Classification structurée pour l'apprentissage par renforcement inverse

Edouard Klein [1,2], **Bilal Piot** [2,3], **Matthieu Geist** [2], **Olivier Pietquin** [2,3]

1. *LORIA – équipe ABC*
   Nancy, France
2. *Supélec – Groupe de recherche IMS-MaLIS*
   Metz, France
   prenom.nom@supelec.fr
3. *UMI 2958 (GeorgiaTech-CNRS)*
   Metz, France

RÉSUMÉ. *Cette contribution traite le problème de l'apprentissage par renforcement inverse (ARI), défini comme la recherche d'une fonction de récompense pour laquelle le comportement d'un expert (connu par le biais de démonstrations) est optimal. Nous introduisons SCIRL, un nouvel algorithme qui utilise la grandeur dénommée attribut moyen de l'expert comme la paramétrisation d'une fonction de score pour un classifieur multiclasse. Cette approche donne une fonction de récompense pour laquelle la politique de l'expert est (nous le démontrons) quasi optimale. Contrairement à la plupart des algorithmes d'ARI existants, SCIRL n'a pas besoin de résoudre le problème direct de l'apprentissage par renforcement. De plus, en utilisant une heuristique, il fonctionne avec uniquement des trajectoires échantillonnées par l'expert. Nous illustrons cela sur un simulateur de conduite.*

ABSTRACT. *This paper adresses the inverse reinforcement learning (IRL) problem, that is inferring a reward for which a demonstrated expert behavior is optimal. We introduce a new algorithm, SCIRL, whose principle is to use the so-called feature expectation of the expert as the parameterization of the score function of a multiclasse classifier. This approach produces a reward function for which the expert policy is provably near-optimal. Contrary to most of existing IRL algorithms, SCIRL does not require solving the direct RL problem. Moreover, with an appropriate heuristic, it can succeed with only trajectories sampled according to the expert behavior. This is illustrated on a car driving simulator.*

MOTS-CLÉS : *apprentissage par renforcement, apprentissage par renforcement inverse.*

KEYWORDS: *reinforcement learning, inverse reinforcement learning.*

DOI:10.3166/RIA.27.155-169

1. Introduction

L'apprentissage par renforcement inverse (ARI) est le problème lié à la recherche, à partir de démonstrations d'un expert, d'une fonction de récompense telle que le comportement de l'expert soit optimal ; historiquement proposé dans (Russell, 1998), ce mécanisme trouve des applications dans divers champs, de la biologie à la neuropsychologie en passant par l'économie et plus récemment la robotique (Abbeel *et al.*, 2010). Beaucoup d'algorithmes d'ARI (que nous décrirons succinctement section 5) cherchent une fonction de récompense dont la politique optimale associée génère une distribution sur les trajectoires (ou une grandeur liée à cette distribution) proche de celle générée par l'expert. Souvent, cette distribution est caractérisée par ce que l'on appelle l'attribut moyen (voir section 2.1) : étant donnée une fonction de récompense linéairement paramétrée par un vecteur d'attributs, il s'agit de l'espérance de la somme pondérée des vecteurs d'attributs sachant que l'on commence dans un certain état en choisissant une certaine action avant de suivre la politique concernée.

Dans cette publication, nous choisissons une autre option. Il est possible d'imiter le comportement de l'expert via un algorithme d'apprentissage supervisé généralisant l'association des actions aux états. Nous considérons ici les classifieurs multiclasses qui à partir d'une base d'entraînement calculent les paramètres d'une fonction de score paramétrée linéairement ; la règle de décision pour un état est l'argument (l'action) qui maximise la fonction de score pour cet état (voir section 2.2). L'idée de base de l'algorithme SCIRL que nous proposons est simplement d'estimer l'attribut moyen de l'expert comme la paramétrisation de la fonction de score (voir section 3.1). Le vecteur de paramètres ainsi calculé définit une fonction de récompense dont on montre qu'elle admet pour politique quasi optimale la politique de l'expert (voir section 3.2).

Un grand avantage de SCIRL est qu'il ne requiert pas, contrairement à la plupart des algorithmes d'ARI, la résolution répétée du problème direct, celui de l'apprentissage par renforcement. Il nécessite d'estimer l'attribut moyen de l'expert, mais ceci est équivalent à un problème d'évaluation de la politique (pour une politique observée, ce qui est moins problématique que l'optimisation répétée d'une politique), voir section 4. De plus, à l'aide d'une heuristique, SCIRL peut s'éxécuter en utilisant uniquement des transitions issues de la politique de l'expert (il n'y a pas besoin d'échantillonner la dynamique complète). Nous illustrons cela sur un simulateur de conduite section 6.

2. Contexte et notations

2.1. Apprentissage par renforcement (inverse)

Un processus décisionnel de Markov (PDM) (Puterman, 1994) est donné par le tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$ où $\mathcal{S}$ est l'espace d'état fini, $\mathcal{A}$ est l'espace d'action fini, $\mathcal{P} = \{P_a = (p(s'|s,a))_{1 \leq s,s' \leq |\mathcal{S}|}, a \in \mathcal{A}\}$ est l'ensemble des probabilités de transition markoviennes, $\mathcal{R} \in \mathbb{R}^{\mathcal{S}}$ est la fonction de récompense sur les états et γ est le facteur d'oubli. Une politique déterministe $\pi \in \mathcal{S}^{\mathcal{A}}$ définit le comportement d'un agent.

La qualité de ce comportement est quantifiée par le biais de la fonction de valeur $v^\pi_\mathcal{R} \in \mathbb{R}^\mathcal{S}$, qui à chaque état associe la récompense pondérée cumulée moyenne recueillie par l'agent lorsqu'il part de cet état et suit la politique π par la suite : $v^\pi_\mathcal{R}(s) = \mathbb{E}[\sum_{t\geq 0} \gamma^t \mathcal{R}(S_t)|S_0 = s, \pi]$. Une politique optimale $\pi^*_\mathcal{R}$ (vis-à-vis de la fonction de récompense $\mathcal{R}$) est une politique dont la fonction de valeur $v^*_\mathcal{R}$ vérifie $v^*_\mathcal{R} \geq v^\pi_\mathcal{R}$ (composante par composante) pour toute politique π.

Soit P_π la matrice stochastique définie par $P_\pi = (p(s'|s, \pi(s)))_{1\leq s,s'\leq|\mathcal{S}|}$. Un léger abus de notation nous permet de nommer a la politique associant l'action a à chaque état s. Les opérateurs d'évaluation (respectivement d'optimalité) de Bellman $T^\pi_\mathcal{R}$ (resp. $T^*_\mathcal{R}$) : $\mathbb{R}^\mathcal{S} \to \mathbb{R}^\mathcal{S}$ sont définis par $T^\pi_\mathcal{R} v = \mathcal{R} + \gamma P_\pi v$ et $T^*_\mathcal{R} v = \max_\pi T^\pi_\mathcal{R} v$. Ces opérateurs sont des contractions qui admettent $v^\pi_\mathcal{R}$ et $v^*_\mathcal{R}$ comme points fixes respectifs : $v^\pi_\mathcal{R} = T^\pi_\mathcal{R} v^\pi_\mathcal{R}$ et $v^*_\mathcal{R} = T^*_\mathcal{R} v^*_\mathcal{R}$. La fonction de qualité $Q^\pi \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ ajoute un degré de liberté dans le choix de la première action : $Q^\pi_\mathcal{R}(s, a) = [T^a_\mathcal{R} v^\pi_\mathcal{R}](s)$. La distribution stationnaire de la politique π est notée ρ_π, elle satisfait $\rho_\pi^\top P_\pi = \rho_\pi^\top$.

