

TD2 – Cours de robotique développementale et cognitive

Algorithmes d'exploration de modèles sensorimoteurs : la librairie Explauto

Pierre-Yves Oudeyer et Pierre Rouanet

Dans le cours de robotique développementale, nous avons discuté de l'importance des mécanismes/algorithmes permettant à des robots d'apprendre des modèles directs et inverses de leur corps. Par exemple, il s'agit pour un robot doté d'un bras à N degrés de libertés (mais à la forme et aux dimensions inconnues) d'apprendre à prédire quelles sont les positions de sa main en fonction des valeurs de ses commandes motrices (angles de ses moteurs par exemple), et d'apprendre le modèle inverse permettant de trouver un ensemble de commandes motrices permettant d'atteindre une cible donnée avec la main. Deux mécanismes en particulier sont mis en œuvre dans ce processus :

- 1) des algorithmes d'exploration (aussi appelés « modèles d'intérêt ») permettant au robot de choisir ses « expériences » et de collecter des données d'apprentissage (associant des valeurs M de ses moteurs à des valeurs S de ses senseurs (e.g. position de la main))
- 2) des algorithmes permettant d'utiliser ces données pour construire/apprendre incrémentalement un modèle direct et/ou un modèle inverse (combinant souvent des méthodes de régression et d'optimisation). On les appelle algorithmes d'apprentissage de modèles sensorimoteurs.

Explauto est une librairie en python permettant d'expérimenter et d'étudier les propriétés de différents algorithmes d'exploration et d'apprentissage de modèles sensorimoteurs. Cette librairie est décrite dans l'article suivant :

[Explauto: an open-source Python library to study autonomous exploration in developmental robotics](#)

Clément Moulin-Frier; Pierre Rouanet; Pierre-Yves Oudeyer
ICDL-Epirob – International Conference on Development and Learning,
Epirob, Oct 2014, Genoa, Italy.

<https://hal.inria.fr/hal-01061708/document>

Elle est disponible sur github : <https://github.com/flowersteam/explauto>

Plusieurs tutoriels expliquent le fonctionnement de cette librairie ainsi que la manière de mettre en œuvre, d'analyser et de comparer différents algorithmes. Ces tutoriels sont basés sur l'utilisation de notebooks Jupyter/Ipython (<https://jupyter.org>), qui permettent de générer des documents mélangeant texte, équations, images et code interactif (pour utiliser Jupyter/Ipython, le plus simple est d'installer Anaconda : <https://www.continuum.io/downloads>).

L'objectif de ce TD consiste à réaliser les exercices suivants, et le rendu se fera sous la forme d'un ou plusieurs notebooks Jupyter, par exemple déposé sur github, et qui présenteront le code mais aussi les visualisations et les explications/justifications de ce que vous aurez réalisés.

Dans ces exercices, vous utiliserez l'environnement 'simple-arm' modélisant un bras plan à n degrés de libertés (voir : http://nbviewer.ipython.org/github/flowersteam/explauto/blob/master/notebook/setting_environments.ipynb).

Afin de préparer les exercices, il vous est recommandé de commencer par faire le tutorial suivant :

http://nbviewer.ipython.org/github/flowersteam/explauto/blob/master/notebook/full_tutorial.ipynb

Exercice 1 :

Pour les différentes configurations du bras (['mid_dimensional', 'default', 'high_dim_high_s_range', 'low_dimensional', 'high_dimensional']), montrer une visualisation de la distribution des positions du bout du bras quand on explore aléatoirement les valeurs des commandes motrices (selon une distribution uniforme, c'est ce qu'on appelle le « random motor babbling/exploration »). Qu'observez vous ? Est-ce que la dimensionnalité du bras a une influence sur ce qu'on observe ? Est-ce qu'une telle méthode d'exploration vous paraît efficace pour collecter des exemples d'apprentissages permettant de construire un modèle direct ? un modèle inverse ? Pourquoi ?

