これからの強化学習

第2章 強化学習の発展的理論

Yuma Yamakura

目次

2.2 強化学習アルゴリズムの理論性能解析とベイズ統計による強化学習のモデル化

2.2節概要

強化学習の根本的な問題

⇒探索と利用のトレードオフ

⇒定量化するのは難しい

しかし, 最悪性能を理論的に評価する方法はある！

⇒リグレット, サンプル複雑性に基づく解析

2.2節概要

強化学習の根本的な問題
⇒探索と利用のトレードオフ
⇒定量化するのは難しい
⇒なぜ？
⇒事前知識がない場合が多いから

ベイズ事前分布の形式で事前知識が得られる
⇒探索と利用のトレードオフが一般的に取り扱える
(計算は指数オーダーなので近似するが…)

2.2節のまとめ図

強化学習中

探索　トレードオフ　利用　⇒　価値関数(報酬和)

評価　　評価

探索コストの最悪性能の評価(2.2.1),(2.2.2)

- 楽観的初期値評価法(2.2.2)
  「悲観的より楽観的に」を表すアルゴリズム
- モデルベースのサンプル複雑性(2.2.2)
  探索コストの指標
  アルゴリズム : Rmax や MoRmax , OIM
- モデルフリーのサンプル複雑性(2.2.2)
  アルゴリズム : Delayed Q-learning

⬇

探索コストの最小化

リグレット上界の評価(2.2.1)

※探索と利用のトレードオフを表す
※報酬和の差

探索コストの最小化(ベイズ主義)

- 通常のDP
  (環境の知識が既知, 強化学習にする必要なし)
- ベイズ適応的MDP(BAMDP)(2.2.3)
  (不確かさを確率的に扱う, POMDPの強化学習へ)

  ※事前分布がきちんとモデル化できる必要

2.2.1 多腕バンディット問題

2.2節のまとめ図

強化学習中

探索 ↔ 利用 ⇒ 価値関数(報酬和)

トレードオフ

評価

評価

探索コストの最悪性能の評価(2.2.1),(2.2.2)

・楽観的初期値評価法(2.2.2)
　「悲観的より楽観的に」を表すアルゴリズム

・モデルベースのサンプル複雑性(2.2.2)
　探索コストの指標
　アルゴリズム : Rmax や MoRmax , OIM

・モデルフリーのサンプル複雑性(2.2.2)
　アルゴリズム : Delayed Q-learning

↓

探索コストの最小化

リグレット上界の評価(2.2.1)

※探索と利用のトレードオフを表す
※報酬和の差

探索コストの最小化(ベイズ主義)

・通常のDP
　(環境の知識が既知, 強化学習にする必要なし)
・ベイズ適応的MDP(BAMDP)(2.2.3)
　(不確かさを確率的に扱う, POMDPの強化学習へ)

　※事前分布がきちんとモデル化できる必要

2.2.1 多腕バンディット問題

探索と利用のトレードオフが生じる最も簡単な問題
(詳細は1.1節で…)

1. K本の腕に[0,1]上の報酬の確率分布$\boldsymbol{\nu_1, \ldots, \nu_K}$(未知)
2. 各時刻$\boldsymbol{t}$でプレイヤーは腕$\boldsymbol{i(t) \in \{1, \ldots, K\}}$を選ぶ
3. 報酬$\boldsymbol{x_{i(t)}(t) \sim \nu_{i(t)}}$を観測

これは状態1個, 行動がK個のMDPと等価

2.2.1 リグレット

リグレット(regret)

⇒「(最適な行動をとらなかったことによる)後悔」

⇒最適解を最初から実行していた場合に比べて，

　行動による損失がどれくらいであったか？

※最適化問題ではよく出てくる用語

2.2.1 多腕バンディット問題のリグレット

強化学習の問題

⇒利得(報酬和 $\sum_t x_{i(t)}(t)$)の最大化を目指す

リグレットを用いると…

⇒リグレット($\bar{R}_T$)の最小化を目指す

$$\bar{R}_T = \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} x_{i*}(t)\right]} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} x_{i(t)}(t)\right]}$$

最適方策を選び続けたときの利得の期待値

ある方策で得た利得の期待値

2.2.1 リグレットの直感的理解

リグレットは,「探索と利用のトレードオフ」をシンプルに表している

- 探索が少なすぎる(利用が多すぎる)

