Apprentissage par démonstrations : vaut-il la peine d'estimer une fonction de récompense?

Bilal Piot, Matthieu Geist, Olivier Pietquin

Apprentissage par démonstrations : vaut-il la peine d'estimer une fonction de récompense?

Bilal Piot[1,2], Matthieu Geist[1], Olivier Pietquin[1,2]

[1] MaLIS research group (SUPELEC)
2 rue Edouard Belin, Metz, 57070 FRANCE prenom.nom@supelec.fr

[2] UMI 2958 (GeorgiaTech-CNRS)

Résumé : Cet article propose une étude comparative entre l'Apprentissage par Renforcement Inverse (ARI) et l'Apprentissage par Imitation (AI). L'ARI et l'AI sont deux cadres de travail qui utilisent le concept de Processus Décisionnel de Markov (PDM) et dans lesquels nous cherchons à résoudre le problème d'Apprentissage par Démonstrations (AD). L'AD est un problème où un agent appelé apprenti cherche à apprendre à partir de l'observation des démonstrations d'un autre agent appelé expert. Dans le cadre de travail de l'AI, l'apprenti essaie d'apprendre directement la politique de l'expert alors que dans le cadre de l'ARI, l'apprenti essaie d'apprendre la récompense qui explique la politique de l'expert. Cette récompense est ensuite optimisée pour imiter l'expert. On peut donc légitimement se demander s'il y a un intérêt à estimer une récompense qui devra ensuite être optimisée ou si l'estimation d'une politique est suffisante. Cette question assez naturelle n'a pas encore été réellement traitée dans la littérature pour l'instant. Ici, des réponses partielles à la fois d'un point de vue théorique et pratique sont produites. **Mots-clés** : Apprentissage par Renforcement Inverse, Apprentissage par Imitation, Apprentissage par Démonstrations.

1 Introduction

Cet article cherche à comparer deux cadres de travail qui utilisent le concept de Processus Décisionnel de Markov (PDM) et où l'on essaie de résoudre le problème d'Apprentissage par Démonstrations (AD). L'AD est un problème où un agent appelé apprenti cherche à apprendre à partir de l'observation des démonstrations d'un autre agent appelé expert. Les deux cadres de travail considérés sont l'Apprentissage par Imitation (AI) (Abbeel & Ng, 2004) et l'Apprentissage par Renforcement Inverse (ARI) (Russell, 1998). Dans le cadre de l'AI, l'apprenti essaie d'apprendre la politique de l'expert ou du moins une politique qui soit aussi bonne que celle de l'expert (par rapport à une fonction de récompense inconnue). Dans le cadre de l'ARI, l'apprenti essaie d'apprendre la fonction de récompense qui pourrait expliquer le comportement de l'expert et que l'agent devra optimiser pour imiter l'expert. L'AI peut être vu comme un problème de classification (Pomerleau, 1989; Atkeson & Schaal, 1997; Langford & Zadrozny, 2005; Syed & Schapire, 2010) où l'apprenti cherche à imiter l'expert via une méthode d'Apprentissage Supervisé (AS) comme la classification. Il existe aussi plusieurs algorithmes d'AI qui sont directement inspirés du cadre de l'ARI comme (Abbeel & Ng, 2004; Syed & Schapire, 2008) mais ces derniers ont besoin de résoudre des PDM de manière itérée, ce qui est en soi un problème difficile lorsque l'espace d'état est grand ou que la dynamique du PDM est inconnue.

L'idée centrale derrière l'ARI est que la fonction de récompense est une représentation compacte de la tâche à effectuer. Cependant, comme le résultat d'un algorithme d'ARI est une fonction de récompense, il est nécessaire de résoudre un PDM pour obtenir la politique optimale correspondant à la récompense trouvée. En ce qui concerne les algorithmes d'AI, le résultat est une politique qui peut directement être utilisée. Cependant, cette politique est fixée et ne peut donc s'adapter à une perturbation de la dynamique, ce qui pourrait être possible si la fonction de récompense était connue, puisque cette dernière est souvent indépendante de la dynamique. Ainsi, il est naturel de se demander : dans quelles circonstances est-il intéressant d'utiliser un algorithme d'ARI, sachant que nous devrons ensuite résoudre un PDM afin d'obtenir une politique ?

Pour tenter de répondre à cette question, une analyse théorique de la différence entre fonctions de valeurs des politiques de l'apprenti et de l'expert quand une méthode de classification est utilisée (dans le cas

à horizon infini) sera menée. Le résultat de cette analyse sera ensuite comparée au seul résultat (à notre connaissance) similaire d'ARI qui quantifie la qualité de l'apprenti lorsque celui-ci est entraîné par l'algorithme SCIRL (Klein *et al.*, 2012). L'analyse comparée des deux résultats nous montre que l'estimation de la récompense ne peut que rajouter de l'erreur dans la politique apprise après optimisation, comparativement à celle apprise via classification. Ensuite, une étude empirique sera menée sur un cadre de travail assez générique que sont les Garnets (Archibald *et al.*, 1995). Cette étude permettra de confirmer si le résultat théorique précédent est vérifié en pratique. Il s'avère en réalité que la qualité de la politique optimisée via la récompense apprise dépend grandement de la récompense inconnue qu'optimise l'expert. En effet, il semble que moins la récompense est informative, plus les algorithmes d'ARI apportent un gain par rapport aux algorithmes d'AI. Finalement, l'étude expérimentale sera poussée un peu plus loin que le cadre théorique étudié, puisque nous observerons la performance des différents algorithmes lorsque la dynamique du PDM est perturbée. Dans ce cas, l'avantage aux algorithmes d'ARI est clair.

2 Contexte et Notations

2.1 Notations générales

Soit $\mathcal{X} = (x_i)_{\{1 \leq i \leq N_\mathcal{X}\}}$ un ensemble fini et $f \in \mathbb{R}^\mathcal{X}$ une fonction, f peut être assimilée à un vecteur colonne et f^T est le vecteur transposé de f. Le support de f est noté $\mathrm{Supp}(f)$. L'ensemble des parties de $\mathcal{X}$ est noté $\mathbb{P}(\mathcal{X})$. L'ensemble des mesures de probabilité sur $\mathcal{X}$ est notée $\Delta_\mathcal{X}$. Soit $\mathcal{Y}$ un ensemble fini, $\Delta_\mathcal{X}^\mathcal{Y}$ est l'ensemble des fonctions de $\mathcal{Y}$ vers $\Delta_\mathcal{X}$. Soit $\zeta \in \Delta_\mathcal{X}^\mathcal{Y}$ et $y \in \mathcal{Y}$, $\zeta(y) \in \Delta_\mathcal{X}$, qui peut être vu comme une mesure de probabilité conditionnelle sachant y, est aussi notée $\zeta(.|y)$ et $\forall x \in \mathcal{X}, \zeta(x|y) = [\zeta(y)](x)$. De plus, soit $A \in \mathbb{P}(\mathcal{X})$, alors $\chi_A \in \mathbb{R}^\mathcal{X}$ est la fonction indicatrice sur l'ensemble $A \subset \mathcal{X}$. Soit $\mu \in \Delta_\mathcal{X}$, $\mathbb{E}_\mu[f]$ est l'espérance de la fonction f par rapport à la mesure de probabilité μ. Soit $x \in \mathcal{X}$, $x \sim \mu$ signifie que x est tiré selon μ. Finalement, nous définissons aussi pour $p \in \mathbb{N}^*$, la $\mathbb{L}_p$-norm de la fonction f : $\|f\|_p = (\sum_{x \in \mathcal{X}}(f(x)^p))^{\frac{1}{p}}$, et $\|f\|_\infty = \max_{x \in \mathcal{X}} f(x)$.

2.2 Processus Décisionnel de Markov

Un Processus Décisionnel de Markov (PDM) fini est un quintuplet $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$ où $\mathcal{S} = (s_i)_{\{1 \leq i \leq N_\mathcal{S}\}}$ est un ensemble fini d'états, $\mathcal{A} = (a_i)_{\{1 \leq i \leq N_\mathcal{A}\}}$ un ensemble fini d'actions, $\mathcal{P} \in \Delta_\mathcal{S}^{\mathcal{S} \times \mathcal{A}}$ est la dynamique markovienne du PDM, $\mathcal{R} \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ est la fonction de récompense et γ est le facteur d'actualisation. Une politique markovienne et stationnaire $\pi \in \Delta_\mathcal{A}^\mathcal{S}$ représente le comportement d'un agent agissant dans un PDM $\mathcal{M}$. L'ensemble des politiques markoviennes et stationnaires est noté $\Pi_{MS} = \Delta_\mathcal{A}^\mathcal{S}$. Quand la politique π est déterministe, elle peut aussi être vue comme un élément de $\mathcal{A}^\mathcal{S}$ et $\pi(s)$ est l'action choisie par la politique π dans l'état s. La qualité d'une politique, dans la cadre d'un horizon infini, est quantifiée par la fonction de valeur $v_\mathcal{R}^\pi \in \mathbb{R}^\mathcal{S}$ qui, à chaque état s, fait correspondre le cumul des récompenses espéré et actualisé en partant de s et en suivant la politique π ensuite :

$$\forall s \in \mathcal{S}, v_\mathcal{R}^\pi(s) = \mathbb{E}[\sum_{t \geq 0} \gamma^t \mathcal{R}(s_t, a_t) | s_0 = s, \pi].$$

Une politique $\pi_\mathcal{R}^*$ est dite optimale (par rapport à la fonction de récompense $\mathcal{R}$) si sa fonction de valeur $v_\mathcal{R}^*$ satisfait $v_\mathcal{R}^* \geq v_\mathcal{R}^\pi$ pour toute politique π (il s'agit ici d'une inégalité composante par composante). Soit P_π la matrice stochastique $P_\pi = (\sum_{a \in A} \pi(a|s)\mathcal{P}(s'|s,a))_{\{(s,s') \in \mathcal{S}^2\}}$ et $\mathcal{R}_\pi \in \mathbb{R}^\mathcal{S}$ la fonction telle que : $\forall s \in \mathcal{S}, \mathcal{R}_\pi(s) = \sum_{a \in A} \pi(a|s)\mathcal{R}(s,a)$. Avec un léger abus de notation, a pourra s'identifier à la politique qui associe l'action a à chaque état s. Les opérateurs d'évaluation (resp. d'optimalité) de Bellman $T_\mathcal{R}^\pi$ (resp. $T_\mathcal{R}^*$) : $\mathcal{R}^\mathcal{S} \to \mathcal{R}^\mathcal{S}$ sont définis tels que :

$$T_\mathcal{R}^\pi v = \mathcal{R}_\pi + \gamma P_\pi v \text{ et } T_\mathcal{R}^* v = \max_\pi T_\mathcal{R}^\pi v.$$

Ces opérateurs sont des contractions et $v_\mathcal{R}^\pi$ et $v_\mathcal{R}^*$ sont leur points fixes respectifs : $v_\mathcal{R}^\pi = T_\mathcal{R}^\pi v_\mathcal{R}^\pi$ et $v_\mathcal{R}^* = T_\mathcal{R}^* v_\mathcal{R}^*$. La fonction de qualité $\mathcal{Q}_\mathcal{R}^\pi \in \mathcal{S} \times \mathcal{A}$ ajoute un degré de liberté sur le choix de la première action, elle est formellement définie comme $\mathcal{Q}_\mathcal{R}^\pi(s,a) = [T_\mathcal{R}^a v_\mathcal{R}^\pi](s)$. On écrit aussi, quand elle existe, $\rho_\pi \in \mathbb{R}^\mathcal{S}$ la mesure de probabilité stationnaire de la politique π (qui satisfait $\rho_\pi^T P_\pi = \rho_\pi^T$). L'existence et l'unicité de ρ_π est garantie quand la chaîne de Markov induite par la matrice stochastique P_π est irréductible, ce qui sera supposé vrai dans la suite de l'article.