Exercice 2 :

Plusieurs algorithmes d'apprentissage de modèles sensorimoteurs sont disponibles dans Explauto. Nous considérerons ici les modèles : 'nearest-neighbour', 'lwlr-CMAES' et 'WNN' tels que décrits dans le notebooks :

https://github.com/flowersteam/explauto/blob/master/notebook/summary_available_models.ipynb

En utilisant la méthode d'exploration « random motor babbling », et pour la configuration du bras 'default', comparer les performances de ces trois modèles: 'nearest-neighbour', 'lwlr-CMAES' et 'WNN'. Pour cela, montrer et comparer les visualisations de :

- 1) l'évolution des performances en prédiction du modèle direct : tous les N pas de temps, tester l'erreur moyenne (et sa déviation standard) du modèle pour prédire la position du bout du bras correspondant à 300 commandes motrices générées aléatoirement et indépendamment des points explorés et utilisés pour construire le modèle (c'est ce qu'on appelle la performance en généralisation). Bien entendu, les données observées pendant les tests du modèle ne doivent pas être utilisées pour mettre à jour le modèle.
- 2) l'évolution des performances du modèle inverse : tous les N pas de temps, tester l'erreur moyenne (et sa déviation standard) du modèle pour atteindre 300 positions du bout du bras générées aléatoirement et indépendamment des points explorés/atteints pendant l'apprentissage. Bien entendu, les données observées pendant les tests du modèle ne doivent pas être utilisées pour mettre à jour le modèle.

Qu'observez vous ? Donner une interprétation.

Exercice 3

Faire la même comparaison que l'exercice 2 mais en utilisant cette fois-ci l'algorithme d'exploration « random goal babbling » tel que décrit dans le notebook :

http://nbviewer.ipython.org/github/flowersteam/explauto/blob/master/notebook/comparing_motor_goal_strategies.ipynb

et dans l'article :

[Exploration strategies in developmental robotics: a unified probabilistic framework](#) Clément Moulin-Frier; Pierre-Yves Oudeyer

ICDL-Epirob – International Conference on Development and Learning, Epirob, Aug 2013, Osaka, Japan.

Exercice 4

Sur la base des résultats des deux exercices précédents, comparez les méthodes de random motor babbling et random goal babbling, et proposez une interprétation.

Exercice 5

Après avoir expérimenté les différents algorithmes d'exploration et d'apprentissage de modèles avec un bras robotique planaire dans les exercices précédents, l'objectif de cet exercice consistera à utiliser certaines de ces méthodes sur le robot Poppy Humanoid. En suivant le notebook qui suit, vous réaliserez dans une première phase une expérience dans laquelle vous utilisez un des algorithmes Explauto afin de permettre au modèle simulé Poppy Humanoid d'apprendre la cinématique inverse de son bras. Dans une seconde phase, vous ré-utiliserez un de ces modèles appris en simulation afin de contrôler la main du robot Poppy Humanoid réel. En particulier vous essaierez de lui faire réaliser une trajectoire circulaire, puis une trajectoire en forme de carré.

Notebook :

<https://goo.gl/Ly55cW>

Exercice 6 - Noté

A l'étape 4 du TD1, il vous était demandé d'imaginer une méthode permettant à un robot humanoïde Poppy Torso d'apprendre comment faire bouger un objet sur une table devant lui (selon des vecteurs de translations tests fixés par l'expérimentateur). En ré-utilisant les méthodes étudiées avec Explauto, proposez une ou plusieurs manières d'utiliser Explauto afin d'améliorer votre méthode initiale, et présentez (et justifiez) des mesures d'évaluation permettant de les comparer.

Le notebook que vous réaliserez pour l'exercice 6 est à rendre (sous la forme d'un fichier publié et visualisable sur github) et donnera lieu à une note qui prendra en compte tant la réalisation technique de l'expérimentation que le caractère pédagogique de ses explications.