⇒最適な腕を見つけられず, リグレットが増える

- 探索が多すぎる(利用が少なすぎる)

⇒最適な腕を選び続けないので, リグレットが増える

バランスが良いときにリグレットが最小になる

2.2.1 $\varepsilon - greedy$方策

$\varepsilon - greedy$アルゴリズム : 方策をどう決めるか？

腕iの報酬の期待値(=価値関数)を$\overline{x}_i$とすると,

$$\pi = \begin{cases} arg\,max_i\ \overline{x}_i & \text{(with probability 1- } \varepsilon\text{) (利用)} \\ \text{ランダムに}i\text{を選択} & \text{(with probability } \varepsilon\text{) (探索)} \end{cases}$$

※貪欲法($greedy$ アルゴリズム)

$\pi = arg\,max_i\ \overline{x}_i$ (利用)

2.2.1 $\varepsilon - greedy$方策のリグレット

貪欲法では利用しかしないが, $\varepsilon - greedy$アルゴリズムでは探索も行う

⇒試行回数を大きくすると, 最適な腕を間違える確率は減る

εを固定にする

⇒リグレットの上界 : $O(T)$ (結構悪い)

εを1/tに比例して減衰

⇒リグレットの上界 : $O(\log T)$ (簡単なのに改善できる！)

2.2.1 不確かなときは楽観的に

$\varepsilon - greedy$アルゴリズムはあくまでランダム探索
⇒探索と利用がハッキリと分離している
⇒両方を同時にこなすアルゴリズムは？
⇒UCB1

※「不確かなときは楽観的に」原理
⇒初期評価を楽観的(高い値)にしておくことで, 最初のうちはいろんな腕を探索する
Ex)すべてのアームの初期値を1億とかにしとくと, すべてのアームを1回ずつ探索するはず！

2.2.1 UCB1アルゴリズム

p.11でやってるはず…

プレイヤーは n_i : T回目までに腕iを選択した回数 として,

$\bar{x}_i + \sqrt{\frac{2 \ln T}{n_i}}$ を最大にする腕を選択

- 最初は第2項の影響大 ⇒ 適当に腕を選びまくる(探索)
- 中盤は第1項と第2項のバランスがとれる
- 終盤は第1項の影響第 ⇒ 良い腕を選びまくる(利用)

2.2.1 UCB1アルゴリズム

p.11でやってるはず…

プレイヤーは n_i : T回目までに腕iを選択した回数 として,

$\bar{x}_i + \sqrt{\dfrac{2\ln T}{n_i}}$ を最大にする腕を選択

- 最初は第2項の影響大 ⇒ 適当に腕を選びまくる(探索)
- 中盤は第1項と第2項のバランスがとれる
- 終盤は第1項の影響第 ⇒ 良い腕を選びまくる(利用)

2.2.1 UCB1アルゴリズムのリグレット

リグレットの上界は$O(log\ T)$
しかも, $\varepsilon - greedy$よりも係数が非常に小さい！
⇒めちゃくちゃ良い利用と探索のトレードオフになる！

2.2.1 UCB1の応用例

- モンテカルロ木探索法

ランダムシミュレーション(プレイアウト)によって,
探索木を構成するアルゴリズム
⇒効率のよいプレイアウトは？
⇒UCB1のようにして求める

- UCTアルゴリズム

探索枝の選択をバンディット問題として考える
⇒UCB1を適用

2.2.1 Tompsonサンプリング

報酬がベルヌーイ分布(確率μで1, それ以外で0)に従う場合のベイズ推論アルゴリズム

1. 腕iのパラメータ$\mu_i \in [0, 1]$に対して, 事前分布として一様分布を考える
2. 時刻tまでの結果を観測した後のμ_iの事後分布$\pi_{i,t}$から, 各腕ごとにサンプル$\theta_{i,t}$を収集
3. サンプル$\theta_{i,t}$が最大となる腕iを時刻$t+1$の行動として選択

2.2.1 Tompsonサンプリング

- 環境をベイズ的にモデル
- 「不確かなときは楽観的に」
  ⇒事後分布のサンプリング
  ⇒楽観的なサンプリングが出たら, その腕を用いる

- 結局, UCB1と同じ考え方
  (性能も似たり寄ったり)

