Least-Squares λ Policy Iteration : optimisme et compromis biais-variance pour le contrôle optimal

Christophe Thiery, Bruno Scherrer

Least-Squares λ Policy Iteration : optimisme et compromis biais-variance pour le contrôle optimal

Christophe Thiery, Bruno Scherrer

LORIA - INRIA Lorraine
Campus Scientifique BP 239
54506 Vandœuvre-lès-Nancy CEDEX
{thierych,scherrer}@loria.fr

Résumé : Dans le contexte des Processus de Décision Markoviens (PDM) à grands espaces d'états avec approximation linéaire de la fonction de valeur, nous proposons un nouvel algorithme, Least-Squares λ Policy Iteration (LSλPI), qui généralise et hérite des propriétés intéressantes de deux algorithmes existants : λ-Policy Iteration (λPI) (Bertsekas & Ioffe, 1996) et Least-Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003). Si le paramètre λ de λPI permet, comme dans la plupart des algorithmes du domaine, de faire un compromis biais-variance dans l'évaluation d'une politique, il introduit également de l'optimisme dans un schéma de type itération sur les politiques. A la manière de LSPI, l'algorithme que nous proposons ne nécessite pas de générer de nouveaux échantillons à chaque changement de politique (il est off-policy), les utilise de manière efficace (c'est une méthode du second ordre) et n'a pas besoin de disposer d'un modèle du PDM. Nous établissons un résultat analytique très général qui montre qu'il est raisonnable d'introduire de l'optimisme dans un schéma PI, dans le sens où il garantit la performance de la politique lorsque l'erreur d'approximation est contrôlée à chaque itération. Ce résultat s'applique en particulier à LSλPI. Finalement, nous vérifions empiriquement sur un problème simple de type chaîne d'états et sur le jeu de Tetris l'intérêt de ce nouvel algorithme, en montrant que le paramètre λ permet d'améliorer la convergence et la performance de la politique obtenues par LSPI.

Introduction

Nous considérons la question de résoudre des Processus Décisionnels de Markov (PDM) de manière approchée, dans le cas où la fonction de valeur est estimée par une architecture linéaire et à l'aide d'échantillons, typiquement lorsque l'espace d'états est trop grand pour une résolution exacte.

TD(λ) avec approximation linéaire (Sutton & Barto, 1998) est un algorithme fondateur de la communauté de l'apprentissage par renforcement. Il estime la fonction de valeur en utilisant les différences temporelles d'une trajectoire échantillonnée, et un paramètre $\lambda \in [0, 1]$ contrôle la profondeur des mises à jour de la fonction de valeur. Avec les grandes valeurs de λ, l'estimation dépend fortement des trajectoires observées et a donc une variance plus importante. Lorsque λ est petit, on accorde au contraire plus de crédit à l'approximation courante de la fonction de valeur et moins aux échantillons observés, ce qui introduit un biais qui ralentit la convergence. Ce **compromis biais-variance** de TD(λ) a été étudié analytiquement dans Kearns & Singh (2000).

L'algorithme TD(λ) est une approximation du premier ordre et présente par conséquent deux inconvénients. Il ne tire pas le meilleur usage des informations données par les échantillons, et requiert l'usage d'un paramètre facteur d'apprentissage qui s'avère souvent difficile à régler. Les méthodes dites du **second ordre** ou aux moindres carrés telles que LSTD(0) (Bradtke & Barto, 1996), LSTD(λ) (Boyan, 2002) et LSPE(λ) (Nedić & Bertsekas, 2003; Yu & Bertsekas, 2009) ont pour but de pallier ces inconvénients. Elles construisent explicitement un système linéaire qui caractérise la solution vers laquelle un algorithme du premier ordre convergerait, mais elles exploitent les informations des échantillons de manière plus efficace et ne nécessitent en général pas de facteur d'apprentissage. Ainsi, le nombre d'itérations nécessaires pour converger est plus faible en pratique, même si chaque itération a une complexité de $O(p^2)$ au lieu de $O(p)$, où p est la dimension de l'architecture linéaire. De plus, il a été argumenté analytiquement et empiriquement (voir par exemple Boyan (2002); Schoknecht (2002); Yu & Bertsekas (2009)) que les méthodes du second ordre sont plus stables et peuvent donner de bien meilleures performances que TD(λ).