L'objectif de l'apprentissage par renforcement (AR) est d'estimer la politique de contrôle optimale lorsque le modèle (les probabilités de transition et la fonction de récompense) est inconnu (mais observé par interaction avec le système à contrôler) et quand l'espace d'état est trop grand pour qu'une représentation exacte des objets concernés (fonctions de valeur, politiques) soit possible (Bertsekas, Tsitsiklis, 1996 ; Sutton, Barto, 1998 ; Szepesvári, 2010). C'est le problème direct. L'apprentissage par renforcement inverse (approché) (Ng, Russell, 2000) consiste à estimer la fonction de récompense pour laquelle une politique observée est optimale. Cette politique est la politique de l'expert, notée π_E. Elle est supposée optimale vis-à-vis d'une certaine fonction de récompense $\mathcal{R}_E$ inconnue. Le but de l'ARI est de calculer une fonction de récompense $\hat{\mathcal{R}}$ telle que la politique de l'expert soit (quasiment) optimale, c'est-à-dire telle que $v^*_{\hat{\mathcal{R}}} \approx v^{\pi_E}_{\hat{\mathcal{R}}}$. Nous appelons cela le problème inverse.

De la même manière que pour le problème direct, l'espace d'état peut être trop grand pour que la fonction de récompense admette une représentation exacte exploitable. La recherche est donc limitée à celle d'une bonne fonction de récompense paramétrée linéairement. Soit $\phi(s) = (\phi_1(s) \dots \phi_p(s))^\top$ un vecteur d'attributs composé de p fonctions de base $\phi_i \in \mathbb{R}^\mathcal{S}$, les fonctions de récompense paramétrées sont définies par $\mathcal{R}_\theta(s) = \theta^\top \phi(s) = \sum_{i=1}^p \theta_i \phi_i(s)$. La recherche d'une bonne récompense est donc ramenée à la recherche d'un bon vecteur de paramètres $\theta \in \mathbb{R}^p$. Nous utiliserons indifféremment $\mathcal{R}_\theta$ et θ comme indices (*e.g.*, v^π_θ pour $v^\pi_{\mathcal{R}_\theta}$). Cette paramétrisation de la récompense implique une paramétrisation similaire de la fonction de qualité :

$$Q^\pi_\theta(s, a) = \theta^\top \mu^\pi(s, a) \text{ avec } \mu^\pi(s, a) = \mathbb{E}[\sum_{t\geq 0} \gamma^t \phi(S_t)|S_0 = s, A_0 = a, \pi]. \quad (1)$$

Conséquemment, la fonction de qualité partage son vecteur de paramètres avec la fonction de récompense, mais en relation avec un vecteur d'attributs μ^π appelé l'attribut moyen. Cette notion est de prime importance pour notre contribution. Chaque composante μ^π_i de ce vecteur d'attributs est en réalité la fonction de qualité de la politique π pour la récompense ϕ_i: $\mu^\pi_i(s, a) = Q^\pi_{\phi_i}(s, a)$. De fait, tout algorithme d'esti-

mation de la fonction de qualité peut être utilisé pour estimer l'attribut moyen, comme une estimation de Monte-Carlo ou un algorithme aux différences temporelles (Klein *et al.*, 2011).

2.2. Classifieurs à fonction de score paramétrée linéairement

Soit $\mathcal{X}$ un ensemble fini ou compact (d'entrées à classifier) et soit $\mathcal{Y}$ un ensemble fini (de labels). Supposons que les entrées $x \in \mathcal{X}$ sont tirées selon une distribution inconnue $\mathbb{P}(x)$ et qu'il existe un oracle qui à chacune de ces entrées associe un label $y \in Y$ tiré selon une distribution de probabilité conditionnelle inconnue $\mathbb{P}(y|x)$. De manière générale, la classification multiclasse cherche, étant donné une base d'entraînement $\{(x_i, y_i)_{1 \leq i \leq N}\}$ tirée selon $\mathbb{P}(x, y)$, une règle de décision $g \in \mathcal{Y}^{\mathcal{X}}$ minimisant l'erreur de classification $\mathbb{E}[\chi_{\{g(x) \neq y\}}] = \mathbb{P}(g(x) \neq y)$, où χ est la fonction indicatrice.

Nous nous préoccupons ici d'un ensemble plus réduit de classifieurs. Nous supposons que la règle de décision associe à l'entrée l'argument qui maximise une certaine fonction de score, celle-ci étant paramétrée linéairement, les paramètres étant appris par le classifieur. Formellement, soit $\psi(x, y) = (\psi_1(x, y) \dots \psi_d(x, y))^\top \in \mathbb{R}^d$ un vecteur d'attributs dont les composantes sont d fonctions de base $\psi_i \in \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$. La fonction de score linéairement paramétrée $s_w \in \mathbb{R}^{\mathcal{X} \times \mathcal{Y}}$ de vecteur de paramètres $w \in \mathbb{R}^d$ est définie par $s_w(x, y) = w^\top \psi(x, y)$. La règle de décision associée $g_w \in \mathcal{Y}^{\mathcal{X}}$ est définie par $g_w(x) \in \operatorname{argmax}_{y \in \mathcal{Y}} s_w(x, y)$. En se basant sur une base d'entraînement $\{(x_i, y_i)_{1 \leq i \leq N}\}$, un classifieur multiclasse (noté CMC) à fonction de score paramétrée linéairement choisit un vecteur de paramètres θ_c. La qualité de ce choix est quantifiée par l'erreur de classification $\epsilon_c = \mathbb{P}(g_{\theta_c}(x) \neq y)$.

Notre étude est valable pour tout algorithme de classification tant qu'il opère en maximisant l'argument d'une fonction de score paramétrée linéairement. Par exemple, un séparateur à vaste marge multiclasse pourrait être choisie (Guermeur, 2007) (en considérant le noyau induit par le vecteur d'attributs) ou encore une approche à vaste marge structurée (Taskar *et al.*, 2005).