2.2.1 その他のバンディット問題

- 敵対バンディット

⇒相手が自由に設定を変えられる, 確率的仮定が通用しない

- 文脈付きバンディット
- 連続バンディット

⇒腕が離散ではなく, d次元実数空間上の点で表現できる

- マルコフ的バンディット

⇒腕がMarkov過程で遷移する

⇒Gittins indexというアルゴリズム

2.2.2 強化学習における探索コスト最小化

2.2節のまとめ図

強化学習中

探索 ⟷ 利用 ⇒ 価値関数(報酬和)

トレードオフ

評価

評価

探索コストの最悪性能の評価(2.2.1),(2.2.2)

・楽観的初期値評価法(2.2.2)
「悲観的より楽観的に」を表すアルゴリズム

・モデルベースのサンプル複雑性(2.2.2)
探索コストの指標
アルゴリズム : Rmax や MoRmax , OIM

・モデルフリーのサンプル複雑性(2.2.2)
アルゴリズム : Delayed Q-learning

⬇

探索コストの最小化

リグレット上界の評価(2.2.1)

※探索と利用のトレードオフを表す
※報酬和の差

探索コストの最小化(ベイズ主義)

・通常のDP
(環境の知識が既知, 強化学習にする必要なし)

・ベイズ適応的MDP(BAMDP)(2.2.3)
(不確かさを確率的に扱う, POMDPの強化学習へ)

※事前分布がきちんとモデル化できる必要

2.2.2 導入

バンディット問題だけではなく, 一般的な強化学習でもリグレットを考えることはできないか？

⇒MDPの上での探索と利用のトレードオフ

MDPは, 探索すべき未来の状態につながる行動を探す必要があり, 単純ではない

2.2.2 楽観的初期価値法

Q-learningなどのモデルフリーな手法において,
各状態の**行動価値の初期値を高く設定**しておく
⇒探索が少ない領域は更新が少ない
⇒高い価値になっているはず
⇒探索される
⇒正確な価値に収束していく

しかし, 効率が良いとはいえないし, 失敗もする

2.2.2 サンプル複雑性：モデルベース手法

定義1

アルゴリズム$\mathcal{A}$をMDP $\mathcal{P}$ 上で実行して生成された履歴$c = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots)$があるとする.

アルゴリズムを非定常な方策と考え, 時刻tでの方策を$\mathcal{A}_t$で表し, 状態s_tから方策$\mathcal{A}_t$を実行したときの期待割引報酬和を$V^{\mathcal{A}_t}(s_t) = \mathbb{E}[r_t + \gamma r_{t+1} + \cdots]$で表す.

$\mathcal{A}$のサンプル複雑性とは, 任意の$\varepsilon > 0$に対して, $\mathcal{A}_t$がε-最適ではない($V^{\mathcal{A}_t}(s_t)$<$V^*(s_t) - \varepsilon$を満たす)を満たすような時刻tの数である.

2.2.2 サンプル複雑性：モデルベース手法

定義1

アルゴリズム$\mathcal{A}$をMDP $\mathcal{P}$ 上で実行して生成された履歴$c = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots)$があるとする.

アルゴリズムを非定

表し, 状態s_tから方 を

$V^{\mathcal{A}_t}(s_t) = \mathbb{E}[r_t +$

MDPなので当然, 列が生成される

$\mathcal{A}$のサンプル複雑性とは, 任意の$\varepsilon > 0$に対して, $\mathcal{A}_t$がε-最適ではない($V^{\mathcal{A}_t}(s_t) < V^*(s_t) - \varepsilon$を満たす)を満たすような時刻tの数である.

2.2.2 サンプル複雑性：モデルベース手法

定義1

アルゴリズム … 歴$c =$
$(s_1, a_1, r_1, s_2$ …

非定常な方策：
時間により方策が異なるということ

アルゴリズムを非定常な方策と考え, 時刻tでの方策を$\mathcal{A}_t$で表し, 状態s_tから方策$\mathcal{A}_t$を実行したときの期待割引報酬和を$V^{\mathcal{A}_t}(s_t) = \mathbb{E}[r_t + \gamma r_{t+1} + \cdots]$で表す.