Algorithme	compromis biais-variance	évaluation optimiste	échantillonnage efficace	off-policy
TD(λ) (Sutton & Barto, 1998)	$\times$			
LSTD(0) (Bradtke & Barto, 1996)			$\times$	
LSTD(λ) (Boyan, 2002)	$\times$		$\times$	
LSPE(λ) (Yu & Bertsekas, 2009)	$\times$		$\times$	
λPI (Bertsekas & Ioffe, 1996)	$\times$	$\times$	$\times$	
LSPI (Lagoudakis & Parr, 2003)			$\times$	$\times$
LSλPI	$\times$	$\times$	$\times$	$\times$

FIGURE 1 – Principales caractéristiques des travaux de l'état de l'art : avec ou sans paramètre de compromis λ, avec évaluation optimiste de la fonction de valeur ou non, dans le contexte des moindres carrés ou non, et avec évaluation off-policy ou non. LSλPI peut être vu comme une généralisation optimiste de LSPI avec un paramètre λ, comme une approximation du second ordre et off-policy de λPI, ou encore comme une variante avec contrôle optimiste et off-policy de LSPE(λ).

Alors que l'algorithme LSTD(λ) (Boyan, 2002) est une version du second ordre de TD(λ), LSPE(λ) (Yu & Bertsekas, 2009) s'inspire d'un algorithme moins connu, λ-Policy Iteration (λPI), proposé par Bertsekas & Ioffe (1996). Si λ permet ici aussi de faire un compromis biais-variance, il joue également un autre rôle. λPI est un algorithme qui généralise les deux algorithmes classiques de la programmation dynamique Value Iteration et Policy Iteration pour calculer une politique optimale : λ correspond ici à la taille du pas effectué en direction de la fonction de valeur de la politique que l'on évalue ($\lambda = 0$ correspond à Value Iteration, tandis que $\lambda = 1$ correspond à Policy Iteration). Ainsi, pour λPI, une valeur de $\lambda < 1$ permet d'introduire une forme d'**optimisme** [1], dans le sens où on ne cherche plus à calculer entièrement la valeur de la politique courante, mais seulement à s'en approcher avant de changer de politique.

Les approches du second ordre que nous venons de mentionner, lorsqu'elles utilisent un paramètre λ pour faire un compromis biais-variance, estiment la fonction de valeur de façon on-policy, c'est-à-dire en utilisant des échantillons générés par la politique à évaluer. Nous nous intéressons dans cet article à l'évaluation **off-policy**, c'est-à-dire l'évaluation d'une politique à partir d'échantillons quelconques. Une telle évaluation est intéressante car elle peut s'inscrire dans un cadre d'itération sur les politiques sans avoir à regénérer des échantillons à chaque itération. C'est une idée qu'on retrouve dans LSPI (Lagoudakis & Parr, 2003), la différence essentielle avec l'algorithme que nous allons présenter ici étant que LSPI n'introduit pas d'optimisme via un paramètre λ.

Après avoir évoqué des caractéristiques qui nous ont paru intéressantes dans les algorithmes de la littérature (compromis biais-variance, optimisme, évaluation du second ordre, off-policy et donc possibilité d'itérer sur les politiques sans regénérer des échantillons), nous pouvons dire de l'algorithme que nous allons présenter, Least-Squares λ Policy Iteration (LSλPI), qu'il est à notre connaissance le premier à toutes les posséder. La figure 1 liste les principaux travaux avec lesquels LSλPI est lié et résume leurs caractéristiques. LSλPI peut être vu comme une généralisation de LSPI avec un paramètre λ qui établit un compromis biais-variance et ajoute de l'optimisme, comme une version du second ordre et off-policy de λPI, ou comme une version avec itération sur les politiques et off-policy de LSPE(λ).

La suite de cet article est organisée de la manière suivante. Dans la section 1, après avoir introduit les notations, nous présentons l'algorithme λPI dans le cas exact. Dans la section 2, nous considérons le cas approximatif : nous détaillons LSλPI et nous décrivons un résultat de convergence général des algorithmes optimistes avec approximation (dont λPI fait partie). Enfin, la section 3 présente des expériences illustrant LSλPI et montrant l'intérêt d'introduire ce nouveau paramètre λ par rapport à LSPI.

1. Une notion d'optimisme analogue (quoiqu'un peu plus extrême) est décrite par Bertsekas & Tsitsiklis (1996), chapitres 5.4 et 6.4. Nous reprenons ce terme dans le sens où l'on change de politique avant d'avoir fini de calculer la valeur de la politique précédente.