3. Classification structurée pour l'apprentissage par renforcement inverse

3.1. Forme générale de l'algorithme

Nous nous trouvons dans le cadre de la classification décrit en section 2.2. L'entrée x peut être vue comme un état et le label y comme une action. Il s'ensuit que la règle de décision $g_w(x)$ est interprétable comme une politique gloutonne vis-à-vis de la fonction de score $w^\top \psi(x, y)$, qui peut elle-même être vue comme une fonction de qualité. Le parallèle avec l'équation (1) est aisé, si $\psi(x, y)$ est l'attribut moyen d'une politique π qui a fourni les labels de la base d'entraînement, et si l'erreur de classification est faible, alors w est le vecteur de paramètres de la fonction de récompense vis-à-vis de laquelle on espère que la politique π est quasi optimale. Ces remarques nous per-

mettent maintenant d'introduire notre algorithme d'ARI par classification structurée (SCIRL pour *Structured Classification-based Inverse Reinforcement Learning*).

Soit π_E la politique de l'expert à partir de laquelle nous souhaitons inférer une fonction de récompense. Soit $\mathcal{D} = \{(s_i, a_i = \pi_E(s_i))_{1 \leq i \leq N}\}$ une base d'entraînement où les états sont échantillonnés selon la distribution stationnaire de l'expert [1] $\rho_E = \rho_{\pi_E}$. Supposons également avoir à disposition une estimée $\hat{\mu}^{\pi_E}$ de l'attribut moyen de l'expert μ^{π_E} défini à l'équation (1). Une description de la manière d'estimer cette quantité en pratique est reportée à la section 4.1; rappelons tout de même qu'estimer μ^{π_E} est simplement un problème d'*évaluation de la politique* (estimation de la fonction de qualité d'une politique), comme signalé section 2.1. Supposons enfin qu'un algorithme de CMC a été choisi. L'algorithme formant notre contribution consiste simplement à choisir $\theta^\top \hat{\mu}^{\pi_E}(s, a)$ comme fonction de score paramétrée linéairement, puis à entraîner le classifieur sur $\mathcal{D}$, ce qui produit un vecteur de paramètres θ_c, puis enfin à renvoyer la fonction de récompense $\mathcal{R}_{\theta_c}(s) = \theta_c^\top \phi(s)$.

Algorithme 1: *SCIRL*

Etant donnée une base d'entraînement $\mathcal{D} = \{(s_i, a_i = \pi_E(s_i))_{1 \leq i \leq N}\}$, une estimée $\hat{\mu}^{\pi_E}$ de l'attribut moyen de l'expert μ^{π_E} et un algorithme de CMC;

Calculer le vecteur de paramètres θ_c en utilisant l'algorithme de CMC auquel sont fournis la base d'entraînement $\mathcal{D}$ et la paramétrisation de la fonction de score : $s_\theta(s, a) = \theta^\top \hat{\mu}^{\pi_E}(s, a)$;

Renvoyer la fonction de récompense $\mathcal{R}_{\theta_c}(s) = \theta_c^\top \phi(s)$;

L'approche proposée est résumé par l'algorithme 1. Le nom de l'algorithme (SCRIL, que l'on peut traduire par Classification structurée pour l'ARI), vient de l'utilisation de l'attribut moyen de l'expert dans le classifieur, ce qui revient d'une certaine manière à prendre en compte la structure du MDP dans le problème de classification et permet le calcul du vecteur de paramètres de la récompense. Contrairement à la plupart des algorithmes d'ARI existants, SCIRL n'a pas besoin de résoudre le problème direct. Cet algorithme requiert une estimation de l'attribut moyen de l'expert mais il ne s'agit là que d'un problème d'évaluation de la politique, notoirement moins difficile que la recherche de politiques optimales qu'implique la résolution du problème direct. Cela est discuté plus en détail section 5.

3.2. Analyse

Dans cette section, nous montrons que la politique de l'expert π_E est quasi optimale vis-à-vis de la fonction de récompense $\mathcal{R}_{\theta_c}$, plus précisément qu'il est possible

1. Par exemple, si la chaîne de Markov induite par la politique de l'expert est à mélange rapide (*fast-mixing*), l'échantillonnage d'une trajectoire donnera rapidement des échantillons tirés selon cette distribution.

de contrôler le terme $\mathbb{E}_{s\sim\rho_E}[v^*_{\theta_c}(s) - v^{\pi_E}_{\theta_c}(s)]$. Avant de présenter le résultat principal, il nous faut introduire quelques notations et définir quelques objets.

Nous allons utiliser le coefficient de concentration C_f (Munos, 2007):

$$C_f = (1-\gamma)\sum_{t\geq 0}\gamma^t c(t) \text{ avec } c(t) = \max_{\pi_1,\dots,\pi_t, s\in\mathcal{S}} \frac{(\rho_E^\top P_{\pi_1}\dots P_{\pi_t})(s)}{\rho_E(s)}.$$

La règle de décision du classifieur est notée $\pi_c(s) \in \operatorname{argmax}_{a\in\mathcal{A}} \theta_c^\top \hat{\mu}^{\pi_E}(s,a)$. L'erreur de classification est donc $\epsilon_c = \mathbb{E}_{s\sim\rho_E}[\chi_{\{\pi_c(s)\neq\pi_E(s)\}}] \in [0,1]$. On écrit $\hat{Q}^{\pi_E}_{\theta_c} = \theta_c^\top \hat{\mu}^{\pi_E}$ la fonction de score calculée à partir de la base d'entraînement $\mathcal{D}$ (celle-ci peut être interprétée comme une fonction de qualité approchée). Soit $\epsilon_\mu = \hat{\mu}^{\pi_E} - \mu^{\pi_E}$: $\mathcal{S}\times\mathcal{A}\to\mathbb{R}^p$ l'erreur sur l'estimation de l'attribut moyen. En conséquence, on définit l'erreur sur la fonction de qualité par $\epsilon_Q = \hat{Q}^{\pi_E}_{\theta_c} - Q^{\pi_E}_{\theta_c} = \theta_c^\top(\hat{\mu}^{\pi_E} - \mu^{\pi_E}) = \theta_c^\top\epsilon_\mu$: $\mathcal{S}\times\mathcal{A}\to\mathbb{R}$. Finalement, on définit l'erreur moyenne d'écart maximal sur la fonction de qualité par $\bar{\epsilon}_Q = \mathbb{E}_{s\sim\rho_E}[\max_{a\in\mathcal{A}}\epsilon_Q(s,a) - \min_{a\in\mathcal{A}}\epsilon_Q(s,a)] \geq 0$.