$\mathcal{A}$のサンプル … がε-最
適ではない(I … な時
刻tの数であ …

$V^{\mathcal{A}_t}(s_t)$は今までやってきた
V値(V関数)とはちょっと違う
(今までのは定常政策だった)

2.2.2 サンプル複雑性：モデルベース手法

定義1

アルゴリ … 歴$c =$
$(s_1, a_1,$ …
アルゴリ … $\mathcal{A}_t$で
表し, 状 … 酬和を
$V^{\mathcal{A}_t}(s_t)$ …

真に最適な方策よりε以上劣るような方策をとることを「間違い」とする
⇒サンプル複雑性=「間違った」時間(回数)

$\mathcal{A}$のサンプル複雑性とは, 任意の$\varepsilon > 0$に対して, $\mathcal{A}_t$がε-最適ではない($V^{\mathcal{A}_t}(s_t)$<$V^*(s_t) - \varepsilon$を満たす)を満たすような時刻tの数である.

2.2.2 サンプル複雑性：モデルベース手法

定義2

アルゴリズム$\mathcal{A}$がPAC-MDPであるとは, 任意の$\varepsilon > \mathbf{0}$と$\mathbf{0} < \delta < \mathbf{1}$に対し, 関係する量$\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-\gamma}, |S|, |A|\right)$の多項式で表される上界に, $\mathcal{A}$のサンプル複雑性が確率$\mathbf{1} - \boldsymbol{\delta}$で抑えられることをいう.

$|S|$: MDPの状態集合のサイズ

$|A|$: MDPの行動集合のサイズ

γ : 割引率(割引利得で出てきた式)

2.2.2 具体的なアルゴリズム

PAC-MDPであることが示されるアルゴリズム

- E^3
- Rmax

(s,a)をm回経験するまで⇒知らない, 楽観的な価値

(s,a)をm回経験した　　⇒経験からP,Rの推定

⇒動的計画法, 最適行動を選択

mを大きくしないと使い物にならない

2.2.2 モデルベース区間推定(MBIE)

(s,a)に対するP,Rの信頼区間を求める
⇒その信頼区間のなかで最大の価値となるような行動を,
　モデル上の計算で解く
　　⇒価値反復法の単純拡張で！簡単！

2.2.2 その他のアルゴリズム

- 区間推定を行う代わりに, 探索ボーナスを価値に加える
- MoRmax(より低い上界を示す)
- OIM(上界は悪いが, 実用的)

2.2.2 サンプル複雑性：モデルフリー手法

モデルベース手法は, サンプル複雑性は効率的
モデルを保持したうえで近似MDPを多数回解く必要
⇒計算量的には効率的ではない
⇒モデルフリー手法で効率化

2.2.2 Delayed Q-learning

Q-learningの更新時のみ変更

1. 楽観的に価値関数を初期化
2. MDP上で状態遷移
3. Q値の更新

・(s,a)の経験がm回集まっていない ⇒ 更新しない

・(s,a)の経験がm回たまった ⇒ 更新

楽観的な手法！

2.2.2 サンプル複雑性とリグレット上界

サンプル複雑性
⇒学習までにかかる時間の上限を表現
　報酬和にどの程度影響するかはわからない
⇒リグレット上界を計算しよう！

※割引すると, 割引総和は実質有限回で近似できる
⇒問題が簡単になる
※割引しない場合を考える

2.2.2 UCRL2

MDPでのリグレットを求める
⇒UCRL2が強力

MBIEと同様に, モデルベースで信頼区間を推定し, 最も楽観的なものを選択

ただし, **信頼区間の幅**はUCB1と同様に, **試行回数とともに広げていく**

細かいところは割愛！

2.2.3 ベイズ主義的アプローチ

2.2節のまとめ図

強化学習中

探索 ⟷ 利用 ⇒ 価値関数(報酬和)

トレードオフ

評価

評価

探索コストの最悪性能の評価(2.2.1),(2.2.2)

・楽観的初期値評価法(2.2.2)
「悲観的より楽観的に」を表すアルゴリズム

・モデルベースのサンプル複雑性(2.2.2)
探索コストの指標
アルゴリズム : Rmax や MoRmax , OIM

・モデルフリーのサンプル複雑性(2.2.2)
アルゴリズム : Delayed Q-learning

⬇

探索コストの最小化

リグレット上界の評価(2.2.1)

