた部分観測マルコフ決定過程（Partially Observable Markov Decision Processes:POMDP）を環境のモデルとして扱い，そのモデル上で平均報酬を最も大きくする方向へ確率的政策を改善していく手法である．

確率的傾斜法による強化学習アルゴリズムの一般形を図 1 に示す．このアルゴリズムにおいて前提となっているのは，すべての観測 X と行動 A において政策 $\pi(a, W, X)$ が内部変数 W のすべての要素に関して偏微分可能ということである．この前提をもとに，適正度 (eligibility[Williams 92]) と呼ばれる，実行した行動に関する情報論的な価値，情報量を計算する．例えば，選ばれる確率の低い行動が行われた時の情報量は大きく，確率の高い行動が行われた時の情報量は小さい．この適正度は過去の行動ほど割引率 γ で減衰させながら，適正度の履歴 (eligibility trace[Singh et al. 94])D として保持されており，この適正度の履歴 D と報酬 r をもとに内部変数 W を更新するため，報酬に遅れのあるような行動も強化される．

1. 観測 $X(t)$ を受け取る
2. $\pi(a(t), W(t), X(t))$ の確率で行動 $a(t)$ を実行する．
3. 環境から報酬 $r(t)$ を受け取る
4. 内部変数 W のすべての要素について
   以下の $e_i(t)$ と $D_i(t)$ を求める
   $e_i(t) = \frac{\delta}{\delta w_i} ln\pi(a(t), W(t), x(t))$
   $\bar{D}_i(t) = e_i(t) + \gamma \bar{D}_i(t-1)$
   ただし γ は割引率 $(0 \leq \gamma < 1)$
5. 以下の式を用いて $\Delta w_i(t)$ を求める
   $\Delta w_i(t) = (r(t) - b)\bar{D}_i(t)$
   ただし b は報酬基底とよばれる定数
6. 政策の改善：以下の式で更新
   $\Delta W(t) = (\Delta w_1(t), \Delta w_2(t), \cdots, \Delta w_i(t), \cdots)$
   $W \leftarrow W + \alpha(1-\gamma)\Delta W(t)$
   (ただし α は非負の学習定数)
7. 時間ステップ $t \leftarrow t+1$ として 1 へ戻る

図 1: 確率的傾斜法のアルゴリズム

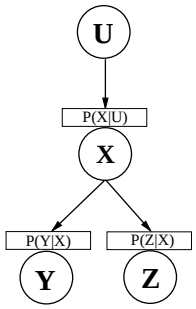

図 2: ベイジアンネットワークの例

2.3 ベイジアンネットワーク

ベイジアンネットワークとは，確率的な知識を有向グラフと条件付き確率で表し，不確実な情報を伝播することで確率的に推論を行うモデルのことで，複数のノードとそれを結ぶアークから構成される非循環の有向グラフである [Pearl 88].

図 2 にベイジアンネットワークの一例を示す．ノードが確率変数に，リンクが確率変数間の依存関係に対応する．アークの根本にあるノードをアークの先にあるノードの親ノード，また，アークの先にあるノードをアークの根本にあるノードの子ノードと呼ぶ．確率変数 U が与えられたとき確率変数 X がどのような分布になっているかは，条件付き確率 $P(X|U)$ を用いて計算できるので，各子ノードにはその親ノードとの関係を表す条件付き確率表 (Conditional Probability Table:CPT) が存在する．

ベイジアンネットワーク上で計算されるのは, 確率分布

	作り込み	BN 学習	LookupTable 学習	NN 学習
既存知識の利用	○	○	△	△
人間の理解	△	○	△	△
汎化能力	-	△	×	○
データの事前処理	-	○	×	×

図 3: 既存の方法との比較 : ○は可能，△は条件つきで可能，×は不可能，-は比較対象外を示す.

が既知である証拠ノードの集合 $\{e_1, \cdots, e_n\}$ が与えられたときの各ノード X の確率分布 $BEL(x) = (P(+x|e_1 \wedge \cdots \wedge e_n), P(-x|e_1 \wedge \cdots \wedge \cdots e_n))$ である．この計算は，証拠ノードからのデータの伝搬により行われる．ここで，$+x$ はノード X の親ノード側から伝搬して得られた確率分布，$-x$ はノード X の子ノード側から伝搬して得られた確率分布である．あるノードの確率分布を求めるには，アークの向きにかかわらず親ノード側と子ノード側から伝搬されてきた情報が使われること注意されたい.

2.4 ベイジアンネットワークの学習

ベイジアンネットワークは確率的現象のモデルとして，症状から病名を推測したりといったような，状況からある事柄を推測するのに用いられる．そのため，ベイジアンネットワークの学習というのは，既に収集されたデータからそのデータが示す現象の正しいモデルを同定するという意味合いで使われてきた．ベイジアンネットワークの学習はネットワークの構造が既知かどうか、また、対象とする領域の完全なデータが得られるかどうかによって分類される [Binder et al. 97] [Heckerman 95]．強化学習においても，[Cassandra et al. 96] においてベイジアンネットワークを環境のモデルとして学習器に持たせ，正しい環境のモデルをオンラインで同定させている.

既存の方法とベイジアンネットワークの学習についての比較を図 3 に示す．既存の方法として，人間が作り込んだもの，ルックアップテーブルの学習，ニューラルネットワークの学習を選んだ.

まず，既存知識の利用に関して比較すると，作り込みとベイジアンネットワークを用いたものでは，既存の知識をそのまま埋め込むことができる．ルックアップテーブルを用いたものは，既存の知識をルックアップテーブルの形として一旦整理する必要があり，そのまま埋め込むことができないという点において条件つき可能とした．ニューラルネットワークの学習においては，[Andrews and Geva 95] や [Thrun 93] のようにあらかじめ知識を入れる方法もあるが埋め込める知識に制限がある.

次に，人間の理解について比較すると，作り込んだものは作り込んだ設計者本人にはどのように動くのかが明確に理解できるが，設計者以外の者によってその挙動を理解することは容易ではない．ベイジアンネットワークの学習では，ベイジアンネットワーク自体が確率変数間の因果関係を表しているのでそのままの形で人間が理解することができる．ルックアップテーブルの学習では，学習後や学習前のルックアップテーブルを理解するためには，既存知識を用いるときに一旦整理する必要があったように，整理する必要があり人間には容易に理解ができない．また，ニューラルネットワークの学習では，[Andrews and Geva 95] や [Thrun 93] のように理解することは不可能ではないが，全てのニューラルネットワークに関して理解可能というわけではないし，ベイジアンネットワークほど明確ではない.

汎化能力に関しては，ニューラルネットワークの学習は未学習データにも対応できるというような汎化能力が見られる．ベイジアンネットワークの学習でも今までに学習したことを条件つき確率表として持っているので未学習データに関しても既存データに対しての類似性において対応できる．一方，ルックアップテーブルを用いたものでは，それぞれの異なる場合を別々に学習しているので汎化能力はない.

最後にデータの事前処理が必要かに関して比較すると，ニューラルネットワークやルックアップテーブルの学習では，一部のデータが欠損しているようなデータ列をあらかじめ事前処理しておく必要がある．一方，ベイジアンネットワークの学習においては，欠損データに関しての事前処理が必要なくそのまま学習に用いることができる.

2.5 ベイジアンネットワーク上の強化学習

[山村 97] で提案されたベイジアンネットワーク上の強化学習では、ベイジアンネットワークを環境の正しいモデルではなく，環境における正しい行動のモデル，すなわち，エージェントの政策として用いている．よって，環境を同定してからその環境上での正しい行動のモデルを学習するというような 2 度手間を取らずに，報酬のみを頼りに正しい行動のモデルを同定することを提案している．また，本手法は先に述べたベイジアンネットワークの学習の分類では，ネットワークの構造が既知で完全なデータが入手できないような困難さの学習に分類される．本手法は Polytree と呼ばれる一つのノードが複数個の親ノードと子ノードを持つようなベイジアンネットワークの構造までを対象とする.

