連 載

—オータムスクール ASCONE06 脳科学への数理的アプローチ—

「神経経済学入門I：自然界で脳はどのように最良な行動を選択していくのだろうか？」

玉川大学工学部 酒井裕先生 講義録 (@ ASCONE 2006 11/25)
講義録担当者 東京大学大学院新領域創成科学研究科 学振 PD 三浦佳二

この講義録は Autumn School for Computational Neuroscience (ASCONE2006) における酒井先生の講義を文章にしたものです。この講義は神経経済学をテーマとして ASCONE2006 の第3日目の午前に行われたもので、同日午後の鮫島先生の講義に含まれる強化学習とゲーム理論の準備も兼ねていました。なお、講義を録音したものを文章に直す際に、読みやすくするために文章を修正している点がありますので、ご了承ください。

1. 前半の講義

自然界における行動選択

経済学というのは、各々が、各自の利益を求める行動を選択していった結果、全体としてどういうことが起こるか、ということを考える学問です。神経経済学というのはその経済学に神経科学・脳科学の知見を応用していくものです。今日はそこまでは全然行かなくて、その一番根本ですね、各自がどうやって行動を選択しているのか、というところだけにスポットをあてた、本当に初歩の神経経済学の入門を行います。今日第一部で、私の話は、自然界で動物がどのように行動を選択しているか、という話をします。第二部の鮫島さんの話との違いは、競合する相手を明確に考えるかどうか、という点にあります。私の話では、相手は自然という漠然としたものを想定しています。明確な相手がいて戦っているという状況ではありません。第二部の鮫島さんの話で、相手がいるという状況を考えることになると思います。

日々、我々は、いろんなものを選択します。例えば昼飯に何を食べようかと思って、まあ、チェーンの牛丼屋に行こうかと考えます。これはまずくはないですが大してうまくないこともわかっています。安いこともわかっています。早いこともわかっています。で、かたや一方、自分の好きな有名なラーメン屋さんにも行きたいとします。相当うまいことはわかっています。ただし、行ったらすごく並んでいるかもしれない。こういう状況で、どっちを選ぼう、ということを日々考えるわけですね。ラーメン屋に行ってみてどのくらい

並んでいるのかは確率的ですね。行ってみなければわかりません。その確率は、ある程度試してみないとわかりません。我々の選択の結果というのは、大抵の場合いつも確率的なんです。選択の結果は確率的であり、その確率は未知である、という状況で、我々は試行錯誤を通して、どう選択すればいいのか、どうするのが得かを学習していかなければなりません。

人間の行動はとても複雑でいろんな事前知識を利用した行動選択を行っています。自然界の動物の基本的な行動をとってみますと、比較的単純です。たとえば、肉食動物の場合を考えます。雌ライオンが、ウサギとシマウマを目の前にしました。さて、どっちを狙いましょうか？ どっちかを狙えば、どっちかは逃げて行ってしまいます。その成功率と報酬の量は次の通りとします。

- ウサギだったら、20 kg で成功率 50%
- シマウマだったら、200 kg で成功率 1%

ここでどういう選択をしたら得でしょうか？ 期待値を計算すれば、ウサギは 10 kg でシマウマは 2 kg なので、そういう意味ではウサギの方が良さそうです。ウサギをずっと選択し続けるのが得です。一般化すると、報酬期待値（報酬量 × 報酬確率）が最大になる行動を選び続けるというのが、最適な行動です。試行錯誤を通して、最終的に報酬期待値が最大になる行動を選び続けるようになれば、最適な行動になっているということです。これが、肉食の餌採りの場合を単純化した例です。このタイプの報酬の与え方を「VR スケジュール」と呼んでいます (Mazur, 2005)。VR という名前の由来は説明しませんが、名前だけ覚えてくだ

さい。報酬の与え方を「強化スケジュール」と呼びます。その与え方によって、「〇〇スケジュール」という名前がついています。「強化」という言葉の由来は、ある行動に報酬を与えることで、その行動の選択を強化する、ということからきています。強化スケジュールというのは、報酬の与え方だと思って下さい。どんな選択をしたら、どういうふうに与えるか、その与え方が強化スケジュールです。また、強化をされる環境で学習することを「強化学習」と呼んでいます。

では次に、**草食動物の場合**を考えます。シマウマが草の生えている牧草地を探すという状況を考えましょう。西に行こうか、東に行こうか、どちらかを選ぶものとなります。

- 西の牧草地は、草が生える頻度は、確率的で平均1日ぐらいかかる
- 東の牧草地は、草が生える頻度は、確率的で平均2.5日ぐらいかかる

生えないこともあれば、生えることもあります。一日に、片方しか行けません。どう選択すると最適でしょうか？行かないうちに、餌が生えることもあれば、生えてないこともあります。行ったときに、草があれば食べられるということです。どちらかを選んだときに、もう片方の選んでなかった方に草が生えていれば、草は残り続けます。そういう状況です。この草食タイプの強化スケジュールを「VI スケジュール」と呼びます。この状況でどうしますか？生える頻度が高いからと言って、西に行き続けるのは、得ではありません。片方に行けば行くほど、もう片方に草が生えている確率は高くなります。ですから、ある程度日にちが経ったら、もう一方に行くのが得ですね。この生える確率とその量に応じて、万遍なく選択するというのが最適です。どのような行動パターンが最適か解いていくと、ある周期的な選び方が最適になることがわかります。先ほどの例でいうと、例えば、2回西に行って、東に1回行くというのを繰り返すのが、最適になります。ただし、動物は交互に選ぶという行動パターンはよく示すのですが、最適な行動パターンの周期が複雑になる場合には、なかなか最適に辿り着かないことが知られています。

