PIAD 2013

Sujet 32 : Semi-supervised Learning Agents

Auteurs : Lan Zhou & Matthieu Zimmer
Encadrants : Paolo Viappiani & Paul Weng

Université Pierre et Marie Curie

14 Mai

PIAD 2013
Sujet 32 : Semi-supervised Learning Agents
Auteurs : Lan Zhou & Matthieu Zimmer
Encadrants : Paolo Viappiani & Paul Weng
Université Pierre et Marie Curie
14 Mai

L'apprentissage semi-supervisé

Plan

Plan

Théorie
Processus de décision Markovien
Apprentissage par Renforcement

Pratique
TORCS
Feedbacks

- et ce qu'on propose dans notre bibliothèque
- quelques algorithmes sarsa , qlearning
- Mise en pratique

Introduction

Problématique

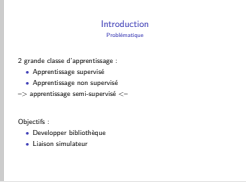
2 grande classe d'apprentissage :

- Apprentissage supervisé
- Apprentissage non supervisé

–> apprentissage semi-supervisé <–

Objectifs :

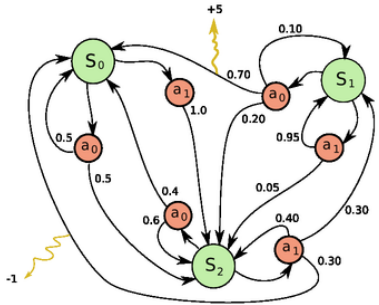
- Developper bibliothèque
- Liaison simulateur



Réseau de neurones, ...
Parfois les agents apprendrons d'eux meme, quelques fois du superviseur
Implémentation C++
Pour arriver à un apprentissage semi supervisé, nous allons partir d'un apprentissage non supervisé : l'APR. Lan va donc vous introduire aux notions de bases.

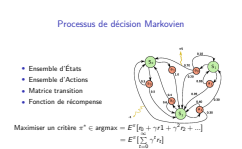
Processus de décision Markovien

- Ensemble d'États
- Ensemble d'Actions
- Matrice transition
- Fonction de récompense



Maximiser un critère $\pi^* \in \text{argmax} = E^{\pi}[r_0 + \gamma r1 + \gamma^2 r_2 + ...]$

$$= E^{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t]$$



Un agent

Les résultats des actions sont indéterminées

Pour finir, le problème revient à trouver une politique qui maximise l'espérance des récompenses.

Apprentissage non supervisé

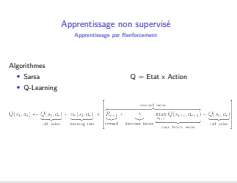
Apprentissage par Renforcement

Algorithmes

- Sarsa
- Q-Learning

Q = Etat x Action

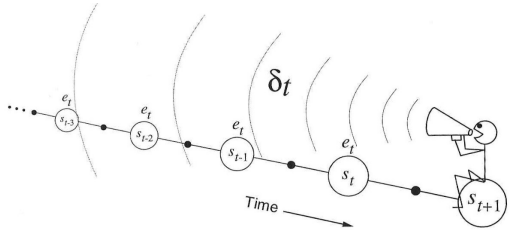
$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})}_{\text{max future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right]$$



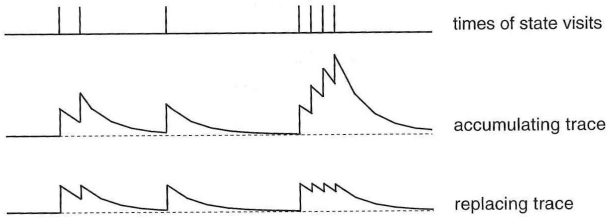
On s'est concentré sur 2 grands algorithmes : SARSA & Q-Learning que nous avons implémenté dans notre bibliothèque
Les 2 utilisent un tableau Q qui permet de facilement retrouver la meilleur action pour un état donné
La formule de mise à jour durant l'apprentissage est la suivante

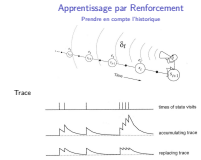
Apprentissage par Renforcement

Prendre en compte l'historique



Trace





Un des problèmes de la version précédente est qu'il n'y a pas de notion d'historique.
Si je fais une erreur ou quelques choses de bien maintenant, c'est uniquement grâce à ma dernière action. Ca prolongue beaucoup la phase d'apprentissage.
Mais ça ne suffit toujours pas.

