HAL
open science

Modélisation des stratégies de génération de choix variables chez la souris

Marwen Belkaid, Jérémie Naudé, Philippe Faure, Olivier Sigaud

To cite this version:

Marwen Belkaid, Jérémie Naudé, Philippe Faure, Olivier Sigaud. Modélisation des stratégies de génération de choix variables chez la souris. Conférence Nationale en Intelligence Artificielle, Jul 2019, Toulouse, France. hal-02328815

HAL Id: hal-02328815

https://hal.science/hal-02328815

Submitted on 23 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modélisation des stratégies de génération de choix variables chez la souris

Marwen Belkaid[1†] Jérémie Naudé[2] Philippe Faure[2*] Olivier Sigaud[1*]

[1] Sorbonne Université, CNRS, ISIR, 75005 Paris, France
[2] Sorbonne Université, INSERM, CNRS, NPS - IBPS, 75005 Paris, France
[*] Contribution égale
[†] Auteur correspondant : marwen.belkaid@upmc.fr

Résumé

Mieux comprendre les processus cognitifs impliqués dans la prise de décision et les comportements adaptatifs est fondamental pour la communauté de l'intelligence artificielle. Dans cet article, nous étudions les stratégies cognitives permettant la génération de choix aléatoires chez la souris. Nous utilisons un algorithme d'apprentissage par renforcement (RL) pour modéliser leur comportement dans une tâche qui récompense des séquences de choix non-répétitives. Nos résultats montrent que les souris sont capables de résoudre la tâche et suggérent qu'elles le font en ajustant les paramètres de leur processus de décision.

Mots Clef

RL, exploration aléatoire, variabilité des choix.

Abstract

Better understanding the cognitive processes involved in decision-making and adaptive behaviors is fundamental for the artificial intelligence community. In this paper, we study the cognitive strategies allowing the generation of random choices by mice. We use a reinforcement learning (RL) algorithm to model the animals' behavior in a task rewarding non-repetitive choice sequences. Our results show that mice can solve the task and suggest they do so by adjusting the parameters of their decision-making process.

Keywords

RL, random exploration, choice variability.

1 Introduction

L'apprentissage par renforcement est un excellent exemple de brassage fructueux entre intelligence artificielle et neurosciences. En effet, les algorithmes de différences temporelles ont été inspirés par le comportement animal dans l'étude de conditionnement [1]. Au cours des deux dernières décennies, ces algorithmes ont à leur tour fourni un cadre formel fondamental pour l'analyse des données comportementales et neurales en neuroscience [2, 3]. Ce cadre décrit bien les mécanismes par lesquels les individus exploitent leur connaissance de l'environnement et répètent des actions qui maximisent les récompenses. Cependant, les principes sous-jacents à la génération de variabilité dans le processus de prise de décision sont encore mal compris. Dans le cadre de l'apprentissage par renforcement, la variabilité des choix est associée au processus d'exploration. L'exploration "dirigée" vise à recueillir des informations sur les contingences environnementales, tandis que l'exploration aléatoire introduit une variabilité indépendamment de celles-ci. Des études ont montré que les animaux sont en mesure d'accroître la variabilité de leurs choix, en particulier lorsque les règles changent [3] ou qu'ils doivent tromper des prédictions faites sur leurs décisions [4, 5]. Cependant, l'utilisation systématique de contingences probabilistes ou d'environnements volatils rend difficile d'isoler expérimentalement une génération intrinsèque de variabilité des conditions environnementales.

Dans cette étude, nous utilisons une nouvelle expérience sur souris et modélisons le comportement des animaux avec un modèle RL pour tester l'hypothèse selon laquelle les animaux peuvent ajuster de manière adaptative le caractère aléatoire de leur comportement.

2 Expériences sur souris

Nous avons entraîné des souris sur une tâche nécessitant la génération d'une séquence complexe de choix [6]. Dans une arène ouverte, les souris pouvaient recevoir une récompense dans trois zones cibles. Ne pouvant pas recevoir deux récompenses successives au même endroit, les animaux devaient réaliser une série de choix en choisissant la prochaine cible parmi les deux alternatives restantes. L'attribution de la récompense dépendait de la complexité de la séquence. Plus précisément, nous avons estimé la mesure de complexité LZ [7] de sous-séquences de choix de taille 10 (9 choix passés + choix suivant) à chaque essai. Malgré sa difficulté, cette tâche est totalement déterministe, ce qui la différencie des autres approches [4, 5]. Nous avons pu déterminer que 25% des séquence de longueur 10 n'étaient pas récompensés au dernier choix. Donc, théoriquement, si une estimation correcte de la complexité des séquences conduit à un taux de réussite de 100%, une sélection aléatoire pure a un taux de réussite de 75%, et une séquence répétitive (e.g. ABCABC...) conduit à un très faible taux de réussite.