今度は、**定常的な競合者がいる場合** (Sakai and Fukai, in press) を考えます。先ほど、一度生えた草は食べに行かなければ残ると言いましたが、競合者がいて、あまりにも草をとりにつかないと、そのうち、他のシマウマに（ある一定の確率で）食われてしまうという状況を考えます。生える頻度と、食べられてしまう頻度によって決まる課題となります。この課題を「競

合的採餌課題」と呼ぶことにします。この状況でも、ある程度のバランスで満遍なく食べに行くのが、最適です。これも、実は、理論的に解いてみると、周期的な行動パターンが最適となります。この課題は、先ほどの草食タイプに、取らなかったら取り除く確率というのを入れるだけです。ちなみに、この課題は私が考えたのですが、今までこんなこと誰も考えなかったのが不思議です。調べてみましたが、誰もやっていないようです。この課題が良いことは、この取り除く確率を0にすると草食タイプに一致して、取り除く確率を1にするとちょうど肉食タイプにできます。肉食タイプと草食タイプの間を連続的に埋めることができるのが利点です。

マッチングの法則

肉食タイプの課題は一番簡単で、動物は長くやれば最適行動に到達できます。ところがもう少し難しくしていくと、なかなか最適にならないことが知られています。難しい強化スケジュールでは、報酬最大化に至らない、ということです。ところが、そのようなとき、ある法則を満たすということが知られています。それが、「**マッチングの法則**」という法則です。Herrnstein がだいたい昔に1961年くらいからハトの実験でこの法則を見つけました (Herrnstein, 1961; Herrnstein, 1997)。このマッチングの法則は、十分学習して定常状態になった後の行動データを解析して得られた法則です。十分学習した後、ある長い期間の選択の頻度を調べると、「**選択頻度 = その選択で得た累積報酬量の割合**」となることを見出されました。つまり、ある期間に選択肢1を選んだ回数、選択肢2を選んだ回数、・・・の割合が、選択肢1からもらった報酬量、選択肢2からもらった報酬量、・・・の割合に一致するというのが、これが、マッチングの法則です (図1)。先ほどのいろんな課題を考えてみても、こんな行動が最適とは全然言えないわけです。全く最適とは直接関係のない法則です。しかし、動物はこういう法則を示してしまうのです。

もう一度行動を定義しておきます。「最適行動」というのは最も報酬が平均的に獲得できるという選択方法です。1回ごとの選択に限らず、先ほどのような周期

図1 マッチングの法則

的な選択パターンも含めてです。最適な選択方法は必ず1つ存在します。一方、「マッチング行動」というのは、マッチングの法則を満たす行動のことです。一般に、こちらの場合は、法則さえ満たせばいいので、複数存在することができます。必ず複数存在することがわかっています。具体的には、例えば、ある行動しか選ばないということをやっていると、その選択肢の割合が100%になって、そこから貰う報酬も100%になります。したがってマッチングの法則を満たしています。このような行動パターンは、各選択肢についてありますから、マッチングの法則が成り立つような行動パターンは、複数個存在することは間違いないですね。このような極端な例だけでなく、マッチングの法則を満たす選択頻度の比率は複数解存在する場合があります。さきほどの肉食タイプの課題では、最適行動がたまたまマッチング行動のうちの一つに一致しています。最適行動がマッチングの法則を満たす例になっています。課題によって、最適行動がマッチング行動になっていることがありますが、一般には異なります。

図2が最初に Herrnstein (1961) が発見したハトの行動の例です。右のレバーか左のレバーを選ぶ課題です。横軸が右を選択して得た報酬の割合、縦軸は右を選択した割合です。いろんなパラメタで実験してみると、この対角線の上ののっています。この直線上がマッチングの法則が成り立つところになります。3つの線は、3羽のハトの結果です。これは、先ほどの草食のタイプで実験した結果で、草が生える確率、つまりここでは報酬をセットする確率をいろいろ変えると、報酬の確率を変えてもマッチングを満たす線上ののっています。もうちょっとダイナミックにみた性質が図3で、最近の Newsome のグループの実験ですが (Sugrue et al., 2004), 同じく草食タイプの課題を、サルにさせた実験結果です。横軸に、片方の選択肢の累積選択回数、縦軸にもう片方の累積選択回数をとって累積値の経過を示してあります。そうすると、ちょうど上がっていく傾きが、選択比率になっています。累積値を使うので、ゆらぎが見えにくくなり、結果が見やすくなります。ここの黒い線は、報酬の比率に対応する傾きを示しています。この傾きで累積選択回数が上がって行けば、マッチングの法則が成り立っていることになります。大体、かなり、マッチングになっていることがわかります。この辺はちょっとずれてますけれどね。このような表示の仕方もあります。後で皆さんにデモをやってもらおう表示はこのタイプを使っていますので、見方を覚えておいてください。

図2 ハトの行動にみられたマッチングの法則 (Herrnstein 1961 より再現)

図3 草食タイプの課題で得られたサルのマッチング行動 (Sugrue et al. 2004) を Actor-Critic で再現した結果。

歴史的な議論～マッチングか？ 最大化か？

マッチングがなんらかの意味で最適になっているのではないとか、ある意味の最適行動と近いのではないとかの議論がずっとあります (Baum and Rachlin, 1969; Herrnstein and Heyman, 1979; Mazur, 1981; Vaughan, 1981; DeCarlo, 1985)。最適行動とマッチング行動が異なるような課題を使ったいろんな実験があります。例えば、先ほどの草食タイプの場合で東西の報酬量を変えないとすると、実はマッチング行動と最適行動の選択頻度は非常に近い値になり、あまり区別がつかせん。ところが報酬量を大きく変えてやると、マッチング行動と最適行動が大きく異なるようにすることが知られています。ただし、自分の中で報酬量