前述した確率的傾斜法において，ベイジアンネットワークは近似関数 $\pi(a, X, W)$ として用いられ，内部変数 W が全ての条件付き確率表に対応する．よって，近似関数 $\pi(a, X, W)$ としてのベイジアンネットワークがすべての条件付き確率表の各要素によって偏微分可能であれば，確率的傾斜法上でベイジアンネットワークを学習させることも可能である.

まず，ベイジアンネットワーク上の強化学習において

はネットワーク上に必ず一つの行動ノードを持つ必要がある．そして，図 1 の 1 で観測した情報 X を確率分布の形に変換し，対応するセンサノードに与えることで，そのノードを証拠ノードとする．この証拠ノードのからの信念の伝搬によって，行動ノードの確率分布が得られる．これが，図 1 の 2 の $\pi(a, W, X)$ に対応する．詳しい信念の伝搬方法については付録 A を参照されたい．

次に，図 1 の 4 に対応する適正度の計算をする必要がある．ベイジアンネットワーク上の強化学習においては，行動ノード Y の信念 $BEL(y)$ のもとで，学習器が行動 $Y = y_d$ を選んだとすると，全ての条件付き確率表の各値に関するその行動の選択確率 $\ln BEL(y_d)$ の傾斜が適正度となる．この傾斜を行動ノードから信念の伝搬の逆向きに伝搬しながら各 CPT の適正度を計算していく．詳しい適正度の伝搬方法に関しては付録 A を参照されたい．

最後に，図 1 の 6 に対応する政策の改善において，確率値の更新を単に変化量と更新前の値の和とすると，確率値が満たすべき 0 以上 1 以下という制約を満たさないという問題が生じる．この問題点を克服するための提案手法は次章で述べる．

以上，確率的傾斜法の近似関数としてベイジアンネットワークを用いる方法について概説した．

3 手法の提案

実際に確率的傾斜法をベイジアンネットワーク上で実装するにあたり，確率値の更新方法と知識の利用法についての提案を以下に説明する．

3.1 確率空間における傾斜法の適用方法の提案

図 1 の 6 において内部変数の更新時に更新前の W の値と変化量を基に更新後の W の値を求めている．ここで内部変数の変わりにベイジアンネットワークの条件つき確率表を更新する場合，確率的傾斜法の一般形の様に更新前の値と変化量の和で変化後の値を決めると，更新後の値が 0 以上 1 以下という確率の範囲に収まらない．これは，確率空間において微分を足すだけの傾斜法を用いる上で避けては通れない根本的な問題である．

これは [Binder et al. 97] においては求められた変化量ベクトルを制約面に投影して制約を満たすような変化量ベクトルを求め，各ベクトルの方向に微小量だけ全ての条件つき確率の値を変更していくアプローチがとられている．しかし，本手法においては適正度に報酬等をかけたものが実際の変化量として用いられるので，毎回微小量づつ変化させるような方法は適していない．

そこで，更新前の値と変化量を基に関数を用いて更新値を求めることで確率空間における制約を満たすような方法を提案する．以下にこの関数が，更新前の値 p，変化量 $delta$，更新後の値 $f(delta, p)$ の 3 値について少なくとも満たしていなければならない必要条件を示す．

1. 変化量 $delta_1 \geq delta_2$ の時，$f(delta_1, p) > f(delta_2, p)$.
   もし，ある変化量 $delta_1$ が別の変化量 $delta_2$ より大きければ，更新後の値も変化量の大きいほう $f(delta_1, p)$ が変化量の小さいもの $f(delta_2, p)$ より大きくなる必要がある．
2. 更新前の値 $p_1 \geq p_2$ の時，$f(delta, p_1) \geq f(delta, p_2)$.
   2 つの異なる変更前の値 p_1 と p_2 が $p_1 > p_2$ の関係にあるとき，同じ変化量 $delta$ だけ変化させられたなら，$f(delta, p_1) \geq f(delta, p_2)$ 更新後の値の大小関係も維持される必要がある．
3. $\lim_{delta \to \infty} f(delta, p) = 1$
   変化量が正の無限大のとき，変化後の値は確率の最大値 1 になるべきである．
4. $\lim_{delta \to -\infty} f(delta, p) = 0$
   変化量が負の無限大のとき，変化後の値は確率の最小値 0 になるべきである．
5. $f(0, p) = p$
   もし変化量がゼロであれば，更新後の値は更新前の値に等しい値であるべきである．

以上の様な条件を満たす関数の一つとして，本論文では以下のような関数を用いることを提案する．

$$f(delta, p) = \frac{1}{\sqrt{2}} \left(t + \frac{\sqrt{2}}{\pi} \left(\frac{2}{1 + exp(-\frac{delta}{\lambda})} - 1 \right) \sin \frac{\pi}{\sqrt{2}} t \right), \tag{1}$$

ここで t は以下の式を満たす値である．

$$p = \frac{1}{\sqrt{2}} \left(t - \frac{\sqrt{2}}{\pi} \left(\frac{2}{1 + exp(-\frac{delta}{\lambda})} - 1 \right) \sin \frac{\pi}{\sqrt{2}} t \right) \tag{2}$$

ここで λ は傾斜率と呼ばれるパラメータで次に示す可視化した図の変化量方向の傾きに対応する．大きいと傾斜が緩やかになり，小さいと傾斜が急になる．この関数は，変更前の値を x 軸，変化量を y 軸，変化後の値を z 軸として可視化すると図 4 のようになる．このグラフの特徴は，上記の条件 5 を満たすので変化量 0 の所の断面が (変化後の値) = (変化前の値) となっていることである．

3.2 知識の利用方法の提案

ベイジアンネットワーク上の強化学習の利点の大きな一つは断片的な知識を与えられることである．ここで，断片的な知識とは環境に関する完全な知識の断片ということである．環境に関して全て分かっているわけでもなく，

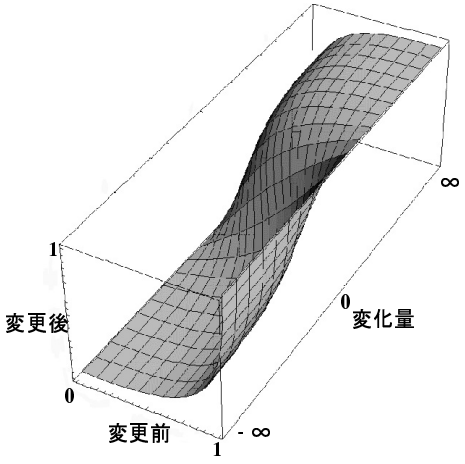

図 4: 更新関数の可視化

同時に環境に関して無知でもないというような実際に強化学習アルゴリズムを応用する時にしばしば起こる設定をそのまま用いるのに断片的な知識が使用可能なことは便利である．

また，現実の応用においては，精度の低いセンサー，すなわち信頼性に欠ける情報しか環境から得られないという厳しい制限のもとで，何とか学習させなければならない．信頼性の低い情報を頼りに学習するには，効率良く知識を用いセンサーの弱点を補う必要がある．

以上のような必要性を満たすために，実際にどのような断片的な知識をどのように埋め込むかという指針を明確にしておく必要がある．

本研究では，ロボットナビゲーション問題において，大まかにわけて３種類の断片的な知識を用いることを提案する．

１つ目は，エージェントにあらかじめ与えておく地図といった外界に関する基本的な知識である．ここで，基本的というのは他の知識がこの知識に依存する，もしくはこの知識を前提として与えられるという意味においてである．例えば，環境の不完全な地図などがこれにあたる．

２つ目は，１つ目に与えた知識に依存して入れることのできる知識である．例えば，地図上で各位置においてどのような観測が得られるかといった知識やスタート地点が地図上でどの位置に属しているかといったものである．

