Apprentissage par renforcement

Yann Ollivier, Facebook AI, Paris

Apprentissage par renforcement : un agent interagit avec un environnement (ou un adversaire). L'agent choisit des actions et l'environnement lui renvoie des récompenses et des observations. L'agent veut sélectionner des actions maximisant sa récompense.

Apprentissage par renforcement

Yann Ollivier, Facebook AI, Paris

Apprentissage par renforcement : un agent interagit avec un environnement (ou un adversaire). L'agent choisit des actions et l'environnement lui renvoie des récompenses et des observations. L'agent veut sélectionner des actions maximisant sa récompense.

Action $a_1 \Longrightarrow$ récompense r_1 et observation $o_1 \Longrightarrow$ action $a_2 \Longrightarrow$ récompense r_2 et observation $o_2 \Longrightarrow$ etc.

Apprentissage par renforcement

Yann Ollivier, Facebook AI, Paris

Apprentissage par renforcement : un agent interagit avec un environnement (ou un adversaire). L'agent choisit des actions et l'environnement lui renvoie des récompenses et des observations. L'agent veut sélectionner des actions maximisant sa récompense.

Action $a_1 \Longrightarrow$ récompense r_1 et observation $o_1 \Longrightarrow$ action $a_2 \Longrightarrow$ récompense r_2 et observation $o_2 \Longrightarrow$ etc.

Problème : trouver une séquence d'actions (sachant les observations précédentes) qui maximise la récompense totale

Apprentissage par renforcement

Yann Ollivier, Facebook AI, Paris

Apprentissage par renforcement : un agent interagit avec un environnement (ou un adversaire). L'agent choisit des actions et l'environnement lui renvoie des récompenses et des observations. L'agent veut sélectionner des actions maximisant sa récompense.

Action $a_1 \Longrightarrow$ récompense r_1 et observation $o_1 \Longrightarrow$ action $a_2 \Longrightarrow$ récompense r_2 et observation $o_2 \Longrightarrow$ etc.

Problème : trouver une séquence d'actions (sachant les observations précédentes) qui maximise la récompense totale, ou l'espérance de la récompense totale si l'environnement est stochastique.

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire
- Marchés financiers

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire
- Marchés financiers
- Robots pour des tâches spécifiques

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire
- Marchés financiers
- Robots pour des tâches spécifiques
- Humain ?

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire
- Marchés financiers
- Robots pour des tâches spécifiques
- Humain ? et intelligence artificielle générique

Apprentissage par renforcement : exemples

- Jeux : échecs, go, jeu vidéo...
- Contrôle optimal de trajectoire
- Marchés financiers
- Robots pour des tâches spécifiques
- Humain ? et intelligence artificielle générique
- Quelles récompenses ?

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

  $\Longrightarrow$ La fonction valeur dans un environnement incertain

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

  $\Longrightarrow$ La fonction valeur dans un environnement incertain

  $\Longrightarrow$ Intelligence artificielle : le meilleur agent rationnel "universel" ?

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

  $\Longrightarrow$ La fonction valeur dans un environnement incertain

  $\Longrightarrow$ Intelligence artificielle : le meilleur agent rationnel "universel" ?

- Comment le calculer ?

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

  $\Longrightarrow$ La fonction valeur dans un environnement incertain

  $\Longrightarrow$ Intelligence artificielle : le meilleur agent rationnel "universel" ?

- Comment le calculer ?

  $\Longrightarrow$ Solutions exactes

Apprentissage par renforcement

- Que veut-on calculer ? Qu'est-ce qu'un agent optimal ?

  $\Longrightarrow$ La fonction valeur dans un environnement incertain

  $\Longrightarrow$ Intelligence artificielle : le meilleur agent rationnel "universel" ?

- Comment le calculer ?

  $\Longrightarrow$ Solutions exactes

  $\Longrightarrow$ Solutions approximatives : apprentissage statistique, méthodes Monte Carlo, réseaux de neurones...

Minimax: le cas des jeux à information totale

Si les règles du jeu (l'environnement) sont connues et que l'état du jeu et visible à tout instant, le comportement optimal est en principe donné par l'algorithme minimax.

Minimax: le cas des jeux à information totale

Si les règles du jeu (l'environnement) sont connues et que l'état du jeu et visible à tout instant, le comportement optimal est en principe donné par l'algorithme minimax.