Nous avons constaté que les souris augmentaient progressi-

Plages des hyperparamètres pour la recherche aléatoire

Label	Plage	Pas	Description
m	$[0, 9]$	1	Taille de la mémoire
τ	$[1, 20]$	continue	Température du Softmax
κ	$[0, 1]$	continue	Coût du demi-tour

Plages des hyperparamètres pour la recherche aléatoire en grille

Label	Plage	Pas	Description
m	$[0, 9]$	1	Taille de la mémoire (pas d'ambigüité)
	$[0, 7]$	1	Taille de la mémoire (ambigüité faible)
	$[0, 5]$	1	Taille de la mémoire (ambigüité moyenne)
α	$2^{-i}, i \in [0, 10]$	1	Taux d'apprentissage
τ	$2^{i}, i \in [-4, 4]$	1	Température du Softmax
κ	$[0.5, 0.95]$	0.05	Coût du demi-tour

TABLE 1 – Plages des valeurs utilisées pour l'optimisation des hyperparamètres

vement la variabilité de leurs séquences de choix [6]. Cela a été démontré par l'augmentation corrélée au fil des sessions de la complexité des séquences et du taux de réussite qui en résulte, ainsi que l'augmentation du taux de demi-tours (i.e. changement de direction) dans les séquences.

3 Méthodes

3.1 Modèle d'apprentissage par renforcement

Nous avons représenté la tâche par un processus de décision markovien (MDP) avec trois états $s \in \{\text{A, B, C}\}$ et trois actions $a \in \{\text{GoToA, GoToB, GoToC}\}$, correspondant respectivement aux emplacements où la récompense est attribuée et aux transitions entre ces emplacements. Les valeurs d'état-action $Q(s, a)$ sont apprises à l'aide de la règle de Rescorla-Wagner [8] :

$$\Delta Q(\mathbf{s_t}, a_t) = \alpha(\mathcal{U}_{t+1} - Q(\mathbf{s_t}, a_t)) \tag{1}$$

où $\mathbf{s_t} = [s_t, s_{t-1}, ..., s_{tm}]$ est l'état en cours qui peut inclure la mémoire jusqu' au m^{ime} emplacement passé, a_t l'action en cours, α la vitesse d'apprentissage et $\mathcal{U}$ la fonction utilitaire définie comme suit :

$$\mathcal{U}_{t+1} = \begin{cases} (1-\kappa).r_{t+1} & \text{si } \ s_{t+1} = s_{t-1} \\ r_{t+1} & \text{sinon} \end{cases} \tag{2}$$

où r est la fonction de récompense et κ le paramètre de coût de demi-tour modélisant le coût moteur ou tout biais en défaveur de l'action menant l'animal à son emplacement précédent. Le coût de demi-tour était nécessaire pour reproduire les trajectoires stéréotypées observées chez la souris à la fin de la phase d'entraînement [6].
La sélection de l'action a été effectuée à l'aide d'une règle softmax, ce qui signifie que, dans l'état $\mathbf{s_t}$, l'action a_t est sélectionnée avec la probabilité :

$$P(a_t|\mathbf{s_t}) = \frac{e^{Q(\mathbf{s_t}, a_t)/\tau}}{\sum\limits_{a} e^{Q(\mathbf{s_t}, a)/\tau}} \tag{3}$$

où τ est le paramètre de température. Ce paramètre réduit la sensibilité à la différence dans les valeurs d'actions, augmentant ainsi la quantité de bruit ou le caractère aléatoire de la prise de décision. Le coût du demi-tour κ a l'effet inverse puisqu'il représente un biais comportemental qui limite le caractère aléatoire des choix. Nous appelons *randomness* l'hyperparamètre défini comme $\rho = \tau/\kappa$.
Une version standard d'apprentissage par renforcement revient à n'inclure aucune mémoire des emplacements précédents et à utiliser un coût du demi-tour nul.
Nous avons également utilisé des variantes du modèle pour manipuler l'ambiguïté de la représentation d'états, i.e. si chacun des emplacements A, B, C pouvait être représenté par $n \geq 1$ états. Pour simplifier, nous avons utilisé trois niveaux d'ambiguïté avec $n = 1$, 2 et 3 pour tous les emplacements. Cela nous a permis de présenter une preuve de l'impact potentiel de l'utilisation d'une représentation d'état parfaite dans notre modèle.