THÉORÈME 1. — *Soit $\mathcal{R}_{\theta_c}$ la fonction de récompense renvoyée par l'Alg. 1. Soient C_f, ϵ_c et $\bar{\epsilon}_Q$ les quantités définies ci-dessus. On a :*

$$0 \leq \mathbb{E}_{s\sim\rho_E}[v^*_{\mathcal{R}_{\theta_c}} - v^{\pi_E}_{\mathcal{R}_{\theta_c}}] \leq \frac{C_f}{1-\gamma}\left(\bar{\epsilon}_Q + \epsilon_c\frac{2\gamma\|\mathcal{R}_{\theta_c}\|_\infty}{1-\gamma}\right).$$

PREUVE (Preuve du Théorème 1). — La démonstration repose uniquement sur la récompense $\mathcal{R}_{\theta_c}$, donc par souci de clarté certains indices relatifs à la récompense sont omis des notations (*e.g.*, v^π pour $v^\pi_{\theta_c} = v^\pi_{\mathcal{R}_{\theta_c}}$ ou $\mathcal{R}$ pour $\mathcal{R}_{\theta_c}$). Tout d'abord, l'erreur $\mathbb{E}_{s\sim\rho_E}[v^*(s) - v^{\pi_E}(s)]$ est reliée au résidu de Bellman $\mathbb{E}_{s\sim\rho_E}[[T^*v^{\pi_E}](s) - v^{\pi_E}(s)]$. Composante par composante :

$$\begin{aligned} v^* - v^{\pi_E} &= T^*v^* - T^{\pi^*}v^{\pi_E} + T^{\pi^*}v^{\pi_E} - T^*v^{\pi_E} + T^*v^{\pi_E} - v^{\pi_E} \\ &\overset{(a)}{\leq} \gamma P_{\pi^*}(v^* - v^{\pi_E}) + T^*v^{\pi_E} - v^{\pi_E} \overset{(b)}{\leq} (I - \gamma P_{\pi^*})^{-1}(T^*v^{\pi_E} - v^{\pi_E}). \end{aligned}$$

L'inégalité (a) est valable car $T^{\pi^*}v^{\pi_E} \leq T^*v^{\pi_E}$ et l'inégalité (b) l'est en vertu de (Munos, 2007, Lemme 4.2). De plus, v^* étant optimale, il est visible que $v^* - v^{\pi_E} \geq 0$ et avec T^* l'opérateur d'optimalité de Bellman, $T^*v^{\pi_E} \geq T^{\pi_E}v^{\pi_E} = v^{\pi_E}$. De plus, on remarque que $(I - \gamma P_{\pi^*})^{-1} = \sum_{t\geq 0}\gamma^t P^t_{\pi^*}$. Donc, d'après la définition du coefficient de concentration C_f :

$$0 \leq \mathbb{E}_{s\sim\rho_E}[v^*(s) - v^{\pi_E}(s)] \leq \frac{C_f}{1-\gamma}\mathbb{E}_{s\sim\rho_E}\left[[T^*v^{\pi_E}](s) - v^{\pi_E}(s)\right]. \qquad (2)$$

Ce résultat est similaire à celui de (Munos, 2007, Theoreme 4.2). Il reste à borner le résidu de Bellman $\mathbb{E}_{s\sim\rho_E}[[T^*v^{\pi_E}](s) - v^{\pi_E}(s)]$. Considérons la décomposition

$$T^*v^{\pi_E} - v^{\pi_E} = T^*v^{\pi_E} - T^{\pi_c}v^{\pi_E} + T^{\pi_c}v^{\pi_E} - v^{\pi_E}.$$

Nous allons borner $\mathbb{E}_{s\sim\rho_E}[[T^* v^{\pi_E}](s) - [T^{\pi_c} v^{\pi_E}](s)]$ et $\mathbb{E}_{s\sim\rho_E}[[T^{\pi_c} v^{\pi_E}](s) - v^{\pi_E}(s)]$.

La politique π_c (la règle de décision du classifieur) est gloutonne vis-à-vis de $\hat{Q}^{\pi_E} = \theta_c^\top \hat{\mu}^{\pi_E}$. Donc, pour chaque couple état-action $(s,a) \in \mathcal{S}\times\mathcal{A}$:

$$\hat{Q}^{\pi_E}(s,\pi_c(s)) \geq \hat{Q}^{\pi_E}(s,a) \Leftrightarrow Q^{\pi_E}(s,a) \leq Q^{\pi_E}(s,\pi_c(s)) + \epsilon_Q(s,\pi_c(s)) - \epsilon_Q(s,a).$$

Par définition, $Q^{\pi_E}(s,a) = [T^a v^{\pi_E}](s)$ et $Q^{\pi_E}(s,\pi_c(s)) = [T^{\pi_c} v^{\pi_E}](s)$. Donc, pour $s \in \mathcal{S}$:

$$\begin{aligned}\forall a \in A,\ [T^a v^{\pi_E}](s) &\leq [T^{\pi_c} v^{\pi_E}](s) + \epsilon_Q(s,\pi_c(s)) - \epsilon_Q(s,a)\\ \Rightarrow [T^* v^{\pi_E}](s) &\leq [T^{\pi_c} v^{\pi_E}](s) + \max_{a\in\mathcal{A}} \epsilon_Q(s,a) - \min_{a\in\mathcal{A}} \epsilon_Q(s,a).\end{aligned}$$

En passant à l'espérance selon ρ_E et tout en rappelant que $T^* v^{\pi_E} \geq v^{\pi_E}$, le premier terme est borné :

$$0 \leq \mathbb{E}_{s\sim\rho_E}\left[[T^* v^{\pi_E}](s) - [T^{\pi_c} v^{\pi_E}](s)\right] \leq \bar{\epsilon}_Q. \tag{3}$$

Il reste enfin à borner le terme $\mathbb{E}_{s\sim\rho_E}[[T^{\pi_c} v^{\pi_E}](s) - v^{\pi_E}(s)]$.

Soit $M \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ la matrice diagonale définie par $M = \mathrm{diag}(\chi_{\{\pi_c(s)\neq\pi_E(s)\}})$. En utilisant cela, l'opérateur de Bellman T^{π_c} peut s'écrire, pour tout $v \in \mathbb{R}^{\mathcal{S}}$:

$$T^{\pi_c} v = \mathcal{R} + \gamma M P_{\pi_c} v + \gamma(I-M)P_{\pi_E} v = \mathcal{R} + \gamma P_{\pi_E} v + \gamma M(P_{\pi_c} - P_{\pi_E})v.$$

En appliquant cet opérateur à v^{π_E} et en utilisant le fait que $\mathcal{R} + \gamma P_{\pi_E} v^{\pi_E} = T^{\pi_E} v^{\pi_E} = v^{\pi_E}$, nous obtenons :

$$\begin{aligned}T^{\pi_c} v^{\pi_E} - v^{\pi_E} &= \gamma M(P_{\pi_c} - P_{\pi_E})v^{\pi_E}\\ \Rightarrow |\rho_E^\top (T^{\pi_c} v^{\pi_E} - v^{\pi_E})| &= \gamma |\rho_E^\top M(P_{\pi_c} - P_{\pi_E})v^{\pi_E}|.\end{aligned}$$

Il est facile de voir que $\|(P_{\pi_c} - P_{\pi_E})v^{\pi_E}\|_\infty \leq \frac{2}{1-\gamma}\|\mathcal{R}\|_\infty$, ce qui permet de borner le dernier terme

$$|\mathbb{E}_{s\sim\rho_E}[[T^{\pi_c} v^{\pi_E}](s) - v^{\pi_E}(s)]| \leq \epsilon_c \frac{2\gamma}{1-\gamma}\|\mathcal{R}\|_\infty. \tag{4}$$

Injecter les bornes des équations (3) et (4) dans l'équation (2) achève la démonstration. ■

Ce résultat montre que si l'attribut moyen de l'expert est bien estimé (dans le sens d'une faible erreur d'estumation ϵ_μ pour les états échantillonnés selon la politique stationnaire de l'expert et pour toutes les actions) et si l'erreur de classification ϵ_c et également faible, alors l'algorithme générique proposé fournit une fonction de récompense $\mathcal{R}_{\theta_c}$ vis-à-vis de laquelle l'expert sera quasi optimal. Un corollaire direct du théorème 1 stipule qu'avec le vrai attribut moyen de l'expert μ^{π_E} et un classifieur parfait ($\epsilon_c = 0$), π_E est l'unique politique optimale pour $\mathcal{R}_{\theta_c}$.