をどうスケールしているのか、わからないんじゃないかという反論があって、この課題では決着がつきませんでした。そこで、先ほどの草食タイプと肉食タイプを組み合わせて、Aの選択肢を草食タイプにして、Bの選択肢を肉食タイプにした課題が提案されました。この場合、与える報酬量が同じでも、最適とマッチングが異なることが知られていて、この課題をハトに行わせた結果、やはりマッチングになるという結果が出ています。これも選択をスイッチするコストなどを考えると最適とみなせる、などという、いろいろないちゃもんがついてなかなか決着がつきませんでした。他にもいろいろ課題が提案されていますが、Vaughan (1981) が提案した課題が一番強力で、おそらく、反論の余地がない課題でしょう。Vaughan 課題でもマッチングになるという結果が出ています。こうした様々な課題での結果から、どうやら動物は、報酬最大化できないのにも関わらず、マッチング行動を示すということがわかってきています。これはいったい何故なのでしょう。別にマッチングというのが生命の目的になるはずがないですね。報酬を最大化したいにきまっています。なのに、なぜか最大化が難しいとマッチングになってしまうようなのです。これが一体なぜなのか、という疑問を、今日の演習の課題にしたいと思います。

マッチングの法則の別形式

マッチングの法則をちゃんと式で書いてみると、このことです。

- N_a : a という選択肢を選択した回数
- R_a : a から得た累積報酬量

というふうに定義しておきますと、一般に2択だけじゃなくて、多択の場合に一般化するとマッチングの法則は、こうなります。

$$\frac{N_a}{\sum_{a'=1}^n N_{a'}} = \frac{R_a}{\sum_{a'=1}^n R_{a'}}$$

この式がすべての a について成り立つ、というのがマッチングの法則になっています。ここで、もう1つ別の量を定義しておきましょう。

- Q_a : a を選んだ時の報酬期待値

肉食タイプの場合は簡単で、報酬量と報酬確率をかければ Q_a になります。他の課題ではもっと違ったものになります。累積報酬量は報酬期待値 × 選択回数と平均的に等しくなりますから、 $R_a = Q_a N_a$ を上の式に代入してみましょう。すると、この式を変形していくと、

- $N_a = 0$ または、 $Q_a = \frac{\sum_{a'=1}^n R_{a'}}{\sum_{a'=1}^n N_{a'}} \equiv V$

となります。右辺は、全ての選択肢からきた報酬を全体の選んだ回数で割ったもので、選択一回あたりの平均の報酬になっています。これを V と定義しておきます。右辺が a に依らない式であることに注意してください。これをまとめると、選択頻度が0でない選択肢、選んでいない選択肢は、必ず報酬期待値が等しくなっているという状態になっているということです。これがマッチングの法則の別の表現になっています。

マッチング行動を導く学習アルゴリズム

このマッチングの法則を満たすように設計された学習則があります。Herrnstein & Vaughan (1980) が提案した方法が最も古く、「Melioration 法 (逐次改良法)」と呼ばれています。これは、先ほど変形したマッチングの法則を満たすように作られています。

1. 報酬期待値 Q_a を何らかの方法で推定
2. 推定した Q_a がより大きい a の選択確率 p_a を上げる

ということを言っています。なぜこれでマッチングの法則を満たすかという、たいいてい選択をたくさんすればするほど報酬期待値が小さくなるような課題を組んでいますね。Herrnstein がやっている実験では少なくともそうになっています。そうじゃない例も、もちろん考えられますけれども、彼らはそういう課題のデザインをしていますので、そのような課題においては、 Q_a が大きい選択肢の選択確率を上げると、 Q_a が小さくなりますので、 Q_a が皆等しくなるところで、定常状態に達し、マッチングになります。ちなみに、Herrnstein & Vaughan (1980) は具体的なアルゴリズムは一切挙げてないですが、その具体的なアルゴリズムをもしこれをもとに考えると、まあ例えば、こんな式になります。 Q_a を推定するにはこんな更新をすればいいです。

$$Q_{a_t} \leftarrow Q_{a_t} + \alpha(r_t - Q_{a_t})$$

何をしているかという、今選んだ選択肢を a_t とします。今選んだ選択肢の報酬期待値を表している Q_{a_t} 、それを $\alpha(r_t - Q_{a_t})$ だけ変更しようということです。そうすると、だいたい定常状態になると、この変化分は平均的にゼロになりますから、この Q_a の平均値と a を選択したときの報酬の平均値というのが等しくなると、これが推定できるということになります。これを使ってどうやって選択確率を変化させるかというと、例えば、最大の Q_a を与える選択肢 $a_{\max}$ の選択確率を上げましょう。確率が1をこえないように、 $1-p$ をかけます。

$$p_{a_{\max}} \leftarrow p_{a_{\max}} + \alpha(1 - p_{a_{\max}})$$

こんな方法が考えられます。Melioration を実現するアルゴリズムはこれ以外にもいろいろ作ることができます。

他には、先ほどのサルの実験を行った Sugrue et al. (2004) が導入した「局所マッチング則」が挙げられます。これはとても直接的です。過去一定期間の各選択肢の平均報酬を推定して、直接次の選択確率を決めます。

1. 各選択肢から過去にどれだけ報酬をもらったかを直接計算し
2. その比率 p_a で選択肢をランダムに選ぶ

各選択肢からもらった平均獲得報酬からどういうふう
に推定するかというと、例えばこんな形になります。

$$R_a \leftarrow R_a + \alpha(r_t \delta_{aa_t} - R_a)$$

先ほどとの違いは、今選択した選択肢だけでなく、ほかの R_a も更新します。その代わり、今選択したもののだけ、今もらった報酬をつけます。 δ_{aa_t} というのは a と a_t が等しい時 1, そうでないとき 0 という記号です。今選択した R_a を更新するときは、ここに報酬をいれて、そうでないときここは 0 になります。そして、この R_a をつかって直接この割合で選択確率をきめていきます。