1 λ-Policy Iteration dans le cas exact

Nous introduisons ici les notations que nous utiliserons et nous présentons une vue d'ensemble de λPI dans le cas exact. Le lecteur pourra se référer à Bertsekas & Ioffe (1996) pour une description plus détaillée de cet algorithme.

1.1 Notations utilisées

Le cadre des Processus Décisionnels de Markov (PDM) permet de formaliser le contrôle optimal stochastique, qui considère un agent devant prendre des décisions afin de maximiser un signal de récompense sur le long terme. Un Processus Décisionnel de Markov est défini comme un quadruplet $(\mathcal{S}, \mathcal{A}, P, R)$ où :

- $\mathcal{S}$ est l'espace d'états ;
- $\mathcal{A}$ est l'espace d'actions ;
- P est la fonction de transition : $P(s, a, s')$ est la probabilité d'arriver dans l'état s' sachant que l'on est dans l'état s et que l'on effectue l'action a.
- R est la fonction de récompense : $R(s, a, s') \in \mathbb{R}$ est la récompense reçue en effectuant l'action $a \in \mathcal{A}$ depuis l'état $s \in \mathcal{S}$ et en arrivant dans l'état s'. On utilisera la notation simplifiée $\mathcal{R}(s, a)$ pour désigner la récompense moyenne d'un couple état-action : $\mathcal{R}(s, a) = \sum_{s' \in S} P(s, a, s') R(s, a, s')$

La dynamique du système vérifie la propriété de Markov, c'est-à-dire que les probabilités de transitions dépendent de l'état courant et de l'action choisie uniquement, et pas des états précédemment visités. Une *politique* est une fonction $\pi : \mathcal{S} \to \mathcal{A}$ qui associe à chaque état l'action correspondante : $\pi(s) = a$. On utilisera également la notation $\pi(s, a)$, avec $\pi(s, a) = 1$ si $\pi(s) = a$ et 0 sinon [2]. La *fonction de valeur* d'une politique π est la fonction $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ qui associe à chaque couple état-action [3] l'espérance du cumul des récompenses que l'on peut obtenir à partir de cet état, en effectuant cette action et en suivant la politique π ensuite :

$$Q^\pi(s, a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \middle| s_0 = s, a_0 = a, a_t = \pi(s_t) \text{ pour } t \geq 1 \right]$$

où $\gamma \in [0, 1]$ est un *facteur d'actualisation* permettant de diminuer l'importance des récompenses lointaines. Si la propriété de Markov est vérifiée, une propriété fondamentale de la fonction de valeur est le fait qu'elle vérifie une équation récursive, *l'équation de Bellman* (Bellman, 1957) :

$$Q^\pi(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') Q^\pi(s', \pi(s')). \tag{1}$$

Ainsi, la valeur d'un couple état-action dépend de la récompense immédiate et de la valeur des états suivants. Cette équation récursive est le fondement de nombreux algorithmes liés aux PDM. On peut la réécrire de manière condensée en introduisant l'opérateur de Bellman B_π défini pour tout vecteur Q par

$$B_\pi Q = \mathcal{R} + \gamma P_\pi Q$$

où $\mathcal{R}$ est le vecteur des récompenses de chaque couple état-action :

$$\mathcal{R} = \begin{pmatrix} \mathcal{R}(s_1, a_1) \\ \vdots \\ \mathcal{R}(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}) \end{pmatrix}$$

et P_π est la matrice de transition du PDM induite par le choix d'une action donnée suivie de la politique π ensuite : $P_\pi(s, a, s', a') = P(s, a, s')\pi(s', a')$. Cet opérateur B_π est contractant (Puterman, 1994) et son unique point fixe est la fonction de valeur Q^π. Ainsi, Q^π est la seule fonction de valeur qui vérifie l'équation de Bellman $B_\pi Q = Q$.

2. On utilise dans cet article uniquement des politiques déterministes.

3. On considère uniquement des fonctions de valeur définies sur l'espace des couples états-actions car nous cherchons à apprendre une politique sans utiliser de modèle du PDM. Cependant, notre description de λPI s'applique également pour des fonctions de valeur définies sur l'espace d'états seul.