L'action optimale à l'instant t est

$$a_t^* = \arg\max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

où les o_t sont les actions de l'adversaire (observations données par l'environnement) et où $r(a_1, o_1, \ldots, a_t, o_t, \ldots)$ est la récompense finale issue de ces actions.

Minimax: le cas des jeux à information totale

Si les règles du jeu (l'environnement) sont connues et que l'état du jeu et visible à tout instant, le comportement optimal est en principe donné par l'algorithme minimax.

L'action optimale à l'instant t est

$$a_t^* = \arg\max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

où les o_t sont les actions de l'adversaire (observations données par l'environnement) et où $r(a_1, o_1, \ldots, a_t, o_t, \ldots)$ est la récompense finale issue de ces actions.

L'action optimale a_t^* est celle qui maximise la fonction valeur du prochain état du jeu.

Minimax: le cas des jeux à information totale

La fonction valeur pour l'état $(a_1, \ldots, o_{t-1})$ est

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

Minimax: le cas des jeux à information totale

La fonction valeur pour l'état $(a_1, \ldots, o_{t-1})$ est

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

(Rq : contient à la fois V et Q)

Minimax: le cas des jeux à information totale

La fonction valeur pour l'état $(a_1, \ldots, o_{t-1})$ est

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

(Rq : contient à la fois V et Q)

La fonction valeur peut être calculée par récurrence en partant de la fin du jeu pour toutes les parties possibles :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} V(a_1, \ldots, o_{t-1}, a_t, o_t)$$

Minimax: le cas des jeux à information totale

La fonction valeur pour l'état $(a_1, \ldots, o_{t-1})$ est

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \max_{a_{t+1}} \min_{o_{t+1}} \ldots \max_{a_T} \min_{o_T} r(a_1, o_1, \ldots, a_t, o_t, \ldots a_T, o_T)$$

(Rq : contient à la fois V et Q)

La fonction valeur peut être calculée par récurrence en partant de la fin du jeu pour toutes les parties possibles :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} V(a_1, \ldots, o_{t-1}, a_t, o_t)$$

L'algorithme correspondant doit parcourir (presque) tout l'arbre des parties possibles dans le jeu : rarement raisonnable.

L'équation de Bellman

S'il peut y avoir des récompenses à chaque instant, la fonction valeur satisfait

L'équation de Bellman

S'il peut y avoir des récompenses à chaque instant, la fonction valeur satisfait l'équation de Bellman

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement adversarial) ou

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement stochastique).
Les o_t encodent les réactions de l'environnement.

L'équation de Bellman

S'il peut y avoir des récompenses à chaque instant, la fonction valeur satisfait l'équation de Bellman

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement adversarial) ou

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement stochastique).
Les o_t encodent les réactions de l'environnement.

En général, la loi de o_t dans $\mathbb{E}_{o_t}$ n'est pas connue.

L'équation de Bellman

S'il peut y avoir des récompenses à chaque instant, la fonction valeur satisfait l'équation de Bellman

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \min_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement adversarial) ou

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \left\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \right\}$$

(environnement stochastique).
Les o_t encodent les réactions de l'environnement.

En général, la loi de o_t dans $\mathbb{E}_{o_t}$ n'est pas connue. La valeur de la récompense r_t sachant les actions et observations précédentes peut aussi être inconnue.

Modèles de l'environnement

Si on dispose d'un modèle de l'environnement (exemple : contrôle optimal stochastique), on peut optimiser les espérances calculées dans le modèle.

Modèles de l'environnement

Si on dispose d'un modèle de l'environnement (exemple : contrôle optimal stochastique), on peut optimiser les espérances calculées dans le modèle.
Un modèle $\mathcal{M}$ est une loi de probabilité $p_{\mathcal{M}}$ sur les prochaines récompenses et les réactions de l'environnement, sachant le passé.

Modèles de l'environnement

Si on dispose d'un modèle de l'environnement (exemple : contrôle optimal stochastique), on peut optimiser les espérances calculées dans le modèle.
Un modèle $\mathcal{M}$ est une loi de probabilité $p_{\mathcal{M}}$ sur les prochaines récompenses et les réactions de l'environnement, sachant le passé.
La fonction valeur est alors

$$V(a_1, \ldots, o_{t-1}) =$$
$$\max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T}$$
$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right)$$

Couvre aussi les environnements connus ou simulables (self-play...).