$$p_a = R_a / \sum_{a'} R_{a'}$$

これが局所マッチング則ですね。これも直接マッチング則が成り立つように作っていますから、定常状態でマッチング則が成り立つようになっていますね。

今度は、全然違う方向から提案されてきた意思決定アルゴリズムの 1 つで、これは強化学習理論 (Sutton and Barto, 1998; Dayan and Abbott, 2001) の中に含まれています。これは、歴史的には、マッチング則とは全く無関係で、工学的に応用するために提案された方法です。一番標準的な「Actor-Critic 法」の例を挙げてみます。Actor-Critic 法とは何かというと、Actor というのは実際に行動するものという意味で、Critic は評価をするものという意味です。まず、Critic の方は、過去一定期間のトータルの平均の獲得報酬の推定をします。

$$V \leftarrow V + \alpha(r_t - V)$$

先ほどの R_a の足し算になっています。その推定誤差を使ってポリシーパラメータ q_a というものを更新します。

$$q_{a_t} \leftarrow q_{a_t} + \alpha(r_t - V)$$

ポリシーパラメータ q_a は選択確率を決定するためのパラメータです。普通は、この q_a を使って、選択確率をこのように決めます。

$$p_a = e^{\beta q_a} / \sum_{a'} e^{\beta q_{a'}}$$

これなぜ便利かというと、 q_a がどんな値を取っても確率の性質を必ず満たすからです。 p_a を全部足すと必ず 1 になります。しかもこれ、指数関数って負にならないですね。確率は必ず正になります。そういう制約条件を満たすために、こんな変数の置き換えをしているわけです。このような置き方をしますと、確率が満たすべき条件を考えずに q_a を自由に更新していいですね。この V は、選択肢ごとではなく、トータルでどれだけ平均的に報酬をもらっているかを表す量ですね。これも先ほどのような解析をすると、もし、定常状態になったとすると、この誤差の部分が 0 になります。したがって V の値は平均的な報酬と一致します。ここの誤差の部分は、現時点での推定誤差を表しています。その同じ推定誤差を使って、ポリシーパラメータを更新するというのがこの Actor-Critic の方法になります。で、実はこれ計算してみると、最適行動ではなくマッチング行動になることが証明できます。これも何故かということ、皆さん演習で考えてください。

他にも強化学習アルゴリズムで、「Direct-Actor」というのがあります。なぜダイレクトかというと、報酬期待値などを推定せずに、とにかくこんなアルゴリズムで、直接ポリシーパラメータを更新していくという方法です。

$$q_a \leftarrow q_a + \alpha r_t (\delta_{aa_t} - p_a)$$

で、これもマッチング行動を示します。

マッチング行動に至らない強化学習アルゴリズム

もう 1 つ、Actor-Critic と並んで有名な強化学習アルゴリズムに「Q-Learning」があります。Q-Learning は何かといいますと、報酬期待値 Q_a を推定し、その報酬期待値を用いて直接、選択確率を決める、という方法です。その決め方は、先ほどのポリシーパラメータで使ったようなやり方で、直接決めます。

$$Q_{a_t} \leftarrow Q_{a_t} + \alpha(r_t - Q_{a_t})$$

$$p_a = e^{\beta Q_a} / \sum_{a'} e^{\beta Q_{a'}}$$

このアルゴリズムでは、実は、マッチングの法則を示さないことがわかります。

他にも、マッチングにならない例があります。これは、原則としては Actor-Critic に入るアルゴリズムで

図4 様々な学習則のシミュレーション．Q-learning と特殊な Actor-Critic 法は、マッチング則にも最適にもなっていない．それ以外の4つはマッチング則を満たす．

ですが、少し「特殊な Actor-Critic 法」です．トータルの平均報酬を推定し、その推定誤差 $\delta = r_t - V$ を使って、選択確率を変更している、という Actor-Critic の原則には従っています．ただし、二択の問題だけに特化して、片方の確率 p_1 だけをパラメタにすることにします．

$$V \leftarrow V + \alpha(r_t - V)$$

$$p_1 \leftarrow \begin{cases} p_1 + \alpha(1 - p_1)\delta & (a_t = 1 \& \delta > 0) \\ p_1 + \alpha p_1 \delta & (a_t = 1 \& \delta \leq 0) \\ p_1 - \alpha p_1 \delta & (a_t = 2 \& \delta > 0) \\ p_1 - \alpha(1 - p_1)\delta & (a_t = 2 \& \delta \leq 0) \end{cases}$$

確率を増やしたい時には、 $1 - p_1$ をかけて、確率が1をこえないようにします．確率を減らしたいときには、 p_1 をかけて、0より小さくならないようにします．まあよく使われる方法なのですが、こういう設計の仕方をする、Actor-Critic でもマッチング行動にはならないということがわかります．

この図4がシミュレーション結果ですね．Vaughan 課題でシミュレーションした結果です．非常に長い時間を4ブロック別のパラメタで続けてやったものです．この点線の横線がマッチングを満たすような選択比率になっていて、実線の横線が最適な選択比率になっています．ですから、実線の辺りにいれば本当はたくさん報酬がもらえるはずなのに、4つのアルゴリズムはちゃんとマッチングの近くに辿り着きます．ブロックが変わると、最適を飛び越えてマッチングの近くに辿り着きます．定常状態でマッチングになっていることがわかります．ところがこの2つのアルゴリズム、特殊な Actor-Critic と Q-learning は真ん中辺を漂っています．マッチングにも最適にもなっていないことがわかります．上の4つのアルゴリズムはマッチングにな

りますが、特殊な Actor-Critic と Q-learning は、マッチングになっていないということがわかります．

これと同じ課題を皆さんにやってもらいたいと思っているので、この課題の中身は教えません．こんなふうに、強化学習でも、最適が保証されてないような、難しい課題をやらせたときに、マッチングになるものもあるし、そうでない場合もあることがわかります．