On note Q^* la fonction de valeur optimale, qui associe à chaque couple état-action la meilleure espérance possible des récompenses.

$$Q^*(s,a) = \max_\pi Q^\pi(s,a).$$

Il peut exister plusieurs politiques optimales, qui partagent alors cette fonction de valeur. Si l'on connaît la fonction de valeur optimale, alors on en déduit facilement une politique optimale π^* en sélectionnant la politique *gloutonne* par rapport à Q^* :

$$\pi^*(s) = \arg\max_a Q^*(s,a).$$

Nous utiliserons la notation $\pi = \text{glouton}(Q)$ pour désigner une politique gloutonne par rapport à Q. La fonction de valeur optimale vérifie elle aussi une équation récursive, *l'équation d'optimalité de Bellman* (Bellman, 1957) :

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \max_{a'} \left(\sum_{s' \in \mathcal{S}} P(s,a,s') Q^*(s',a') \right). \tag{2}$$

Là aussi, on peut introduire un opérateur, noté B :

$$[BQ](s,a) = \mathcal{R}(s,a) + \gamma \max_{a'} \left(\sum_{s' \in \mathcal{S}} P(s,a,s') Q(s',a') \right) \tag{3}$$

et l'équation (2) peut se réécrire de manière condensée : $BQ^* = Q^*$. L'opérateur B est contractant (Puterman, 1994) et son unique point fixe est la fonction de valeur optimale Q^*. Q^* est donc la seule fonction de valeur vérifiant $BQ = Q$. Ces deux opérateurs permettent notamment d'exprimer le fait qu'une politique π est gloutonne par rapport à une fonction de valeur Q : on a dans ce cas $BQ = B_\pi Q$. Nous passons maintenant à la présentation des algorithmes qui permettent de calculer la fonction de valeur optimale.

1.2 Value Iteration

L'algorithme Value Iteration (Bellman, 1957), issu de la programmation dynamique, est l'un des algorithmes standards des PDM. Nous le présentons ici (Algorithme 1) sous un angle particulier, qui permettra de mieux mettre en évidence le lien avec λPI.

Algorithme 1 Value Iteration

répéter
  $\pi_{k+1} \leftarrow \text{glouton}(Q_k)$
  $Q_{k+1} \leftarrow B_{\pi_{k+1}} Q_k$
  $k \leftarrow k+1$
jusqu'à $\|Q_{k+1} - Q_k\|_\infty < \epsilon$

$\|\cdot\|_\infty$ désigne la norme infinie sur l'espace des fonctions de valeur, c'est-à-dire $\|Q\|_\infty = \max_{(s,a)} |Q(s,a)|$. À chaque itération, la politique π_{k+1} est choisie comme la politique gloutonne par rapport à Q_k, puis la valeur suivante Q_{k+1} est calculée en appliquant une fois l'opérateur de Bellman sur la valeur courante Q_k. Comme $B_{\pi_{k+1}} Q_k = BQ_k$, chaque itération revient en fait à appliquer l'opérateur d'optimalité de Bellman B présenté plus haut. Comme cet opérateur est contractant et que son unique point fixe est la fonction de valeur optimale Q^*, l'algorithme converge vers la valeur optimale.

1.3 Policy Iteration

Avec l'algorithme Policy Iteration (Bellman, 1957), la politique π_{k+1} est choisie comme la politique gloutonne par rapport à Q_k, puis Q_{k+1} est calculée comme la valeur de la politique π_{k+1} (Algorithme 2). Pour cela, on peut appliquer successivement l'opérateur $B_{\pi_{k+1}}$ jusqu'à atteindre son point fixe qui est la valeur de la politique π_{k+1}, ce qui est équivalent à résoudre l'équation de Bellman (équation (1)) analytiquement. La phase d'évaluation est plus coûteuse en général que celle de Value Iteration puisqu'il faut appliquer un grand nombre de fois l'opérateur de Bellman. En contrepartie, Policy Iteration nécessite en général moins d'itérations pour converger.

Algorithme 2 Policy Iteration

répéter
$\pi_{k+1} \leftarrow \text{glouton}(Q_k)$
$Q_{k+1} \leftarrow B^{\infty}_{\pi_{k+1}} Q_k$
$k \leftarrow k+1$
jusqu'à $\pi_{k+1} = \pi_k$

1.4 Modified Policy Iteration

Un algorithme intermédiaire entre Value Iteration et Policy Iteration consiste à appliquer l'opérateur de Bellman un nombre déterminé de fois m. Ainsi, on ne calcule pas entièrement la valeur de la politique courante π_k (contrairement à Policy Iteration), mais on peut s'en approcher plus rapidement qu'avec Value Iteration. Cette méthode est intitulée Modified Policy Iteration (Puterman, 1994) et est détaillée à l'Algorithme 3. Lorsque $m = 1$, on retrouve Value Iteration, et lorsque $m \to \infty$, on retrouve Policy Iteration.