2.3 AI et ARI

L'AI et l'ARI sont deux cadres de travail où l'on essaie de résoudre le problème d'AD via le paradigme des PDM. Plus précisément, dans le cadre de l'AI, l'apprenti, à partir d'observations de la politique experte π_E, essaie d'apprendre une politique π_A qui est aussi bonne que celle de l'expert par rapport à la récompense inconnue $\mathcal{R}$ que l'expert essaie d'optimiser (souvent, l'expert est considéré optimal : $v_{\mathcal{R}}^{\pi_E} = v_{\mathcal{R}}^*$). Cela peut être exprimé numériquement ; l'apprenti cherche une politique π_A telle que la quantité $\mathbb{E}_\nu[v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_A}]$ soit la plus petite possible, où $\nu \in \Delta_S$. En général, $\nu = \rho$ où ρ est la mesure de probabilité uniforme sur $\mathcal{S}$ ou $\nu = \rho_{\pi_E}$ (ρ_{π_E} est aussi noté ρ_E). Dans le cadre de l'ARI, l'apprenti essaie d'apprendre une récompense $\hat{\mathcal{R}}$ qui pourrait expliquer la politique de l'expert. Plus précisément, à partir d'observations de la politique experte π_E, l'apprenti essaie d'apprendre $\hat{\mathcal{R}}$ telle que $\pi_E \approx \pi_{\hat{\mathcal{R}}}^*$. Cela peut aussi être exprimé numériquement, l'apprenti essaie de trouver une fonction de récompense $\hat{\mathcal{R}}$ telle que les quantités $\mathbb{E}_\nu[v_{\hat{\mathcal{R}}}^{\pi_{\hat{\mathcal{R}}}^*} - v_{\hat{\mathcal{R}}}^{\pi_E}]$ ou $\mathbb{E}_\nu[v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_{\hat{\mathcal{R}}}^*}]$ soient les plus petites possibles.

3 Etude théorique

Cette section donne certains éléments de réponse théoriques à la question : est-il intéressant d'estimer la fonction de récompense ? Tout d'abord, nous présentons un résultat inédit en ce qui concerne la classification vue comme une méthode d'AI dans le cadre d'un horizon infini. Une preuve de ce résultat est donné dans l'annexe A. Ce résultat est une borne supérieure sur la différence entre les fonctions de valeurs de l'expert et de l'apprenti. Comme une borne pour la classification vue comme une méthode d'ARI a été précédemment trouvée dans le cadre de l'horizon fini par Syed & Schapire (2010), une comparaison informelle de ces deux résultats sera faite. De plus, sachant qu'il existe une borne sur la performance de l'algorithme d'ARI appelé SCIRL (Klein *et al.*, 2012), nous pourrons comparer les performances de méthodes d'ARI et d'AI d'un point de vue théorique. Nous avons choisi de nous intéresser aux bornes précédentes car la classification et SCIRL sont des algorithmes qui ne nécessitent pas de résoudre des PDM de manière itérative. Ainsi, il n'y a pas de terme d'erreur de programmation dynamique approchée à prendre en compte et à propager pour obtenir la performance de ce genre d'algorithmes.

3.1 AI vu comme un problème de classification dans le cas à horizon infini

Une façon assez simple de réaliser un algorithme d'AI est par pure imitation via un algorithme d'AS comme la classification. Plus précisément, nous supposons disposer d'un ensemble d'exemples experts $D_E = (s_i, a_i)_{\{1 \leq i \leq N\}}$ où $a_i \sim \pi_E(.|s_i)$. Sans perte de généralité, nous supposons que les états s_i sont tirés selon une mesure de probabilité $\nu \in \Delta_{\mathcal{S}}$. Ainsi les exemples (s_i, a_i) sont tirés selon μ_E telle que : $\mu_E(s, a) = \nu(s)\pi_E(a|s)$. Ensuite, un classifieur est appris à partir des exemples (lorsque les actions sont discrètes, il s'agit d'un problème de classification multi-classes) grâce à un algorithme d'AS. Le résultat de l'algorithme est une politique $\pi_C \in \mathcal{A}^{\mathcal{S}}$, qui associe à chaque état une action. La qualité du classifieur est quantifiée par l'erreur de classification :

$$\epsilon_C = \mathbb{E}_{\mu_E}\big[\chi_{\{(s,a) \in \mathcal{S} \times \mathcal{A}, \pi_C(s) \neq a\}}\big].$$

La qualité de l'expert (par rapport à la fonction de récompense $\mathcal{R}$) peut être quantifiée avec $v_{\mathcal{R}}^{\pi_E}$. Usuellement, l'expert est supposé optimal (c'est-à-dire, $v_{\mathcal{R}}^{\pi_E} = v_{\mathcal{R}}^*$), mais cela n'est pas nécessaire pour l'analyse suivante (l'expert pourra être sous-optimal par rapport à $\mathcal{R}$). La qualité de la politique π_C est quantifiée par sa fonction de valeur $v_{\mathcal{R}}^{\pi_C}$. Ainsi, nous allons borner $\mathbb{E}_\nu[v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_C}]$ qui représente la différence entre la qualité de l'expert et celle de la politique du classifieur. Si cette quantité est négative, cela est bon, car en moyenne cela voudra dire que π_C est meilleur que π_E. C'est pourquoi seule une borne supérieure est nécessaire. Cette borne supérieure montre l'efficacité d'une simple méthode de classification comme algorithme d'AI dans le cas à horizon infini.

Soit le coefficient de concentration suivant : $C_\nu = (1-\gamma)\sum_{t \geq 0} \gamma^t c_\nu(t)$ où pour tout $t \in \mathbb{N}$, $c_\nu(t) = \max_{s \in \mathcal{S}} \frac{(\nu^T P_{\pi_E}^t)(s)}{\nu(s)}$. On remarque que si $\nu = \rho_E$, ce qui est une hypothèse raisonnable, alors $C_\nu = C_{\rho_E} = 1$.

Théorème 1
Soit π_C la politique du classifieur (entraîné sur l'ensemble d'exemples experts D_E pour imiter la politique de l'expert π_E). Soit aussi ϵ_C l'erreur de classification et C_ν le coefficient de concentration défini plus haut. Alors $\forall \mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$:

$$\mathbf{E}_\nu[v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_C}] \leq \frac{2C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C. \quad (1)$$

De plus, il est intéressant de noter que pour une récompense $\mathcal{R}$ qui dépend uniquement des états, c'est-à-dire $\forall (s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{A}, \mathcal{R}(s,a) = \mathcal{R}(s,b)$, la borne donnée par l'Eq. (1) est améliorée. En effet, pour une récompense qui dépend uniquement des états, la borne devient :

$$\mathbf{E}_\nu[v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_C}] \leq \frac{2\gamma C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C.$$

Il y a donc une amélioration d'un facteur γ entre les récompenses dépendant uniquement des états et les autres fonctions de récompense. La preuve du Th. 1 est donnée en annexe A et est basée sur la propagation de l'erreur de classification. Syed & Schapire (2010) ont établi une borne similaire dans le cas à horizon fini. Cependant, comme la plupart des algorithmes d'AI et d'ARI considèrent le cadre à horizon infini, nous pensons notre résultat utile pour comparer les différentes méthodes.

3.2 Une borne dans le cas à horizon fini

Dans cette section, nous introduisons des notations spécifiques au cas à horizon fini et les résultat de Syed & Schapire (2010) sont interprétés. Considérons le PDM fini $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$ avec pour horizon H et sans facteur d'actualisation γ. Une politique markovienne et non-stationnaire est un élément de Π_{MS}^H ; si π est non stationnaire, alors π^t réfère à la politique stationnaire qui est égal à la t-ième composante de π. D'une manière similaire au cas à horizon infini, la fonction de valeur de la politique π au temps t est définie par :

$$\forall s \in \mathcal{S}, v_{t,\mathcal{R}}^{\pi}(s) = \mathbb{E}[\sum_{t'=t}^{H} \mathcal{R}(s_{t'}, a_{t'})|s_t = s, \pi].$$

On définit aussi $D_{\pi,\nu}^t$ la mesure de probabilité sur les paires états-actions au temps t induite par la politique π. c'est-à-dire qu'un exemple (s,a) est tiré de $D_{\pi,\nu}^t$ en tirant d'abord $s_1 \sim \nu \in \Delta_{\mathcal{S}}$, puis en suivant la politique π, à partir du pas de temps 1 jusqu'au pas de temps t, nous obtenons ainsi une trajectoire $(s_1, a_1, \ldots, s_t, a_t)$ et nous posons $(s,a) = (s_t, a_t)$. D'une manière plus formelle, nous avons :

$$\forall 1 \leq t \leq H, \forall (s,a) \in \mathcal{S}\times\mathcal{A}, D_{\pi,\nu}^t(s,a) = (\nu^T(P_{\pi^1} \times \cdots \times P_{\pi^{t-1}}))(s)\pi^t(a|s).$$