３つ目の知識は，壁の方向へは進めないとか地図上で離れた位置にある状態には一回の行動では移動できないといった物理的な制約からの知識である．

以上のような知識のベイジアンネットワークへの埋め込み方として以下のような方法を提案する．

1. **ベイジアンネットワークの構造として埋め込む**
   外界に関する基本的知識はこの構造として埋め込む必要がある．例えば，本研究で用いるエージェントに持たせる地図は，他の知識を埋め込むための基本的な知識であると同時に，他の知識に依存している知識である．このような知識はベイジアンネットワークの構造として埋め込む必要がある．
2. **条件付き確率表の初期値として埋め込む**
   例えば，地図上の各位置においてどのような観測が得られるかといったようなことはあらかじめ知ることができる場合のある情報なので，対応する条件付き確率表の初期値としてベイジアンネットワークに埋め込むことができる．
3. **条件付き確率表 (CPT) の値に０や１を使う**
   物理的な制約に由来する知識の場合は，あらかじめ不可能か可能かということがはっきりと分かっているので，確率の値として０や１を用いることができる．例えば，地図上のある位置から別の位置へ一行動で移動できないことは０や１の確率値を用いて明確に埋め込むことができる．また，０や１という確率の値は，学習プロセスにおいて変更されることが無いので，このような物理的な制約など不変的な知識を埋め込むのに適している．

4　グリッドワールドにおける実験

4.1　問題設定

ベイジアンネットワーク上の強化学習を応用するにあたり，エージェントのセンサから得られる情報が少ないために隠れ状態問題が生じるような環境において，知識の効果を確認することを試みる．そのために，強化学習をするベイジアンネットワークを制御器に用いたロボットのナビゲーション問題を扱う．本研究におけるロボットナビゲーション問題とは，ロボットがスタート地点からゴール地点まで到達するための道のりを学習する問題とする．ケペラシミュレータ上においてロボットナビゲーションの実験をする前にまずグリッドワールドにおいて実験を行い，本手法の有効性と挙動を確認する．

4.2　実験方法

環境として図 5 のような，アルファベットのエイチの形をしたものを用いる．図中の点線で囲まれた小さなマスを環境の各状態とし，エージェントが S で記されたスタート地点から G で記されたゴール地点までたどり着く政策を学習することを目的とする．

エージェントは，S で記された 4 つのマスからランダムに選ばれた状態から出発し，ゴール地点に到着すると 100 の報酬が与えられ，またスタートの 4 地点からランダムに選ばれた状態から試行を繰り返す．また，壁のある方向へ進もうとすると-10 の報酬すなわち罰を受ける．

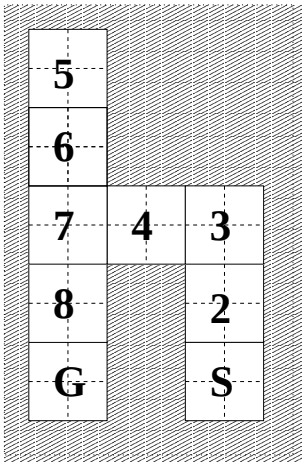

図 5: 環境の地図 (点線) とそれより精度の粗いエージェントが持っている地図 (実線)

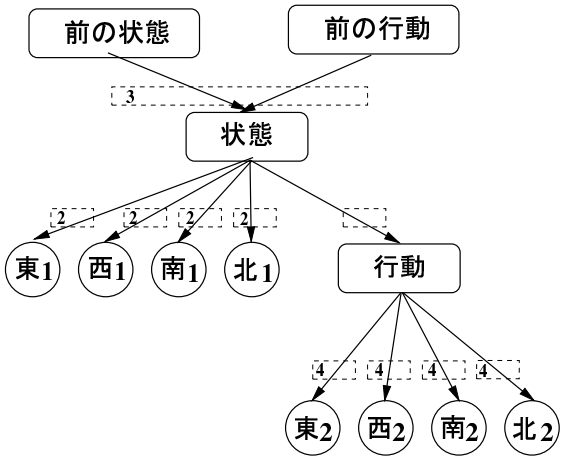

図 6: エージェントのベイジアンネットワーク：図中の東西南北は各方向にあるセンサの値に対応する．また，点線の四角は CPT に対応し，四角内の番号は付与した知識に対応する．

エージェントは東西南北方向それぞれに２マス先に壁があるかどうかを識別できるセンサーが付いているものとし，東西南北方向に１マス動くという４つの行動を選択できるものとする．

4.2.1　エージェントに与えた知識

エージェントには知識を入れるために図 6 にあるようなベイジアンネットワークを用いる．東西南北ノードはそちらの方向のノードに壁があるかどうかの２値をとり，前の状態と状態ノードは環境の地図上の数字に対応する８値をとる．前の行動と行動のノードでは，東西南北４方向への行動に対応する４値を取る．

このベイジアンネットワークでは，まず，自分の持っている地図上で自分で居ると信じていた前の状態と実際に選んだ前回の行動，そして東西南北方向のセンサからの値をもとに現在の地図上の位置を推測する．次に，現在の地図上の状態と東西南北センサの値を再度使い，実際に自分が取るべき行動を推測する．

エージェントにはベイジアンネットワークの構造として埋め込んだ知識の他に以下のような知識が最初から与えられているものとする．別の言い方をすれば，以下のような設計者が既に知っている断片的な知識をエージェントに入れるためにこのようなベイジアンネットワークの構造が用いられている．具体的な実際に入れた条件つき確率表の初期値は付録 B に示す．

1. 環境のおおまかな地図：図 5 の太い線で区切られたような実際の環境よりも粗い精度の地図で，必ずしも一行動で地図上のとなりの位置に移動できるとは限らない．
2. 地図上の番号付けられた各位置においてどのようなセンサ情報が得られるかという知識．
3. １行動では，地図上で隣接している状態か今居た状態のどちらかにしか動けないという物理的制約．
4. 壁の方向には進めないという制約．

4.2.2　センサ情報のベイジアンネットへの受渡し方

エージェントのベイジアンネットワークは行動毎に親ノードの無いノードには事前確率を，子ノードの無いノードには尤度を設定する必要がある．そのため，外界からのセンサ情報はベイジアンネットワークに載せる前に以下のように確率の表現に変換する．

東西南北のセンサノード　ここで用いるセンサは，２マス先に壁があったらオンになり，無ければオフになるという理想的なセンサを用いている．センサがオンの時には，オンに対応する確率が１，オフに対応する確率が０となるような確率ベクトルを用いる．逆の場合は，この逆となる．

前の状態ノード　前回の状態ノードの確率ベクトルをそのまま用いる．

前の行動ノード　前回選択した行動を確率１とし，それ以外の行動の確率を０とした確率ベクトルを用いる．

4.2.3　ベイジアンネットワークからの行動決定

エージェントの行動選択にはベイジアンネットワーク上の行動ノードの確率分布ベクトルをそのまま用いた．そのため選ばれる確率の小さい行動もその確率に比例した確率で選択される．

このロボットナビゲーション問題がどれほど困難な問題かを知るための指標として，ランダムに行動を選択す

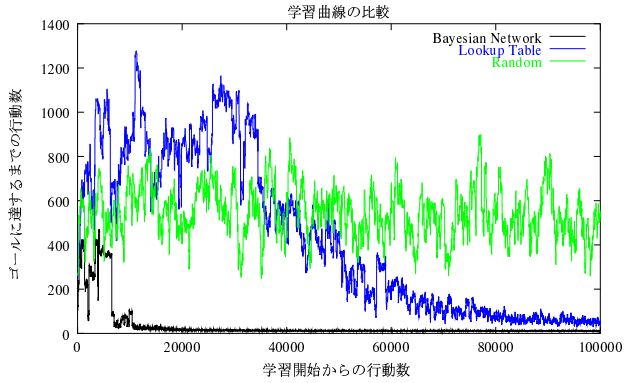

図 7: ベイジアンネットワークを用いたものとルックアップテーブルを用いたものとランダムに行動を選択するエージェントの学習曲線の比較：それぞれ１０実験の平均値

るエージェントと各センサの値から行動を選択するルックアップテーブルを持ったエージェントの学習も行った．それぞれエージェントが 10 万回行動を選択するまで学習をさせた．本研究では，一試行をスタートからゴールにたどりつくまでと定義する．この実験のパラメータには，割引率 $\gamma = 0.97$，ベース $b = 10$，学習率 $\alpha = 0.05$，傾斜率 $\lambda = 40$ を用いた．