報酬最大化の方法

もう少し考える材料として、どうしたら報酬最大化できるようになるのかという観点で考えていきましょう．先ほど、トータルの平均報酬が

- Q_a : a を選んだときの報酬期待値（環境）
- p_a : a を選ぶ確率（動物）

を用いてこんな風に書けると言いました：

$$\langle r \rangle = \sum_{a'=1}^n Q_{a'} p_{a'}$$

この平均報酬 $\langle r \rangle$ をなんとか最大にしたい、最大になるような選択確率 p_a を求めたいわけです．これをどうやるかという、トータルの平均報酬 $\langle r \rangle$ が大きくなるように選択確率 p_a を変化させていけば、いつか大体最大になるでしょう．極大値にいくことももちろんありますが、極大値がもし無ければ、最大値に必ず行きます．ということは最急勾配法というのをを使って、一番坂道を登るように変化させていけば、必ず山のてっぺんにいける、ということになります．ですがこれを、もし、真面目にやろうとすると、相当な試行を繰り返さないとなかなかどちらに変化させたいか、わからない形をしています．何回も試してトータルの平均報酬を推定します．その推定値を覚えておいて、また、 p_a をちょっと変更してまた推定し比較する、ということは何回もやっていかないとできないことがわかりますね．まあ、もし2択ぐらいだったら、動

かすのは一個の変数なのでいいのですが、もし選択肢が増えていったら大変なことが想像できますね。こんなことをやっていたら、とてつもなく時間がかかります。ですから、たぶん動物はそんなことをやっていないだろうということが予想されます。で、もうちょっと今の方針で考えてみますと、式の形を見て微分を利用したらいいのではないかと思います。〈 r 〉をそのまま微分して、勾配を直接求めると、こんな形になります。

$$\Delta p_a \propto \frac{\partial \langle r \rangle}{\partial p_a} = Q_a + \sum_{a'=1}^n p_{a'} \frac{\partial Q_{a'}}{\partial p_a}$$

右辺第1項の Q_a は何とか推定できますね。 Q_a の推定方法は Melioration や Q-learning のときに紹介しました。ところが、右辺第2項は知らないわけです。ここを推定しようと思えば、先ほど言ったこと、ちょっと動かして差分を見るってことをしなきゃいけなくなります。さあではどうしますか？

ちなみに、 p_a をそれぞれ動かすと足して1にならなくなるので、先ほどのポリシーパラメータを使って、 q_a を動かすのが合理的でしょう。で、このへんの材料をもとに、動物はなぜ、マッチング行動に至っているのかということ、皆さんで議論して結論を出していただきます。

演習課題：

動物は、報酬最大化できないにもかかわらず、マッチング行動を示す。いったいなぜか？

先ほどの課題の Matlab デモがあります。A か B かどちらかを選んでその結果が表示されます。実際にはお金はあげませんが、なるべくたくさん貰えるようにやってみてください。1ブロック(100回)終わると、結果が、累積の選択比率がでます。このプログラムの中身に何が仕込んであるか今は言いませんが、どういうやり方するとマッチングになるのか、どうやってやれば最適になるか、というのを考えながらやってみてください。

(注：前半の講義の後に演習の時間がありました。酒井先生が準備された Matlab プログラムで、実際に Vaughan 課題を体験しつつ、演習課題に関して班毎に分かれて議論を行いました。以下は、演習の成果の発表です。ただし、紙面の都合上、大幅に編集してあります。)

2. 演習課題の発表

A 班

わたしたちは、マッチングが出るアルゴリズムと出ないアルゴリズムとでは、何が違うのかを考えます。

Actor-Critic 法と Q-learning とを比較すると、両方オンラインにも関わらず、効率が悪いように思われるマッチングが出る方法 (Actor-Critic 法) を動物は選ぶ、これはなんでなのだろうかと考えました。私たちは、報酬を得るための行動は、決められた回数しかできない、言葉をかえると、行動にある一定のコストがかかると考えました。一方の選択はいつも平均的に安定した報酬がえられる、もう一つの選択はリスクでたまに膨大な報酬が得られる、ということがあったときには、行動にコストがかかるので、なるべくコストがかかっても、安定的にとれるほうの選択をしがちになるだろうと考えて、ひょっとしたら、学習が完了するまでの報酬の累積量が最適な行動を取るほうより大きいんじゃないか、と考えました。有限回数しかできない場合、例えば、10日間食べなかったら死んでしまいますよね。そしたら、100回やったら膨大な報酬がもらえるよりも、毎日ご飯を食べられたほうがいいですよ。

それを示す例として、一日一回確実に1または、百日に一回の確率で101の報酬が得られるとします。肉食タイプで考えます。これを、千回試行したときに、どれだけ報酬がもらえるかを考えました。プリントの2番目のアルゴリズムと Q ラーニングの2つでプログラムで乱数を発生させてやってみました。

当然のことながら、Q-learningの方が、1000回後に(報酬が)大きくなる確率が高い。平均1100とかになります。ただ、Q-learningの分散が非常に大きくなりました。700とかになります。逐次だと、平均960分散100のようになります。逐次でないと、死ぬ確率高いんですね。そういう意味で、マッチングになるタイプのものを選ぶでないかと我々は結論づけました。

Q：分散が小さくなければ生存に有利な理由は？

A：例えば、1000回の時点で総報酬が900以下だと死ぬという状況を考えて時に、平均総獲得報酬がほぼ同じならば、分散が小さいときに有利になります。

なんでマッチングをとるかという理由ですけれども、マッチングは、選択肢を幅広く持つておくための戦略なんじゃないかと思いました。環境の変動に対して柔軟に対応できる。その一つとしては、各選択肢の期待

値が同じになるように行動してるので、ある選択肢に変化が起こったときに期待値の変動が少ない。それは、わかんないですけど、(講義中にでていた) 様々な学習則のシミュレーションをみても、状況が変化するときの、傾きをみていただくと、傾きが急峻だということが、変化する率が高いことを示してるのかなと思います。