※探索と利用のトレードオフを表す
※報酬和の差

探索コストの最小化(ベイズ主義)

・通常のDP
(環境の知識が既知, 強化学習にする必要なし)

・ベイズ適応的MDP(BAMDP)(2.2.3)
(不確かさを確率的に扱う, POMDPの強化学習へ)

※事前分布がきちんとモデル化できる必要

2.2.3 導入

実際に, "問題が完全に未知である"なんてことはあるか？
エキスパートの知識を利用することは考えられないか？
⇒このような"事前知識"を利用すれば,
　最小限の探索で効率的な学習ができるのでは？

「不確かさ」の確率論 ⇒ ベイズ主義的アプローチ

2.2.3 ベイズ主義的アプローチ

事前確率分布 : 事前知識が含む不確かさ
(パラメータがありそうな範囲)
事後確率分布 : データ観測後の知識
(真の値の近傍に範囲が狭まった確率分布)
共役事前分布 : データ観測過程と対応する形式で
確率分布を表現

事後確率分布は複雑だが, 共役事前分布が表現できれば,
少数のハイパーパラメータで表現できる

2.2.3 探索と利用のトレードオフの本質

探索と利用のトレードオフは, なぜ起こる？

⇒最適な探索と利用の方法を知らないから

⇒なぜ？

⇒環境の情報がわからないから

⇒なぜ？

⇒データが足りないから

つまり, ベイズで補える部分！

2.2.3 ベイジアン強化学習

環境 : k次元のパラメータベクトル$\theta \in \mathbb{R}^k$によって決まる

MDP $\mathcal{P}_\theta$として記述

ベイズ環境モデル : ありうる環境の集合のなかでどれが

ありそうか : $\mathbb{R}^k$上の確率分布$p(\theta)$

ベイジアン強化学習 : ベイズ環境モデル上での強化学習

割引報酬の期待値 : $\mathbb{E}_{p(\theta)}[\sum_{t=0}^{\infty} \gamma^t r_t]$

2.2.3 ベイズ適応MDP(BAMDP)

BAMDPは,

「MDPの学習過程の最適化」を

環境パラメータθと現在の状態sにおいて,

$[\theta, s]$を状態とする新しいMDPを構成

観測できない部分 ⇒ POMDP

「POMDPの方策最適化」問題に帰着させる発想

遷移確率Tは

$$\hat{T}([\theta, s], a, [\theta', s']) = P_{\theta}(s'|s, a) \cdot \delta(\theta, \theta')$$

2.2.3 BAMDPの環境

POMDP(1.5章)は, 観測できない要素θに関する信念を確率分布の形で保持

⇒BAMDPをPOMDPに変換すると,

θ上の信念が, 環境の事前分布･事後分布に相当！

しかも, 事前分布$p(\theta)$に相当する信念を初期状態とする

⇒事後分布$p(\theta|s_1, a_1, \ldots, s_t, a_t)$は信念状態を用いて導出可能

POMDP上で学習⇒最適方策を得ることができる！

2.2.3 BAMDPにおけるトレードオフ

BAMDPは**オフライン学習**

(今までのアルゴリズムはオンライン学習)

⇒環境上で**試行錯誤なしに**方策を求めるアプローチ

⇒**探索と利用を行わず**, 最適解を求めることができる！

⇒**探索と利用のトレードオフの最適解**

2.2.3 最適な理由

<なぜ？>

ベイズ主義アプローチ

⇒生じうるすべての可能性と, 観測後の不確かさを,
　事前分布と事後分布で明示的に表現できるから！

⇒決して戻れない可能性がある問題の探索でも, 探索なしに
　状態がわかる！

⇒最適な探索が実現

2.2.3 注意点

POMDPの厳密解を求める計算量⇒指数オーダー
⇒近似しなきゃ使い物にならない
　しかも, PAC-MDPを満たさない場合も･･･

<近似方法>
- ベイズ環境モデルの共役分布表現を直接利用
- 環境モデルをサンプリングする手法
- モンテカルロ木探索法を利用

2.2.3 ベイズ環境モデルの共役分布表現を直接利用

- BEETLEアルゴリズム

信念空間上の価値関数の構成要素を, 適切に選んだ基底関数の線形和で近似することにより, POMDPソルバを使える形式に帰着

※近似精度は基底関数とPOMDPソルバによる

2.2.3 BVR(Bounded Variance Reward)

- BVRアルゴリズム

事後分布から計算される報酬･遷移確率の分散に応じて, 探索ボーナスを与える

信念発展の木構造を探索せずに済ませる考え方

2.2.3 PAC-MDPとBAMDP

PAC-MDPは, 事前分布に表現されている知識をどれだけ活用できるか, といった点についてはうまく表現できない

(PAC-MDPはMDPに対するアルゴリズムの効率指標)

