科学哲学科学史特殊講義（京都大学）

補足3：強化学習

鈴木貴之

（東京大学大学院総合文化研究科）

tkykszk@g.ecc.u-tokyo.ac.jp

マルコフ決定問題：

- 行動が非決定論的な状況で、最善の方策（policy）を発見する。
- エージェントは行為aによるsからs'への遷移によって報酬$R(s,a,s)$を受け取るとする。
- 問題は、行為の集合$A(s)$、遷移モデル$P(s'|s,a)$、報酬関数$R(s,a,s')$によって定義される。

マルコフ決定問題の例

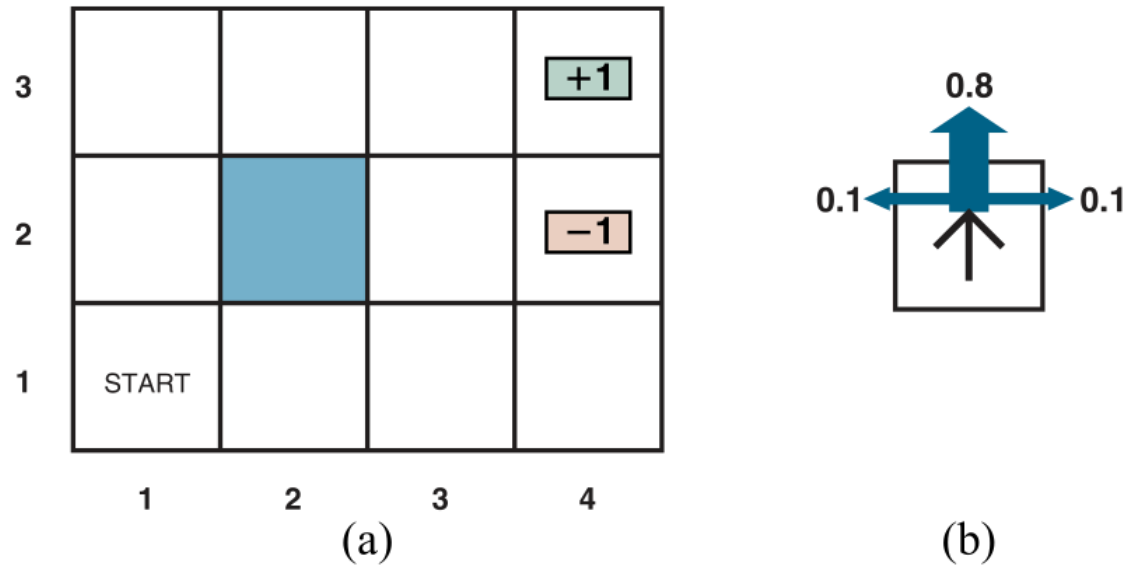


Figure 17.1 (a) A simple, stochastic 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. Transitions into the two terminal states have reward +1 and –1, respectively, and all other transitions have a reward of –0.04.