あとは、生物学的な制約が当然あるだろうってのが、100年に一回ものすごい食べものがもらえるよりも、1か月食べないというわけにもいかない。この話ってのは、食べものに依存するので制約ができたわけで、お金のことならだいたいが話が変わる可能性もあると思うんですけど。以上です。

C 班「マッチング≒妥協？」

まず、実際に、配布された Matlab のプログラムを実行してみました。一人は、29000 円を獲得することができて、もう一人は 2000 円しか獲得できなかった。2000 円の人を選択比率がマッチングの傾きに近かったの、あまりいい結果ではなかったのかなと思います。

なぜ動物はマッチング行動しているのか、ということが問題になっていたと思います。私たちは、動物が生きていく行動を選択するのになぜマッチング行動がおこるのかというのを論じていきました。脳はさばるというタイトルは言いすぎかもしれないんですけど、Q-learning では脳にとって計算の負担が大きすぎるというのが講義でも言われていたと思います(注：これは誤解で、正確には「最大化をするのは負担が大きすぎる」ということです)。そこで、過去一定期間の履歴から得られた報酬をもとに、一番もっともらしい選択をするということで、コストダウンをはかりたい。そのために、マッチング戦略があるのかなと思いました。

ですが、報酬期待値を求めるのが、脳にとって計算負荷が大きい。でも(Matlab の) 結果を見るとマッチングでも最適でもない戦略ができていると思います。数多く戦略のなかで、なぜマッチング行動をとるのかという説明にはなっていないので、別の戦略を考えてみました。コストを下げるということだけを考えて、グリーディー戦略というのを考えてみました。グリーディー戦略とは、その地点だけをみて、一番いい行動だけを選択し続けるという戦略のことです。例えば、講義の中で例にあがっていた、左と右に餌があるという場合に、左の方に餌が多くある確率が高いと知っていたら、ずっと左だけをとりつづけるという戦略です。ですが、この場合、環境が変化するととんでもなく悪い結果になるということがわかると思います。環境が

まったく逆転してしまったときに、左を選択するのは悪い選択になってしまう。つまりグリーディー戦略だと環境が変化した際に弱いということが言えると思います。ですが、マッチング戦略であれば、一定期間さかのぼった情報をもとに、次の戦略を決めるってことで、環境の変化に柔軟に対応できるのではないかと考えました。生物が生き残るために、環境の変化に対応するために、マッチング戦略が一番いい戦略だから、だから、動物はマッチング戦略をとるのではないかと考えました。

3. 後半の講義

マッチングは学習のコストを下げるための妥協か？

あまり時間ないので、さくっと、まあ一つの答えを提示します。マッチングは妥協というか、学習にかかる時間、ラーニングのコストを下げるための方策で、別に真の最適行動に至らなくても、まあ、マッチングで大体速く良い所に行ければ、良いだろう、というのが(演習の発表の)結論だったと思います。実は、私もそう思って、いろいろ解析しています。

この図 5 は何かというと、競合者のシマウマがいるタイプの課題で、あらゆるパラメタを検索した結果です。マッチングと最適でどのくらい得られる報酬が違うのかをパラメタ空間で理論的に全部調べ尽くしました。最適報酬の何割以上もらえるようなパラメタ領域が全体の何%あるか、ということを図 6 で示しています。マッチングをやっていると、大体 100%に近いですね。最適の 9 割以上の報酬がマッチングでもらえるパラメ

図5 競合的採餌課題(定常的な競合者がいる草食タイプの場合)においてパラメタ空間を網羅して最適よりもどのくらい悪いかを調べた図(Sakai and Fukai, in press)

図6 Vaughan 課題での報酬確率のデザイン

タ領域は93%あることがわかります。2分の1でランダムに選んでしまうと、どんどん落ちていきまして、最適報酬の9割以上得られるパラメタ領域は23%になってしまいます。交互に選ぶ選び方もそんなに良くありません。したがって、少なくともこの課題に関して言えば、マッチングの戦略ってのはそんなに悪くない。しかも、本当に最適に辿り着くよりも遥かに速い、ということが言えるのです。しかし、あくまでも課題依存の話です。先ほどのような課題であればこの結論で構いませんが、違う課題では同じ結論が得られるとは限りません。世の中には、マッチングでは最適報酬よりはるかに少ない報酬しか得られない課題を組むことも実は可能です。その例は、実は今回皆さんにやってもらった課題です。やってもらった課題は、Vaughanが提案した課題でして、基本的には肉食タイプをベースにしています。ただし、Aを選んだ時に与える確率を、過去の一定期間の選択頻度に依存して変化させるという課題になっています。これはまさにマッチング行動をみるために、考え出された課題なんですね。で、この課題では、その、今の関数ですね、どういう選択頻度の時に与える確率をいくつにするか、というのが自由にデザインできるんです(図6)。選択肢1をとったときに報酬を与える確率を、今までの頻度に対してこんな関数にしてみました。選択肢2をとったときにはこんな関数にしてみました。この交点がマッチングになります。平均獲得報酬はこの2つの関数の間くらいになっていて、最適な選択頻度はここにあることがわかります。この関数いくらでもデザインすることができますから、マッチングと最適で得られる報酬の比率を自由に変えることができます。実は皆さんにこの課題をやってもらいました。先ほどの、マッチングで妥協していいかどうかという問題ですが、こういう課

