近似価値関数を用いた経路探索アルゴリズムの計算効率性について*

Efficiency of Shortest-path Search Algorithms Based on Approximate Value Functions *

遠藤雅人**・宮城俊彦***

By Masato ENDO**・Toshihiko MIYAGI***

１．はじめに

近年，GPS を利用したカーナビゲーションや料金所渋滞の解消を目的とした ETC(自動料金収受システム)といった ITS(高度交通情報システム)が発展してきている．特に，カーナビゲーションにおける経路誘導機能は出発地と目的地間の最短経路情報をドライバーに提供し，渋滞の抑制につながるため，交通システム全体にも望ましい影響を与えている．しかし，この機能は最短距離となる経路を検索することは可能であるが，走行時間情報を組み入れたものではないため，必ずしも最短経路が最小時間経路であるとは限らない．

実際に同じ経路を同じ時間帯に反復走行した場合においても，道路特性や交通特性に依存して所要時間に大きなバラつきが発生することが実証されている．

時々刻々と変化する交通状況に対応して，リアルタイムな交通状況を観測し，経路誘導を行うシステムとして VICS が存在する．VICS は客観的な交通情報に基づく経路誘導を行えるため，現時点では最も優れた経路情報提供システムと言えるが，高速道路や都市部の主要道路など観測地点が限定され，またドライバーに提供されるデータは事前情報にすぎないという特徴を持つ．

このように現在の経路誘導システムは，ドライバーにとって有効で詳細な走行情報である自身の走行経験を考慮していない．つまり，問題はその走行経験を蓄積し，解析し，そして有利な指示を与える機能を現在のカーナビゲーション・システムが備えていないという点である．GPS は車両の現在位置を特定することができ，ドライバー自身の経路走行データを取得することも可能である．この点に着目するならば，自己の走行経験を１つの交通情報データとしてナビゲーターシステムに取り込み，より優れた経路を探索する手がかりを得ることができるのではないかという発想も生まれる．本研究は次世代カーナビシステムを念頭に，その基礎を成す，走行経験のデータを利用した確率的最短経路探索アルゴリズムついて研究したものである．

*キーワーズ：経路選択，交通手段分析，活動分析

**非会員，学(工)，東北大学情報科学研究科

***正員，博(工)，東北大学情報科学研究科

(宮城県仙台市青葉区荒巻字青葉6-6-06,

TEL:022-795-7504,

E-mail: toshi_miyagi@plan.civil.tohoku.ac.jp)

２．確率的最短経路問題の定式化

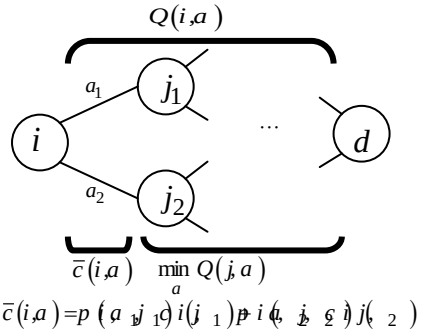

図１　想定するネットワーク

ここでは，図１のようなネットワークを想定する．ノード i を始点とし，終点 d まで方策 π に基づいて行動 a をとった場合の期待最小所要時間を $Q^*(i,a)$ とすると，確率的最短経路問題は，以下の Bellman 方程式を解く問題に帰着できる．

$$Q^*(i,a)=\bar{c}(i,a)+\beta\sum_{j\in S}p(i,a,j)Q^*\left(j,\pi(j)\right) \qquad (1)$$

ここで，$i,j\in S$ はノードを，また，S はノードの集合を表す．A を行動 a の有限集合，$V(i)$ はノード i から目的地までの所要時間関数の期待値であり，状態価値関数と呼ぶ．$\bar{c}(i,a)$ はノード i で行動 a をとって次のノードに移動するときの所要時間 $c(i,a)$ の期待値である．$p(i,a,j)$ はノード i から行動 a をとってノード j に推移する確率であり，それ以前の推移の履歴には依存しないと仮定する（マルコフ性）．β は割引率である．