Algorithme 3 Modified Policy Iteration

répéter
$\pi_{k+1} \leftarrow \text{glouton}(Q_k)$
$Q_{k+1} \leftarrow B^{m}_{\pi_{k+1}} Q_k$
$k \leftarrow k+1$
jusqu'à $\|Q_{k+1} - Q_k\|_\infty < \epsilon$

1.5 λ-Policy Iteration

λ-Policy Iteration (λPI), introduit par Bertsekas & Ioffe (1996), propose une autre manière de généraliser Value Iteration et Policy Iteration. Comme dans les algorithmes précédents, la nouvelle politique π_{k+1} est choisie comme la politique gloutonne sur Q_k, puis on calcule une nouvelle fonction de valeur Q_{k+1}. Un paramètre $\lambda \in [0, 1]$ spécifie si la mise à jour de la fonction de valeur est plus proche de Policy Iteration ($\lambda = 1$) ou de Value Iteration ($\lambda = 0$). λ correspond à la taille du pas effectué en direction de $Q^{\pi_{k+1}}$. Les auteurs de l'algorithme ont introduit un opérateur noté M_k et défini à chaque itération k pour tout vecteur Q par

$$M_k Q = (1-\lambda) B_{\pi_{k+1}} Q_k + \lambda B_{\pi_{k+1}} Q. \tag{4}$$

Intuitivement, cet opérateur peut être vu comme une application *amortie* de l'opérateur de Bellman $B_{\pi_{k+1}}$. Ils ont établi que l'opérateur M_k est contractant de facteur $\gamma\lambda$. L'algorithme λPI calcule son point fixe en effectuant des applications successives de M_k (voir Algorithme 4). Lorsque $\lambda = 1$, on a $M_k = B_{\pi_{k+1}}$ et

Algorithme 4 λ-Policy Iteration

répéter
$\pi_{k+1} \leftarrow \text{glouton}(Q_k)$
$Q_{k+1} \leftarrow M^{\infty}_{k} Q_k$
$k \leftarrow k+1$
jusqu'à $\|Q_{k+1} - Q_k\|_\infty < \epsilon$

l'algorithme se ramène à Policy Iteration. Plus λ est grand, et plus le vecteur Q_{k+1} calculé s'approche de $Q^{\pi_{k+1}}$. A l'inverse, lorsque $\lambda = 0$, on a $Q_{k+1} = B_{\pi_{k+1}} Q_k$ et on retrouve Value Iteration.

Bertsekas & Ioffe (1996) ont montré que la mise à jour de la fonction de valeur correspond, lorsque $\lambda < 1$, à une moyenne géométrique de termes identiques à ceux de Modified Policy Iteration. On a en effet $Q_{k+1} = T_\lambda Q_k$, où l'opérateur T_λ est défini pour tout vecteur Q par

$$T_\lambda Q = (1-\lambda) \left(\sum_{i=1}^{\infty} \lambda^{i-1} B^{i}_{\pi_{k+1}} Q \right). \tag{5}$$

λPI converge vers la fonction de valeur optimale pour tout $\lambda \in [0, 1]$ (Bertsekas & Ioffe, 1996). La vi-