2つの問題：

- 予測問題：効用関数（各状態の価値を表す関数）を推定する。
- 制御問題：最適方策（行動方針）を発見する。

この問題における最適方策

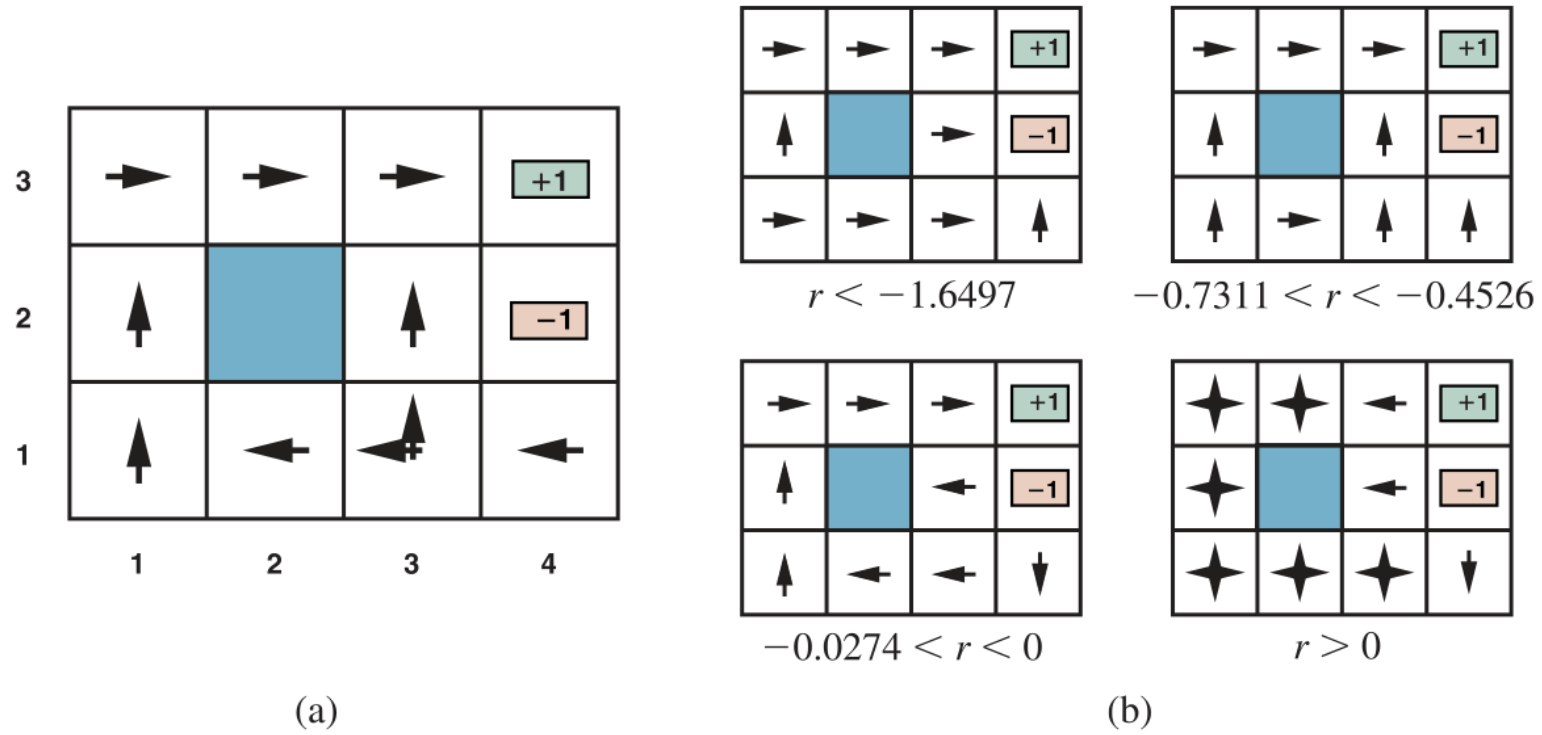


Figure 17.2 (a) The optimal policies for the stochastic environment with $r = -0.04$ for transitions between nonterminal states. There are two policies because in state (3,1) both *Left* and *Up* are optimal. (b) Optimal policies for four different ranges of r.

2つの困難：

- スパースなフィードバック→**信頼割り当て問題（credit assignment problem）**
- 相対的なフィードバック→**活用（exploitation）と探索（exploration）の問題**

- ある状態の効用$U(s)$は、その状態から出発し、最適方策を実行した際の報酬の総和の期待値として定義できる。
- 最適方策は、正しい効用関数によって決定できる。

  →効用関数の推定と最適方策の推定を繰り返すことで、両者を特定できる。

強化学習（reinforcement learning）：

- 遷移モデルや報酬関数が未知である場合に、方策を学習する。

強化学習がAI研究において重要な理由：

- チェスでは、グランドマスターの手を訓練データに利用することも可能だが、利用できるデータは可能な盤面状況に対するごく一部でしかない。
- 実世界の問題では、この制約はより深刻なものとなる。
- 報酬信号を与えることは、個々の状況における正解を与えることよりも容易。

強化学習の分類：

- モデルベースの強化学習
  - 遷移モデルを利用する。
- モデルフリーの強化学習
  - 遷移モデルを利用しない。

時間的差分（TD）学習：

- 行為の結果得られた報酬の値と予測値との誤差に基づいて効用関数をアップデートする。
- $U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha[R(s, \pi(s), s') + \gamma U^{\pi}(s') - U^{\pi}(s)]$
- 例：バスによる移動

Q−学習：

- 行為の価値を表す関数（Q−関数）を直接学習する。
- $Q(s,a)$は、エージェントがsにおいて行為aをし、その後は最適に行為する場合の報酬の総和の期待値。
- Q関数に関してもTD更新式が得られる。
- $Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a,s') + \gamma \max_{a'}(s',a') - Q(s,a)]$

活用(exploitation)と探索(exploration)のトレードオフ：

- 遷移モデル、報酬モデルが不明だとすれば、現在最適だと思われる方策は、じつは最適ではないかもしれない。
  - スロットマシンの例
- エージェントは、一定の頻度で未知の状態を探索する必要がある。