Incertitude sur les modèles de l'environnement

Un modèle d'un environnement réaliste est forcément approximatif.
Les observations consécutives fournissent de plus en plus d'information sur l'environnement.

Incertitude sur les modèles de l'environnement

Un modèle d'un environnement réaliste est forcément approximatif. Les observations consécutives fournissent de plus en plus d'information sur l'environnement.

$\Longrightarrow$ Prior bayésien sur l'environnement $\mathcal{M}$:

Incertitude sur les modèles de l'environnement

Un modèle d'un environnement réaliste est forcément approximatif. Les observations consécutives fournissent de plus en plus d'information sur l'environnement.

$\Longrightarrow$ Prior bayésien sur l'environnement $\mathcal{M}$:

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T} \sum_{\mathcal{M}}$$

$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right) \times \text{prior}(\mathcal{M})$$

Incertitude sur les modèles de l'environnement

Un modèle d'un environnement réaliste est forcément approximatif. Les observations consécutives fournissent de plus en plus d'information sur l'environnement.

$\Longrightarrow$ Prior bayésien sur l'environnement $\mathcal{M}$:

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T} \sum_{\mathcal{M}}$$

$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right) \times \text{prior}(\mathcal{M})$$

Peut être vu comme un problème non bayésien avec environnement connu, en passant dans l'espace des lois postérieures.

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?
- L'exploration acquiert automatiquement une valeur canonique :

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?
- L'exploration acquiert automatiquement une valeur canonique : des actions qui réduisent l'incertitude sur l'environnement augmentent la fonction valeur.

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?
- L'exploration acquiert automatiquement une valeur canonique : des actions qui réduisent l'incertitude sur l'environnement augmentent la fonction valeur.
- Non-oubli asymptotique du prior si certaines actions ont des conséquences permanentes.

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?
- L'exploration acquiert automatiquement une valeur canonique : des actions qui réduisent l'incertitude sur l'environnement augmentent la fonction valeur.
- Non-oubli asymptotique du prior si certaines actions ont des conséquences permanentes.
  Les actions perçues comme suicidaires par le prior ne sont jamais explorées.

Approche bayésienne : conséquences

- Seule approche raisonnable en environnement très incertain
- Choix du prior ?
- L'exploration acquiert automatiquement une valeur canonique : des actions qui réduisent l'incertitude sur l'environnement augmentent la fonction valeur.
- Non-oubli asymptotique du prior si certaines actions ont des conséquences permanentes.
  Les actions perçues comme suicidaires par le prior ne sont jamais explorées.
  (Disparaît en environnement stationnaire.)

Priors bayésiens pour l'environnement

- Situations simples (ex : bandits manchots) : priors explicites.

Priors bayésiens pour l'environnement

- Situations simples (ex : bandits manchots) : priors explicites. Origine historique des compromis exploration-exploitation type UCB.

Priors bayésiens pour l'environnement

- Situations simples (ex : bandits manchots) : priors explicites. Origine historique des compromis exploration-exploitation type UCB.

- Environnements réels, intelligence artificielle générale : on doit capturer toutes les régularités de l'environnement (lois de la physique...)

Priors bayésiens pour l'environnement

- Situations simples (ex : bandits manchots) : priors explicites. Origine historique des compromis exploration-exploitation type UCB.

- Environnements réels, intelligence artificielle générale : on doit capturer toutes les régularités de l'environnement (lois de la physique...)

  Approche : lister tous les univers possibles en privilégiant les explications les plus simples.

Un prior bayésien universel ?

Prior universel de Solomonoff [Solomonoff 1964]

$$\text{prior}(\mathcal{M}) = 2^{-K(\mathcal{M})}$$

où la complexité de Kolmogorov $K(\mathcal{M})$ est la longueur du plus petit programme (machine de Turing) qui simule l'environnement $\mathcal{M}$.

Un prior bayésien universel ?

Prior universel de Solomonoff [Solomonoff 1964]

$$\text{prior}(\mathcal{M}) = 2^{-K(\mathcal{M})}$$

où la complexité de Kolmogorov $K(\mathcal{M})$ est la longueur du plus petit programme (machine de Turing) qui simule l'environnement $\mathcal{M}$.