3.2 Optimisation des hyperparamètres

Les principaux résultats d'ajustement de modèle (*model fitting*) présentés dans cet article ont été obtenus en optimisant les hyperparamètres vis-à-vis du comportement des souris session par session indépendamment. Ce processus visait à déterminer quelles valeurs des deux hyperparamètres m et $\rho = \tau/\kappa$ font que le modèle se comporte comme une souris en termes de taux de réussite (pourcentage d'actions récompensées) et de complexité (variabilité des décisions). Notre objectif principal était de distinguer deux stratégies : répéter des séquences récompensées ou choisir au hasard. Par conséquent, nous avons momentanément laissé de côté la question de la vitesse d'apprentissage et n'avons considéré que le comportement du modèle après la convergence. Le taux d'apprentissage α a ici été fixé à 0.1.
Les hyperparamètres ont été sélectionnés par recherche aléatoire (*random search*) [9] (voir les plages répertoriées dans le tableau 1). Le modèle a été exécuté pour 2.10^6 itérations pour chaque jeu de paramètres. Le score de *fitness*

relatif aux données moyennes des souris à chaque session a été calculé comme suit :

$$fitness = 1 - \frac{1}{2}(|\widehat{S} - \overline{S}| + |\widehat{C} - \overline{C}|) \quad (4)$$

où $\overline{S}$ et $\overline{C}$ sont respectivement le taux de réussite moyen et la complexité moyenne chez la souris et $\widehat{S}$ et $\widehat{C}$ le taux de réussite et la complexité du modèle - tous les quatre $\in [0, 1]$. Les simulations étaient suffisamment longues pour que l'apprentissage converge. Ainsi, au lieu de plusieurs exécutions pour chaque jeu de paramètres, ce qui aurait été coûteux en calcul, $\widehat{S}$ et $\widehat{C}$ ont été moyennés sur les 10 dernières sessions simulées. Nous avons considéré qu'une session simulée = 200 itérations, ce qui représente la limite supérieure du nombre d'essais réalisés par les souris au cours d'une session réelle.

Étant donné que les souris étaient systématiquement récompensées au cours de l'entraînement, leur taux de réussite dans cette condition n'était pas indicatif (i.e. toujours 100%). Ainsi, pour évaluer la capacité du modèle à reproduire des trajectoires stéréotypées circulaires lors de la dernière séance d'entraînement, nous avons remplacé $\overline{S}$ et $\widehat{S}$ dans l'équation (5) par $\overline{U}$ et $\widehat{U}$ représentant les taux moyens de demi-tour pour le modèle et les souris, respectivement.

Des simulations supplémentaires ont été réalisées avec deux objectifs : 1) tester si un seul jeu de paramètres pouvait reproduire le comportement de la souris sans qu'il ne soit nécessaire de modifier les valeurs des paramètres au cours des sessions, 2) tester l'influence de l'ambiguïté de la représentation d'état sur l'utilisation de la mémoire dans le modèle de calcul. Ces nouvelles simulations visaient à reproduire le comportement de la souris au fil des sessions, de la phase d'entraînement à la condition de complexité. Par conséquent, le taux d'apprentissage α a été optimisé en plus des hyperparamètres m et ρ susmentionnés (voir les plages répertoriées dans le tableau 1). Chaque jeu de paramètres a été testé sur 20 exécutions différentes. Chaque série est une simulation de 4000 itérations, ce qui correspond à 10 séances d'entraînement et 10 séances de complexité, car les séances simulées consistent en 200 itérations. Le score de *fitness* a été calculé comme le score moyen de la dernière séance d'entraînement et des 10 séances de complexité utilisant les équations (4) et (5). L'utilisation d'une recherche en grille (*grid search*) garantissait des valeurs comparables pour les trois niveaux d'ambiguïté. Vu le coût de calcul supplémentaire induit par des niveaux d'ambiguïté plus élevés, nous avons progressivement diminué la limite supérieure de la plage de taille de la mémoire afin d'éviter des calculs longs et inutiles dans des régions sans intérêt de l'espace de recherche.