Dans leur article, Syed & Schapire (2010) supposent l'accès à un ensemble de trajectoires $D_E = (\omega_i)_{\{1\leq i\leq N\}}$ où $\omega_i = (s_1^i, a_1^i, \ldots, s_H^i, a_H^i)$ avec $s_1^i \sim \nu \in \Delta_{\mathcal{S}}$ et $(s_t^i, a_t^i) \sim D_{\pi_E,\nu}^t$ où $1 \leq t \leq H$ et π_E est la politique experte qui est supposée non-stationnaire et markovienne. Dans le cas à horizon fini, l'AI vu comme classification consiste à apprendre une politique $\pi_C = (\pi_C^t)_{\{1\leq t\leq H\}}$ grâce à H classifieurs entraînés sur les ensembles $D_E^t = (s_t^i, a_t^i)_{\{1\leq i\leq N\}}$. Ainsi, pour chaque ensemble $D_E^t = (s_t^i, a_t^i)_{\{1\leq i\leq N\}}$, un classifieur multi-classe est entraîné et une politique déterministe π_C^t est apprise avec pour erreur de classification :

$$\epsilon_C^t = \mathbb{E}_{D_E^t}[\chi_{\{(s,a)\in\mathcal{S}\times\mathcal{A}, \pi_C^t(s)\neq a\}}].$$

On note $\epsilon_C = \max_{1\leq t\leq H} \epsilon_C^t$. Avec ces notations, nous avons le théorème suivant :

Théorème 2
Soit π_E la politique experte supposée non-stationnaire et markovienne, D_E un ensemble de N trajectoires avec $s_1^i \sim \nu \in \Delta_{\mathcal{S}}$ et π_C la politique apprise par H classifieurs, alors :

$$\mathbb{E}_\nu[v_{1,\mathcal{R}}^{\pi_E} - v_{1,\mathcal{R}}^{\pi_C}] \leq \min(2\sqrt{\epsilon_C}H^2, 4\epsilon_C H^3 + \delta_{\pi_E})\|\mathcal{R}\|_\infty,$$

où $\delta_{\pi_E} = \frac{\mathbb{E}_\nu[v_{1,\mathcal{R}}^* - v_{1,\mathcal{R}}^{\pi_E}]}{\|\mathcal{R}\|_\infty}$ représente la sous-optimalité de l'expert.

Il est possible de comparer ces résultats avec notre borne, même si les cadres sont légèrement différents, en remarquant d'une manière informelle que l'introduction d'un facteur d'actualisation γ dans le cas à horizon infini correspond à un horizon moyen de longueur $\frac{1}{1-\gamma}$: $\sum_{t\geq 0}\gamma^t = \frac{1}{1-\gamma}$. En remplaçant H par $\frac{1}{1-\gamma}$ dans la borne précédente, nous obtenons :

$$\mathbb{E}_\nu[v_{1,\mathcal{R}}^{\pi_E} - v_{1,\mathcal{R}}^{\pi_C}] \leq \min(\frac{2\sqrt{\epsilon_C}}{(1-\gamma)^2}, \frac{4\epsilon_C}{(1-\gamma)^3} + \delta_{\pi_E})\|\mathcal{R}\|_\infty.$$

Ainsi, en identifiant d'une manière informelle les erreurs de classification et l'horizon H à $\frac{1}{1-\gamma}$, notre borne est légèrement meilleure soit par un facteur $\sqrt{\epsilon_C}$, soit par un facteur $\frac{2}{1-\gamma}$. De plus, comme notre borne correspond au cadre à horizon infini, elle est plus adaptée aux algorithmes d'AI et d'ARI puisque la plupart d'entre eux considèrent le cas à horizon infini.

3.3 SCIRL et sa borne de performance

Klein *et al.* (2012) supposent que la fonction de récompense inconnue est linéairement paramétrée par un vecteur de fonctions de base (on parle aussi d'attribut vectoriel). Plus précisément, soit $\phi(s,a) = (\phi_1(s,a),\ldots,\phi_p(s,a))^T$ un vecteur composée de $p \in \mathbb{N}^*$ attributs vectoriels $\phi_i \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, la fonction de récompense est alors $\mathcal{R}_\theta(s,a) = \theta^T\phi(s,a) = \sum_{1\leq i\leq p}\theta_i\phi_i(s,a)$. Trouver une bonne fonction de récompense revient alors à trouver un bon vecteur de paramètres $\theta \in \mathbb{R}^p$. Le choix des attributs vectoriels est pour cet algorithme fait par l'utilisateur. De plus, SCIRL a besoin de l'estimation de l'attribut vectoriel moyen ω_{π_E} qui correspond au cumul moyen et actualisé de chaque attribut vectoriel vu comme fonction de récompense, en partant d'un état donné, en choisissant une action donnée et en suivant ensuite la politique experte :

$$\omega_{\pi_E}(s,a) = \mathbb{E}[\sum_{t\geq 0}\gamma^t\phi(s_t,a_t)|s_0 = s, a_0 = a, \pi_E].$$

On remarque alors que : $\mathcal{Q}_{\mathcal{R}_\theta}^{\pi_E}(s,a) = \theta^T\omega_{\pi_E}(s,a)$. Ensuite, une estimation de l'attribut vectoriel moyen $\hat{\omega}_{\pi_E}$ est réalisée via un ensemble d'exemples expert : D_E. Le problème d'estimation de l'attribut vectoriel moyen de l'expert est en fait un problème d'évaluation de la politique. Puis, SCIRL utilise l'estimation de l'attribut vectoriel $\hat{\omega}_{\pi_E}$ comme base de fonctions d'un classifieur multi-classes basé sur une fonction de score qui est paramétrée linéairement par rapport à $\hat{\omega}_{\pi_E}$. Ce classifieur est entraîné grâce à l'ensemble D_E. L'erreur de classification est notée $\epsilon_C = \mathbb{E}_{\mu_E}[\chi_{\{(s,a)\in\mathcal{S}\times\mathcal{A},\pi_C(s)\neq a\}}]$ avec $\pi_C(s) = \text{argmax}_{a\in\mathcal{A}}\,\theta_C^T\hat{\omega}_{\pi_E}(s,a)$ et θ_C la sortie du classifieur. La sortie de SCIRL est la fonction de récompense $\mathcal{R}_C = \theta_C^T\phi$. La performance de l'algorithme est quantifiée par la borne suivante :

$$0 \leq \mathbb{E}_{\rho_E}[v_{\mathcal{R}_C}^* - v_{\mathcal{R}_C}^{\pi_E}] \leq \frac{C_f}{(1-\gamma)}\left(\frac{2\|\mathcal{R}_C\|_\infty\epsilon_C}{1-\gamma} + \bar{\epsilon}_Q\right),$$

avec $C_f = (1-\gamma)\sum_{t\geq 0}\gamma^t c_f(t)$ où pour tout $t \in \mathbb{N}$, $c_f(t) = \max_{s\in\mathcal{S}}\frac{(\rho_E^T P_{\pi_{\mathcal{R}_C}^*}^t)(s)}{\rho_E(s)}$. De plus, $\bar{\epsilon}_Q = \mathbb{E}_{\rho_E}[\max_{a\in\mathcal{A}}\epsilon_Q(.,a) - \min_{a\in\mathcal{A}}\epsilon_Q(.,a)]$, où $\epsilon_Q(s,a) = \theta_C^T(\hat{\omega}_{\pi_E}(s,a) - \omega_{\pi_E}(s,a))$, est une mesure de l'erreur de l'estimation de l'attribut vectoriel moyen. Cette borne est spécifique à la fonction de récompense $\mathcal{R}_C$ et la constante C_f n'est pas égale à 1 quand $\nu = \rho_E$, ce qui la rend probablement beaucoup moins fine que celle obtenue pour la classification pure, cela même si l'attribut vectoriel moyen est parfaitement estimé ($\bar{\epsilon}_Q = 0$). Cela semble indiquer que cet algorithme d'ARI est moins intéressant qu'un simple algorithme de classification, en théorie. Cependant, en pratique, nous verrons que pour des fonctions de récompense ayant une structure spécifique, SCIRL peut avoir une bien meilleure performance qu'un simple algorithme de classification (voir Sec. 4).

4 Etude empirique

Cette section montre à travers divers expériences que les bornes théoriques précédentes ne sont pas entièrement représentatives des performances réelles des algorithmes d'ARI et d'AI. Ici, nous menons plusieurs expériences qui montrent, dans une certaine mesure, l'intérêt d'estimer une fonction de récompense et cela grâce à un cadre de travail assez général qu'est celui des Garnets (Archibald *et al.*, 1995). Nous avons choisi

plus particulièrement ce cadre de travail, où les problèmes sont des PDM finis avec une représentation tabulaire des fonctions de valeurs, car ils permettent de pouvoir comparer les différents algorithmes. En effet, même si ces problèmes sont assez faciles à appréhender, ils évitent le problème de biais induit par le choix de la représentation et permettent de traiter une variété importante de situations. La comparaison est faite entre un algorithme de classification pure et deux récents algorithmes d'ARI que sont SCIRL et Relative Entropy IRL (RE) (Boularias *et al.*, 2011), pour lequel il n'y a pas d'analyse de performance connue. L'algorithme de classification a été choisi comme référence pour le cadre de travail de l'AI car il possède une garantie théorique de performance et ne résout pas itérativement des PDM contrairement à la plupart des algorithmes d'AI. SCIRL et RE ont été choisis comme références pour le cadre de travail de l'ARI car eux aussi ne doivent pas résoudre de PDM itérativement, ce qui réduit l'impact de la programmation dynamique approchée (PDA) dans l'interprétation des résultats, même si la fonction de récompense estimée devra ensuite être optimisée. L'optimisation se fera d'ailleurs via l'algorithme d'itération de la politique, pour ne pas introduire d'erreur de PDA et pouvoir avoir une interprétation des résultats qui ne dépende pas de l'algorithme d'optimisation de la récompense (même s'il faut garder en tête que la performance réelle des algorithmes d'ARI est dégradée par cette phase d'optimisation). Ces expériences montrent que le choix de la structure de la fonction de récompense inconnue, qui est utilisée pour pouvoir obtenir la politique experte à imiter via l'algorithme d'itération de la politique, est crucial. En effet, quand la récompense inconnue est une fonction gaussienne pour chaque état-action, la classification a plutôt une bonne performance alors qu'elle a plutôt une faible performance quand la récompense est parcimonieuse ou dépend uniquement de l'état. L'idée intuitive derrière ces résultats est : quand la récompense est très informative, l'impact de l'horizon d'optimisation est réduit, ce qui favorise la classification.