状態 i における近似価値関数が最小となるときの行動を選択するため，最適方策は次式によって求められる．

$$\pi^*(i)=\arg\min_{a\in A(i)}Q(i,a) \qquad (2)$$

Bellman 方程式は推移確率が与えられる場合には効率的に解を求めることができる．しかし，現実の問題に応用するとき，一般に，状態空間は巨大であり，マルコ

フ推移確率行列の次元は膨大なものになる．また，交通ネットワーク問題では，ノード間推移確率を求めること自体、不可能なケースがほとんどである．そこで，本研究では，推移確率は未知のものとして取り扱う．しかし，日々の走行データを蓄積すれば，これを有効に利用することができ，線形関数近似手法を用いることで，状態の汎化による状態数の減少によって学習を高速化することが可能になる．

３．Q-Routing

TD 学習をベースとし，各状態で取れるすべての行動を十分に実行することで，最適な方策を学習できるアルゴリズムを Q 学習という[Watkins,1992]．各状態でどの行動が良いかを評価するために，各状態と行動のすべての組み合わせに評価値である行動価値関数 Q を持たせる．この場合，学習で獲得される行動価値関数 Q は，使われている方策とは独立に最適行動価値関数Q^*を直接近似する．

Littman and Boyan (1993)は，Q 学習を確率的最短経路問題に適用した適応型ルーティングアルゴリズムを提案した．Q-Routing では，エージェントをパケットとして置き換えて考える．Q-Routing アルゴリズムは，ルーティングテーブルとして行動価値関数 Q を使用する．これは終点ノードが d であるパケットをノード i から隣接ノード j に送信した場合の転送時間の推定値である．つまり，パケットが i に到着してから d に到着するまでの時間だといえる．パケットが移動する隣接ノードの選択としては，単に Q 値が最小となるノードが選ばれる．そして，パケットが i から j へと移動したとき，即座に j から d までの送信時間の最小推定値 t が i に返される．

$$t = \min_{z \in neighbors\ of\ j} Q_j(z,d) \tag{3}$$

パケットの i における待ち時間をq_iとすると，図２のように表せる．i は次のように Q 値を更新する．

$$Q_i(j,d) \leftarrow Q_i(j,d) + \alpha\left(q_i + t - Q_i(j,d)\right) \tag{4}$$

ここで，αは学習率(0<α<1)と呼ばれ，更新の強さを制御するパラメータである．(4)式を図１のような確率的最短経路問題に応用すると，以下の式が得られる．

$$Q(i,a) \leftarrow Q(i,a) + \alpha\left(c(i,a) + \min_b Q(j,b) - Q(i,a)\right) \tag{5}$$

特徴として，ランダムに行動選択を行うため計算効率性が悪いことがあげられる．しかしながら，あらゆる状態や行動を選択するため，必ず最適値に収束する．

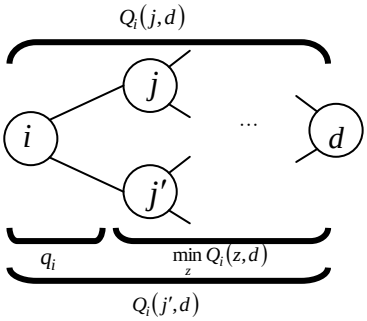

図２　Q-Routing

４．LSPI法

LSPI(Least-Squares Policy Iteration)法とは，最小二乗不動点近似法を用いて価値関数を近似(方策評価)し，近似価値関数が最小となるときの行動を選択（方策改善）することで，最適方策を解析的に求める学習アルゴリズムのことである．現実の問題に強化学習を適用する際，多くの場合に状態空間は巨大であり，かつ状態間の所要時間や推移確率は未知である．このとき，価値関数を評価するために大量のサンプルを必要とする．また，Q-Routing によってあらゆる行動を選択することでランダムに推移確率を発生させ，価値関数を近似していく方法は非効率的であり，多くの時間を要する．そこで，複数の基底関数にそれぞれ重み係数ベクトルを掛け合わせることで，真の価値関数を近似する．基底関数を用いた状態行動価値関数の表現は以下のようである．