⇒真のBAMDPの解との近さを評価する**PAC-BAMDP**

2.2.3 PAC-MDPとPAC-BAMDP

- PAC-MDP

⇒学習完了後の方策と,
アルゴリズムにより得る探索を含めた方策の差を見る
⇒探索の分, 決して0にならない

- PAC-BAMDP

⇒直接計算が困難なBAMDPの最適方策解を
多項式計算時間で近似する精度を評価
⇒ゼロに近づけることができる

2.2.3 BOLT(Bayesian Optimistic Local Transitions)

- BOLTアルゴリズム

各状態から任意の都合のいい状態にη回遷移したという観測
⇒事後分布がわかる
⇒予測分布を計算
⇒遷移確率の楽観的な信頼区間の上限を得る

※PAC-BAMDPを満たす
※探索ボーナスの計算がOIMアルゴリズム(2.2.2)と近い

2.2.3 環境モデルのサンプリングに基づく手法

ベイズ主義的機械学習

⇒モンテカルロ法により, 事後確率分布から抽出した

サンプルを利用してよい近似を実現

⇒ベイジアン強化学習にも使えないか？

※複雑な事後分布 ⇒ 共役分布が書けない

※サンプリングはどんな事後分布でも実現可能

※サンプリング結果は, 動的計画法(DP)で取り扱える

⇒シンプルなアルゴリズム

2.2.3 Bayesian DP

- Bayesian DP

MDPのパラメータθを事前分布から1つサンプリング
⇒そのθで表されるMDP $\mathcal{P}_\theta$の最適方策をDPで計算
⇒その方策でNステップ行動
⇒経験から事後分布を使って新たなθをサンプリング
⇒…(繰り返し)

※Thompsonサンプリング(2.2.1)をMDPに拡張
※探索コストは比較的大きい

2.2.3 MBBE(Model-Based Bayesian Exploration)

- **MBBE**

n組のMDPパラメータ$\theta_1, \dots, \theta_n$をサンプリング
⇒得られたn個のMDP $\mathcal{P}_{\theta_1}, \dots, \mathcal{P}_{\theta_n}$をDPで解く
⇒各(s,a)ペアに対するQ値の分布の幅を計算
⇒各(s,a)ペアについて情報を収集する価値を計算
⇒Q値の期待値の和を最大化する行動を選択

※探索コストの上限不明, 計算コスト減少のために複雑
⇒比較が難しい

2.2.3 BOSS(Best of Sampled Sets)

- BOSSアルゴリズム

n組のMDPパラメータ$\theta_1, \dots, \theta_n$をサンプリング
⇒n個のMDPを合成した合成MDPを作成
⇒その合成MDP上で最適解を考える(楽観的)

※MDP上の各状態で, エージェントがn個の世界(MDP)から
　好きな世界を選んで行動できるという考え方
※一定量の経験を集めることで, 事後分布から
　再サンプリングして探索範囲を狭める

2.2.3 BOSSのメリット･デメリット

<メリット>

- PAC-BAMDPは証明されていないが,
  Rmaxと同等の上界を持つPAC-MDPと証明されている
- 複雑な事前確率を扱え, 実装が簡単で, 計算コストも大きくない

<デメリット>

- 各状態の楽観的見積もりを混合して探索
  ⇒問題によっては楽観的になりすぎる

2.2.3 MC-BRL(Monte-Carlo Bayesian RL)

- MC-BRLアルゴリズム

n組のMDPパラメータ$\theta_1, \ldots, \theta_n$をサンプリング
⇒どれか1つが正しいと考え, 観測できないn個の世界の
　インデックスでMDPの状態を拡張したPOMDPを作る
⇒POMDPソルバで解く