4.1 Description des algorithmes d'AL et d'ARI utilisés

Le premier algorithme utilisé est un algorithme de pure classification. Plus précisément, il s'agit d'une classification multi-classe entraînée sur l'ensemble D_E et qui utilise une approche de large marge structurée (Taskar *et al.*, 2005) qui consiste à minimiser le critère suivant par rapport à $\mathcal{Q} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$:

$$\mathfrak{L}_0(\mathcal{Q}) = \frac{1}{N}\sum_{i=1}^{N} \max_{a\in\mathcal{A}}[\mathcal{Q}(s_i,a) + l(s_i,a)] - \mathcal{Q}(s_i,a_i) + \lambda\|\mathcal{Q}\|_2^2,$$

où $l(s,a) = 0$ quand $\exists 1 \leq i \leq N, (s,a) = (s_i,a_i)$ et $l(s,a) = 1$ sinon. La minimisation est réalisée via une descente de sous-gradient (Shor *et al.*, 1985). La politique obtenue par l'algorithme est une politique déterministe, telle que $\pi_C(s) \in \operatorname{argmax}_{a\in\mathcal{A}} \mathcal{Q}^*(s,a)$ où $\mathcal{Q}^*$ est la sortie de la minimisation du critère $\mathfrak{L}_0$ via la descente de sous-gradient. Les deux autres algorithmes sont des algorithmes d'ARI. SCIRL (présenté en Sec. 3.3) a besoin aussi de l'ensemble D_E pour être implémenté et a pour sortie la récompense $\mathcal{R}_C$. L'instantiation de SCIRL utilisée dans nos expériences est celle décrite dans l'article d'origine. Afin d'obtenir une politique π_C, cette récompense est optimisée via l'algorithme d'itération de la politique par rapport à la récompense $\mathcal{R}_C$. Comme nous l'avons dit, l'algorithme d'itération de la politique a besoin de la connaissance de toute la dynamique du PDM pour être exécuté mais permet une comparaison qui ne dépend pas du choix de l'algorithme de PDA (nous avons besoin de résoudre un PDM pour connaître la performance de la fonction de récompense estimée, mais pas pour obtenir cette estimation). Comme SCIRL, l'algorithme RE suppose une paramétrisation linéaire de la récompense. Le principe de la méthode de Relative Entropy est basé sur la minimisation de l'entropie relative (divergence de Kullbach-Leibler) entre la distribution empirique des trajectoires états-actions induite par une politique aléatoire et celle induite par une politique qui aurait les mêmes attributs vectoriels moyens que l'expert (voir Boularias *et al.*, 2011, pour les détails). L'algorithme RE utilisé dans cet article est celui décrit dans l'article d'origine. Il a besoin en entrée de l'ensemble D_E et aussi d'un ensemble D_P de trajectoires tirées selon une politique non-experte. Dans les expériences, une politique aléatoire sera choisie pour générer l'ensemble D_P (voir Sec. 4.3). La sortie de l'algorithme RE est une récompense $\mathcal{R}_C$ et l'algorithme d'itération de la politique est aussi utilisé pour obtenir une politique π_C relative à la récompense $\mathcal{R}_C$.

4.2 Le cadre de travail des Garnets

Les Garnets sont une classe de PDM construits de manière aléatoire, qui sont totalement abstraits bien que représentatifs du genre de PDM finis que l'on peut rencontrer en pratique (voir Archibald *et al.*, 1995)). La routine pour créer un Garnet stationnaire est caractérisée par 3 paramètres et s'écrit $Garnet(N_S, N_A, N_B)$.

Les paramètres N_S et N_A sont le nombre d'états et d'actions respectivement, et N_B est un facteur de branchement qui spécifie le nombre d'états suivants pour chaque couple état-action. Les états suivants sont choisis aléatoirement dans l'espace d'état sans replacement pour chaque couple état-action. La probabilité de transiter vers chaque état suivant est générée en partitionnant l'intervalle unité avec $N_B - 1$ points de coupure choisis aléatoirement. La récompense $\mathcal{R}(s, a)$ sera choisie en fonction des expériences. Pour chaque Garnet, il est possible de calculer la politique expert π_E grâce à la récompense $\mathcal{R}$ via l'algorithme d'itération de la politique. Finalement, le facteur d'actualisation est fixé à $0,99$.

4.3 Pure classification versus SCRIL et RE

L'idée, pour obtenir un résultat qui soit assez général, est d'exécuter la même expérience sur des centaines de PDM et de regrouper à la fin les résultats obtenus. Tous les algorithmes utilisés ont pour entrée des ensembles du type : $D_E = (s_i, a_i)_{\{1\leq i\leq N\}}$ où $a_i \sim \pi_E(.|s_i)$. Plus précisément, $D_E = (\omega_j)_{\{1\leq i\leq K_E\}}$ où $\omega_j = (s_{i,j}, a_{i,j})_{\{1\leq i\leq H_E\}}$ est une trajectoire obtenue en commençant par un état choisi aléatoirement $s_{1,j}$ (distribution uniforme) et en suivant la politique π_E H_E fois ($s_{i+1,j} \sim P(.|s_{i,j}, a_{i,j})$). Donc, D_E est composé par K_E trajectoires de la politique π_E de taille H_E et nous avons $K_E H_E = N$. On procure aussi à l'algorithme RE un ensemble de transitions $D_P = (s_i, a_i, s'_i)_{\{1\leq i\leq N'\}}$ où $a_i \sim \pi_R(.|s_i)$ avec π_R la politique aléatoire (distribution uniforme sur les actions pour chaque état) et où $s'_i \sim P(.|s_i, a_i)$. En fait, D_P a la forme particulière suivante $D_P = (\tau_j)_{\{1\leq j\leq K_P\}}$ où $\tau_j = (s_{i,j}, a_{i,j}, s'_{i,j})_{\{1\leq i\leq H_P\}}$ est une trajectoire obtenue en commençant par un état choisi aléatoirement $s_{1,j}$ (distribution uniforme) et en suivant la politique π_R H_P fois ($s'_{i,j} = s_{i+1,j} \sim P(.|s_{i,j}, a_{i,j})$). Donc, D_P est composé par K_P trajectoires de la politique π_R de taille H_P et nous avons $K_P H_P = N'$. Ainsi, si pour un Garnet on a π_E et π_R, l'ensemble de paramètres (K_E, H_E, K_P, H_P) est suffisant pour instantier les ensembles de type D_E et D_P.

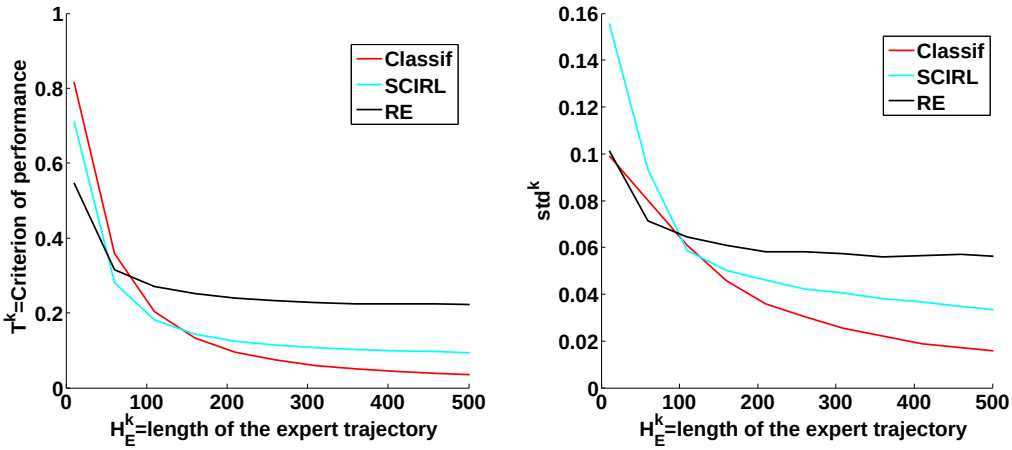
La première expérience montre la performance des différents algorithmes quand H_E (longueur des trajectoires expertes) augmente et quand la récompense $\mathcal{R}$ *inconnue* de chaque Garnet est choisie par une variable gaussienne normal centrée-réduite pour chaque couple état-action. Elle consiste à générer 100 Garnets de type $Garnet(N_S, N_A, N_B)$, où N_S est choisie uniformément entre 50 et 100, N_A uniformément entre 5 et 10 et N_B uniformément entre 2 et 5. Cela nous donne l'ensemble de Garnets $\mathfrak{G} = (G_p)_{\{1\leq p\leq 100\}}$. Sur chaque Garnet G_p de l'ensemble $\mathfrak{G}$, nous calculons π_E^p (via l'itération de la politique appliquée à $\mathcal{R}$) and π_R^p. Le paramètre H_E prend ses valeurs dans l' ensemble $(H_E^k)_{\{1\leq k\leq 11\}} = (50, 100, 150, .., 500)$, $K_E = 1$, $H_P = 10$, $K_P = 50$. Ensuite, pour chaque ensemble de paramètres (K_E, H_E^k, K_P, H_P) et pour chaque G_p, nous générons 100 ensembles $(D_E^{i,p,k})_{\{1\leq i\leq 100\}}$ et 100 ensembles $(D_P^{i,p,k})_{\{1\leq i\leq 100\}}$. Le critère de performance choisie pour chaque couple $(D_E^{i,p,k}, D_P^{i,p,k})$ est le suivant :

$$T^{i,p,k} = \frac{\mathbb{E}_\rho[v_{\mathcal{R}}^{\pi_E^p} - v_{\mathcal{R}}^{\pi_C^{i,p,k}}]}{\mathbb{E}_\rho[v_{\mathcal{R}}^{\pi_E^p}]},$$

où π_E^p est la politique experte, $\pi_C^{i,p,k}$ la politique induite par l'algorithme qui a pour entrée $(D_E^{i,p,k}, D_P^{i,p,k})$ et ρ est la distribution (mesure de probabilité) uniforme sur l'espace d'état $\mathcal{S}$. Pour l'algorithme de classification pure, nous avons $\pi_C^{i,p,k}(s) \in \operatorname{argmax}_{a\in A} \hat{Q}^*(s, a)$ où $\hat{Q}^*$ est le minimiseur trouvé par la descente de gradient sur le critère $\mathfrak{L}_0$. Pour les algorithmes SCIRL et RE, $\pi_C^{i,p,k}$ est la politique obtenue en optimisant la récompense $\mathcal{R}_C$ sortie par les algorithmes d'ARI via l'algorithme d'itérations de la politique. Le critère moyen de performance est donc pour chaque ensemble de paramètres (K_E, H_E^k, K_P, H_P) : $T^k = \frac{1}{10000} \sum_{1\leq p\leq 100, 1\leq i\leq 100} T^{i,p,k}$. Ensuite pour chaque algorithme, nous traçons $(H_E^k, T^k)_{\{1\leq k\leq 15\}}$. Un autre critère qui est également utile pour interpréter les résultats est une information sur la variance. Pour chaque Garnet et chaque ensemble de paramètres, l'écart-type $std^{p,k}$ est pour chaque algorithme :

$$std^{p,k} = \left\{ \frac{1}{100} \sum_{1\leq i\leq 100} [T^{i,p,k} - \sum_{1\leq j\leq 100} T^{j,p,k}]^2 \right\}^{\frac{1}{2}}.$$