$\Longrightarrow$ Le problème général de l'induction est mathématiquement bien posé.

L'agent optimal universel (AIXI)

Ce prior définit l'agent optimal universel [AIXI, Hutter 2005]

L'agent optimal universel (AIXI)

Ce prior définit l'agent optimal universel [AIXI, Hutter 2005]

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T} \sum_{\mathcal{M}}$$

$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right) \times 2^{-K(\mathcal{M})}$$

L'agent optimal universel (AIXI)

Ce prior définit l'agent optimal universel [AIXI, Hutter 2005]

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T} \sum_{\mathcal{M}}$$

$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right) \times 2^{-K(\mathcal{M})}$$

. . . une optimisation sur toutes les séquences d'actions possibles, d'une somme sur tous les univers possibles.

L'agent optimal universel (AIXI)

Ce prior définit l'agent optimal universel [AIXI, Hutter 2005]

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \int_{r_t, o_t} \max_{a_{t+1}} \int_{r_t, o_t} \ldots \max_{a_T} \int_{r_T, O_T} \sum_{\mathcal{M}}$$

$$(r_t + \cdots + r_T) \times p_{\mathcal{M}} \left(r_{[t;T]}, o_{[t;T]} \mid a_{[1:T]}, o_{[1:t-1]}, r_{[1:t-1]} \right) \times 2^{-K(\mathcal{M})}$$

. . . une optimisation sur toutes les séquences d'actions possibles, d'une somme sur tous les univers possibles.

Meilleure définition mathématique, à ce jour, d'une intelligence artificielle générique.

Apprentissage par renforcement

- Que veut-on calculer ?

- Comment le calculer ?

Calcul de la fonction de Bellman

Environnements simples : solutions exactes

Calcul de la fonction de Bellman

Environnements simples : solutions exactes

- Jeux de petite taille (morpion...) : exploration exhaustive de l'arbre

Calcul de la fonction de Bellman

Environnements simples : solutions exactes

- Jeux de petite taille (morpion...) : exploration exhaustive de l'arbre
- Solutions algébriques exactes :
  - Contrôle optimal pour les environnements stochastiques linéaires avec coûts quadratiques

Calcul de la fonction de Bellman

Environnements simples : solutions exactes

- Jeux de petite taille (morpion...) : exploration exhaustive de l'arbre
- Solutions algébriques exactes :
  - Contrôle optimal pour les environnements stochastiques linéaires avec coûts quadratiques
  - Modèles financiers simples

Calcul de la fonction de Bellman

Environnements simples : solutions exactes

- Jeux de petite taille (morpion...) : exploration exhaustive de l'arbre
- Solutions algébriques exactes :
  - Contrôle optimal pour les environnements stochastiques linéaires avec coûts quadratiques
  - Modèles financiers simples
- Solutions algébriques asymptotiques : bandits bayésiens

Calcul de la fonction de Bellman

Environnements complexes : solutions approchées

Calcul de la fonction de Bellman

Environnements complexes : solutions approchées

- Encodage à la main d'une fonction valeur approximative (approche historique : échecs...)

Calcul de la fonction de Bellman

Environnements complexes : solutions approchées

- Encodage à la main d'une fonction valeur approximative (approche historique : échecs...)
- Exploration aléatoire dirigée de l'arbre des actions (méthodes Monte Carlo)

Calcul de la fonction de Bellman

Environnements complexes : solutions approchées

- Encodage à la main d'une fonction valeur approximative (approche historique : échecs...)
- Exploration aléatoire dirigée de l'arbre des actions (méthodes Monte Carlo)
- Utilisation de modèles (réseaux de neurones...) pour extrapoler des cas connus vers les situations nouvelles

Calcul de la fonction de Bellman

Environnements complexes : solutions approchées

- Encodage à la main d'une fonction valeur approximative (approche historique : échecs...)
- Exploration aléatoire dirigée de l'arbre des actions (méthodes Monte Carlo)
- Utilisation de modèles (réseaux de neurones...) pour extrapoler des cas connus vers les situations nouvelles
- Recherche directe d'un modèle paramétrique (réseau de neurones) satisfaisant approximativement l'équation de Bellman, par modifications sucessives.