$$\hat{Q}(i,a;w) = \sum_{i=1}^{k} \phi_i(i,a) w_i = \phi(i,a)^T w \tag{6}$$

ここで，$\hat{Q}(i,a;w)$は状態i，行動aのときの状態行動価値関数，wは重み係数ベクトル，$\phi(i,a)$は状態i，行動aにおける基底関数，kは基底関数の個数をそれぞれ表す．重み係数ベクトルwは最小二乗不動点近似法により，次式を解くことで求められる．

$$\mathbf{A}w = b \tag{7}$$

このとき，

$$\mathbf{A} = \mathbf{\Phi}^T\left(\mathbf{\Phi} - \beta \mathbf{P}\mathbf{\Pi}_\pi \mathbf{\Phi}\right)$$
$$b = \mathbf{\Phi}^T \mathbf{C} \tag{8}$$

である．推移確率行列 P や所要時間則行列 C が未知である場合，サンプルによって収集されたデータから，(9)式のように A，b を近似することができる．

$$\tilde{\mathbf{A}} = \frac{1}{L}\sum_{t=1}^{L}\left[\phi(i_t,a_t)\left(\phi(i_t,a_t) - \beta\phi(j_t,\pi(j_t))\right)^T\right]$$
$$\tilde{b} = \frac{1}{L}\sum_{t=1}^{L}\left[\phi(i_t,a_t)c(i_t,a_t)\right] \tag{9}$$

ここで，基底関数の個数を k ，サンプル数を L とすると，$\mathbf{A}$ は $k \times k$ 行列，$\mathbf{b}$ は $k \times 1$ 行列，$\mathbf{\Phi}$ は $k \times L$ 行列として表わされる．

したがって，(7)式により重み係数ベクトルを求め，(6)式により行動価値関数を求める．このときの最適方策は次式によって求められる．

$$\pi^*(i) = \arg\min_{a \in A} \hat{Q}(i,a) = \arg\min_{a \in A} \phi(i,a)^T w \quad (10)$$

(10)式で求めた方策 $\pi(i)$ を用いて，(9)式より基底関数の値を更新する．この反復計算により，最適値に収束する．本研究で用いた LSPI アルゴリズムを表１に示す．

表１　LSPI アルゴリズム

(Step 1)　t=1 とする．w^0 ，π^0 を初期値とし，基底関数と許容誤差 ε を設定．

(Step 2)　全サンプルデータによって構成された基底関数行列 $\mathbf{\Phi}(i,a)$ ，$\mathbf{\Phi}(j,\pi(j))$ ，そして全試行における所要時間行列 $\mathbf{C}(i,a)$ を用いて $\tilde{\mathbf{A}}$ と $\tilde{b}$ を求める．

$$\tilde{\mathbf{A}} = \frac{1}{L}\mathbf{\Phi}(i,a)^T[\mathbf{\Phi}(i,a) - \beta\mathbf{\Phi}(j,\pi(j))]$$