評価関数の導入：

- 状態空間が巨大な場合には、すべての状態を訪問することは困難。
- 効用関数の近似として、評価関数を導入する必要がある。
- 評価関数は、教師あり学習によって学習可能。

深層強化学習：

- 効用関数やQ−関数は、線形関数ではうまく近似できないかもしれない。
- そのような場合には、深層ニューラルネットワークで評価関数を表現すればよい。
- ただし、この手法はシステムの挙動が必ずしも安定しないため、商業的な利用は困難。

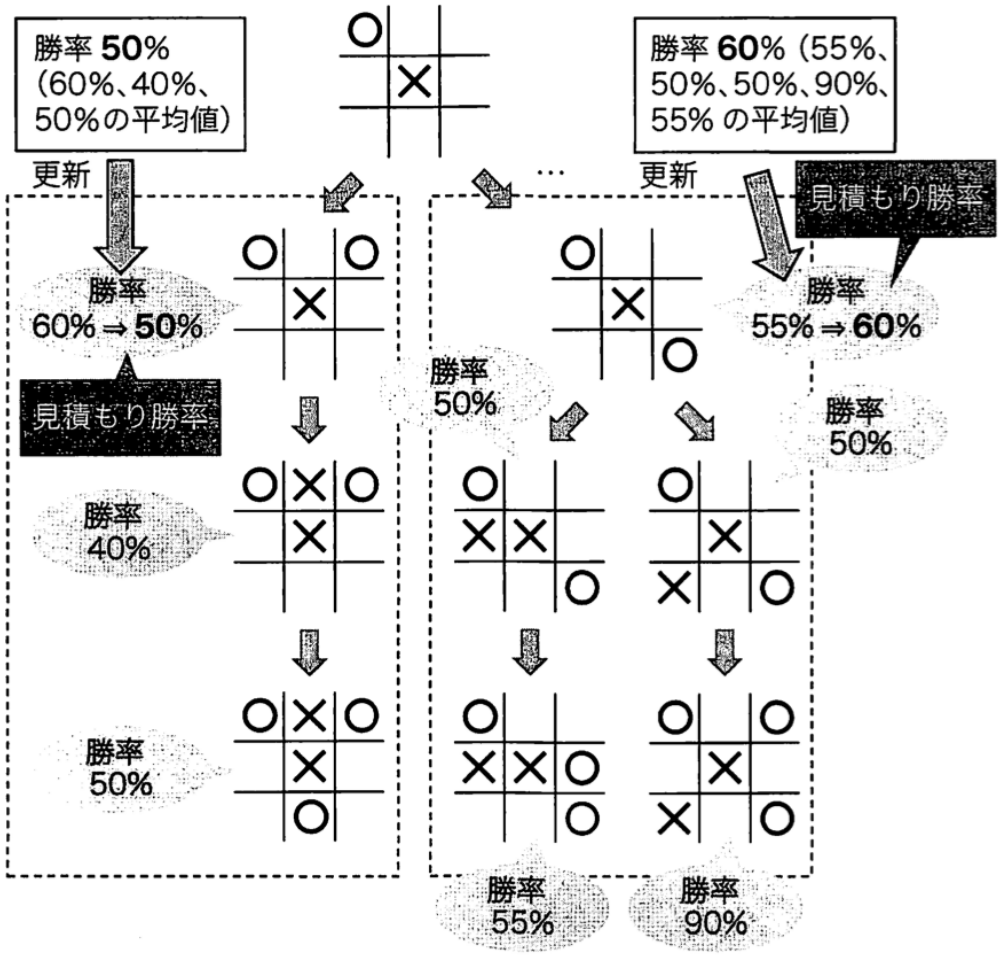
DeepMind：

- 深層Q-学習：深層ニューラルネットワークを用いてQ-関数を表現。
- Atariのビデオゲームの多くをエキスパートレベルでプレイ可能。
- ただし、報酬がスパースなゲームでは学習が困難。

AlphaGo：

- 評価関数を学習し、どの手が探索に値するかを予測する。
  - 探索を進めたあとの評価値によって評価値を調整する。
  - 探索回数が少ない手は優先して探索する。
- AlphaGo Zeroは自己対戦で学習。
- AlphaZeroはAlphaGo Zeroの手法を一般化。

AlphaGoによる盤面の評価



考察：

- 強化学習には数多くの試行が必要。
  - 現実世界での試行錯誤が必要な課題では利用が困難。
  - 試行の失敗が致命的となる場合にも利用が困難。

- ある程度の頻度で報酬を得られることも必要。
  - 報酬が未知の課題には適用できない。
- ゲームは強化学習に適した題材だと考えられる。
  - 行為や報酬が明確
  - 文脈が明確に限定されている