Certains allègueraient que ces bornes sont trivialement valables pour la fonction de récompense nulle (fonction régulièrement citée comme exemple de la nature mal posée du problème de l'ARI) correspondant au cas $\theta_c = 0$. Cependant il faut se rappeler que le vecteur de paramètres θ_c est choisi par le classifieur. Avec $\theta_c = 0$, la règle de décision serait une politique aléatoire uniforme et nous aurions $\epsilon_c = \frac{|\mathcal{A}|-1}{|\mathcal{A}|}$, c'est-à-dire la pire erreur de classification possible. Ce cas est très improbable, l'objectif du classifieur étant de minimiser ϵ_C. De fait, nous affirmons que notre approche permet, d'une certaine manière, de lever l'ambiguïté du problème de l'ARI (au moins, l'algorithme ne renvoie pas de récompense triviale comme la récompense nulle). Cette borne est invariante par dilatation. Il est possible d'imposer $\|\theta_c\| = 1$ ou de normaliser la fonction de valeur (et de qualité) par $\|\mathcal{R}_{\theta_c}\|_\infty^{-1}$.

Il existe une dépendance cachée de l'erreur de classification ϵ_c à l'estimation de l'attribut moyen de l'expert $\hat{\mu}^{\pi_E}$. En effet, l'erreur de classification minimale dépend de l'espace d'hypothèses généré par les fonctions de base de la fonction de score de l'algorithme de CMC (ici, $\hat{\mu}^{\pi_E}$). Néanmoins, avec une bonne représentation de la fonction de récompense (c'est-à-dire un choix judicieux de fonctions de base ϕ_i) et une faible erreur d'estimation, cela ne devrait pas poser de problème en pratique.

Finalement, si notre borne se base sur les erreurs en généralisation ϵ_c et $\bar{\epsilon}_Q$, le classifieur n'utilisera $(\hat{\mu}^{\pi_E}(s_i, a))_{1\leq i\leq N, a\in\mathcal{A}}$ que lors de la phase d'entraînement, où les s_i sont les états présents dans $\mathcal{D}$. Il renvoie θ_c, vu comme une fonction de récompense, donc l'estimée de l'attribut moyen $\hat{\mu}^{\pi_E}$ n'est plus nécessaire après l'étape de CMC. Ainsi, en pratique, il est suffisant d'estimer $\hat{\mu}^{\pi_E}$ correctement uniquement pour les couples état-action $(s_i, a)_{1\leq i\leq N, a\in\mathcal{A}}$, ce qui permet d'envisager, par exemple, une simple estimation de Monte-Carlo.

4. Mise en pratique

4.1. Estimation de l'attribut moyen de l'expert

SCIRL a besoin d'une estimée $\hat{\mu}^{\pi_E}$ de l'attribut moyen de l'expert. C'est un problème similaire à l'évaluation d'une politique. Répétons l'observation-clef : chaque composante de μ^{π_E} est la fonction de qualité pour π_E vis-à-vis de la fonction de récompense ϕ_i: $\mu_i^{\pi_E}(s,a) = Q_{\phi_i}^{\pi_E}(s,a) = [T_{\phi_i}^a v_{\phi_i}^{\pi_E}](s)$. Nous présentons une revue rapide de méthodes de calcul exacte et approchées, ainsi qu'une heuristique.

Si le modèle (les probabilités de transition P) est connu, il est possible de calculer l'attribut moyen de manière exacte. Soit $\Phi \in \mathbb{R}^{|\mathcal{S}|\times p}$ la matrice d'attributs dont les lignes sont indexées par $s \in \mathcal{S}$ et contiennent les vecteurs d'attributs $\phi(s)^\top$.

Pour un certain $a \in A$, soit $\boldsymbol{\mu}_a^{\pi_E} \in \mathbb{R}^{|\mathcal{S}|\times p}$ la matrice des attributs moyens dont les lignes sont les attributs moyens de l'expert, c'est-à-dire $(\mu^{\pi_E}(s,a))^\top$ pour chaque $s \in \mathcal{S}$. Ces notations nous permettent d'écrire $\boldsymbol{\mu}_a^{\pi_E} = \Phi + \gamma P_a (I - \gamma P_{\pi_E})^{-1}\Phi$. Ajoutons que le coût computationnel de cette méthode est du même ordre de grandeur

que l'évaluation d'une seule politique, en effet la partie coûteuse (le calcul de $(I - \gamma P_{\pi_E})^{-1}$) est partagée par toutes les composantes.

Si le modèle est inconnu, tous les algorithmes d'apprentissage par différences temporelles peuvent être utilisés pour obtenir une estimation de l'attribut moyen de l'expert (Klein *et al.*, 2011), comme par exemple LSTD (*Least-Squares Temporal Differences*) (Bradtke, Barto, 1996). Soit $\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ un vecteur d'attributs composé de d fonctions de base $\psi_i \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. Chaque composante $\mu_i^{\pi_E}$ de l'attribut moyen de l'expert est paramétée par un vecteur $\xi_i \in \mathbb{R}^d$: $\mu_i^{\pi_E}(s,a) \approx \xi_i^\top \psi(s,a)$. Supposons l'existence d'une base d'entraînement $\{(s_i, a_i, s'_i, a'_i = \pi_E(s'_i))_{1\leq i\leq M}\}$ dont les actions a_i ne sont pas nécessairement échantillonnées selon la politique π_E (*e.g.*, on peut utiliser des trajectoires obtenues par un agent suivant une politique ϵ-gloutonne), le but étant d'obtenir une meilleure représentativité des données (des actions sous-optimales devraient être essayées). Soit $\tilde{\Psi} \in \mathbb{R}^{M\times d}$ (resp. $\tilde{\Psi}'$) la matrice d'attributs dont les lignes sont les vecteurs d'attributs $\psi(s_i,a_i)^\top$ (resp. $\psi(s'_i,a'_i)^\top$). Soit $\tilde{\Phi} \in \mathbb{R}^{M\times p}$ la matrice d'attributs dont les lignes sont les vecteurs d'attributs de la récompense $\phi(s_i)^\top$. Enfin, soit $\Xi = \begin{bmatrix}\xi_1 & \dots & \xi_p\end{bmatrix} \in \mathbb{R}^{d\times p}$ la matrice de tous les vecteurs de paramètres. Appliquer LSTD sur chacune des composantes de l'attribut moyen donne l'algorithme LSTD-μ (Klein *et al.*, 2011): $\Xi = (\tilde{\Psi}^\top(\tilde{\Psi} - \gamma\tilde{\Psi}'))^{-1}\tilde{\Psi}^\top\tilde{\Phi}$ et $\hat{\mu}^{\pi_E}(s,a) = \Xi^\top\psi(s,a)$. De la même manière que pour le cas exact, la partie coûteuse de l'algorithme (inverser la matrice) est partagée par toutes les composantes. Le coût reste donc raisonnable (du même ordre que LSTD).