4.3 結果

実験結果を図 7 に示す．このグラフにおいて x 軸は学習開始からの行動数，y 軸はゴールそれぞれの試行において要したスタート地点からゴール地点到着までの行動数とした．同じ試行に属する学習開始からの行動数に対しては，同じゴールまでに要した行動数がプロットされることに注意されたい．これは，ベイジアンネットワークを用いたもの，ルックアップテーブルを用いたもの，ランダムに行動を選択するものの 3 つのデータ系列が同じタイムスケールで比較されるようにしたためである．すなわち，x 軸は学習開始からの時間に対応する．

4.4 考察

この問題の難しさを知るために用いたルックアップテーブルを使ったものとランダムに行動を選択したものを比較すると，ルックアップテーブルを用いたものは学習過程でランダムに行動選択をするエージェントよりゴールに到達するまでの行動数が大きくなることが約 4 万行動目まであることが観察できる．よって，知識を持っていないルックアップテーブルを用いたものはランダムよりも性能が悪化することが学習途中において発生する．

一方，ベイジアンネットワークを用いて知識を入れたものは，ランダムよりゴールに達するまでの行動数が大きくなることはなく，約 1 万行動選択目以降は急速に性能を回復している．また，ルックアップテーブルを用いたものとの性能を比較すると，常に少ない行動数でゴールまで達しているのが観察される．

これは，知識を入れたものはその分だけ政策の探索空間が狭められ同時に，その探索空間において性能の良いと思われるところから学習を始めているためだと考えられる．

知識を入れた効用として以下の 2 点が考えられる．

1. **学習までに要する時間が短縮される**．学習開始からの行動数がルックアップテーブルを用いたものでは学習曲線が漸近状態に達するまでに約１０万ステップ要しているのに対して，ベイジアンネットワークを用いたものでは約 1 万ステップで学習曲線が漸近状態に達している．
   本論文では，知識を入れていないルックアップテーブルを問題の難しさの指標として用いているが，同じだけ知識を付与したルックアップテーブルを用いたものに関しても考察すると，これはベイジアンネットワークを用いたものよりも学習すべき値の数が大きくなる．これは，ベイジアンネットワークにおいては確率変数の独立性から条件つき確率表ののサイズを小さくできるのに対して，ルックアップテーブルにおいては独立性を埋め込むことができないからである．一般に，学習時に変更しなければいけない値の数が多ければ多い程学習に要する時間がかかるので，例え同じだけ知識を付与したルックアップテーブルを用いたとしても，学習に要する時間はベイジアンネットワークを用いたものの方が短くなる．また，同じ知識をルックアップテーブルに埋め込むことはベイジアンネットワークほど知識を入れるのが容易ではない．
2. **学習曲線の漸近状態におけるゴールに到達するまでに要するステップ数を下げることができる**．これは，ルックアップテーブルを用いたものでは，図 5 において同じセンサ情報しか得られない状態，状態２，８，６が同じ状態としてしか認識できないのに対して，ベイジアンネットワークを用いたものでは知識を用いてそれらの状態を区別し，隠れ状態問題を克服しているからである．左にベイジアンネットワークの行動ノードと状態ノードの間の条件付き確率表を図 8 に示す．右にルックアップテーブルを地図上の状態に対応する用に並び変えたものを．ルックアップテーブルにおいては，状態２，６，８を区別することができないので，これらの全ての状態において南に進む行動を選択する確率が 0.65 と北に進む行動を選択する確率 0.15 より大きくなっている．このため，ルックアップテーブルをもちいたエージェントはスタート地点から状態 2 を通って状態 3 へ移動するのに，必要以

状態	ベイジアンネット				ルックアップテーブル			
	北	西	南	東	北	西	南	東
1	0.00	0.26	0.74	0.00	0.80	0.07	0.00	0.13
2	0.12	0.84	0.00	0.03	0.15	0.00	0.65	0.20
3	0.00	0.99	0.00	0.00	0.09	0.57	0.35	0.00
4	0.00	0.40	0.60	0.00	0.11	0.68	0.05	0.16
5	0.00	0.26	0.74	0.00	0.13	0.04	0.70	0.13
6	0.00	0.21	0.79	0.00	0.15	0.00	0.65	0.20
7	0.00	0.15	0.84	0.00	0.15	0.00	0.65	0.20
8	0.00	0.18	0.81	0.00	0.15	0.00	0.65	0.20

図 8: 学習後の CPT(条件付き確率表) の比較 : 左がベイジアンネットワークを用いたもの，右がルックアップテーブルを用いたものをベイジアンネットワークの対応するセンサ入力に合わせて整理したもの．

上の試行錯誤が行われる必要がある．そのため，ベイジアンネットワークを用いたものの学習時のゴールに到達するまでに要するステップ数のほうが少なくてすむ．

以上，グリッドワールドにおいてベイジアンネットワーク上の強化学習の有効性を確認した．

5 ケペラロボットシミュレータにおける学習

本章では，グリッドワールドより現実的なケペラロボットシミュレータの環境において本手法の有効性を確認することを目的とする．ケペラロボットシミュレータにおいては，センサにノイズが入るために同じ状態にいても違う状態と認識してしまう不完全知覚問題が生じる．ケペラロボットには精度の低い近接センサと光センサがそれぞれ 8 つずつ付けられているのみである．少ない数の信頼性に欠けるセンサを通してではロボットは環境から少ない情報量しか得られない．また，グリッドワールドにおいては既知であった自分の現在向いている方向も環境からは明示的に与えられない．

ベイジアンネットワーク上の強化学習に期待される利点として知識の活用と，ケペラシミュレータの以上のような厳しい制約が非常にマッチしていると考えられる．このようなグリッドワールドに比べてより学習困難な環境において本手法を応用する．

5.1 ケペラロボットシミュレータ

ケペラロボットは，研究用に開発された小型ロボットで，駆動系として DC モーターを左右に一つずつ，センサとして 8 方向の赤外線近接センサと光センサを図 9 の番号がつけられている位置に持つが，近接センサは性能が低く極めて近距離の情報しか得られない．ケペラロボットシミュレータは，Olivier Michel によって開発されたケペラロボットが動き回れる仮想的な 2 次元の 1 メートル四方の環境を提供する [Olivier 96]．このシミュレータを用いて障害物や光源を配置してロボットの動き回る環境を作り以下の実験を行う．

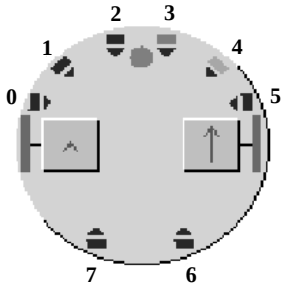

図 9: ケペラロボットのセンサ

シミュレータ上のケペラロボットは以下のようなノイズをセンサに与えることでより現実の精度の低いセンサの不確定性を埋め込まれている．ここでこのノイズは必ずしも実際のケペラロボット上で受け取るものとは同じとは限らないが，このようにすることで，同じ状態にいても異なる観測情報を受け取るという不完全知覚問題の生じる非マルコフ性の環境を実現している．まず，モーターにはロボットが送った回転数の± 10%のランダムノイズが，また，2 個のモーターの速度の違いから生じるロボットの回転方向には± 5%のランダムノイズが入る．赤外線近接センサは，各センサ前方の三角系上の１５点上の障害物の有無を調べ，0(障害物なし) から 1023(障害物あり) の整数値を返す．このとき，値の± 10%のランダムノイズが加えられる．光センサの値は，センサと光源の角度と距離のみから計算され 48(明るい) から 525(暗い) の整数を返す．このとき，その値の± 5 %のランダムノイズが付け加えられる．