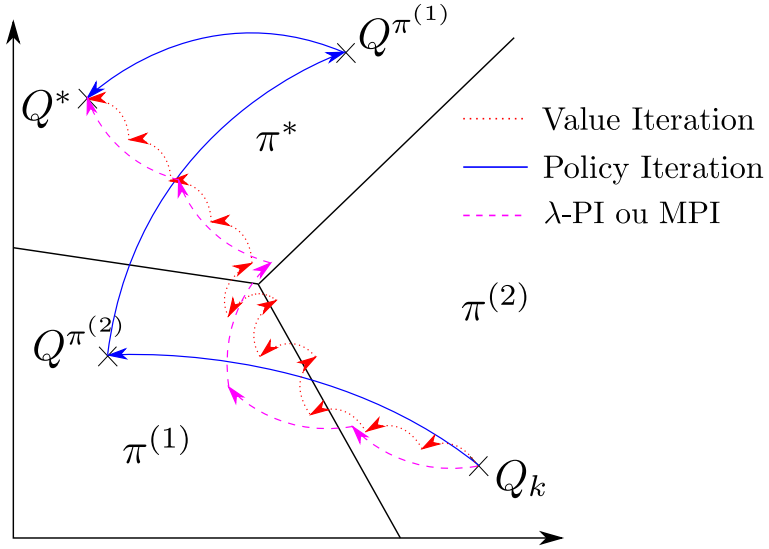

FIGURE 2 – **Vue intuitive de λPI dans la partition des politiques gloutonnes :** D'après Bertsekas & Tsitsiklis (1996), on peut décomposer l'espace des fonctions de valeur en un ensemble de polyhèdres, où chaque polyhèdre correspond à une région où une politique est gloutonne. Policy Iteration calcule un pas d'une seule étape directement vers $Q^{\pi_{k+1}}$ tandis que Value Iteration réalise plusieurs petits pas en direction de $Q^{\pi_{k+1}}$. Modified Policy Iteration (MPI) et λPI sont intermédiaires : ils réalisent une étape en direction de $Q^{\pi_{k+1}}$, dont la longueur est paramétrable par m et λ respectivement.

tesse de convergence asymptotique a été caractérisée analytiquement par ses auteurs. Nous rappelons ici ce résultat :

Proposition 1 (Convergence de λ-Policy Iteration (Bertsekas & Ioffe, 1996))

Soit (Q_k, π_k) la séquence de fonctions de valeurs et de politiques générées par λ-Policy Iteration. On a alors :

$$\lim_{k \to +\infty} Q_k = Q^*.$$

De plus, pour tout k plus grand qu'un certain index $\bar{k}$,

$$\|Q^* - Q_{k+1}\|_\infty \leq \frac{\gamma(1-\lambda)}{1-\lambda\gamma} \|Q^* - Q_k\|_\infty.$$

On voit ici que le facteur $\beta = \frac{\gamma(1-\lambda)}{1-\lambda\gamma}$ est compris entre 0 (lorsque $\lambda = 1$) et γ (lorsque $\lambda = 0$). La convergence asymptotique est donc plus rapide pour les valeurs de λ proches de 1. Les petites valeurs de λ introduisent ainsi un biais, dû au fait que l'on ne calcule plus la fonction de valeur de la politique courante, mais que l'on se contente de s'en approcher. L'ensemble des algorithmes que nous venons de présenter, ainsi que leur comportement en termes de biais, est illustré sur la figure 2.

2 Cas approché : Least-Squares λ Policy Iteration

Nous venons de présenter λPI (Bertsekas & Ioffe, 1996) dans le cas exact et de rappeler son principal résultat de convergence, qui montre que lorsque $\lambda < 1$, un biais dû à l'évaluation incomplète dégrade la vitesse de convergence asymptotique. Dans le cas exact, le paramètre λ n'a que peu d'utilité car rien ne vient compenser ce biais : la convergence asymptotique est plus lente et il n'y a pas de contrepartie. En pratique, c'est alors Modified Policy Iteration (Algorithme 3), plus simple que λPI, qui induit la meilleure convergence (Thiery & Scherrer, 2009c).

C'est dans le cas approché, lorsque la fonction de valeur est approximative et qu'elle est estimée à l'aide d'échantillons, que diminuer λ va s'avérer intéressant. D'abord, notons que la convergence asymptotique correspond au moment où la politique obtenue est optimale et où il ne reste plus qu'à affiner la fonction de

valeur ; ici, le fait que la vitesse de convergence asymptotique se dégrade lorsque $\lambda < 1$ va s'avérer peu crucial dans la mesure où de toute manière, on ne peut en général pas atteindre une politique optimale.