Si l'on dispose d'un simulateur permettant notammment d'échantillonner selon la politique à imiter, l'attribut moyen de l'expert peut également être estimé via une méthode de Monte-Carlo pour chaque couple état-action (comme signalé en section 3.2, $\hat{\mu}^{\pi_E}$ ne doit être connu que pour $(s_i,a)_{1\leq i\leq N, a\in\mathcal{A}}$). Si K trajectoires sont échantillonnées pour chaque couple, cette méthode requiert $KN|\mathcal{A}|$ simulations.

Pour minimiser l'erreur $\bar{\epsilon}_Q$, il est nécessaire d'utiliser des transitions dont l'état de départ est tiré selon ρ_E et dont les actions sont uniformément distribuées. Cependant, il est possible que seules les transitions issues de l'expert soient disponibles : $\mathcal{T} = \{(s_i, a_i = \pi_E(s_i), s'_i)_{1\leq i\leq N}\}$. Bien que le couple état-action (s_i,a_i) puisse être exploité par le classifieur, les transitions (s_i,a_i,s'_i) seules ne sont pas seules suffisantes pour une estimation précise de l'attribut moyen. Il est toujours possible de rester précis sur l'estimation de $\mu^{\pi_E}(s,\pi_E(s))$, mais il y a peu d'espoir de l'être pour $\mu^{\pi_E}(s, a \neq \pi_E(s))$, ces actions (et la dynamique résultante) n'étant pas représentées dans les données. Il est heureusement possible de recourir à des heuristiques ; ce cas ne rentre pas dans l'analyse présentée en section 3.2, mais peut malgré tout fournir de bons résultats expérimentaux comme illustré en section 6.

Nous proposons une telle heuristique. Supposons que $\mathcal{T}$ contienne les seules données disponibles, que nous utilisons pour fournir une estimation $\hat{\mu}^{\pi_E}(s,\pi_E(s))$ (cela revient à estimer non plus une fonction de qualité comme décrit ci-dessus, mais simplement une fonction de valeur). Un point de vue optimiste suppose que choisir une action différente de celle de l'expert ne fait que retarder l'effet de l'action de l'expert. Plus formellement, nous associons à chaque état s un état virtuel s_v pour le-

quel $p(s_v|s, a \neq \pi_E(s)) = 1$ et $p(.|s_v, a) = p(.|s, \pi_E(s))$ pour toute action a et pour lequel l'attribut (de récompense) moyen est le vecteur nul, $\phi(s_v) = 0$. Dans ce cas, on a $\mu^{\pi_E}(s, a \neq \pi_E(s)) = \gamma\mu^{\pi_E}(s, \pi_E(s))$. Appliquer cette idée sur l'estimation effectivement disponible (rappelons que le classifieur n'a besoin d'évaluer $\hat{\mu}^{\pi_E}$ qu'en $(s_i, a)_{1\leq i\leq N, a\in\mathcal{A}}$) fournit l'heuristique que nous proposons : pour $1 \leq i \leq N$, $\hat{\mu}^{\pi_E}(s_i, a \neq a_i) = \gamma\hat{\mu}^{\pi_E}(s_i, a_i)$.

Il est également possible de pousser cette idée plus loin afin d'obtenir une estimation plus simple (mais offrant de moindres garanties) de l'attribut moyen de l'expert. Supposons que $\mathcal{T}$ consiste en une longue trajectoire, c'est-à-dire $s'_i = s_{i+1}$ (donc $\mathcal{T} = \{s_1, a_1, s_2, \ldots, s_{N-1}, a_{N-1}, s_N, a_N\}$). L'attribut moyen $\mu^{\pi_E}(s_i, a_i)$ est estimé en utilisant la seule trajectoire disponible et en utilisant l'heuristique précedente pour les autres actions :

$$\forall 1 \leq i \leq N,\ \hat{\mu}^{\pi_E}(s_i, a_i) = \sum_{j=i}^{N} \gamma^{j-i}\phi(s_j) \text{ et } \hat{\mu}^{\pi_E}(s_i, a \neq a_i) = \gamma\hat{\mu}^{\pi_E}(s_i, a_i). \quad (5)$$

Pour résumer, l'attribut moyen de l'expert peut être vu comme un vecteur de fonctions de qualité (pour une même politique π_E et pour différentes fonctions de récompense ϕ_i). En conséquence, tout algorithme d'évaluation de la fonction de qualité peut être utilisé pour estimer $\mu^{\pi}(s, a)$. Selon la quantité et la nature des données disponibles, une heuristique peut-être employée pour évaluer l'attribut moyen pour une action différente de celle de l'expert. Cette estimation n'est nécessaire que pour entraîner le classifieur, il est donc suffisant de disposer de valeurs uniquement pour les couples état-action $(s_i, a)_{1\leq i\leq N, a\in\mathcal{A}}$. En tous cas, estimer μ^{π_E} n'est pas plus coûteux que d'estimer la fonction de qualité d'une politique donnée, dans le cas *on-policy*, ce qui est rappelons-le moins coûteux que de trouver la politique optimale pour une fonction de récompense arbitraire (comme l'exigent la plupart des algorithmes d'ARI existants, voir section 5).

4.2. Instanciation

Comme précisé précedemment, tout algorithme de CMC peut être utilisé. Ici nous choisissons l'approche à marges structurées de (Taskar *et al.*, 2005). Soit $\mathcal{L} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}_+$ une fonction de marge définie par l'utilisateur satisfaisant $\mathcal{L}(s, \pi_E(s)) \leq \mathcal{L}(s, a)$ (ici, $\mathcal{L}(s_i, a_i) = 0$ et $\mathcal{L}(s_i, a \neq a_i) = 1$). L'algorithme CMC résout :

$$\min_{\theta,\zeta} \frac{1}{2}\|\theta\|^2 + \frac{\eta}{N}\sum_{i=1}^{N}\zeta_i \quad \text{t.q.} \quad \forall i, \theta^\top\hat{\mu}^{\pi_E}(s_i, a_i) + \zeta_i \geq \max_a \theta^\top\hat{\mu}^{\pi_E}(s_i, a) + \mathcal{L}(s_i, a).$$

De façon similaire à (Ratliff *et al.*, 2006), nous fournissons la forme *hinge-loss* équivalente (avec les variables d'ajustement ζ_i serrées, ce qui permet de déplacer les contraintes dans la fonction objectif) :

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \max_a \theta^\top\hat{\mu}^{\pi_E}(s_i, a) + \mathcal{L}(s_i, a) - \theta^\top\hat{\mu}^{\pi_E}(s_i, a_i) + \frac{\lambda}{2}\|\theta\|^2.$$

La fonction objectif est minimisée en utilisant une descente de sous-gradient. L'attribut moyen de l'expert est estimé en utilisant le principe décrit par l'équation (5).