5.2 実験

5.2.1 実験方法

グリッドワールドで用いた環境に似ている図 10 に示すようなアルファベットのエイチの形をした環境を用いる．この環境において，スタート地点は図中に S で記した部分で，ゴール地点は図中に G と記した部分とし，ロボットがスタート地点からゴール地点に到着するための政策を学習することを目的とする．ロボットは S で記された付近の 50 × 50 の正方形からランダムに選ばれた地点に北向きに置かれた状態から出発し，ゴール地点に到着すると 500 の報酬が与えられ，また S の付近でランダムに選ばれたスタート地点から試行を繰り返す．また，ロボットはいずれかの近接センサの値が 700 を越えると壁にぶ

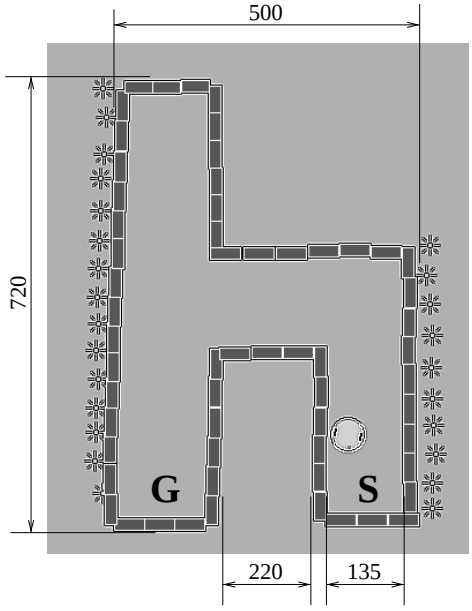

図 10: ケペラロボットシミュレータの環境:東西にある※の記号が光源を，エイチの中にある円形のものが環境に比例した大きさのロボットである．

つかったとして-50 の負の報酬を受けとる．

ここで，ロボットは 8 つの近接センサと 8 つの光センサを図 9 の各番号の位置に持ち，それぞれのノイズモデルは前節で説明した通りである．ケペラロボットの近接センサは遠くまで見通すことができないので，ロボットは各方向に壁があるかないかで地図上の位置を推測することは不可能である．

さらに，本実験では，ロボットの選択できる行動は直進，右に 30 度回転，左に 30 度回転の 3 つの行動のみとする．

よって，ロボットはまず方向が分からない事には．地図を与えたとしてもその地図上での位置を推測することはできない．グリッドワールドのように自分が進みたい方角に進むためには，そちらの方向を向いて直進をする必要がありその分グリッドワールドより困難なクラスの問題である．

そこで，地図上の東側では光が東側からさしてきて，地図上の西側では西側から光がさしてくるという設定にした．このようにすることで，ロボットが光の来る方向と自分の地図上で居る位置から自分の向いている方角を推測し行動できるようにする．

ロボットに与えた知識　エージェントには推論のために図 12 にあるようなベイジアンネットワークを用いる．各光センサノードは各光センサが光を感じたかどうかの 2 値をとり，各近接センサノードは各近接センサが障害物を検知したかどうかの 2 値をとる．光の有無のノードは少なくとも一つの光センサが光を感じていればオンになり，そうでなければオフになる 2 値をとる．状態ノードと前の状態ノードはケペラロボットが持っている地図上の数字に対応する 4 値をとる．前の行動は，前回の行動が直進であればオン，それ以外であればオフの 2 値をとる．

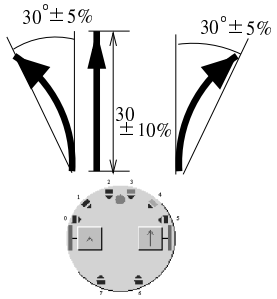

図 11: ケペラロボットの行動

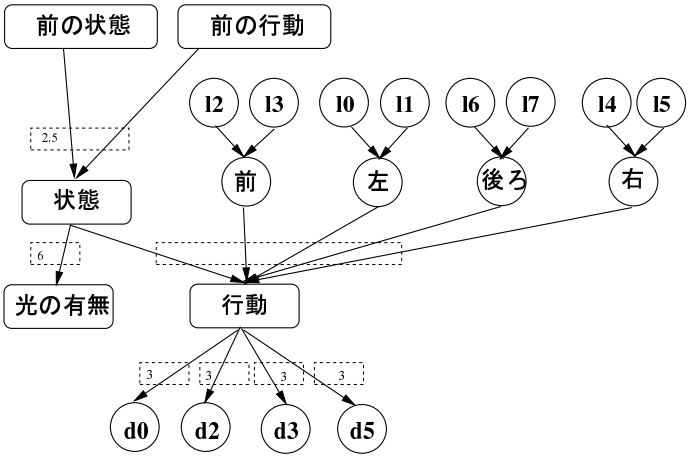

図 12: ケペラロボットのベイジアンネットワーク : l で始まるノードは各光センサを，d で始まるノードは各近接センサを意味する．また，点線の四角は CPT に対応し，四角内の番号は付与した知識に対応する．

このベイジアンネットワークでは，まず，自分の持っている地図上で自分で居ると信じていた前の状態と実際に選んだ前回の行動，そして各光センサからの値を元に現在の地図上の位置を推測する．同時に各光センサの値からロボットの前後左右に光があるかを推測し，前後左右からの情報と現在居る地図上の状態，そして，近接センサの値からふさわしい行動を推測する．

エージェントにはベイジアンネットワークの構造として埋め込んだ知識の他に以下のような知識が最初から与えられているものとする．別の言い方をすれば，以下のような設計者が既に知っている断片的な知識をエージェントに入れるためにこのようなベイジアンネットワークの構造が用いられている．実際に用いた，具体的な条件つき確率表の初期値は付録 B に示す．

1. 環境のおおまかな地図 : 図 13 に示すような地図で図 10 の実際にケペラロボットが動き回る環境よりも粗

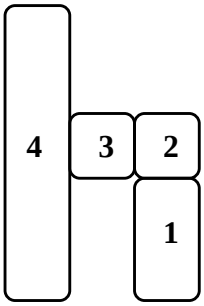

図 13: ケペラロボットの知識として持っている地図

い精度で，その上に全く同じ形状でなく，必ずしも１行動で地図上のとなりの位置に移動できるとは限らない．

2. １行動では，地図上で隣接している状態か今居た状態のどちらかにしか動けないという物理的制約．
3. 壁の方向には進めないという制約．具体的には，前方に付けられた近接センサが障害物を検知したときには，直進の行動をとれないということ．壁が左右どちらかのすぐそばにあるときには壁がある方向には回らないということ．
4. スタート地点が地図上でどこにあるのかということ
5. 直進の行動を取るときのみ自分の状態が変化し，右折と左折しているときには状態は変化しないということ．
6. 光は地図上の状態１，２においては東から，状態４においては西から差していて，状態３においては見えないということ．

センサ情報のベイジアンネットへの受渡し方　グリッドワールドで行ったのと同じようにロボットのベイジアンネットワークで証拠ノードとなりうるノードに観測データを確率の形で受け渡す必要がある．以下に各ノードにおいて観測情報を証拠ノードに受け渡す方法を示す．

前の状態　前回の状態ノードの確率ベクトルをそのまま事前確率ベクトルとして用いる．

前の行動　前回選択した行動が直進であればオンに対応する確率が１に，それ以外の時には０になるような事前確率ベクトルを用いる．

光の有無　全ての光センサの値が 300 以下であったら光が無い状態に対応する確率を１，それ以外の場合は光がある状態に対応する確率を１にした尤度ベクトルを用いる．

各光センサ　実際に観測される値を x とすると x の値域は $(0 \leq x \leq 1023)$ で，この値を以下の式を用いてセンサがオンの確率を求め，これを事前確率ベクトルとして用いる．

$$P(sensor = on) = \begin{cases} 1 & x > 300 \\ \frac{x}{300} & \text{それ以外} \end{cases}$$

