Imitation et Personnalisation de Comportements en Apprentissage par Renforcement Profond

Encadrants : Sylvain Lamprier (SU) - Sylvain.Lamprier@lip6.fr
Olivier Sigaud (SU) - Olivier.Sigaud@upmc.fr
Guillaume Gaudron (Ubisoft) - guillaume.gaudron@ubisoft.com
Lieu du Stage : Jussieu
Durée : 6 mois
Compétences souhaitées : de bonnes compétences en apprentissage par renforcement profond sont requises. Un bon niveau de programmation et la maîtrise d'une librairie de *deep learning* (pyTorch de préférence) sont également demandés.

Thème : Apprentissage par renforcement; Imitation Learning; Decision Transformers

Contexte

Dans un jeu vidéo compétitif en équipe, la perte d'un équipier fragilise naturellement sa "team". C'est d'autant plus injuste lorsque cette perte est liée à un problème de réseau rencontré par un joueur, et peut être temporaire. C'est donc un enjeu de pouvoir, dans cette situation, permettre à un "bot" de prendre le relais, qui dans la mesure du possible va adopter un comportement proche de celui du joueur qui a quitté la partie. La capacité de pouvoir imiter des joueurs humains va encore plus loin, par exemple pour pouvoir tester la faisabilité ou l'équilibre d'une nouvelle zone pour plusieurs styles de joueurs avant son lancement opérationnel.

Une grande difficulté de ce contexte applicatif réside dans le fait que, si l'on dispose généralement de volumes conséquents de traces d'utilisateurs pour l'apprentissage d'un modèle moyen, on n'a souvent accès qu'à un très faible nombre de données pour la personnalisation propre à chaque individu, les traces disponibles étant réparties sur un très grand nombre d'utilisateurs de la plateforme considérée. Par ailleurs, les règles de "privacy" peuvent empêcher de relier des parties à un profil et un historique de parties stockées. La personnalisation des modèles comportementaux doit alors pouvoir être faite sur la base d'un historique très limité d'actions, typiquement le seul historique de la partie en cours. Cela implique d'imaginer des modèles dérivés d'une politique générale, capable de s'adapter rapidement à des comportements spécifiques, ou bien encore d'identifier des hiérarchies de comportements typiques dont l'identification peut se faire sur la base des parties considérées.

Ce cadre du *few-shot imitation learning* a été en premier lieu introduit par le travail présenté dans [1], qui utilise un mécanisme d'attention sur les trajectoires d'entrée pour s'adapter à des nouvelles tâches en exploitant une unique démonstration d'entrée. Lors de l'apprentissage, l'idée du modèle proposé est d'échantillonner des paires de démonstrations

issues d'une même tâche, avec pour objectif d'utiliser la première comme entrée pour identifier la tâche et la seconde comme sortie attendue. En test, une seule démonstration et un état initial sont donnés en entrée du réseau de l'agent pour interagir avec le système. Alors que, dans cette approche, les politiques sont apprises selon des coûts supervisés en fonction des actions attendues à chaque étape des démonstrations (i.e., *Behavioral Cloning* BC) à l'instar de [5], le travail présenté dans [2] reprend la même idée d'utilisation d'une trajectoire de démonstration pour spécifier la tâche, mais travaille par apprentissage par renforcement de politiques, avec une architecture de type *Transformer Network*. L'idée est de pouvoir éventuellement dépasser les performances de la démonstration, qui ne servent donc qu'à fournir des indications sur la tâche, plutôt que de se cantonner à reproduire le comportement présenté. D'autres approches travaillent par apprentissage par renforcement inverse pour découvrir les fonctions de récompenses associées aux démonstrations présentées, et faire en sorte d'apprendre des politiques maximisant ces récompenses cumulées sur chaque tâche. C'est le cas par exemple de [6], qui infère un vecteur de paramètres lié à la tâche en fonction de la démonstration présentée. Ce vecteur est ensuite utilisé par un discriminateur contrastif pour former la fonction de récompense, et par une politique conditionnelle apprise selon un principe de maximum d'entropie. Le stage s'inscrit dans cette ligne de travaux en cherchant à appliquer des méthodes d'imitation efficaces sur des comportements de joueurs en ligne. Lors du déploiement de la politique apprise pour remplacer un joueur déconnecté, le modèle se sert du début de la partie pour inférer le type de joueur (discret ou continu) et conditionner les choix d'action en fonction. On suppose que l'on n'a pas accès à un simulateur pour explorer en phase de test (trop coûteux et dépend souvent d'autres joueurs humains qu'on ne sait pas simuler).