5. Travaux connexes

La notion d'ARI a été pour la première fois introduite dans (Russell, 1998) puis formalisée dans (Ng, Russell, 2000). Une approche classique, initiée dans (Abbeel, Ng, 2004), consiste à trouver une politique (à travers une recherche dans l'espace des fonctions de récompense) telle que son attribut moyen (ou de manière plus générale une mesure de la distribution sous jacente aux trajectoires) s'approche de celui de la politique de l'expert. Voir (Neu, Szepesvari, 2009) pour un état de l'art partiel. Certains algorithmes mentionnés ne sont pas capables de renvoyer une fonction de récompense, bien qu'ils utilisent l'ARI comme étape. Ils rentrent dans le champ généralement appelé l'apprentissage par imitation.

Plus proches de notre contribution, quelques approches introduisent également de la structure dans la classification (Melo, Lopes, 2010 ; Ratliff *et al.*, 2006). Dans (Melo, Lopes, 2010), une métrique induite par le PDM est utilisée pour construire un noyau qui sera utilisé par l'algorithme de classification, permettant des améliorations par rapport à un noyau non structuré. Cette approche n'est cependant pas un algorithme d'ARI, et plus important l'évaluation d'une métrique dans un PDM n'est pas triviale. Dans (Ratliff *et al.*, 2006), un algorithme de classification est utilisé pour fournir une fonction de récompense. Au lieu d'associer des actions à des états comme nous le faisons, cet algorithme associe des politiques optimales (qui jouent le rôle de labels) à des PDM (entrées), ce qui permet d'incorporer la structure, au prix de la résolution d'un grand nombre de PDM.

Tous les algorithmes d'ARI à notre connaissance requièrent la résolution du problème direct de l'AR de manière répétée, à l'exception de (Dvijotham, Todorov, 2010 ; Boularias *et al.*, 2011). Le travail présenté dans (Dvijotham, Todorov, 2010) ne s'applique qu'aux PDM solvables linéairement (où le contrôle se fait en imposant une dynamique au système). Dans (Boularias *et al.*, 2011), en utilisant l'argument de l'entropie relative, une fonction objectif est maximisée via une montée de sous-gradient. Estimer la valeur du sous-gradient demande des trajectoires échantillonnées selon la politique optimale pour la fonction de récompense courante. On peut contourner le problème grâce à l'échantillonnage préférentiel. Cela requiert cependant d'échantillonner des trajectoires selon une politique différente de celle de l'expert, et le problème direct se maintient au coeur de l'approche (même si sa résolution est évitée).

SCIRL n'a pas besoin de résoudre le problème direct, mais uniquement d'estimer l'attribut moyen de la politique de l'expert. Autrement dit, plutôt que de résoudre plusieurs fois le problème de l'optimisation d'une politique, nous ne résolvons qu'une seule fois un problème d'évaluation de politique [2]. Cela amène des garanties théo-

2. La résolution d'un MDP (en utilisant par exemple un algorithme d'*itération de la valeur*) implique en effet de calculer de manière répétée la valeur de politiques arbitraires. Résoudre plusieurs fois le MDP pour

riques (ce qui n'est pas le cas de tous les algorithmes d'ARI, par exemple (Boularias *et al.*, 2011)). De plus, par l'utilisation d'une heuristique qui dépasse le cadre de notre analyse, il est possible à SCIRL de se contenter de données fournies par l'expert. La prochaine section présente une démonstration empirique de cela. Nous ne connaissons aucun autre algorithme d'ARI en mesure de fonctionner dans des conditions si drastiques.

6. Expériences

Nous illustrons SCIRL sur un simulateur de conduite similaire à (Abbeel, Ng, 2004 ; Syed, Schapire, 2008). Le but est de conduire une voiture sur une autoroute à trois voies dont le trafic est généré aléatoirement (les sorties de route sont possibles des deux côtés). La voiture peut se déplacer vers la gauche ou la droite, accélérer ou ralentir et conserver sa vitesse. L'expert optimise une récompense définie par nos soins, $\mathcal{R}_E$, qui récompense la vitesse, punit les sorties de route, punit sévèrement les collisions et ne donne pas d'information dans les autres cas.

Nous avons comparé SCIRL, tel qu'instancié comme décrit en section 4.2, à un classifieur non structuré (en utilisant le même algorithme de classification que celui placé au cœur de SCIRL) ainsi qu'à l'algorithme de (Abbeel, Ng, 2004) (appelé ici PIRL pour *Projection Inverse Reinforcement Learning*). Nous nous préoccupons également du comportement optimal vis-à-vis d'une récompense tirée aléatoirement (en utilisant la même paramétrisation que SCIRL et PIRL, le vecteur de paramètres est tiré selon une loi uniforme) afin de disposer d'un autre point de comparaison.

Pour SCIRL et PIRL on discrétise l'espace d'état en un vecteur d'attribut pour la récompense, $\phi \in \mathbb{R}^{729}$: 9 positions horizontales pour la voiture du joueur, 3 positions horizontales et 9 verticales pour la voiture la plus proche de celle du joueur et 3 vitesses. Ces attributs sont bien moins informatifs que ceux utilisés dans (Abbeel, Ng, 2004 ; Syed, Schapire, 2008). Les attributs de (Syed, Schapire, 2008) sont si informatifs que tirer aléatoirement un vecteur θ de paramètres positifs pour la récompense donne lieu à un comportement acceptable. Le facteur d'oubli est $\gamma = 0.9$. Le classifieur utilise le même vecteur d'attributs reproduit pour chaque action.