各赤外線近接センサ　実際に観測される値を x とすると x の値域は $(48 \leq x \leq 525)$ で，この値を以下の式を用いてセンサがオフの確率を求め，これを尤度ベクトルとして用いる．

$$P(sensor = off) = \frac{1}{477}(x - 48)$$

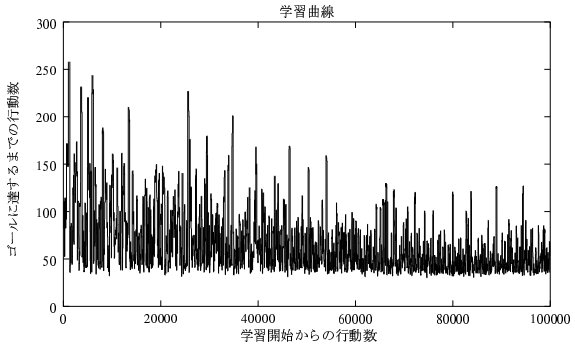

図 14: 学習曲線：１０実験の平均のプロット

5.2.2　ベイジアンネットワークからの行動決定

ロボットの行動選択にはグリッドワールドにおいての実験と同様に，ベイジアンネットワーク上の行動ノードの確率分布ベクトルをそのまま用いた．そのため選ばれる確率の小さい行動もその確率に比例した確率で選択される．

この実験のパラメータには，割引率 $\gamma = 0.97$，ベース $b = 10$，学習率 $\alpha = 0.05$，傾斜率 $\lambda = 40$ を用いた．

5.2.3　結果

10 実験の結果の平均値を図 14 に示す．x 軸は学習開始からの行動数，y 軸は各試行においてスタート地点からゴールに到達するまでの行動数とした．

また，図 15 に１０万行動学習した後の１０試行をプロットした．

5.2.4　考察

学習初期において約 250 もあった最悪時の行動数は 10 万回行動を行った後では約 100-150 とかなり性能が向上しているのが分かる．

また，グリッドワールドにおける実験に比べてゴールに到達するまでの行動数の変化が激しい．確率的傾斜法

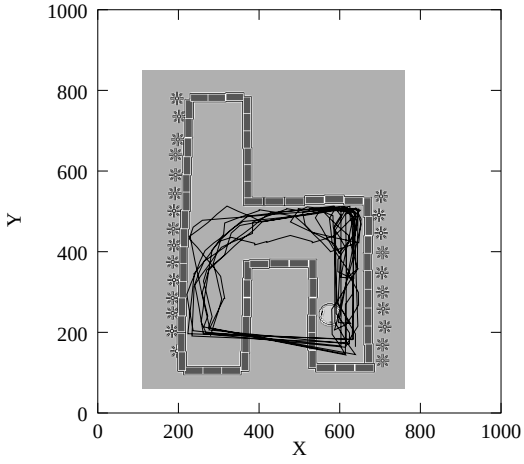

図 15: 10 万行動後の１０試行のプロット:ゴールからスタート地点への線は環境によって移動させられたものである．

の特徴として確率的に政策を選択するので，スタートからゴールに到着するまでの行動数が多い程，低い確率の行動が選ばれることも多くなる．本実験では，最短でもスタート地点からゴール地点に到達するまでに 50 行動を選択する必要があるので，その分だけ確率の低い行動が選択される確率が大きくなっている．そのために，図 14 のようなグリッドワールドにおいての実験に比べて安定しないグラフとなると考えられる．

この環境では，スタート地点から内側の壁沿いに進めばゴールに到達できるが，図 15 の学習後のゴールまでの道のりから分かるように，ロボットは壁沿いに移動する行動は獲得していない．これは，壁にあたると負の報酬が与えられるためそのような政策が強化されないためである．

ロボットは現在自分の持っている地図上でどこにいるのかを推測しているが，センサとアクチュエータのノイズがどれほどこの推測に対して影響を与えているかを考察する．各センサに対応する確率変数が 2 値を取るため，10%程のノイズはそれほど状態推測に対して影響が出ていない．しかし，ロボットが地図上の各状態の境の辺りにいる場合には，このノイズの影響によりどちらの状態にいるかが分からなくなることがあり得る．また，このベイジアンネットワークにおいては前回居た状態と選択した行動と現在の光の有無をもとに次の状態を推測しているので，一旦状態が分からなくなると光の有無がはっきりと分からない限り現在の状態を推測できないという累積誤差が生じる．

この実験において，もしルックアップテーブルを用いた場合を考えてみる．８つの光センサの値をオンかオフの２値をとるとして，４つの近接センサも同じように２値をとるとした場合，$2^{12}(=4096)$ の各状態について３つの行動を選ぶので，ルックアップテーブルの要素数が 12288 にもなる．これだけの値を学習を通して変更しながら短い行動数でゴールに到達できるような政策を獲得するにはかなりの時間を要すると考えられる．一方，ベイジアンネットワークを用いた本研究では，行動ノードの CPT は $2^4 \times 4(=64)$ の状態に対して３つの行動を選ぶので要素数 192 の大きさになる．一般的に変更する変数の数が大きい程学習に時間がかかるので，ベイジアンネットワークを用いたものの方が短時間で学習できることが分かる．

この実験においては，右折と左折と直進しか行動を選択できないようなエージェントに方向という知識を入れることで今までルックアップテーブルなどで学習させることがとても困難な問題において本手法の有効性を確認した．

6　おわりに

本論文では，[山村 97] で提案されたベイジアンネットワーク上の強化学習を実際に応用する際に生じた問題点を克服する方法として，確率空間で微分を足すような傾斜法を用いる方法を提案し，またベイジアンネットワーク上の強化学習で利用できる知識の種類とその利用方法を提案した．そして，ケペラシミュレータに応用し，実験を通して本手法の有効性と挙動を分析した．

今後の課題として，現在の学習で変化しうるのはベイジアンネットワーク中の条件付き確率表のみであるが，[Heckerman 95] で研究されているように，各ノードの確率変数のとりうる値の数を動的に変化させる枠組を作り上げることで，より柔軟に不確実な与えられた知識を変化させることが挙げられる．

公表論文

小野塚卓，山村雅幸 :“ベイジアンネットワーク上の強化学習のロボットナビゲーションへの応用”，人工知能学会全国大会 (第 11 回) 論文集，pp.421-424(1997).

Masayuki Yamamura, Takashi Onozuka: ” Reinforcement Learning with Knowledge by using a Stochastic Gradient Method on a Bayesian Network” In Proceedings of the 1998 IEEE International Joint Conference on Neural Networks, (1998).

謝辞

本研究を行うにあたり終始多大なる御指導および御教示を頂きました山村雅幸助教授に深く感謝の意を表します．また，休学中にお世話になったエディンバラ大学人

工知能学科の Rina Dechter 博士と John Holland 博士に深く感謝すると同時に、奨学金を通して支えて下さったロータリー財団にも感謝致します．本研究を進める上で多大な御教示と御意見を頂きました山村研究室の皆様に御礼申し上げます．最後に、１９年間の学生生活を暖かく見守ってくれた両親に感謝します．

参考文献

[山村 97] 山村雅幸, Bayesian Network 上の強化学習, 第24回知能システムシンポジウム, pp.61－66 (1997).

[Pearl 88] Pearl, J., “ Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. ”, Morgan Kaufmann, San Mateo, California, (1988).

[Kimura et al. 95] Kimura, H., Yamamura, M., and Kobayashi, S., ”Reinforcement Learning by Stochastic Hill Climbing on Discounted Reward,” Proc. of the 12th International Conference on Machine Learning, pp.295-303 (1995).

[Binder et al. 97] Binder, J., Koller, D., Russel, S. and Kanazawa, K., “Adaptive Probabilistic Networks with Hidden Variables,” Machine Learning Journal, 1997.