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

- Commencer avec des parties aléatoires.

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

- Commencer avec des parties aléatoires.
- Petit à petit, choisir plus souvent l'action qui a statistiquement donné de meilleurs résultats.

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

- Commencer avec des parties aléatoires.
- Petit à petit, choisir plus souvent l'action qui a statistiquement donné de meilleurs résultats.
  (Critères inspirés des bandits manchots. Les détails jouent beaucoup.)

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

- Commencer avec des parties aléatoires.
- Petit à petit, choisir plus souvent l'action qui a statistiquement donné de meilleurs résultats.
  (Critères inspirés des bandits manchots. Les détails jouent beaucoup.)
- Asymptotiquement, on se concentre sur une sous-partie efficace de l'arbre.

Monte Carlo Tree Search

Monte Carlo Tree Search [Coulom 2006] est une exploration aléatoire dirigée de l'arbre, pour approcher la recherche minimax.

- Commencer avec des parties aléatoires.
- Petit à petit, choisir plus souvent l'action qui a statistiquement donné de meilleurs résultats.
  (Critères inspirés des bandits manchots. Les détails jouent beaucoup.)
- Asymptotiquement, on se concentre sur une sous-partie efficace de l'arbre.

$\Longrightarrow$ Premières réussites au Go. [MOGO, Gelly et al 2007].

Monte Carlo Tree Search : commentaires

Bien adapté sur des problèmes discrets comme les jeux, avec des arbres relativement peu profonds.

Monte Carlo Tree Search : commentaires

Bien adapté sur des problèmes discrets comme les jeux, avec des arbres relativement peu profonds.

Moins adapté sur des problèmes physiques continus, avec beaucoup d'actions par seconde et où on ne passe jamais deux fois exactement dans le même état.

Monte Carlo Tree Search : commentaires

Bien adapté sur des problèmes discrets comme les jeux, avec des arbres relativement peu profonds.

Moins adapté sur des problèmes physiques continus, avec beaucoup d'actions par seconde et où on ne passe jamais deux fois exactement dans le même état.

Possibilités d'heuristiques pour extrapoler aux états non visités.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

$\Longrightarrow$ Quelques étapes de minimax améliorent une fonction V médiocre.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

$\Longrightarrow$ Quelques étapes de minimax améliorent une fonction V médiocre.
On peut utiliser MCTS plutôt que le vrai minimax.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

$\Longrightarrow$ Quelques étapes de minimax améliorent une fonction V médiocre.
On peut utiliser MCTS plutôt que le vrai minimax.

$\Longrightarrow$ Utiliser les valeurs améliorées pour apprendre une nouvelle fonction V (par régression/apprentissage supervisé) et des choix d'action.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

$\Longrightarrow$ Quelques étapes de minimax améliorent une fonction V médiocre.
On peut utiliser MCTS plutôt que le vrai minimax.

$\Longrightarrow$ Utiliser les valeurs améliorées pour apprendre une nouvelle fonction V (par régression/apprentissage supervisé) et des choix d'action.

$\Longrightarrow$ Itérer.

De Shannon à DeepMind

Approche classique pour les échecs : minimax à horizon faible + fonction V codée à la main.

$\Longrightarrow$ Quelques étapes de minimax améliorent une fonction V médiocre.
On peut utiliser MCTS plutôt que le vrai minimax.

$\Longrightarrow$ Utiliser les valeurs améliorées pour apprendre une nouvelle fonction V (par régression/apprentissage supervisé) et des choix d'action.

$\Longrightarrow$ Itérer. C'est le principe d'AlphaGoZero.

Solutions approchées de l'équation de Bellman (1)

Retour sur l'équation de Bellman :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \Big\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \Big\}$$

(V-learning)

Solutions approchées de l'équation de Bellman (1)

Retour sur l'équation de Bellman :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \Big\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \Big\}$$

(V-learning) ou bien

$$V(a_1, \ldots, o_{t-1}, a_t) = \mathbb{E}_{o_t} \Big\{ r_t + \max_{a_{t+1}} V(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \Big\}$$

(Q-learning).