題を考えると、最適に比べると、マッチングで相当損することになります。

Q: 日常生活的に、こういう状況ってどんな時に考えられますか

A: わかりませんが、例えば、相手がいて、相手が今までの選択頻度で選択を変える時、そういうゲームをやったとしたら、一方から見れば、こういう課題に見えます。

マッチング行動の理由

なぜマッチングになるか、という問に対する答えはたぶん一つではないと思います。一つの解として、こういうことを考えてみましょう。簡単に、2択の場合を考えましょう。平均獲得報酬はこんなふうに計算できます。

$$\langle r \rangle = Q_1 p_1 + Q_2 p_2 = Q_1 p_1 + Q_2 (1 - p_1)$$

これを、最大化しようと思ったら、微分して勾配を求めてそれをもとに選択確率を変化させていけばいいわけです。

$$\Delta p_1 \propto \frac{d\langle r \rangle}{dp_1} = Q_1 - Q_2 + p_1 \frac{\partial Q_1}{\partial p_1} + (1 - p_1) \frac{\partial Q_2}{\partial p_1}$$

先ほどもちょっと言いましたが、微分を含む項(右辺の後半の3項と第4項)を知らないわけです。知らないから、ここを推定しようと思うと、 p_1 をちょっと動かして推定した結果との差を見なければいけないわけですね、で、ここを妥協しましょう。つまり、環境が決めている Q_1 と Q_2 は、自分が戦略 p_1 を変えたとしても、変わらないものだと、思い込んでしまいましょう。そしたら、この3項4項は0なんですね。0だと思い込んでしまうと、更新式はこんな形がです。

$$\Delta p_1 \propto Q_1 - Q_2$$

Δ は更新量という意味です。そうすると、もし定常状態にいたつたとすれば、更新量が平均的に0なわけですから、 Q_1 、 Q_2 が等しくなる。ですから、自分が戦略を変えても環境が変わらないと思い込んで、勾配法を使って最大化していけば、必ずマッチングに行くことがわかりますね。多択の場合にも、同様に計算していくと、定常状態では必ず、選択確率が0か、期待値が全体の平均に等しいというのが導かれることがわかります。ですから、マッチングに対する一つの答えとして、どこで妥協しているかという、微分値を計算するのを妥協しているのではない、という可能性が挙げられます。環境が変わらないという想定で最大化

をしている、というのが、一つの答えじゃないかと考えられます。

先ほどの Actor-Critic アルゴリズムはどうやって出てきたかという、ポリシーパラメタ q_a を使って選択確率の定義をすると、

$$p_a = \frac{e^{\beta q_a}}{\sum_{a'=1}^n e^{\beta q_{a'}}} \quad (\text{確率なので足して 1 になる})$$

となります。平均的な獲得報酬は、多択の場合、このようになります。

$$\langle r \rangle = \sum_{a=1}^n Q_a p_a$$

ここで、自分が戦略を変えても Q_a が変わらないと思って、勾配法の式を導き出すと、

$$\Delta q_a \propto \beta p_a (Q_a - \langle r \rangle) \quad (1)$$

となります。実際にこれをアルゴリズムとして、実装しようと思ったら、どうなるかという、右辺に p_a がかかっています、ここの部分を a を選んだ時にだけ更新することにすれば、平均的にはこの p_a をかけたこととなりますね。その時には、 a を選んだ時の報酬に、 Q_a を入れ替えてもいいでしょう。で、全体の平均値 $\langle r \rangle$ は、別で推定しましょう。こういう置き換えをおこなうと、Actor-Critic になるんですね。

$$q_{a_t} \leftarrow q_{a_t} + \alpha(r_t - V)$$

次に、この更新式 (1) を展開しておきましょう。

$$\Delta q_a \propto p_a Q_a - p_a \langle r \rangle$$

展開しておいて、右辺第 1 項の p_a の代わりに、 a を選んだ時だけ更新するとします。同じように、 Q_a の所を a を選んだ時の報酬 r_t にします。第 2 項の p_a は、そのままにしておいて、この部分は毎回更新することになると、

$$\Delta q_a \propto \delta_{aa_t} r_t - p_a r_t$$

となります。これは Direct-Actor の更新式になります。ですので、報酬期待値の変化を無視して最大化を行うという原則から、マッチングになる全てのアルゴリズムはこんなふうに全部導けます。なぜ Q-learning や特殊な Actor-Critic はマッチングにならないかというと、式 (1) の形に押し込むことはできないからです。

なぜマッチングになるかという疑問に対する一つの解答：

報酬期待値が一定だという前提で、効率的に報酬

最大化をしてしまうと、

報酬期待値が一定ではない時には、最適行動にならずマッチング行動になる。

ちなみに、報酬期待値が本当に一定のとき、つまり、肉食タイプの課題ですが、その場合には、マッチングと最適が一致します。

真の報酬最大化と状態依存マッチング

さて、本当は、報酬最大化したいはずで。では、どうしたらできる、ということを考えていきますと、報酬期待値 Q_a が本当に定数に決まるような、十分な情報源をもとに行動すれば、マッチングで最適になることがわかります。この情報源というのは何かというと、肉食タイプの場合は、情報はなくても報酬期待値は一定ですので、情報は必要ありません。草食タイプの場合は、最後にある選択肢 a を選択してから、どのぐらい経過しているか、という間隔を、情報源として使っていたら、その条件の下に計算すれば、報酬期待値は一定になります。私が提案した競合者がいるタイプも同様です。先ほど皆さんにやってもらった Vaughan 課題では、実験者が選択頻度を計算している期間もわかって、その期間での選択頻度を情報源として用いれば、報酬期待値 Q_a が一定になります。このように情報源を使って、その状態に依存して行動を選べば良いということです。

情報源 S を用いた上で、先ほどと同様に、自分が戦略をかえても Q_a が変わらないと思い込んで、報酬最大化すると、自分の用いている情報源の値を条件としたマッチングの法則が成り立つという状況になります。このとき、用いている情報源が十分だったら、最大化になります。ですが、情報源がもし不十分なら、最大化はできずに、情報源 S に依存したマッチング行動が顕れます。報酬期待値が決まるような十分な情報源をもとに行動をしていると思い込んで最大化すると、本当にそれに十分な情報であれば最適になりますが、そうでなかったら、状態依存マッチングの法則がでてくる、ということになります。皆さんも Matlab デモをやりながら無意識に情報源を探すという過程を行っていたと思います。それが、もし、どういう情報源をもとにしたら報酬期待値一定になるのか、という基準で情報源を探していったとすると、その情報源が探し出せたとき、最適になるわけです。動物はそういう戦略を取っているのではないかと考えられるわけです。