Ensuite, pour chaque algorithme, nous calculons la moyenne des écarts-types sur les 100 Garnets pour chaque ensemble de paramètre : $std^k = \frac{1}{100} \sum_{1\leq p\leq 100} std^{p,k}$. Pour chaque algorithme, nous traçons $(H_E^k, std^k)_{\{1\leq k\leq 15\}}$. Les résultats sont reportés sur la Fig. 1. Nous y observons que l'algorithme de pure classification a une meilleure performance que les algorithmes d'ARI lorsque le nombre de données expertes augmentent. Cela peut s'expliquer par la forme particulière de la fonction de récompense qui a une

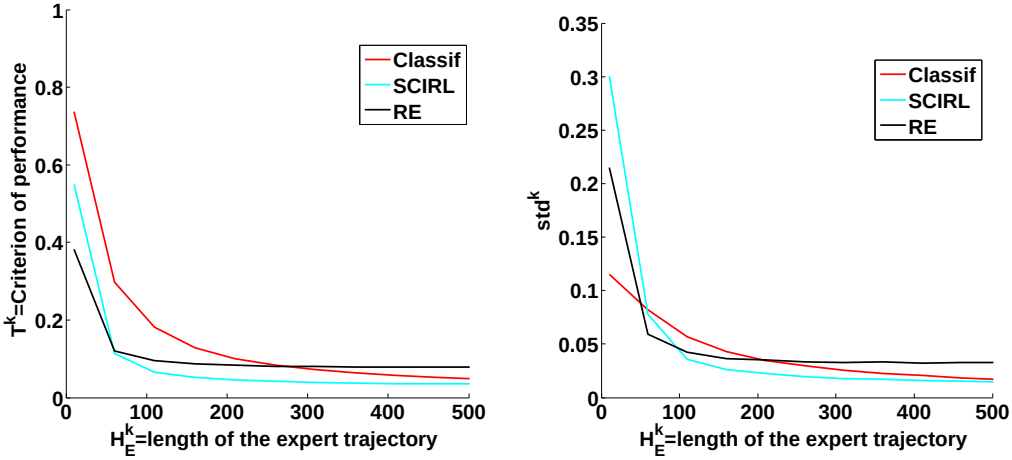


(a) Performance (b) Écart-type moyen

FIGURE 1 – Expérience sur les Garnets : récompense normalement distribuée.

structure qui favorise particulièrement l'algorithme de classification. En effet, comme il y a pour chaque couple état-action une récompense qui est choisie aléatoirement via une gaussienne centrée-réduite, faire une erreur de classification n'est pas tant important car il y aura par la suite d'autres récompenses similaires dans les états suivants. En ce qui concerne les algorithmes d'ARI, comme il y a une récompense très informative, un nombre de données important est nécessaire pour pouvoir estimer cette récompense. Une autre interprétation complémentaire de ces résultats est : comme la fonction de récompense est très informative, le choix de l'action n'a pas un impact très important sur les états futurs et l'impact sur l'horizon d'optimisation est donc fortement réduit. Plus l'horizon d'optimisation est réduit et plus la classification aura un avantage sur les méthodes d'ARI.

La seconde expérience est exactement la même que la première, excepté que nous changeons la forme de la fonction de récompense. En effet, cette dernière ne sera plus normalement distribuée mais parcimonieuse. Pour chaque Garnet, nous allons générer une récompense avec un petit support : $\mathrm{Supp}(\mathcal{R}) \leq \frac{N_S N_A}{50}$ en choisissant de manière aléatoire entre 1 and $\frac{N_S N_A}{50}$ couples (s, a) (tirage sans remise) telle que $\mathcal{R}(s, a) \neq 0$ (récompense choisie uniformément entre 0 and 1). Pour les autres couples (s, a), $\mathcal{R}(s, a) = 0$. Les résultats de cette expérience sont reportés sur la Fig. 2. Ces courbes montrent que les algorithmes d'ARI



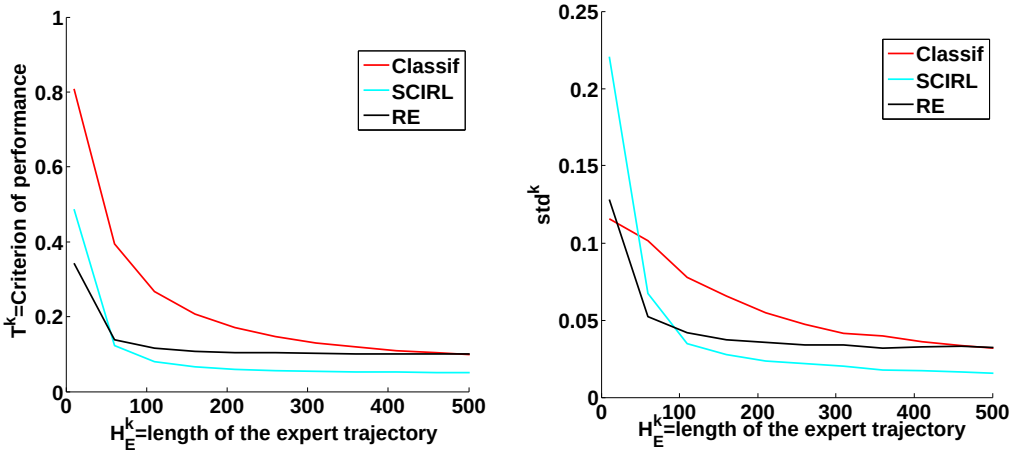
(a) Performance (b) Écart-type moyen

FIGURE 2 – Expérience sur les Garnets : récompense parcimonieuse.

sont plus performants que précédemment et que l'algorithme de classification pure a une performance qui est légèrement détérioré par rapport à l'expérience précédente. Cela peut encore s'expliquer par la

forme particulière de la récompense inconnue. En effet, comme la récompense inconnue est parcimonieuse, faire une erreur de classification dans un état où l'expert choisit une action qui donne une récompense est grave puisque il n'y a que très peu de couples états-actions avec des récompenses. Ainsi, l'algorithme de classification pure peut avoir quelques problèmes lorsque le nombre de données est faible, c'est exactement ce que nous observons sur la Fig. 2(a). De plus, nous remarquons avant tout que les algorithmes d'ARI ont une meilleure performance que précédemment, peut-être parce que la structure de la récompense est plus simple à apprendre. Ici, comme la récompense est peu informative, l'impact de l'horizon d'optimisation peut être plus important que celui de la fonction de récompense précédente ce qui peut aussi détériorer la performance de la classification.

La troisième expérience est encore la même que la première, sauf que nous allons choisir une fonction de récompense qui est uniquement état-dépendante. Pour construire une fonction de cette forme, nous choisissons aléatoirement pour chaque état s une valeur $\mathcal{R}(s)$ tiré selon une distribution centrée-réduite, puis $\forall(s,a) \in \mathcal{S}\times\mathcal{A}, \mathcal{R}(s,a) = \mathcal{R}(s)$. Les résultats sont reportés sur la Fig. 3. Elle montre que les performances



(a) Performance (b) Écart-type moyen

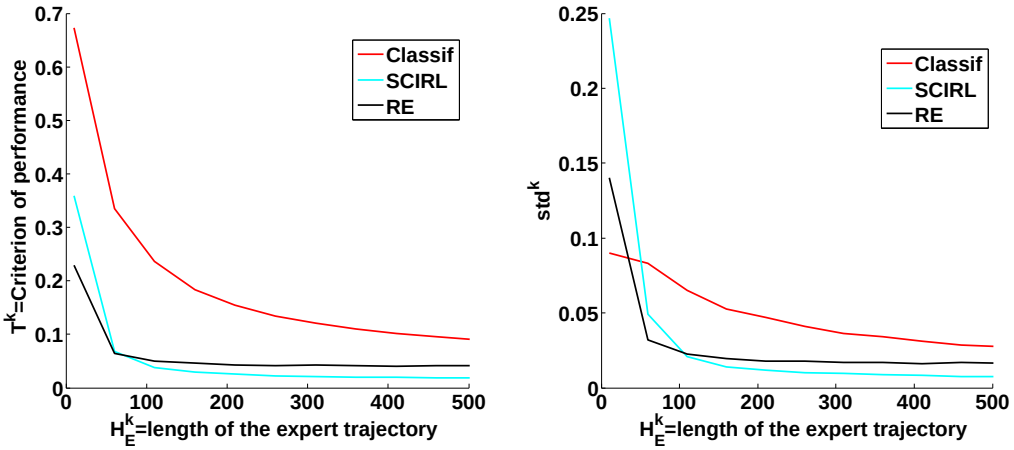
FIGURE 3 – Expérience sur les Garnets : récompense uniquement état-dépendant.

des algorithmes d'ARI sont légèrement meilleures que pour la seconde expérience. On remarque aussi que les algorithmes d'ARI, notamment SCIRL, ont une meilleure performance que la classification pure. Cela peut être expliqué, encore une fois, par la structure de la fonction de récompense qui est encore plus simple que dans les expériences précédentes. La classification pure voit sa performance détériorée par rapport à la seconde expérience. Comme la récompense inconnue dépend uniquement de l'état et non de l'action, il semble assez important de suivre la trajectoire de l'expert pour obtenir une bonne performance. Ainsi, une erreur de classification sur un état donné qui entraînerait l'agent sur une mauvaise trajectoire peut être assez dommageable et amener à une mauvaise performance globale.

Enfin, pour la dernière expérience, nous choisissons une récompense qui est à la fois parcimonieuse et uniquement état-dépendante. Pour cela, nous choisissons aléatoirement et de manière uniforme un nombre d'état s entre 1 et $\frac{N_S}{10}$ via un tirage sans remise. Pour chacun de ces état s, nous choisissons une valeur $\mathcal{R}(s)$ entre 0 et 1 de manière uniforme. Pour les autres états s, nous posons $\mathcal{R}(s) = 0$. Ensuite, nous posons : $\forall(s,a) \in \mathcal{S}\times\mathcal{A}, \mathcal{R}(s,a) = \mathcal{R}(s)$. Les résultats sont reportés sur la Fig 4. Les conclusions que l'ont peut tirer de cette dernière expérience est une combinaison des conclusions tirées des expériences Fig. 2 et Fig. 3. Ici, il y a un clair avantage des algorithmes d'ARI sur la classification pure. L'horizon d'optimisation doit être assez grand, ce qui défavorise la classification.