Solutions approchées de l'équation de Bellman (1)

Retour sur l'équation de Bellman :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \Big\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \Big\}$$

(V-learning) ou bien

$$V(a_1, \ldots, o_{t-1}, a_t) = \mathbb{E}_{o_t} \Big\{ r_t + \max_{a_{t+1}} V(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \Big\}$$

(Q-learning).

Souvent, le max sur l'action est remplacé par une action prise selon une politique approximativement optimale.

Solutions approchées de l'équation de Bellman (1)

Retour sur l'équation de Bellman :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \Big\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \Big\}$$

(V-learning) ou bien

$$V(a_1, \ldots, o_{t-1}, a_t) = \mathbb{E}_{o_t} \Big\{ r_t + \max_{a_{t+1}} V(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \Big\}$$

(Q-learning).

Souvent, le max sur l'action est remplacé par une action prise selon une politique approximativement optimale.
$\Longrightarrow$ Fonction valeur de cette politique

Solutions approchées de l'équation de Bellman (1)

Retour sur l'équation de Bellman :

$$V(a_1, \ldots, o_{t-1}) = \max_{a_t} \mathbb{E}_{o_t} \Big\{ r_t + V(a_1, \ldots, o_{t-1}, a_t, o_t) \Big\}$$

(V-learning) ou bien

$$V(a_1, \ldots, o_{t-1}, a_t) = \mathbb{E}_{o_t} \Big\{ r_t + \max_{a_{t+1}} V(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \Big\}$$

(Q-learning).

Souvent, le max sur l'action est remplacé par une action prise selon une politique approximativement optimale.
$\Longrightarrow$ Fonction valeur de cette politique
$\Longrightarrow$ politique améliorée en modifiant la politique pour favoriser les actions ayant une meilleure valeur (policy gradient, policy improvement).

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.
  $\Longrightarrow$ Théorie bien comprise (Monte Carlo)

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.
  $\Longrightarrow$ Théorie bien comprise (Monte Carlo)
  $\Longrightarrow$ Convergence très lente

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.
  $\Longrightarrow$ Théorie bien comprise (Monte Carlo)
  $\Longrightarrow$ Convergence très lente
  $\Longrightarrow$ uniquement pour des états discrets

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.
  $\Longrightarrow$ Théorie bien comprise (Monte Carlo)
  $\Longrightarrow$ Convergence très lente
  $\Longrightarrow$ uniquement pour des états discrets
  $\Longrightarrow$ pas de généralisation aux états non visités

Solutions approchées de l'équation de Bellman (2)

Solutions approchées : recherche d'une fonction paramétrique

$$V_\theta(a_1, \ldots, o_{t-1}, a_t)$$

parmi une famille de fonctions paramétrées par θ

$\Longrightarrow$ Ajuster θ pour vérifier approximativement l'équation de Bellman.

Exemples :

- Cas "tabulaire" : stocker une valeur distincte pour chaque état.
  $\Longrightarrow$ Théorie bien comprise (Monte Carlo)
  $\Longrightarrow$ Convergence très lente
  $\Longrightarrow$ uniquement pour des états discrets
  $\Longrightarrow$ pas de généralisation aux états non visités
  $\Longrightarrow$ nécessité de visiter tous les états un grand nombre de fois

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.
$\Longrightarrow$ Théorie mal comprise

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.
$\Longrightarrow$ Théorie mal comprise
$\Longrightarrow$ Approximateurs universels capables en principe de représenter la vraie fonction valeur

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.
$\Longrightarrow$ Théorie mal comprise
$\Longrightarrow$ Approximateurs universels capables en principe de représenter la vraie fonction valeur

$\Longrightarrow$ Modules existants :

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.
  $\Longrightarrow$ Théorie mal comprise
  $\Longrightarrow$ Approximateurs universels capables en principe de représenter la vraie fonction valeur

  $\Longrightarrow$ Modules existants :
  - Gestion des images par un réseau convolutionnel, du son, du texte par des réseaux adaptés...

Solutions approchées de l'équation de Bellman (3)

- Modèles linéaires :

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) = \sum_i \theta_i \, \varphi_i(a_1, \ldots, o_{t-1}, a_t)$$

où les φ_i sont un ensemble de fonctions de l'état du système.

Les φ_i peuvent être fixés à l'avance (choisies à la main, ondelettes, réseaux de neurones préentraînés...) ou appris.