行動選択の結果が遅れて得られる場合

さて、今までは考慮してきませんでしたが、選択の結果はすぐ得られるとは限らない、という問題に直面することがよくあります。先ほど導入した情報源に依存した選択をすると、状態遷移を通じて、結果が遅れていたとしても、その原因となった行動選択に結びつけられることが知られています。それが強化学習理論の一番の肝になっています。どういうことかという、情報源から情報が得られて、それがどんどん、移り変わっていくわけです。移り変わりというのは、自分の行動にも、もらった報酬にも依存する。そこで、例えば、この時点の行動とそれから数ステップ先の報酬とが、実は環境を通じて、因果関係があったとしましょう。すぐ結果が得られなかったんですね。この場合にも、実は状態遷移を通して、過去の原因の行動を強化できます。まず、ある時刻以降に得られた報酬の足し合わせ、という量を定義しましょう。

$$\nu_t^\gamma \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots$$

ここで γ という係数は、割引率 (discount factor) と呼ばれていて、段々と重みが減っていく形で、将来報酬を足し合わせた量となっています。この量を、先ほどまでの話において選択の直後に得られる即時報酬 r_t と、全部入れ替えてやって、

$$Q_{a_t}^\gamma(s_t) \equiv \langle \nu_t^\gamma | s_t, a_t \rangle, \quad V^\gamma(s_t) \equiv \langle \nu_t^\gamma | s_t \rangle$$

と定義します。この将来平均報酬 $\langle \nu_t | s \rangle$ を最大化するように、同じように報酬期待値の変化を無視して勾配法を構成すると、状態依存マッチングみたいな式が出てきます。

$$p_a = 0 \quad \text{あるいは} \quad Q_a^\gamma(s) = \langle \nu^\gamma | s \rangle$$

この関係式は γ にも依存します。割引率に依存するようなマッチング法則になっています。動物が割引率を導入して将来報酬まで考慮した行動選択を行っているとしたら、この割引率に依存するマッチング法則が観測される可能性があります。そのような見方をすることによって逆に動物が用いている割引率を推定することが可能かもしれません。

γ つきの Q や V の値は、こんな方法で学習できるんですね。

$$\begin{aligned} Q_{a_t}^\gamma(s_t) &\leftarrow Q_{a_t}^\gamma(s_t) \\ &\quad + \alpha (r_t + \gamma Q_{a_{t+1}}^\gamma(s_{t+1}) - Q_{a_t}^\gamma(s_t)) \\ V^\gamma(s_t) &\leftarrow V^\gamma(s_t) \\ &\quad + \alpha (r_t + \gamma V^\gamma(s_{t+1}) - V^\gamma(s_t)) \end{aligned}$$

これを Temporal Difference (TD) learning といいます (Sutton and Barto, 1998; Dayan and Abbott, 2001)。一個先の状態の Q の値や V に γ をかけたものと、今貰えた報酬と、今の状態の Q の値や V の値の差分になっているので、Temporal Difference 学習と言っています。一個先の推定値を使うだけで、将来にわたる累積報酬の期待値が推定できるというのが、強化学習の一番の肝です。これをする、遅れた報酬にも対応できるということになります。

以上でこの講義を終わりたいと思います。

参 考 文 献

- 1) Baum, W. and Rachlin, H. (1969): Choice as time allocation, *Journal of the Experimental Analysis of Behavior*, Vol.12, pp.861-874
- 2) Dayan, P. and Abbott, L. (2001): *Theoretical Neuroscience*. Cambridge, MA: The MIT Press
- 3) DeCarlo, L.T. (1985): Matching and maximizing with variable-time schedules, *Journal of the Experimental Analysis of Behavior*, Vol.43, pp.75-81
- 4) Herrnstein, R.J. (1961): Relative and absolute strength of response as a function of frequency of reinforcement, *Journal of the Experimental Analysis of Behavior*, Vol.4, pp.267-272
- 5) Herrnstein, R.J. (1997): *The Matching Law: Papers in Psychology and Economics*. Cambridge, MA: Harvard Univ. Press
- 6) Herrnstein, R.J. and Heyman, G.M. (1979): Is matching compatible with reinforcement maximization on concurrent variable interval, variable ratio? *Journal of the Experimental Analysis of Behavior*, Vol.31, pp.209-223
- 7) Herrnstein, R.J. and Vaughan, W.J. (1980): Melioration and behavioral allocation. In J. Staddon (Ed.), *Limits to Action: The Allocation of Individual Behavior*. New York: Academic Press
- 8) Mazur, J. (1981): Optimization theory fails to predict performance of pigeons in a two-response situation, *Science*, Vol.214, No.4522, pp.823-825
- 9) Mazur, J.E. (2005): *Learning and Behavior*. 6th edition. Prentice Hall (日本語訳「メイザーの学習と行動」J.E. メイザー, 二瓶社)
- 10) Sakai, Y. and Fukai, T. (in press): The actor-critic learning is behind the matching law: Matching vs. optimal behaviors, *Neural Computation*
- 11) Sugrue, L.P., Corrado, G.S., and Newsome,

- W.T. (2004): Matching behavior and the representation of value in the parietal cortex, *Science*, Vol.304, pp.1782–1787
- 12) Sutton, R.S. and Barto, A.G. (1998): *Reinforcement Learning*. Cambridge, MA: MIT Press (日本語訳「強化学習」Sutton & Barto, 森北出版)
- 13) Vaughan, W.J. (1981): Melioration, matching, and maximization, *Journal of the Experimental Analysis of Behavior*, Vol.36, pp.141–149