5 Perturbations de la dynamique

Dans cette section, nous allons un peu plus loin que le cadre théorique étudié jusqu'alors et nous cherchons à savoir s'il est intéressant d'estimer une récompense afin d'être plus stable face aux changements de dynamique du PDM. Comme la récompense est souvent vue comme la représentation la plus succincte



(a) Performance (b) Écart-type moyen

FIGURE 4 – Expérience sur les Garnets : récompense uniquement état-dépendant et parcimonieuse.

de la tâche à accomplir, nous pouvons souhaiter que l'optimisation de la récompense estimée par les algorithmes d'ARI donne une politique quasi-optimale, même s'il y a une perturbation de la dynamique. Les perturbations de la dynamique que nous considérons ici sont celles qui laissent invariante la structure du PDM. La structure du PDM est, pour un couple état-action donné (s, a), les différents états atteignables en étant dans l'état s puis en effectuant l'action a, c'est-à-dire $\mathrm{Supp}(P_{s,a})$. La structure du PDM est donc l'ensemble $(\mathrm{Supp}(P_{s,a}))_{\{(s,a)\in\mathcal{S}\times\mathcal{A}\}}$. Ainsi une perturbation de la dynamique est le choix d'une dynamique $\tilde{\mathcal{P}}$ différente de $\mathcal{P}$ telle que : $(\mathrm{Supp}(P_{s,a}))_{\{(s,a)\in\mathcal{S}\times\mathcal{A}\}} = (\mathrm{Supp}(\tilde{P}_{s,a}))_{\{(s,a)\in\mathcal{S}\times\mathcal{A}\}}$.

La première expérience consiste à générer 100 Garnets du type type $Garnet(N_S, N_A, N_B)$, où N_S est choisi uniformément entre 50 et 100, N_A uniformément entre 5 et 10 et N_B uniformément entre 2 et 5. Cela nous donne l'ensemble de Garnets $\mathfrak{G} = (G_p)_{\{1\leq p\leq 100\}}$. La récompense inconnue qui sert à générer les données expertes D_E est ici la récompense normalement distribuée, comme pour la première expérience de la Sec. 4.3. Ensuite pour chaque G_p, nous réalisons 50 perturbations de la dynamique et nous obtenons un ensemble de Garnets $\tilde{\mathfrak{G}} = (G_{p,q})_{\{1\leq p\leq 100, 1\leq q\leq 50\}}$. Sur chaque problème p, q de l'ensemble $\tilde{\mathfrak{G}}$, nous calculons $\pi_E^{p,q}$ et $\pi_R^{p,q}$ et pour chaque problème p du set $\mathfrak{G}$, nous calculons π_E^p et π_R^p. Le paramètre H_E prend ses valeurs dans l'ensemble $(H_E^k)_{\{1\leq k\leq 15\}} = (50, 100, 150, .., 500)$, $K_E = 1$, $H_P = 10$, $K_P = 50$. Puis, pour chaque ensemble de paramètre (K_E, H_E^k, K_P, H_P) et pour chaque G_p, nous calculons 100 ensembles de données expertes $(D_E^{i,p,k})_{\{1\leq i\leq 100\}}$ et 100 ensembles de données aléatoires $(D_P^{i,p,k})_{\{1\leq i\leq 100\}}$. Le critère de performance pour chaque couple $(D_E^{i,p,k}, D_P^{i,p,k})$ sur chaque $G_{p,q}$ est le suivant :

$$T^{i,p,q,k} = \frac{\mathbb{E}_\rho[v_{\mathcal{R}}^{\pi_E^{p,q}} - v_{\mathcal{R}}^{\pi_C^{i,p,k}}]}{\mathbb{E}_\rho[v_{\mathcal{R}}^{\pi_E^{p,q}}]},$$

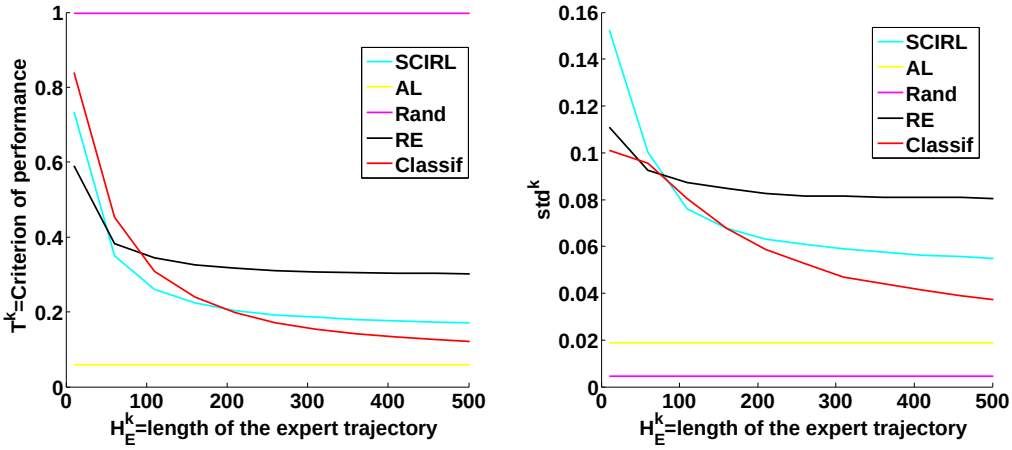
où $\pi_E^{p,q}$ est la politique experte sur $G_{p,q}$, $\pi_C^{i,p,k}$ est la politique induite par l'algorithme choisi avec pour entrées $(D_E^{i,p,k}, D_P^{i,p,k})$ et ρ est la distribution uniforme. Pour la classification pure, nous avons $\pi_C^{i,p,k}(s) \in \mathrm{argmax}_{a\in A}\, \hat{Q}^*(s, a)$ où $\hat{Q}^*$ est le minimiseur obtenu par la descente de sous-gradient sur le critère $\mathfrak{L}_0$. Pour les algorithmes SCIRL et RE, $\pi_C^{i,p,k}$ est la politique obtenue en optimisant la fonction de récompense $\mathcal{R}_C$, obtenue par l'algorithme d'ARI, via l'algorithme d'itération de la politique. De plus, quand $\pi_C^{i,p,k} = \pi_E^p$ alors $T^{i,p,q,k}$ représente la meilleure performance possible qu'il est possible d'atteindre par un algorithme d'AI : cette courbe sera notée AL (pour Apprenticeship Learning) dans nos figures. Finalement, $\pi_C^{i,p,k} = \pi_R^p$, alors $T^{i,p,q,k}$ représente la performance de la politique aléatoire et cette courbe sera notée Rand (pour Random) dans nos figures. Notre critère moyen de performance pour chaque ensemble de paramètres (K_E, H_E^k, K_P, H_P) est :

$$T^k = \frac{1}{500000} \sum_{1\leq p\leq 100, 1\leq q\leq 50, 1\leq i\leq 100} T^{i,p,q,k}.$$

Pour chaque algorithme, nous traçons $(H_E^k, T^k)_{\{1\leq k\leq 15\}}$. Un autre critère intéressant est celui de la variance. Pour chaque Garnet G_p et chaque ensemble de paramètre, nous allons calculer l'écart-type $std^{p,k}$ pour chaque algorithme :

$$std^{p,k} = \left\{ \frac{1}{5000} \sum_{1\leq i\leq 100}^{1\leq q\leq 50} [T^{i,p,q,k} - \sum_{1\leq j\leq 100}^{1\leq q'\leq 50} T^{j,p,q',k}]^2 \right\}^{\frac{1}{2}}.$$

Ensuite, nous calculons l'écart-type moyen pour les 100 Garnets, pour chaque ensemble de paramètres : $std^k = \frac{1}{100}\sum_{1\leq p\leq 100} std^{p,k}$. Pour chaque algorithme, nous traçons $(H_E^k, std^k)_{\{1\leq k\leq 15\}}$. Les résultats sont reportés sur les Fig. 5 à 8. Sur la Fig. 5, la récompense est normalement distribuée et une perturbation



(a) Performance (b) Écart-type moyen

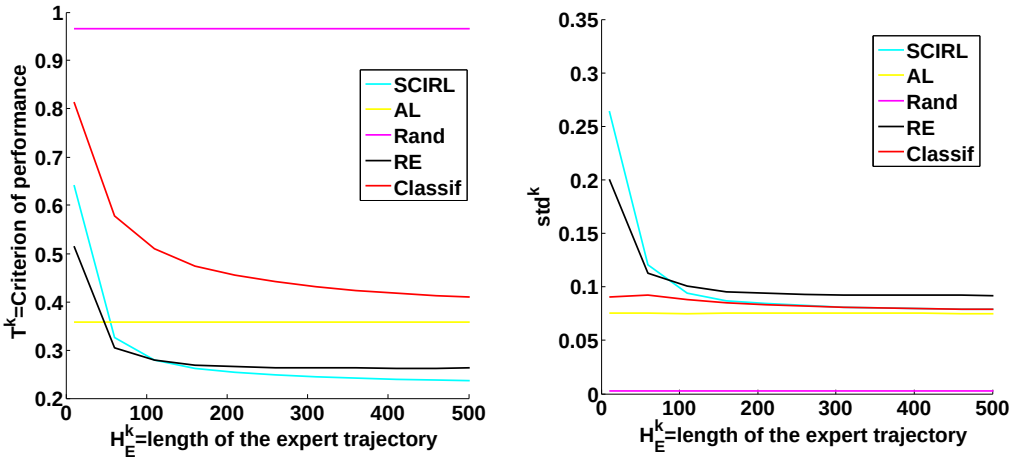
FIGURE 5 – Perturbations de la dynamique : récompense normalement distribuée.

de la dynamique semble ne peut pas détériorer la performance de la politique experte (en jaune sur le graphique). En effet, comme la récompense est très informative, l'impact de l'horizon d'optimisation doit être assez faible et la perturbation de la dynamique ne changera pas trop la politique optimale. On peut observer cela sur la Fig. 5(a), où la courbe jaune est assez proche de 0. Avec cette structure de récompense, il semble plus judicieux d'utiliser un algorithme de classification pure pour garantir cette propriété de stabilité.