[Heckerman 95] Heckerman, D., “A tutorial on Learning With Bayesian Networks,” Microsoft Technical report, MSR-TR-95-06.

[Singh et al. 94] Singh, S.P., Jaakkola, T. and Jordan, M.I., “Learning without State-Estimation in Partially Observable Markovian Decision Processes,” Proc. 11th ICML, pp.284-292(1994).

[Michel 96] Michel, O., ” Khepera simulator package version 2.0,” Freeware mobile robot simulator written at the university of nice sophia-antipolis by Olivier Michel, Available at http://diwww.epfl.ch/lami/team/michel/khepsim/index.html.

[Cassandra et al. 96] Cassandra, A.R., Kaelbling, L. P., Kurien, J.A., “Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation,” Proc. of 1996 IEEE/ RSJ IROS, 963-972(1996).

[Watkins and Dayan 92] Watkins, C.J.C.H. and Dayan, O., “Technical Note: Q-learning, Machine Learning 8,” pp.55-68(1992).

[Williams 92] Williams, R.J., “Simple Statistical Gradient Following Algorithms for Connectionist Reinforcement Learning,” Machine Learning 8, pp.229-256(1992).

[Thrun 93] Thrun, S., “Extracting provably correct rules from artificial neural networks,” Technical Report AI-TR-93-5,University of Bonn,Bonn, Germany,1993.

[Andrews and Geva 95] Andrews, R. and Geva S., “RULEX and CEBP networks as ther basis for a rule refinement system,” in ”Hybrid Problems, Hybrid Solutions”, John Hallam (Ed), 1995.

A　polytree 構造のベイジアンネットワークにおける伝搬則 [山村 97]

本研究で用いた polytree 構造のベイジアンネットワークの伝搬則を以下に説明する．polytree は任意の 2 ノード間に無向パスが一つしかない構造である．図 16 に模式図を示す.

A.1　準備

ベクトル $\boldsymbol{x} = [x_1 x_2 \cdots x_m]$ と関数 f に関して，

$$f(\boldsymbol{x}) = [f(x_1)\, f(x_2) \cdots f(x_m)]$$

以下，例えば $f(\boldsymbol{x})g(\boldsymbol{x})^T = \sum f(x_i)g(x_i)$ のように，和をコンパクトに表現するために転置やトレースといった行列演算を用いる．

条件つき確率表は次の行列形式で書く．

$$P(\boldsymbol{y}|\boldsymbol{x}) = \begin{bmatrix} P(y_1|x_1) & P(y_2|x_1) & \cdots & P(y_n|x_1) \\ P(y_1|x_2) & P(y_2|x_2) & \cdots & P(y_n|x_2) \\ \vdots & & \ddots & \vdots \\ P(y_1|x_m) & P(y_2|x_m) & \cdots & P(y_n|x_m) \end{bmatrix}$$

伝搬手続きでは次の正規化オペレータをしばしば用いる．

$$\alpha f(\boldsymbol{x}) = \frac{1}{\sum_i f(x_i)} f(\boldsymbol{x})$$

次のような 1 を含むベクトルを用いる．

$$\delta(\boldsymbol{x}, k) = [\delta_1 \delta_2 \cdots \delta_{|\boldsymbol{x}|}], \quad \delta_i = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

$$\mathbf{1}(\boldsymbol{x}) = [1\, 1 \cdots 1], \quad |\mathbf{1}(\boldsymbol{x})| = |\boldsymbol{x}|$$

さらにベクトル $\boldsymbol{u} = [u_1\, u_2 \cdots u_m]$ と $\boldsymbol{v} = [v_1\, v_2 \cdots v_n]$ の組合せについて次の演算を定義する．

$$f(\boldsymbol{u} \times \boldsymbol{v}) = [f(u_1 v_1)\, f(u_2 v_2) \cdots f(u_m v_n)]$$

$$\sum_{\boldsymbol{u}} = [f(\boldsymbol{u}v_1)\mathbf{1}(\boldsymbol{u})^T\, f(\boldsymbol{u}v_2)\mathbf{1}(\boldsymbol{u})^T \cdots f(\boldsymbol{u}v_n)\mathbf{1}(\boldsymbol{u})^T]$$

A.2　信念の伝搬

図 16 の右側半分が尤度ベクトルの伝搬に関する変数を，左側半分が事前確率ベクトルの伝搬に関する変数を示す．以下が [Pearl 88] からの信念の伝搬則である．

$$BEL(\boldsymbol{x}_i) = \alpha \pi(\boldsymbol{x}_i)\lambda(\boldsymbol{x}_i) \tag{3}$$

$$\lambda(\boldsymbol{x}_i) = \prod_j \lambda_{Y_j}(\boldsymbol{x}_i) \tag{4}$$

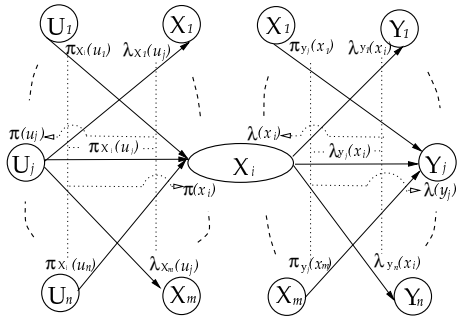

図 16: polytree における信念と適正度の伝搬

$$\pi(\boldsymbol{x}_i) = \sum_{v \in \underset{k}{\times} \boldsymbol{u}_k} P(\boldsymbol{x}_i|v) \prod_{u_{kh} \in v} \pi_{x_i}(u_{kh}) \tag{5}$$

$$\lambda_{Y_j}(\boldsymbol{x}_i) = \beta \lambda(\boldsymbol{y}_j) \left[\sum_{w \in \underset{k \neq i}{\times} \boldsymbol{x}_k} P(y_j|\underset{k}{\times}\boldsymbol{x}_k) \prod_{x_{kh} \in w} \pi_{Y_j}(x_{kh}) \right]^T \tag{6}$$

$$\pi_{X_i}(\boldsymbol{u}_j) = \alpha \pi(\boldsymbol{u}_j) \prod_{k \neq i} \lambda_{X_k}(\boldsymbol{u}_j) \tag{7}$$

上の数式の [ ] 内は次のような行列となる．

	y_{j1} $\cdots$ y_{jg} $\cdots$ y_{jn}
x_{i1}	
$\vdots$	
x_{il}	$\sum_{w = \underset{k \neq i}{\times} \boldsymbol{x}_k} P(y_{ig}|x_{il}, w) \prod_{x_{kh} \in w} \pi_{Y_j}(x_{kh})$
$\vdots$	
x_{im}	

A.3　適正度の伝搬

3 式より，

$$\varepsilon\pi(\boldsymbol{x}) = \frac{1}{\pi(x_d)}\delta(\boldsymbol{x}, d) - \frac{\lambda(\boldsymbol{x})}{\pi(\boldsymbol{x})\lambda(\boldsymbol{x})^T} \tag{8}$$

$$\varepsilon\lambda(\boldsymbol{x}) = \frac{1}{\lambda(x_d)}\delta(\boldsymbol{x}, d) - \frac{\pi(\boldsymbol{x})}{\pi(\boldsymbol{x})\lambda(\boldsymbol{x})^T} \tag{9}$$

4 式からの適正度の伝搬則は以下のようになる．

$$\varepsilon\lambda_{Y_j}(\boldsymbol{x}_i) = \varepsilon\lambda(\boldsymbol{x}_i) \prod_{k \neq j} \lambda_{Y_k}(\boldsymbol{x}_i) \tag{10}$$

同様に，5 式からの適正度の伝搬則は以下のようになる．

$$\varepsilon\pi_{X_j}(\boldsymbol{u}_k) = \varepsilon\pi(\boldsymbol{x}_i) \left[\sum_{v \in \underset{k \neq i}{\times} \boldsymbol{u}_k} P(\boldsymbol{x}_i|\underset{k}{\times}\boldsymbol{u}_k) \prod_{u_{kh} \in v} \pi_{X_j}(u_{kh}) \right]^T \tag{11}$$

$$\varepsilon P(\boldsymbol{x}_i|\underset{k}{\times}\boldsymbol{u}_k) = \varepsilon\pi(\boldsymbol{x}_i)^T \underset{k}{\times} \pi_{x_j}(\boldsymbol{u}_k) \tag{12}$$