※楽観主義原理は使っていない
※事前分布のサンプルを使ってBAMDPを直接近似

2.2.3 MC-BRLのメリット·デメリット

<メリット>

• PAC-BAMDPを満たす

<デメリット>

• パラメータ次元数が大きくなると多数のサンプルが必要

⇒拡張したPOMDPが複雑になる

⇒ソルバの計算時間が増大

※BOSSのように最サンプリングを組み合わせるとよい？

2.2.3 モンテカルロ木探索法

サンプリングを使う別の方法

⇒BAMDPの探索木上のパスをサンプリング

※BAMDPの困難は,

　指数的にノードが増える信念木の探索

　⇒信念木の空間をサンプリングで代用

2.2.3 Sparse Sampling

状態数の多いMDPに対して, 価値の評価のために,
ランダムシミュレーションによる疎なサンプルの
平均を用いる学習手法
(ひとつ前のスライドと類似！)
⇒計算コストの問題から一般的にはならなかった

⇒UCT(高効率なモンテカルロ木探索をできるアルゴリズム)により, 応用可能に？

2.2.3 Sparse Sampling

状態数の多いMDPに対して, 価値の評価のために,
ランダムシミュレーションによる疎なサンプルの
平均を用いる学習手法
(ひとつ前のスライドと類似！)
⇒計算コストの問題から一般的にはならなかった

⇒高効率なモンテカルロ木探索をできる
アルゴリズムであるUCTなど,
基礎研究が発展してきた

2.2.3 FSSS(Forward Search Sparse Sampling)

UCTはPOMDPにおける信念木の探索に応用
⇒MDPの方策選択に適用すると,
　最適解収束までの最悪計算時間が超指数関数的

- FSSSアルゴリズム

UCB1のかわりに漸近的にSparse Samplingに到達する
木探索アルゴリズムを利用

2.2.3 BFS3(Bayesian FSSS)

- BFS3

FSSSを使ってベイジアン強化学習

PAC-MDPである

2.2.3 ベイジアン強化学習の限界

- ベイズ主義的手法は, 事前分布で問題を適切に表現できない場合は効果的ではない！
  ⇒この場合はモデルフリーな手法で…
- 割引なしのケースをうまく扱う手法は見つかっていない
  ⇒リグレット上界が扱えない

しかし, 事前知識を与える発想は効率的な計算には大事！

2.2.4 おわりに

2.2節のまとめ図

強化学習中

探索 ⟷ 利用 ⇒ 価値関数(報酬和)

トレードオフ

評価 ↓ 評価 ↓

探索コストの最悪性能の評価(2.2.1),(2.2.2)

・楽観的初期値評価法(2.2.2)
「悲観的より楽観的に」を表すアルゴリズム

・モデルベースのサンプル複雑性(2.2.2)
探索コストの指標
アルゴリズム : Rmax や MoRmax , OIM

・モデルフリーのサンプル複雑性(2.2.2)
アルゴリズム : Delayed Q-learning

↓

探索コストの最小化

リグレット上界の評価(2.2.1)

※探索と利用のトレードオフを表す
※報酬和の差

探索コストの最小化(ベイズ主義)

・通常のDP
(環境の知識が既知, 強化学習にする必要なし)

・ベイズ適応的MDP(BAMDP)(2.2.3)
(不確かさを確率的に扱う, POMDPの強化学習へ)

※事前分布がきちんとモデル化できる必要

2.2.4 キーワードまとめ

- バンディット問題(UCB, UCT, Thompson)
- 探索と利用のトレードオフ(楽観的初期評価法)
- モデルベース手法のサンプル複雑性

  ⇒評価指標 : PAC-MDP

  ⇒アルゴリズム : E^3, Rmax ,MBIE ,MoRmax ,OIM
- モデルフリー手法のサンプル複雑性

  ⇒Delayed Q-learning
- リグレット上界(UCRL2)

2.2.4 キーワードまとめ

- ベイズ適応的MDP(BAMDP)

  ・BAMDP解の近似精度

  ⇒評価指標 : PAC-BAMDP

  ・共役分布表現を直接利用する方法

  (BEETLE, BVR, BOLT)

  ・環境モデルのサンプリングに基づく手法

  (Bayesian MDP, MBBE, BOSS, MC-BRL)

  ・モンテカルロ木探索

  (FSSS, BFS3)