La seconde expérience est exactement la même que la première, excepté que la récompense choisie est parcimonieuse. Les résultats sont reportés sur la Fig. 6. Comme la récompense est parcimonieuse, nous pouvons nous attendre à ce qu'une perturbation de la dynamique apporte une détérioration importante de la performance de la politique experte puisque l'horizon d'optimisation est plus important que dans l'expérience précédente. Ici, nous voyons que les algorithmes d'ARI sont en dessous de la courbe jaune quand le nombre de données expertes augmente, ce qui montre qu'aucun algorithme d'AI ne sera capable d'atteindre un tel niveau de stabilité. En ce sens et pour ce type de récompenses inconnues, nous pouvons dire que les méthodes d'ARI sont supérieures à celles d'AI. Ainsi, il semble qu'estimer une fonction de récompense dans ce cas peut être utile car cela garantit un niveau de stabilité qu'aucun algorithme d'AI n'est capable d'apporter.

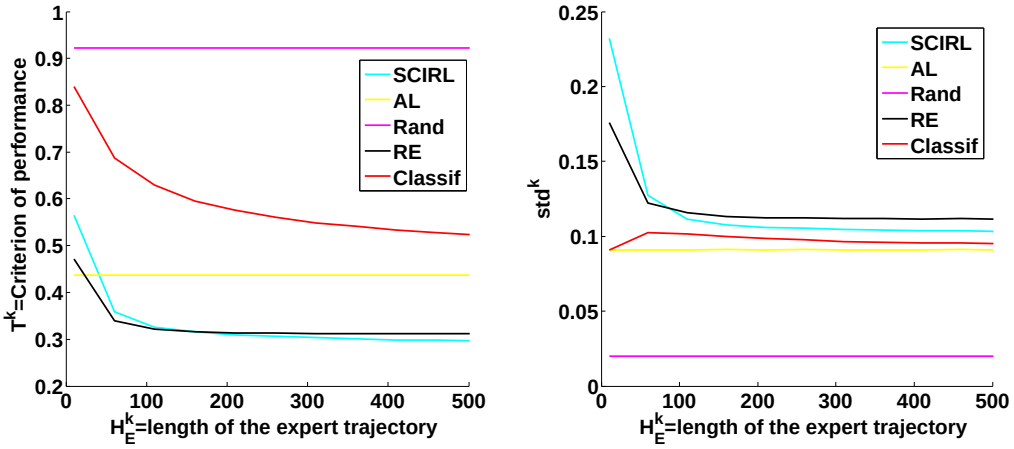
La troisième expérience est encore une fois la même que la précédente, sauf que la récompense est ici uniquement état-dépendante. Les résultats sont reportés sur la Fig. 7. Pour la Fig. 7, la structure de la récompense est assez simple. Il semble que les algorithmes sont encore plus stables et avec moins de données que sur l'expérience précédente. Encore une fois, cela peut s'expliquer par la taille de l'horizon d'optimisation qui devient grand et favorise grandement les méthodes d'ARI comparées aux méthodes d'AI.

Finalement, la dernière expérience consiste à choisir une récompence à la fois parcimonieuse et uniquement état-dépendante. Les résultats sont reportés sur la Fig. 8. Pour cette dernière expérience, les résultats sont une combinaison de ce qui a été dit pour les deux précédentes expériences. La stabilité face aux changements de dynamique est une propriété intéressante des algorithmes d'ARI et une piste de réflexion pour réaliser du transfert de tâches.



(a) Performance (b) Écart-type moyen

FIGURE 6 – Perturbations de la dynamique : récompense parcimonieuse.

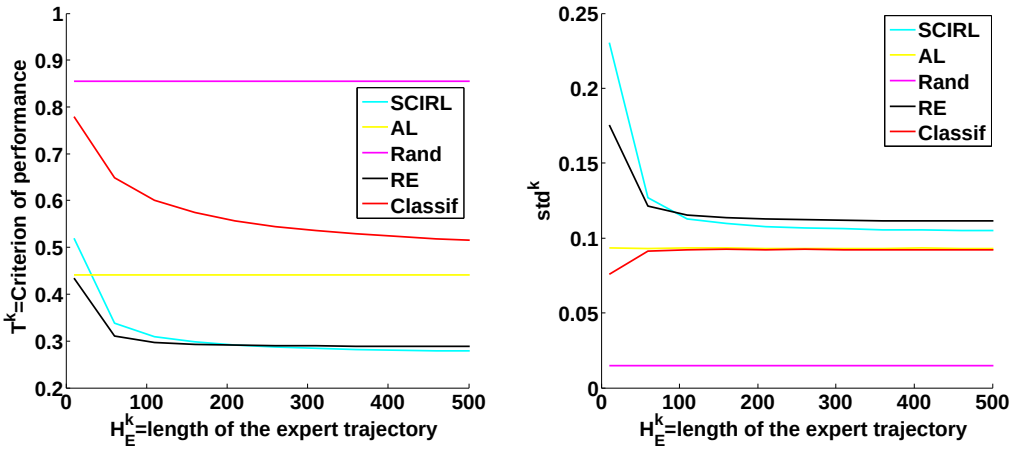


(a) Performance (b) Écart-type moyen

FIGURE 7 – Perturbations de la dynamique : récompense uniquement état-dépendante.

6 Conclusion et perspectives

Dans cet article, nous avons essayé de donner des éléments théoriques et empiriques qui nous aident à mieux appréhender la question : y a-t-il un intérêt à estimer une fonction de récompense ? Dans un premier temps, nous avons montré une borne supérieure pour la différence de fonctions de valeurs entre la politique experte et celle de l'apprenti pour un algorithme de classification pure (vue comme une méthode d'AI) dans le cas à horizon infini. Ce résultat a été comparé à la performance théorique de SCIRL et informellement comparé à une borne similaire dans le cas à horizon fini. Nous avons conclu qu'en théorie, il n'y a pas de raisons spécifiques d'utiliser un algorithme d'ARI, qui nécessite de plus par la suite une optimisation de la récompense trouvée pour obtenir une politique. Cependant, en pratique, les expériences réalisées dans cet article sur une tâche assez générale (les Garnets) montrent que certaines récompenses inconnues avec une structure particulière favorisent considérablement les méthodes d'ARI face aux méthodes d'AI et vice-versa. Plus exactement, il semble que les structures de récompenses peu informatives favorisent les algorithmes d'ARI. La raison que nous mettons en avant pour expliquer ce phénomène est la suivante : moins la récompense est informative et plus important est l'horizon d'optimisation. C'est donc un désavantage clair pours les algorithmes de classification, qui ne prennent pas en compte cet horizon d'optimisation. Cependant, nous n'apportons aucune justification théorique en ce qui concerne la meilleure performance



(a) Performance (b) Écart-type moyen

FIGURE 8 – Perturbations de la dynamique : récompense uniquement état dépendante et parcimonieuse.

des algorithmes d'ARI pour des récompenses particulières. Cela pourrait d'ailleurs être une perspective intéressante pour donner plus de crédit à nos expériences. Enfin, nous avons voulu aussi montrer la capacité de stabilité des algorithmes d'ARI pour certaines formes de récompenses face aux perturbations de dynamique. D'autre part, il semble intéressant de pouvoir créer un algorithme qui prendrait en compte des données de différentes dynamiques perturbées du même PDM pour apprendre une récompense qui serait encore moins sensible aux changements de dynamique. Cela pourrait être intéressant avec des applications où l'humain serait impliqué : chaque personne représenterait alors une version perturbée d'un même PDM.

Références

ABBEEL P. & NG A. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*.

ARCHIBALD T., MCKINNON K. & THOMAS L. (1995). On the generation of markov decision processes. *Journal of the Operational Research Society*.

ATKESON C. & SCHAAL S. (1997). Robot learning from demonstration. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*.

BOULARIAS A., KOBER J. & PETERS J. (2011). Relative entropy inverse reinforcement learning. In *JMLR Workshop and Conference Proceedings Volume 15 : AISTATS 2011*.

KLEIN E., GEIST M., PIOT B. & PIETQUIN O. (2012). Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems 25 (NIPS)*.

LANGFORD J. & ZADROZNY B. (2005). Relating reinforcement learning performance to classification performance. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*.

POMERLEAU D. (1989). *Alvinn : An autonomous land vehicle in a neural network*. Rapport interne, DTIC Document.

RUSSELL S. (1998). Learning agents for uncertain environments. In *Proceedings of the 11th annual conference on Computational Learning Theory (COLT)*.

SHOR N., KIWIEL K. & RUSZCAYNSKI A. (1985). *Minimization methods for non-differentiable functions*. Springer-Verlag.

SYED U. & SCHAPIRE R. (2008). A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems 21 (NIPS)*.

SYED U. & SCHAPIRE R. (2010). A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems 23 (NIPS)*.

TASKAR B., CHATALBASHEV V., KOLLER D. & GUESTRIN C. (2005). Learning structured prediction models : A large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*.

A Preuve du Théorème 1

Soit π_C la politique du classifieur (entraîné sur l'ensemble D_E pour imiter la politique de l'expert π_E). Soit aussi ϵ_C l'erreur de classification C_ν le coefficient de concentration défini plus haut. Alors pour tout $\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$:

$$\mathbf{E}_\nu[v_\mathcal{R}^{\pi_E} - v_\mathcal{R}^{\pi_C}] \leq \frac{2C_\nu\|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C.$$

Pour cette preuve, nous introduisons quelques notations. $P \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times\mathcal{S}}$ est une matrice rectangulaire telle que :

$$\forall(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}, P((s,a),s') = \mathcal{P}(s'|(s,a)).$$

$\Pi_E \in \mathbb{R}^{\mathcal{S}\times(\mathcal{S}\times\mathcal{A})}$ est une matrice rectangulaire telle que :

$$\forall(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}, \Pi_E(s',(s,a)) = \chi_{\{\tilde{s}\in\mathcal{S},\tilde{s}=s\}}(s')\pi_E(a|s).$$

$M \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})^2}$ est une matrice diagonale telle que :

$$\forall(s,a) \in \mathcal{S}\times\mathcal{A}, M((s,a),(s,a)) = \chi_{\{b\in\mathcal{A},\pi_C(s)\neq b\}}(a).$$

$\tilde{M} \in \mathbb{R}^{\mathcal{S}^2}$ est une matrice diagonale telle que :

$$\forall(s,a) \in \mathcal{S}\times\mathcal{A}, \tilde{M}(s) = \sum_{a\in\mathcal{A}} \pi_E(a|s)\chi_{\{b\in\mathcal{A},\pi_C(s)\neq b\}}(a).$$