On donne à SCIRL n trajectoires de longueur n (débutant dans un état choisi aléatoirement), n allant de 3 à 20 (ce qui donne de 9 à 400 transitions). Chaque expérience est répétée 50 fois. Le classifieur utilise les mêmes données que SCIRL. PIRL est un algorithme itératif dont chaque itération implique la résolution du PDM pour une fonction de récompense arbitraire. Nous l'avons fait fonctionner pendant 70 itérations, tous les objets requis (l'attribut moyen de politiques différentes de celles de l'expert et la politique optimale pour une fonction de récompense à chaque itération) ont été calculés de manière exacte en utilisant le modèle. Nous mesurons les performances de chacune des approches grâce à $\mathbb{E}_{s\sim\mathcal{U}}[v^{\pi}_{\mathcal{R}_E}(s)]$, avec $\mathcal{U}$ la distribution uniforme (ce

---

des récompenses arbitraires est donc beaucoup plus dur que d'estimer une fois pour toutes l'attribut moyen de l'expert (sans mentionner le problème de l'apprentissage *off-policy*).

qui permet de tester la capacité de généralisation de chaque approche même pour des états infréquemment rencontrés), $\mathcal{R}_E$ étant la récompense de l'expert et π est l'une des politiques suivantes: la politique optimale pour $\mathcal{R}_E$ (point de référence haut), la politique optimale pour une récompense aléatoire (point de référence bas), la politique optimale pour $\mathcal{R}_{\theta_c}$ (SCIRL), la politique produite par PIRL et la règle de décision du classifieur.

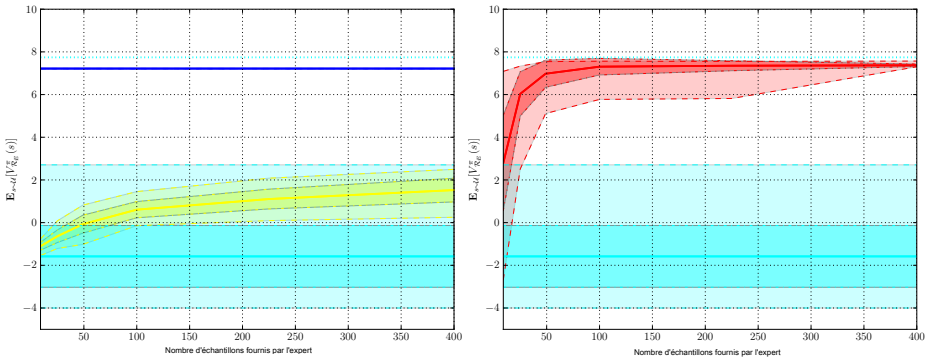

Figure 1. Problème de la conduite. La ligne la plus haute est la valeur de l'expert. Pour chaque courbe, nous dessinons la moyenne (trait plein), l'écart-type (foncé) et les valeurs min et max (plus clair). La politique correspondant à une récompense aléatoire occupe le bas des deux graphes, la politique retournée par le classifieur donne la courbe croissante du graphe de gauche et la politique optimale pour la récompense trouvée par SCIRL donne la courbe croissante du graphe de droite. La ligne solitaire constante à gauche correspond à PIRL.

La figure 1 montre la performance de chacune des approches en fonction du nombre de transitions de l'expert utilisées (sauf pour PIRL qui a utilisé le modèle). On peut voir que le classifieur ne fonctionne pas bien sur ce problème. Augmenter le nombre de transitions améliorerait ses performances, mais après 400 transitions, il ne fonctionne pas aussi bien que SCIRL avec uniquement une dizaine de transitions. Cela illustre la pertinence d'utiliser μ et non ϕ dans la paramétrisation de la fonction de score, l'attribut moyen contient en effet de l'information quant à la dynamique engendrée par la politique de l'expert. SCIRL fonctionne particulièrement bien ici : après seulement une centaine de transitions il atteint les performances de PIRL. Ces deux algorithmes sont proches de la valeur de l'expert. Nous ne fournissons pas de données exactes en ce qui concerne les temps de calcul, mais faire fonctionner SCIRL une fois avec 400 transitions en entrée est environ une centaine de fois plus rapide que de faire fonctionner PIRL pour 70 itérations.

7. Conclusion

Nous avons introduit une nouvelle approche de résolution du problème de l'ARI en structurant un classifieur à fonction de score linéairement paramétrée avec une estimée de l'attribut moyen de l'expert. Il renvoie une fonction de récompense pour laquelle nous avons montré que l'expert est quasi optimal, pour peu que l'erreur de classification soit faible et que l'estimation de l'attribut moyen de l'expert soit bonne. Des méthodes pratiques pour l'estimation de cette grandeur ont été présentées et nous avons introduit une heuristique pour le cas où les seules données disponibles proviennent de l'expert, ainsi qu'une instanciation spécifique de l'algorithme SCIRL. Nous avons exhibé, grâce à un simulateur de conduite sur autoroute, que l'approche proposée fonctionne bien (même combinée à l'heuristique discutée), bien mieux qu'un classifieur non structuré et aussi bien que la méthode de l'état de l'art qui avait accès au modèle (et avec un coût computationnel moindre). Nous prévoyons de poursuivre notre analyse des propritétés théoriques de SCIRL (notamment en vue de l'utilisation avec des heuristiques) et de l'appliquer à des problèmes robotiques réels.

Bibliographie

Abbeel P., Coates A., Ng A. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research*, vol. 29, n° 13, p. 1608–1639.

Abbeel P., Ng A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine learning (ICML).*

Bertsekas D. P., Tsitsiklis J. N. (1996). *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3)*. Athena Scientific. Hardcover.

Boularias A., Kober J., Peters J. (2011). Relative entropy inverse reinforcement learning. In *JMLR Workshop and Conference Proceedings Volume 15: AISTATS 2011.*

Bradtke S. J., Barto A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, vol. 22, n° 1-3, p. 33-57.

Dvijotham K., Todorov E. (2010). Inverse optimal control with linearly-solvable MDPs. In *Proceedings of the 27th International Conference on Machine Learning (ICML).*

Guermeur Y. (2007). VC theory of large margin multi-category classifiers. *Journal of Machine Learning Research*, vol. 8, p. 2551-2594.

Klein E., Geist M., Pietquin O. (2011). Batch, off-policy and model-free apprenticeship learning. In *Proceedings of the European Workshop on Reinforcement Learning (EWRL).*

Melo F. S., Lopes M. (2010). Learning from demonstration using MDP induced metrics. In *Proceedings of the European Conference on Machine Learning (ECML).*

Munos R. (2007). Performance bounds in L_p norm for approximate value iteration. *SIAM journal on control and optimization*, vol. 46, n° 2, p. 541-561.

Neu G., Szepesvari C. (2009). Training parsers by inverse reinforcement learning. *Machine Learning*, vol. 77, n° 2-3, p. 303-337.

Ng A. Y., Russell S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning (ICML).*

Puterman M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

Ratliff N., Bagnell A. D., Zinkevich M. (2006). Maximum margin planning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML).*

Russell S. (1998). Learning agents for uncertain environments (extended abstract). In *Proceedings of the 11th annual Conference on Computational Learning Theory (COLT).*

Sutton R. S., Barto A. G. (1998). *Reinforcement Learning: An Introduction* (3rd éd.). The MIT Press. Hardcover.

Syed U., Schapire R. (2008). A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems 20 (NIPS).*

Szepesvári C. (2010). *Algorithms for Reinforcement Learning*. Morgan and Claypool.

Taskar B., Chatalbashev V., Koller D., Guestrin C. (2005). Learning structured prediction models: a large margin approach. In *Proceedings of 22nd International Conference on Machine Learning (ICML).*