$$\tilde{b} = \frac{1}{L}\mathbf{\Phi}(i,a)^T\mathbf{C}(i,a)$$

(Step 3)　$\tilde{\mathbf{A}}$ ，$\tilde{b}$ の値から以下の式に基づいて重み係数ベクトル w を求める

$$w^{t+1} \leftarrow \tilde{\mathbf{A}}^{-1}\tilde{b}$$

(Step 4)　以下の式より価値関数を求める．

$$\hat{Q}^{t+1} = \mathbf{\Phi}w^{t+1}$$

(Step 5)　最適方策を決定

$$\pi^{t+1}(i) \leftarrow \arg\min_{a \in A(i)} \hat{Q}^{t+1}(i,a)$$

(Step 6)　以下の式を満足しない場合、t=t+1 として、(Step 2)に戻る．

$$\left\| w^{t+1} - w^t \right\| < \varepsilon$$

５，シミュレーション実験

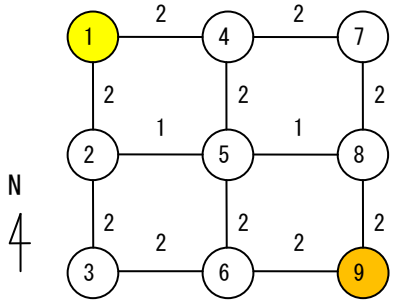

図３　シミュレーションモデル

図３のネットワークモデルを用いて収束効率性を検証した．状態集合を $\mathbf{S} = \{1,2,3,4,5,6,7,8,9\}$ ，行動集合を $\mathbf{A} = \{$東へ進む, 南に進む$\}$ ，割引率 β を 1.0 とする．始点ノードを 1，終点ノードを 9 とおく．リンク上に表示された数字はリンク所要時間の期待値を表す．ここで，リンク間の所要時間はノード 2・5 間において 60%の確率で 1，20%の確率で 0.5，20%の確率で 1.5 となるように推移確率を設定した．同様に，それ以外のリンクでは 60%の確率で 2，20%の確率で 1.5，20%の確率で 2.5 となるようにした．このように確率的に変動するリンク所要時間のもとで，エージェントは始点から終点まで推移する．

このとき，現在のノード数 i ，行動 a ，所要時間 c ，推移後のノード数 j ，終点ノード到達時出力数 ab ，経験頻度 ex を毎試行ごとに出力する．終点ノード 9 に到達すると，また始点に戻る．これを１エピソードとし，100 回繰り返す．ここで，過剰推定を防止するため，1 エピソード当たりの試行回数が 7 回を超えるものは解析に含めない．累積したデータを以下の式により基底関数に変換する．

$$\mathbf{\Phi} = (1 \quad i \quad ab \quad i*j \quad c*j \quad ex)^T \quad (11)$$

基底変数の選択は，F 検定による逐次変数選択法と相関分析の２つの方法を検討した．その結果，相関分析を用いたほうが良い結果が得られることが明らかとなった．図２は相関分析で得られた基底変数を用いた場合の価値関数近似の結果を示したものである．

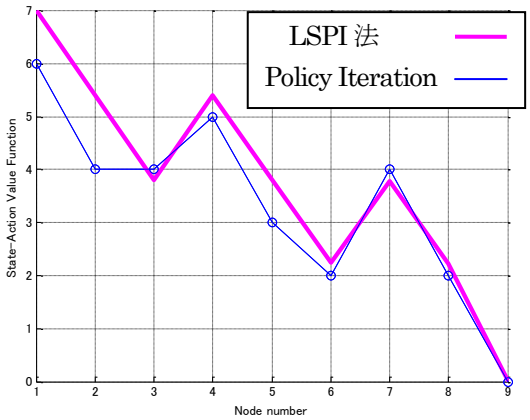

図４　ノード数と近似価値関数の関係

図４は横軸にノード数，縦軸に価値関数 Q を取っている．推移確率を既知としたときの真の価値を”o”で示し，LSPI における結果を重ねて表示した．図４より，近似価値関数は真の値を十分な精度で近似していることが分かる．しかし，価値関数の近似に関しては問題ないが，そこから求められた最適方策は正確なものではなかったため，この方法はまだ発展途上であるといえる．

また，推移確率が未知な場合における代表的アルゴリズムである Q-Routing と LSPI 法との収束特性を比較した．結果を図５に示す．

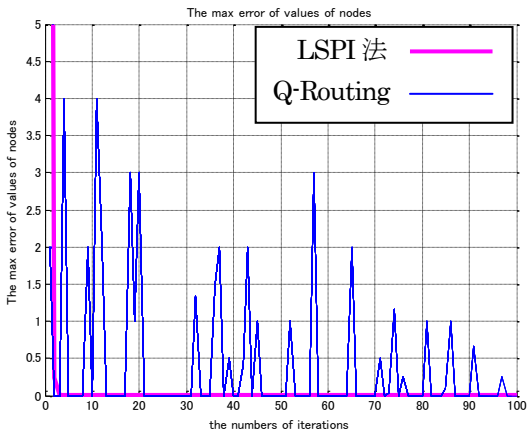

図５　Q-Routing と LSPI 法の収束特性の比較

図５は横軸に反復計算回数，縦軸に価値関数の TD 誤差を取っている．分散の大きい Q-Routing に対し，LSPI 法はわずかな反復回数で収束している．よって，LSPI 法の収束効率性の高さを実証することができた．