4 Résultats

En définissant des états en tant que vecteurs incluant l'historique des localisations précédentes au lieu de la seule localisation actuelle, nous avons pu faire varier la taille de la mémoire des agents simulés et obtenir différentes solutions du modèle en conséquence. Nous avons constaté

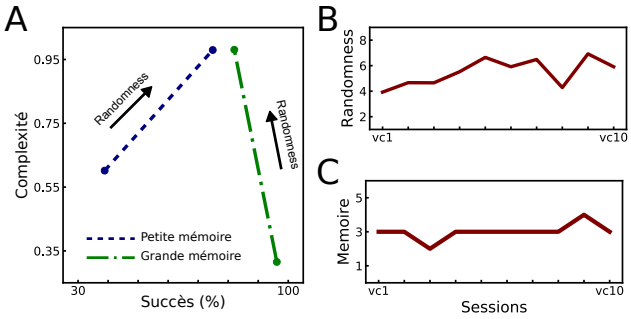

FIGURE 1 – A) Deux régimes de fonctionnement du modèle. **B)** et **C)** Valeurs optimisées des paramètres ρ et m respectivement.

qu'en l'absence de mémoire (i.e. état = emplacement actuel), le modèle avait appris des valeurs égales pour les deux cibles dans presque tous les états. En revanche, et en accord avec les prédictions théoriques pour l'apprentissage par renforcement, avec l'historique des neuf derniers choix stockés en mémoire, le modèle était capable d'apprendre à enchainer des choix récompensés (100% succès).

Par ailleurs, la tendance à effectuer des choix aléatoires dépendait non seulement des valeurs associées aux choix actuels, mais également de la température du softmax. Cet hyperparamètre a eu des effets opposés sur le comportement du modèle selon que la taille de mémoire utilisée soit petite ou grande. Tandis que l'augmentation de la température augmentait toujours la complexité des séquences de choix, elle augmentait également le taux de réussite pour les petites tailles de mémoire, mais le diminuait pour les plus grandes mémoires (Fig 1A). Cela indique que le modèle peut trouver la solution optimale de la tâche si une mémoire de grande taille est utilisée, alors qu'une petite mémoire nécessite un niveau d'aléatoire élevé. Une limite entre les deux régimes a été trouvée entre des tailles de mémoire 3 et 4.

En optimisant le modèle vis-à-vis du comportement de la souris, nous avons constaté que l'amélioration de leurs performances au fil des sessions s'expliquait mieux par une augmentation du facteur aléatoire (paramètre *randomness*) en utilisant une petite mémoire (voir Fig 1B et 1C) – Notons que la baisse à la 8ième session provient des données expérimentales. Ce modèle a permis de mieux expliquer les données que l'utilisation d'un jeu de paramètres fixe tout au long des sessions (Facteur de Bayes = 3,46).

Le modèle avec une mémoire de taille 3 reproduisait au mieux le comportement de la souris, mais à peine mieux que les versions avec des mémoires plus petites. Du point de vue computationnel, une explication possible du fait que, bien qu'elle soit théoriquement suffisante, une mémoire de taille 1 obtient des résultats inférieurs à la taille 3, c'est que la représentation d'état est trop simplifiée dans le modèle. Ainsi, avec seulement trois états représentant parfaitement et sans ambiguïté chacune des cibles, l'algorithme ne peut pas prendre en compte le bruit comportemental, les erreurs et/ou les biais de la souris. Par consé-

quent, modifier la représentation de l'état du modèle pour le rendre plus réaliste devrait réduire la taille de la mémoire nécessaire pour reproduire les performances de la souris. Pour tester cette hypothèse, nous avons utilisé une variante du modèle dans laquelle nous avons manipulé l'ambiguïté de la représentation d'états : chacun des emplacements A, B, C pourrait être représenté par $n \geq 1$ états, avec $n = 1$ correspondant à des états non ambigus. Comme attendu, le modèle avait besoin d'une mémoire plus petite au fur et à mesure que l'ambiguïté de la représentation augmentait. Nous avons également constaté que le taux d'apprentissage le plus adapté était plus élevé avec des représentations ambiguës, tandis que le facteur aléatoire *randomness* demeurait inchangé quel que soit le niveau d'ambiguïté. Cela corrobore le fait que l'utilisation d'une capacité de mémoire supplémentaire par le modèle est due aux propres limites du modèle plutôt qu'à la nécessité réelle de mémoriser les choix précédents. Par conséquent, cette analyse computationnelle suggère que les souris ont adapté le paramètre *randomness* de leur système de prise de décision afin d'obtenir plus de variabilité au cours des sessions plutôt que de se souvenir des séquences de choix récompensées. Cette conclusion a été renforcée par une série d'arguments comportementaux confirmant l'absence de mémorisation de l'historique des choix dans leur stratégie [6].