M et M' sont deux matrices qui encodent la différence entre les politiques π_C et π_E. On a :

$$\begin{aligned} v_\mathcal{R}^{\pi_E} - v_\mathcal{R}^{\pi_C} &\overset{(a)}{=} T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_E} - T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_C} + T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_C} - v_\mathcal{R}^{\pi_C}, \\ &\overset{(b)}{=} \gamma P_{\pi_E}(v_\mathcal{R}^{\pi_E} - v_\mathcal{R}^{\pi_C}) + T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_C} - v_\mathcal{R}^{\pi_C}, \\ &\overset{(c)}{=} (I - \gamma P_{\pi_E})^{-1}(T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_C} - v_\mathcal{R}^{\pi_C}), \end{aligned}$$

L'égalité (a) est vrai car $T_\mathcal{R}^{\pi_E}v^{\pi_E} = v_\mathcal{R}^{\pi_E}$, l'égalité (b) est obtenue par définition de $T_\mathcal{R}^{\pi_E}$ et l'égalité (c) est vrai par inversibilité de $I - \gamma P_{\pi_E}$ où $I \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ est la matrice identité. La prochaine étape est de travailler sur le terme $T_\mathcal{R}^{\pi_E}v_\mathcal{R}^{\pi_C} - v_\mathcal{R}^{\pi_C}$. Pour chaque fonction $v \in \mathbb{R}^\mathcal{S}$, par définition de $T_\mathcal{R}^{\pi_E}$:

$$T_\mathcal{R}^{\pi_E}v = \mathcal{R}_{\pi_E} + \gamma P_{\pi_E}v.$$

De plus, il est clair que :

$$\mathcal{R}_{\pi_E}(s) = \sum_{a\in A} \pi_E(a|s)\chi_{\{b\in A,\pi_C(s)\neq b\}}(a)\mathcal{R}(s,a) + (1 - \sum_{a\in\mathcal{A}} \chi_{\{b\in\mathcal{A},\pi_C(s)\neq b\}}(a)\pi_E(a|s))\mathcal{R}(s,\pi_C(s)).$$

Avec des notations vectorielles, cela peut s'écrire :

$$\mathcal{R}_{\pi_E} = \Pi_E M\mathcal{R} + (I - \tilde{M})\mathcal{R}_{\pi_C}.$$

Pour chaque fonction $v \in \mathbb{R}^\mathcal{S}$, et par définition de P_{π_E} :

$$\begin{aligned} P_{\pi_E}v(s) = &\sum_{a\in\mathcal{A}}\sum_{s'\in\mathcal{S}} \chi_{\{b\in\mathcal{A},\pi_C(s)\neq b\}}(a)\pi_E(a|s)\mathcal{P}(s'|s,a)v(s') \\ &+ (1 - \sum_{a\in\mathcal{A}} \chi_{\{b\in\mathcal{A},\pi_C(s)\neq b\}}(a)\pi_E(a|s)) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,\pi_C(s))v(s'). \end{aligned}$$

Avec des notations vectorielles, cela peut s'écrire :

$$P_{\pi_E}v = \Pi_E MPv + (I - \tilde{M})P_{\pi_C}v.$$

Donc :

$$\begin{aligned} T_{\mathcal{R}}^{\pi_E} v &= \Pi_E M(\mathcal{R} + \gamma P v) + (I - \tilde{M})(\mathcal{R}_{\pi_C} + \gamma P_{\pi_C} v), \\ &\overset{(d)}{=} T_{\mathcal{R}}^{\pi_C} v + \Pi_E M(\gamma P v + \mathcal{R}) - \gamma \tilde{M}(\gamma P_{\pi_C} v + \mathcal{R}_{\pi_C}), \end{aligned}$$

où l'égalité (d) tient car $T_{\mathcal{R}}^{\pi_C} v = \mathcal{R}_{\pi_C} + \gamma P_{\pi_C} v$ pour chaque fonction $v \in \mathbb{R}^{\mathcal{S}}$. Donc, pour $v = v_{\mathcal{R}}^{\pi_C}$ et en utilisant le fait que $T_{\mathcal{R}}^{\pi_C} v_{\mathcal{R}}^{\pi_C} = v_{\mathcal{R}}^{\pi_C}$:

$$T_{\mathcal{R}}^{\pi_E} v_{\mathcal{R}}^{\pi_C} - v_{\mathcal{R}}^{\pi_C} = \Pi_E M(\gamma P v_{\mathcal{R}}^{\pi_C} + \mathcal{R}) - \tilde{M}(\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C} + \mathcal{R}_{\pi_C}).$$

Ainsi, nous avons :

$$\nu^T (v_{\mathcal{R}}^{\pi_E} - v_{\mathcal{R}}^{\pi_C}) = \nu^T (I - \gamma P_{\pi_E})^{-1} [\Pi_E M(\gamma P v_{\mathcal{R}}^{\pi_C} + \mathcal{R}) - \tilde{M}(\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C} + \mathcal{R}_{\pi_C})]$$

Par l'inégalité d' Hölder :

$$|\nu^T (I - \gamma P_{\pi_E})^{-1} \Pi_E M(\gamma P v_{\mathcal{R}}^{\pi_C} + \mathcal{R})| \leq \|\nu^T (I - \gamma P_{\pi_E})^{-1} \Pi_E M\|_1 \|\gamma P v_{\mathcal{R}}^{\pi_C} + \mathcal{R}\|_\infty,$$

et aussi :

$$|\nu^T (I - \gamma P_{\pi_E})^{-1} \tilde{M}(\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C} + \mathcal{R}_{\pi_C})| \leq \|\nu^T (I - \gamma P_{\pi_E})^{-1} \tilde{M}\|_1 \|\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C} + \mathcal{R}_{\pi_C}\|_\infty,$$

De plus, comme $(I - \gamma P_{\pi_E})^{-1} = \sum_{t \geq 0} \gamma^t P_{\pi_E}^t$ et par définition de C_ν :

$$\begin{aligned} \|\nu^T (I - \gamma P_{\pi_E})^{-1} \Pi_E M\|_1 &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t \geq 0} \gamma^t \frac{(\nu^T P_{\pi_E}^t)(s)}{\nu(s)} \nu(s) \pi_E(a|s) M((s,a),(s,a)), \\ &\overset{(e)}{\leq} \frac{C_\nu}{1-\gamma} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mu_E(s,a) \chi_{\{b \in \mathcal{A}, \pi_C(s) \neq b\}}, \\ &\overset{(f)}{=} \frac{C_\nu}{1-\gamma} \mathbb{E}_{\mu_E}[\chi_{\{(s,a) \in \mathcal{S} \times \mathcal{A}, \pi_C(s) \neq a\}}], \\ &\overset{(g)}{=} \frac{C_\nu}{1-\gamma} \epsilon_C, \end{aligned}$$

où l'inégalité (e) est vraie car $\sum_{t \geq 0} \gamma^t \frac{(\nu^T P_{\pi_E}^t)(s)}{\nu(s)} \leq \frac{C_\nu}{1-\gamma}$ par définition de C_ν, $\nu(s)\pi_E(a|s) = \mu_E(s,a)$ par définition de μ_E et $M((s,a),(s,a)) = \chi_{\{b \in \mathcal{A}, \pi_C(s) \neq b\}}$ par définition de M. L'égalité (f) tient en écrivant la somme comme une espérance et l'égalité (g) tient par définition d' ϵ_C. Et par les même arguments :

$$\begin{aligned} \|\nu^T (I - \gamma P_{\pi_E})^{-1} \tilde{M}\|_1 &= \sum_{s \in \mathcal{S}} \sum_{t \geq 0} \gamma^t \frac{(\nu^T P_{\pi_E}^t)(s)}{\nu(s)} \nu(s) \tilde{M}(s,s), \\ &\leq \frac{C_\nu}{1-\gamma} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \nu(s) \pi_E(a|s) \chi_{\{b \in \mathcal{A}, \pi_C(s) \neq b\}}, \\ &= \frac{C_\nu}{1-\gamma} \mathbb{E}_{\mu_E}[\chi_{\{(s,a) \in \mathcal{S} \times \mathcal{A}, \pi_C(s) \neq a\}}], \\ &= \frac{C_\nu}{1-\gamma} \epsilon_C. \end{aligned}$$

Finalement, il est clair que :

$$\|\gamma P v_{\mathcal{R}}^{\pi_C} + \mathcal{R}\|_\infty \leq \frac{\gamma \|\mathcal{R}\|_\infty}{1-\gamma} + \|\mathcal{R}\|_\infty = \frac{\|\mathcal{R}\|_\infty}{1-\gamma},$$

et que :

$$\|\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C} + \mathcal{R}_{\pi_C}\|_\infty \leq \frac{\gamma \|\mathcal{R}\|_\infty}{1-\gamma} + \|\mathcal{R}\|_\infty = \frac{\|\mathcal{R}\|_\infty}{1-\gamma},$$

Ainsi, nous avons :

$$|\nu^T(I-\gamma P_{\pi_E})^{-1}\Pi_E M(\gamma P v_{\mathcal{R}}^{\pi_C}+\mathcal{R})| \leq \frac{C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C,$$

et :

$$|\nu^T(I-\gamma P_{\pi_E})^{-1}\tilde{M}(\gamma P_{\pi_C} v_{\mathcal{R}}^{\pi_C}+\mathcal{R}_{\pi_C})| \leq \frac{C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C.$$

Donc, si nous regroupons les différents résultats obtenus :

$$\mathbb{E}_\nu[v_{\mathcal{R}}^{\pi_E}-v_{\mathcal{R}}^{\pi_C}] \leq \frac{2C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C.$$

De plus, il est intéressant de remarquer que pour des récompenses uniquement état-dépendante $\mathcal{R} : \forall(s,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{A}, \mathcal{R}(s,a)=\mathcal{R}(s,b)$, la borne décrite par l'équation Eq. (1) peut être améliorée. En effet, pour une récompense uniquement état-dépendante, comme $\mathcal{R}_{\pi_E}=\mathcal{R}_{\pi_C}$ alors le terme $\Pi_E M\mathcal{R}-\tilde{M}\mathcal{R}_{\pi_C}$ est nulle et la borne devient :

$$\mathbb{E}_\nu[v_{\mathcal{R}}^{\pi_E}-v_{\mathcal{R}}^{\pi_C}] \leq \frac{2\gamma C_\nu \|\mathcal{R}\|_\infty}{(1-\gamma)^2}\epsilon_C.$$

Il y a donc une amélioration d'un facteur γ dans la borne entre les récompenses uniquement état-dépendantes et les autres récompenses.