６，結論

本研究では，集約状態を用いた基底関数の最適選択と LSPI 法を組み合わせることで，新しいモデルフリーの強化学習法を適用した最短経路探索法を提案した．本研究で提案する手法はネットワークにおけるノード間の推移で得られる情報のみを対象として数値解析を行い，より複雑な価値の推定を可能としている．これは，事前知識を全く必要としておらず，理論的にも正当性が証明されている．また，方策評価において任意に基底関数の数を設定することができる．そのため，ある方策評価における最適な基底関数を適切に設定することで，より効率的な学習が行えると考えられる．しかしながら，シミュレーション実験での結果は価値関数を近似するという点で，考察することができたが，最適方策は不安定であった．このような失敗例から，本研究で行った基底関数設定方法は必ずしも最適な方法であるといえず，依然として研究の可能性に満ちている．

従来，ネットワークにおける状態間移動は行動と一対一の関係にあった．つまり，ある交差点 A で左折をしたら交差点 B に到着するといった関係である．しかし，本研究では確率的な推移関数を用いることで，状態間移動において異なる行動を取り扱うことが可能となった．これにより，行動 A と行動 B をとったときの状態間移動は異なる結果が得られることになる．この手法を応用することで，交差点間移動に例えた場合，加速する，減速する，停止するといった行動をとることで，推移関数により目的地に到着するまでの期待時間をより高精度に求めることが可能となる．また，交通だけなく，物流や通信といったネットワークが用いられる分野において，この方法は応用できる可能性があるといえる．

最新の GPS は現時点での座標データが 1 秒間隔で取得できる．そのため，車両に GPS 受信機を搭載し，最寄りの電子基準点データを利用して車両の移動軌跡を解析処理することで，速度や加速度といった走行特性が明らかとなっている．移動に伴って得られるこれらのデータから，相関係数によって最適な基底関数を設定することで，実際の道路網における価値関数の近似に応用できると考えられる．

今後の課題として，最適方策を確実に推定できる基底関数の検証，計算時間の短縮，データ収集方法などがあげられる．また，実験方法を変化させることで，さらに本研究の有効性の検証を行っていく必要がある．現実の道路網では，本研究で用いたシミュレーションモデルと比較してはるかに複雑となっている．実用性を考慮したカーナビゲーションに応用するとした場合，アルゴリズムの効率化などを行って莫大なデータをより短縮した計算時間で解を導くことが重要な課題となる．データの収集方法，収集したデータの扱い方などに関しても，今後検討していく必要がある．

参考文献

1)Lagoudakis and Parr: Least-Squares Policy Iteration,Department of computer Science,Duke University, ,2003

2) R. S. Sutton & A.G. Barto , Reinforcement Learning : An Introduction.MIT Press , 1998

3)Michail G. Lagoudakis. and Ronald Parr.,Least-Squares Policy Iteration,Department of Computer Science,Duke University,Durham,NC27708,USA, 2003

4)薮内佳孝，加藤昇平，犬塚信博：強化学習に基づいたルーティングアルゴリズムの一手法，The 18th Annual Conference of the Japanese Society for Artificial Intelligence,2004

5)Bellman , R . E.:Dynamic Programming , Princeton University Press , 1957

6)Dimitri P .Bertsekas & John N. Tsitsiklis : Neuro-Dynamic Programming , Athena Scientific , 1996

7)Michael Littman & Justin Boyan :A distributed Reinforcement Learning Scheme for Network Routing . Technical Report CMU-CS-93-165,School of Computer Science , Carnegie Mellon University , 1993

8)Watkins , C.J.C.H & Dayan , P.: Learning from Delayed Rewards. PhD thesis, Cambridge University,1989