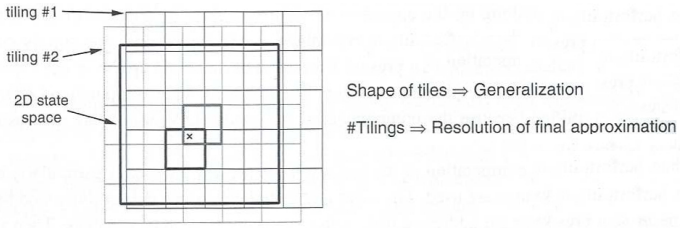
Fonction d'approximation

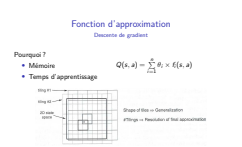
Descente de gradient

Pourquoi ?

- Mémoire
- Temps d'apprentissage

$$Q(s,a) = \sum_{i=1}^{n} \theta_i \times f_i(s,a)$$





l'apprentissage par renforcement «de base» n'est pas extensible à très grand espace d'état, car il faut maintenir dans la mémoire très grandes matrices
On va donc chercher maintenant à approximer Q(s,a) par cette fonction.
On notera que les états ne sont alors plus discrétisés.
On gagne en mémoire, plus qu'on tableau teta, et en généralisation, ce qu'il apprend dans une configuration il peut le généraliser.

Voici qui clos la partie sur l'APR.

Semi supervisé

3 idées

- Agir sur les récompenses
- Agir sur le choix des actions
- Agir sur la stratégie (exploitation/exploration)

Semi supervisé
3 idées
• Agir sur les récompenses
• Agir sur le choix des actions
• Agir sur la stratégie (exploitation/exploration)

Comment intégrer une supervision ?
Un tuteur va surveiller l'agent de temps en temps et complémenter la fonction de récompense
contrôler l'agent

Plan

Plan
Théorie
Processus de décision Markovien
Apprentissage par Renforcement
Pratique
TORCS
Feedbacks

Nous passons donc maintenant à la mise en pratique et à la liaison avec le simulateur.

TORCS

TORCS

Le simulateur que nous avons choisi avec nos encadrants est TORCS.
Il permet de simuler des courses de voitures en 3D pour faire de jolies démos.
Il est open-source et écris en C/C++, très personnalisable, il permet d'intégrer des modules indépendants pour les agents.
Il permet également d'être simuler sans GUI pour que les agents puissent apprendre.

Les entrées

Capteurs locaux -> généralisation des pistes

- distance au centre
- angle tangent
- vitesse
- longueur segment
- angle prochain virage

Les entrées
Capteurs locaux -> généralisation des pistes
- distance au centre
- angle tangent
- vitesse
- longueur segment
- angle prochain virage

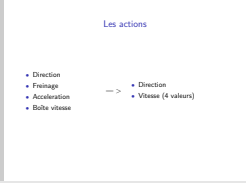
[Utiliser l'image]

Les actions

- Direction
- Freinage
- Acceleration
- Boîte vitesse

— >

- Direction
- Vitesse (4 valeurs)



Dans TORCS il y a en 4 actions possible, qu'on a fusionné en 2 pour simplifier le problème.
Le calcul du changement de boîte à vitesse est automatique.
4 valeurs : acc, ne rien faire, freiné, reculé.

Les récompenses

Sur route et avance :

- vitesse $\in [100; 6000]$

Avance pas : - 500
Avance dans un mur : -1000
Bloqué et Recule : 15
Sens inverse

- vitesse $\in [-4000; -2000]$

Sort de route

- écart $\in [-1000; -4000]$

Les récompenses
Sur route et avance :
• vitesse ∈ [100; 6000]
Avance pas : - 500
Avance dans un mur : -1000
Bloqué et Recule : 15
Sens inverse
• vitesse ∈ [−4000; −2000]
Sort de route
• écart ∈ [−1000; −4000]

Demo

+ Quelques stats

Demo
+ Quelques stats

L'autre objectif du projet était d'intégrer des feedbacks d'un superviseur à TORCS.

Mode superviseur

Agir sur les récompenses

Mode superviseur
Agir sur les récompenses

L'autre objectif du projet était d'intégrer des feedbacks d'un superviseur à TORCS.

Mode contrôle

Agir sur les actions
Montrer à l'agent comment conduire Démo

Mode contrôle
Agir sur les actions
Montrer à l'agent comment conduire Démo

Merci aux encadrants, ...

Conclusion

Aller plus loin...

Thank you for listening. We'll be pleased to try to respond to any of your questions.

References & Remerciements

- Reinforcement Learning : An Introduction, 1998
  Sutton, Richard S and Barto, Andrew G
- Image Wikipédia : Apprentissage par renforcement

References & Remerciements

- Reinforcement Learning : An Introduction, 1998
  Sutton, Richard S and Barto, Andrew G
- Image Wikipédia : Apprentissage par renforcement