- Réseaux de neurones paramétrés par θ.
  $\Longrightarrow$ Théorie mal comprise
  $\Longrightarrow$ Approximateurs universels capables en principe de représenter la vraie fonction valeur

  $\Longrightarrow$ Modules existants :
  - Gestion des images par un réseau convolutionnel, du son, du texte par des réseaux adaptés...
  - Gestion de l'historique $s_t = (a_1, o_1, \ldots, a_t, o_t)$ par un réseau récurrent (si nécessaire)

Solutions approchées de l'équation de Bellman (4)

$\Longrightarrow$ On cherche θ tel que

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Solutions approchées de l'équation de Bellman (4)

$\Longrightarrow$ On cherche θ tel que

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Données d'entraînement : un certain nombre de trajectoires $a_1, o_1, \ldots, a_t, o_t, \ldots$ issues de l'environnement.

Solutions approchées de l'équation de Bellman (4)

$\Longrightarrow$ On cherche θ tel que

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Données d'entraînement : un certain nombre de trajectoires $a_1, o_1, \ldots, a_t, o_t, \ldots$ issues de l'environnement.

$\Longrightarrow$ Descente de gradient sur θ pour réduire l'écart entre le terme de gauche et le terme de droite dans l'équation ?

Solutions approchées de l'équation de Bellman (4)

$\Longrightarrow$ On cherche θ tel que

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Données d'entraînement : un certain nombre de trajectoires $a_1, o_1, \ldots, a_t, o_t, \ldots$ issues de l'environnement.

$\Longrightarrow$ Descente de gradient sur θ pour réduire l'écart entre le terme de gauche et le terme de droite dans l'équation ?
Impossible en général.

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)
   $\Longrightarrow$ Vraie descente de gradient stochastique.

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)
   $\Longrightarrow$ Vraie descente de gradient stochastique.

2. On a accès à un grand nombre de trajectoires réelles (données peu coûteuses).

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)
   $\Longrightarrow$ Vraie descente de gradient stochastique.

2. On a accès à un grand nombre de trajectoires réelles (données peu coûteuses).
   $\Longrightarrow$ Temporal difference, Q-learning

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)
   $\Longrightarrow$ Vraie descente de gradient stochastique.

2. On a accès à un grand nombre de trajectoires réelles (données peu coûteuses).
   $\Longrightarrow$ Temporal difference, Q-learning

3. On a accès à un petit nombre de trajectoires réelles (données coûteuses).

Données d'apprentissage

Trois cas bien différents :

1. On a directement accès à un simulateur exact de l'environnement (jeux en self-play, jeux vidéo...)
   $\Longrightarrow$ Vraie descente de gradient stochastique.

2. On a accès à un grand nombre de trajectoires réelles (données peu coûteuses).
   $\Longrightarrow$ Temporal difference, Q-learning

3. On a accès à un petit nombre de trajectoires réelles (données coûteuses).
   $\Longrightarrow$ Modèle(s) du monde pour fournir des trajectoires imaginaires supplémentaires.

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Gradient de l'erreur : impossible (sauf cas d'un simulateur exact)

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Gradient de l'erreur : impossible (sauf cas d'un simulateur exact)

Temporal difference/Q-learning : ajuster le terme de gauche uniquement pour le rapprocher du terme de droite.

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Gradient de l'erreur : impossible (sauf cas d'un simulateur exact)

Temporal difference/Q-learning : ajuster le terme de gauche uniquement pour le rapprocher du terme de droite.

$\Longrightarrow$ Perte des garanties théoriques de la descente de gradient.

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Gradient de l'erreur : impossible (sauf cas d'un simulateur exact)

Temporal difference/Q-learning : ajuster le terme de gauche uniquement pour le rapprocher du terme de droite.

$\Longrightarrow$ Perte des garanties théoriques de la descente de gradient.
mais

Temporal difference

$$V_\theta(a_1, \ldots, o_{t-1}, a_t) \approx \mathbb{E}_{o_t}\left\{ r_t + \max_{a_{t+1}} V_\theta(a_1, \ldots, o_{t-1}, a_t, o_t, a_{t+1}) \right\}$$

Gradient de l'erreur : impossible (sauf cas d'un simulateur exact)

Temporal difference/Q-learning : ajuster le terme de gauche uniquement pour le rapprocher du terme de droite.