5 Discussion

L'exploration et la variabilité des choix sont généralement étudiées en utilisant des tâches avec environnements stochastiques et/ou volatils. Nos résultats permettent d'avancer dans la compréhension des processus sous-jacents à la génération de la variabilité indépendamment des conditions environnementales. Confrontées à une tâche déterministe qui privilégie les séquences de choix complexes, les souris ont évité les répétitions en se rapprochant d'une sélection aléatoire.

Savoir si et comment des séquences aléatoires pourraient être générées par le cerveau a toujours été une énigme. Une première hypothèse est que, chez l'homme, le processus exploiterait la mémoire pour assurer l'égalité d'utilisation des réponses. Une seconde hypothèse suggère que le manque de mémoire pourrait aider à éliminer les biais contre-productifs. Nos résultats plaident en faveur de ce dernier point de vue : les souris n'ont pas utilisé leur mémoire, mais ont plutôt adapté leurs paramètres de prise de décision afin de maximiser le caractère aléatoire des choix.

Notons toutefois que dans notre modèle, la mémoire sert à retenir les séquences réalisées dans un horizon limité dans le but d'apprendre les associations état-action sur la base de ces représentations. D'autres modèles pourraient faire intervenir la mémoire d'une autre manière et expliquer différemment le processus d'exploration ou de génération de variabilité observé dans nos expériences. Dans de futurs travaux, nous testerons donc d'autres modèles afin d'évaluer la capacité d'une stratégie purement déterministe à générer des comportements proches de l'aléatoire et le coût computationnel minimal d'une solution de ce type.

Remerciements

Nous aimerions remercier Elise Bousseyrol, Romain Durand-de Cuttoli, Malou Dongelmans, Etienne K. Duranté, Tarek Ahmed Yahia, Steve Didienne, Bernadette Hanesse and Maxime Come qui ont réalisé les tâches expérimentales. Ce travail a été financé par le Centre National de la Recherche Scientifique CNRS UMR 8246, la Fondation pour la recherche médicale (FRM, Equipe FRM DEQ2013326488 pour P.F), l'Institut national du cancer (Grant TABAC-16-022 pour P.F.) et le Labex SMART (ANR-11-LABX-65) soutenu par les fonds du programme d'Investissements d'Avenir (ANR-11-IDEX-0004-02). Le laboratoire de P.F. et J.N. fait partie du réseau RTRA de l'École des neurosciences de Paris Île-de-France. P.F. et J.N. sont membres du LabEx Bio-Psy.

Références

[1] Richard S Sutton and Andrew G Barto. Toward a modern theory of adaptive networks : expectation and prediction. *Psychological review*, 88(2) :135, 1981.

[2] Wolfram Schultz. Getting formal with dopamine and reward. *Neuron*, 36(2) :241–263, 2002.

[3] Nathaniel D Daw, John P O'doherty, Peter Dayan, Ben Seymour, and Raymond J Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095) :876, 2006.

[4] Daeyeol Lee, Michelle L Conroy, Benjamin P McGreevy, and Dominic J Barraclough. Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive Brain Research*, 22(1) :45–58, 2004.

[5] Dougal GR Tervo, Mikhail Proskurin, Maxim Manakov, Mayank Kabra, Alison Vollmer, Kristin Branson, and Alla Y Karpova. Behavioral variability through stochastic choice and its gating by anterior cingulate cortex. *Cell*, 159(1) :21–32, 2014.

[6] Marwen Belkaid, Elise Bousseyrol, Romain Durand-de Cuttoli, Malou Dongelmans, Etienne K Duranté, Tarek Ahmed Yahia, Steve Didienne, Bernadette Hannesse, Maxime Come, Alexandre Mourot, Jérémie Naudé, Olivier Sigaud, and Philippe Faure. Mice adaptively generate choice variability in a deterministic task. *BioRxiv*, 2019.

[7] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on information theory*, 22(1) :75–81, 1976.

[8] Robert A Rescorla and Allan R Wagner. A theory of pavlovian conditioning : The effectiveness of reinforcement and non-reinforcement. *Classical conditioning II : Current research and theory*, 1972.

[9] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb) :281–305, 2012.