6 式より，$B = \lambda_{Y_j}(\boldsymbol{x}_i)\mathbf{1}\boldsymbol{x}_i)$ を正規化係数とすると，

$$\varepsilon\lambda(\boldsymbol{y}_j) = -\frac{1}{B^2}\varepsilon\lambda_{Y_j}(\boldsymbol{x}_i)\sum_{\substack{w\in\times\boldsymbol{x}_k \\ k\neq i}} P(\boldsymbol{y}_j|\underset{k}{\times}\boldsymbol{x}_k)\prod_{x_{kh}\in w}\pi_{Y_j}(x_{kh}) \tag{13}$$

$$\varepsilon\pi_{Y_j}(\boldsymbol{x}_l) = -\frac{1}{B^2}\varepsilon\lambda_{Y_j}(\boldsymbol{x}_i)\sum_{\substack{w\in\times\boldsymbol{x}_k \\ k\neq l}} P(\boldsymbol{y}_j|\underset{k}{\times}\boldsymbol{x}_k)\prod_{x_{kh}\in w}\begin{Bmatrix} 1 & k=i \\ \pi_{Y_j}(x_{kh}) & k\neq i \end{Bmatrix} \tag{14}$$

$$\varepsilon P(\boldsymbol{y}_j|\underset{k}{\times}\boldsymbol{x}_k) = -\frac{1}{B^2}\left[\left(\overset{i-1}{\underset{k=1}{\times}}\pi_{Y_j}(\boldsymbol{x}_k)\right)\times\mathbf{1}(\boldsymbol{x}_i)\times\left(\overset{m}{\underset{k=i+1}{\times}}\pi_{Y_j}(\boldsymbol{x}_k)\right)\right]^T\lambda(\boldsymbol{y}_j) \tag{15}$$

7 式より，$A = \pi_{X_i}(\boldsymbol{u}_j)\mathbf{1}(\boldsymbol{u}_j)^T$ を正規化係数とすると，

$$\varepsilon\pi(\boldsymbol{u}_j) = -\frac{1}{A^2}\varepsilon\pi_{X_i}(\boldsymbol{u}_j)\prod_{k\neq i}\lambda_{X_k}(\boldsymbol{u}_j) \tag{16}$$

$$\varepsilon\lambda_{X_j}(\boldsymbol{u}) = -\frac{1}{A^2}\varepsilon\pi_{X_i}(\boldsymbol{u}_j)\pi(\boldsymbol{u}_j)\prod_{k\neq i,j}\lambda_{X_k}(\boldsymbol{u}_j) \tag{17}$$

B 実際に用いた条件つき確率表の初期値

本文中で各実験で用いた知識を実際にどのような条件つき確率表としてベイジアンネットワークに埋めこんだかを以下に記す。以下の表は，確率の和が 1 になるように正規化されて使われている．

B.1 グリッドワールドにおける実験

B.1.1 地図上の各状態において各センサがどのようになっているかという知識

エージェントが地図上の各状態にいるときに東西南北の各方向センサがどのようになるかを表で示す．

状態	東方向センサ 1	
	off	on
1	0	1
2	0	1
3	0	1
4	1	0
5	0	1
6	0	1
7	0	1
8	0	1

状態	西方向センサ 1	
	off	on
1	0	1
2	0	1
3	1	0
4	1	0
5	0	1
6	0	1
7	0	1
8	0	1

状態	南方向センサ 1	
	off	on
1	0	1
2	1	0
3	1	0
4	0	1
5	1	0
6	1	0
7	1	0
8	1	0

状態	北方向センサ 1	
	off	on
1	1	0
2	1	0
3	0	1
4	0	1
5	0	1
6	1	0
7	1	0
8	1	0

B.1.2 一行動では地図上の隣接している地点か今居た状態のどちらかにしか移動できないとい知識

各前の状態と各前にとった行動のそれぞれの場合にエージェントが現在どの状態にいるかという確率を表で示す．

前の状態	前の行動	状態							
		1	2	3	4	5	6	7	8
1	n	1	1	0	0	0	0	0	0
1	w	1	0	0	0	0	0	0	0
1	s	1	0	0	0	0	0	0	0
1	e	1	0	0	0	0	0	0	0
2	n	0	1	1	0	0	0	0	0
2	w	0	1	0	0	0	0	0	0
2	s	1	1	0	0	0	0	0	0
2	e	0	1	0	0	0	0	0	0
3	n	0	0	1	0	0	0	0	0
3	w	0	0	1	1	0	0	0	0
3	s	0	1	1	0	0	0	0	0
3	e	0	0	1	0	0	0	0	0
4	n	0	0	0	1	0	0	0	0
4	w	0	0	0	1	0	0	1	0
4	s	0	0	0	1	0	0	0	0
4	e	0	0	1	1	0	0	0	0
5	n	0	0	0	0	1	0	0	0
5	w	0	0	0	0	1	0	0	0
5	s	0	0	0	0	1	1	0	0
5	e	0	0	0	0	1	0	0	0
6	n	0	0	0	0	1	1	0	0
6	w	0	0	0	0	0	1	0	0
6	s	0	0	0	0	0	1	1	0
6	e	0	0	0	0	0	1	0	0
7	n	0	0	0	0	0	1	1	0
7	w	0	0	0	0	0	0	1	0
7	s	0	0	0	0	0	0	1	1
7	e	0	0	0	1	0	0	1	0
8	n	0	0	0	0	0	0	1	1
8	w	0	0	0	0	0	0	0	1
8	s	0	0	0	0	0	0	0	1
8	e	0	0	0	0	0	0	0	1

B.1.3 壁の方向には進めないという知識

各行動を選択したそれぞれの場合に以下のセンサがどのような値になっているかという確率を表で示す.

行動	東方向センサ 2	
	off	on
北	1	1
西	1	1
南	1	1
東	1	0

行動	西方向センサ 2	
	off	on
北	1	1
西	1	0
南	1	1
東	1	1

行動	南方向センサ 2	
	off	on
北	1	1
西	1	1
南	1	0
東	1	1

行動	北方向センサ 2	
	off	on
北	1	0
西	1	1
南	1	1
東	1	1

B.2 ケペラロボットシミュレータにおける実験

B.2.1 一行動では地図上の隣接している状態か今居た状態のどちらかにしか動けないという物理的制約と直進の行動を選択した時のみ状態が変化しそれ以外の行動では変化しないという知識

各前の状態を前の行動からそれぞれの場合にどの状態にいるかという確率を表で示す.

前の状態	前の行動	状態			
		1	2	3	4
1	回転	1	0	0	0
1	直進	1	1	0	0
2	回転	0	1	0	0
2	直進	1	1	1	0
3	回転	0	0	1	0
3	直進	0	1	1	1
4	回転	0	0	0	1
4	直進	0	0	1	1

B.2.2 光は地図上の状態 1，2においては東側から，状態 4においては西側から，状態 3においては見えないという知識

各状態に居る時に光があるかどうかの確率を示す.

状態	光の有無	
	無し	有り
1	1	10
2	1	10
3	1	0
4	1	10

B.2.3 壁の方向には進めないという制約

各行動を選択したときに各センサがオンかオフかの確率を表で示す.

行動	近接センサ 0	
	off	on
直進	1	1
右折	1	1
左折	1	0

行動	近接センサ 2	
	off	on
直進	1	0
右折	1	3
左折	1	3

行動	近接センサ 3	
	off	on
直進	1	0
右折	1	3
左折	1	3

行動	近接センサ 5	
	off	on
直進	1	1
右折	1	0
左折	1	1