$\Longrightarrow$ Perte des garanties théoriques de la descente de gradient.
mais
$\Longrightarrow$ plus proche du minimax : les valeurs sont propagées de l'avenir vers le passé.

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

- Si la famille V_θ est linéaire. [Tsitsiklis–Van Roy 1997]

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

- Si la famille V_θ est linéaire. [Tsitsiklis–Van Roy 1997]
  $\Longrightarrow$ Convergence, mais vers un point qui n'est pas forcément le minimiseur de l'erreur

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

- Si la famille V_θ est linéaire. [Tsitsiklis–Van Roy 1997]
  $\Longrightarrow$ Convergence, mais vers un point qui n'est pas forcément le minimiseur de l'erreur

- Pour une famille V_θ quelconque, si la dynamique de l'environnement est réversible [Ollivier 2018]

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

- Si la famille V_θ est linéaire. [Tsitsiklis–Van Roy 1997]
  $\Longrightarrow$ Convergence, mais vers un point qui n'est pas forcément le minimiseur de l'erreur

- Pour une famille V_θ quelconque, si la dynamique de l'environnement est réversible [Ollivier 2018]
  $\Longrightarrow$ L'algorithme est bien une descente de gradient, mais d'une erreur différente, venant des chaînes de Markov (forme de Dirichlet).

Temporal difference : garanties théoriques

Cet algorithme peut diverger, même dans des cas simples.

Convergence connue dans deux cas :

- Si la famille V_θ est linéaire. [Tsitsiklis–Van Roy 1997]
  $\Longrightarrow$ Convergence, mais vers un point qui n'est pas forcément le minimiseur de l'erreur

- Pour une famille V_θ quelconque, si la dynamique de l'environnement est réversible [Ollivier 2018]
  $\Longrightarrow$ L'algorithme est bien une descente de gradient, mais d'une erreur différente, venant des chaînes de Markov (forme de Dirichlet).
  ... ou en environnement "assez réversible"

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

- Propagation de l'information du futur vers le passé.

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

- Propagation de l'information du futur vers le passé.
- Non-unicité de la solution (sur des espaces infinis) et

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

- Propagation de l'information du futur vers le passé.
- Non-unicité de la solution (sur des espaces infinis) et existence de nombreuses quasi-solutions très différentes de la vraie solution.

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

- Propagation de l'information du futur vers le passé.
- Non-unicité de la solution (sur des espaces infinis) et existence de nombreuses quasi-solutions très différentes de la vraie solution.
- Nécessité de modèles du monde, priors bayésiens etc, pour gérer l'incertitude, l'exploration et l'identification des causes

La fonction valeur : problèmes théoriques et pratiques

Même sans divergence, la convergence est souvent lente et fragile.

Nombreux problèmes pratiques + raisons plus fondamentales :

- Propagation de l'information du futur vers le passé.
- Non-unicité de la solution (sur des espaces infinis) et existence de nombreuses quasi-solutions très différentes de la vraie solution.
- Nécessité de modèles du monde, priors bayésiens etc, pour gérer l'incertitude, l'exploration et l'identification des causes, et pour réduire le nombre d'essais-erreurs nécessaires.

Bibliographie

Approches classiques de l'apprentissage par renforcement :

Richard Sutton, Andrew Barto, *Reinforcement learning: an introduction*, 2nd edition, 2018.

Priors universels sur les environnements, complexité de Kolmogorov pour l'induction et la prévision :

Marcus Hutter, *On universal prediction and Bayesian confirmation*, 2007.

Ray Solomonoff, *A formal theory of inductive inference*, parts I and II, 1964.

Marcus Hutter, *Universal artificial intelligence*, 2005.

AlphaGoZero, Monte Carlo Tree Search :

Sylvain Gelly, David Silver, *Monte-Carlo tree search and rapid action value estimation in computer Go*, 2011.

David Silver et al., *Mastering the game of Go without human knowledge*, 2017.

Convergence de l'algorithme Temporal Difference :

John Tsitsiklis, Benjamin Van Roy, *An Analysis of Temporal-Difference Learning with Function Approximation*, 1997.

Yann Ollivier, *Approximate Temporal Difference Learning is a Gradient Descent for Reversible Policies*, 2018.