

Apprentissage par Renforcement Hors Ligne (BATCH)

Alexandre Gerussi, Léo Pérard, Lucas Seguinot

M2 MOCAD - IIR

15 décembre 2015

Plan de l'exposé

- 1 Introduction
- 2 Principes généraux
- 3 Kernel Based Approximate Dynamic Programming
- 4 Fitted Q-iteration
- 5 Least-Squares Policy Iteration

Pourquoi batch ?

- en ligne: interactions libres, voir illimitées avec l'environnement
- pas toujours possible
 - sondages
 - "conduite de vélo": nécessite un opérateur humain
 - ?? : casse du matériel en cas d'échec

Types de batch

- pure batch
- growing batch
- semi-batch

Principes généraux

- nombreux algorithmes, différant essentiellement sur des détails
- itérations dans SARSA ou Q-learning: exploration + convergence
- \Rightarrow **experience replay**: faire converger sans explorer
- itérations en-ligne: locales, propagation aléatoire par les itérations
- \Rightarrow **fitting**: accélérer et stabiliser la propagation en globalisant les MAJ

Kernel Based Approximate Dynamic Programming

- ??

Fitted Q-iteration

Itération sur la valeur

$$Q^{i+1}(s, a) = \sum_{s' \in S} \mathcal{T}_{ss'}^a (\mathcal{R}_{ss'}^a + \gamma \max_{a' \in A} Q^i(s', a'))$$

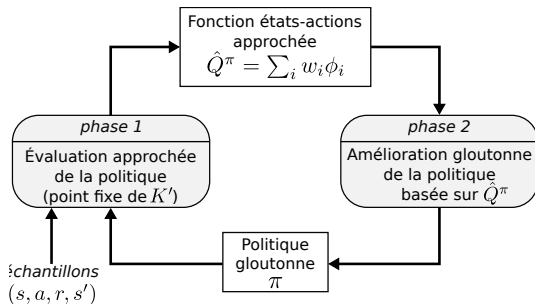
- \mathcal{F} ensemble de transitions (s, a, r, s')
- Base de donnée d'apprentissage P :

$$(s, a) \rightarrow r + \gamma \max_{a' \in A} \hat{Q}^i(s', a')$$

- Apprentissage supervisé $\rightarrow \hat{Q}^{i+1}$

Least-Squares Policy Iteration

$$Q = K(Q) \longleftrightarrow Q = K'(Q) = \mathcal{P}_Q^\perp(K(Q))$$



Application: équilibre et conduite d'un vélo

- rester debout et atteindre un but en vélo
- valeurs sous contrôle:
 - force rotatoire à appliquer au guidon
 - placement du centre de masse par rapport au vélo
- pûrement hors-ligne à partir de quelques milliers de trajectoires effectuées aléatoirement
- experience replay: quelques passes de l'ensemble des données font converger