Description

Dans le cadre de ce stage, on propose de considérer l'utilisation de *Decision Transformers* [7] pour ce cadre de *few-shot imitation learning*. L'introduction très récente de ces nouveaux modèles bouscule fortement la manière d'aborder les tâches de renforcement, du moins à partir de données hors-ligne. L'idée est, à partir d'une collection de trajectoires issues d'un environnement, d'apprendre à prédire l'action à appliquer à chaque étape t en fonction de l'historique d'états rencontrés ainsi que de la récompense cumulée désirée à partir de l'état courant. En test, il s'agit de fournir la récompense cumulée maximale à chaque pas de temps, de manière à ce que la politique apprise tende à fournir les actions optimales en fonction des états d'entrée et de la tâche considérée. Nous proposons d'étendre ce type d'architecture au cadre multi-tâches, où chaque tâche est un joueur ou un groupe de joueurs à imiter. L'idée est de considérer des vecteurs de récompenses désirées pour spécifier la tâche, en considérant chaque dimension comme représentant un type de profil. Soit un ensemble de profils prédéfinis $W = (w_1, ...w_m)$ pour chacun desquels on dispose de traces d'exécution D_i (pour tout $i \in \{1; ...; m\}$). On pourra par exemple considérer un vecteur de récompenses cumulées $R_t = (R_{t,1}, ..., R_{t,m})$ tel que, pour toute démonstration $\tau \in \cup_i D_i$ et pour tout i et tout t: $R_{t,i} = \sum_{s=0}^{T-s} \gamma^s P_\theta(\tau \in D_i | \tau_{1:t+s})$, avec P_θ un réseau *Transformer* probabiliste appris pour prédire le profil de τ par entropie croisée sur le corpus de

démonstrations. L'idée est ensuite d'apprendre une politique de type *Decision Transformer* en utilisant ce type de récompenses multivariées en entrée. Lors du déploiement, on pourra soit choisir le type de comportement que l'on souhaite produire en ajustant le R cible manuellement, soit inférer le vecteur de récompenses le plus adapté à l'historique de partie observé selon le réseau P_θ. Une variante pourra également considérer la formation dynamique des différents profils selon le corpus de démonstrations, par exemple par maximisation de l'information mutuelle entre le vecteur de récompenses choisi pour chaque τ et le profil w_i dont elle est issue, à la manière de [6].

Le stage pourra donc s'articuler sur les différentes étapes suivantes :

1. Implémentation de méthodes de l'état de l'art telles que [1] et [6] et expérimentations sur des environnements de référence à définir (par exemple des environnements tels que Mario ou vizDoom et idéalement des environnements issus de l'univers de Ubisoft). Les profils de joueurs à considérer pourront être issus de bots scriptés plus ou moins performants, appris ou suivant divers types de stratégies.

2. Apprentissage de *Decision Transformers* sur un profil de joueur unique et expérimentations.

3. Extension au cadre multi-profils selon un réseau P_θ appris sur un corpus de démonstrations partitionné selon les différents types de profils connus. Étude de l'espace de vecteurs R en fonction des comportements obtenus. Expérimentations en situation pour le remplacement de joueurs déconnectés avec inférence de profil à partir de l'historique de la partie en cours.

4. Extension au cadre de profils dynamiques et expérimentations.

Le stage s'inscrit dans le cadre d'une collaboration informelle avec "La Forge France", le laboratoire français de recherche d'Ubisoft. Il est prévu de consacrer, outre le temps passé à Jussieu, un certain temps à l'échange avec les chercheurs et ingénieurs de l'entreprise à Montreuil. L'objectif est la réalisation d'un prototype permettant la publication de l'approche dans une conférence de premier plan type ICLR, ainsi que le renforcement des relations de collaboration entre SU et Ubisoft pouvant ouvrir à des perspectives de recherche fructueuses dans les prochaines années (dans le cadre de thèses CIFRE par exemple).

Références

1. Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., ... & Zaremba, W. (2017). One-shot imitation learning. *arXiv preprint arXiv:1703.07326*.
2. Cachet, T.. R., Perez, J., & Dance, C. (2021, July). Conditioned Reinforcement Learning for Few-Shot Imitation. In *International Conference on Machine Learning* (pp. 2376-2387). PMLR.
3. Yang, R., Sun, X., & Narasimhan, K. (2019). A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *arXiv preprint arXiv:1908.08342*.

4. Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.
5. Dasari, S., & Gupta, A. (2020). Transformers for one-shot visual imitation. *arXiv preprint arXiv:2011.05970*.
6. Yu, L., Yu, T., Finn, C., & Ermon, S. (2019). Meta-inverse reinforcement learning with probabilistic context variables. *arXiv preprint arXiv:1909.09314*.
7. Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., ... & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*.