Ensuite, l'estimation qui est calculée a une certaine variance qui va pouvoir être diminuée grâce à λ. Bertsekas & Ioffe (1996) ont en effet montré que la mise à jour dans λPI pouvait s'écrire de la manière incrementale suivante : $Q_{k+1} \leftarrow Q_k + \Delta_k$, avec

$$\Delta_k(s,a) = E\left[\sum_{t=0}^{\infty}(\lambda\gamma)^t \delta_k(s_t,a_t,s_{t+1}) \Big| s_0 = s, a_0 = a, a_t = \pi_{k+1}(s_t) \text{ pour } t \geq 1\right] \quad (6)$$

où les δ_k sont les différences temporelles définies par $\delta_k(s,a,s') = R(s,a,s') + \gamma Q_k(s', \pi_{k+1}(s')) - Q_k(s,a)$. Cette expression sous forme d'espérance met en évidence le fait que λ a une influence de type compromis biais-variance. Lorsque $\lambda = 1$, on peut voir que $\Delta_k = (I - \gamma P_{\pi_{k+1}})^{-1}(\mathcal{R} + \gamma P_{\pi_{k+1}} Q_k - Q_k) = Q^{\pi_{k+1}} - Q_k$. Ainsi, on calcule la vraie fonction de valeur de π_{k+1} et il n'y a pas de biais. Cependant, lorsque l'on recourt à de l'échantillonnage pour estimer Δ_k, on peut voir sur l'équation (6) que l'horizon de la somme à estimer est plus important pour les grandes valeurs de λ. La variance de l'estimation risque alors de pénaliser l'algorithme. En revanche, lorsque $\lambda < 1$, cette variance est réduite. En contrepartie, on retrouve le biais lié au fait que le vecteur Q_{k+1} calculé à chaque itération n'est plus la valeur de π_{k+1}. Le nombre d'itérations nécessaires sera donc plus important, mais chaque itération sera moins sensible à la variance de l'estimation.

2.1 Architecture d'approximation

Avant de détailler la manière dont nous allons exploiter ce compromis biais-variance dans le cadre des moindres carrés, nous présentons les notations spécifiques au cas approché, et en particulier aux méthodes d'itération sur les politiques. On considère une architecture d'approximation linéaire classique. À chaque itération k, on maintient à jour une politique π_k et une fonction de valeur $\widehat{Q}_k$. La politique suivante π_{k+1} est alors la politique gloutonne par rapport à $\widehat{Q}_k$, puis on représente $\widehat{Q}_{k+1}$ avec une combinaison linéaire de fonctions de base :

$$\widehat{Q}_{k+1}(s,a) = \sum_{i=1}^{p} \phi_i(s,a) w_{k+1,i}.$$

Les termes $\phi_i(s,a)$ sont p fonctions de base arbitraires et les $w_{k+1,i}$ sont les paramètres de l'architecture. Comme en général $p \ll |\mathcal{S}||\mathcal{A}|$ lorsque l'espace d'états est grand, stocker une fonction de valeur ainsi représentée demande beaucoup moins d'espace qu'une représentation tabulaire. En notant $\phi(s,a)$ le vecteur de taille p dont les éléments sont les fonctions de base appliquées au couple (s,a) :

$$\phi(s,a) = \begin{pmatrix} \phi_1(s,a) \\ \vdots \\ \phi_p(s,a) \end{pmatrix}$$

et Φ la matrice de taille $|\mathcal{S}||\mathcal{A}| \times p$ composée de tous ces vecteurs :

$$\Phi = \begin{pmatrix} \phi(s_1,a_1)^{\mathrm{T}} \\ \phi(s_1,a_2)^{\mathrm{T}} \\ \vdots \\ \phi(s_{|\mathcal{S}|},a_{|\mathcal{A}|})^{\mathrm{T}} \end{pmatrix},$$

$\widehat{Q}_{k+1}$ peut être noté $\widehat{Q}_{k+1} = \Phi w_{k+1}$, où w_{k+1} est le vecteur des paramètres $(w_{k+1,1}, ..., w_{k+1,p})$ caractérisant la fonction de valeur $\widehat{Q}_{k+1}$.

2.2 Idée générale

Dans Least-Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003), qui est une version approximative de Policy Iteration, le vecteur w_{k+1} est calculé à chaque itération de manière à ce que $\widehat{Q}_{k+1}$ approche la fonction de valeur de π_{k+1}, c'est-à-dire le point fixe de $B_{\pi_{k+1}}$. En d'autres termes, LSPI détermine un w_{k+1} qui vérifie

$$B_{\pi_{k+1}} \Phi w_{k+1} \simeq \Phi w_{k+1}.$$

La démarche que nous proposons ici, intitulée Least-Squares λ Policy Iteration (LSλPI) consiste à généraliser LSPI en y ajoutant le paramètre λ de λPI. On peut remarquer dans les Algorithmes 2 et 4 que la seule différence entre Policy Iteration et λPI en version exacte est l'opérateur dont on calcule le point fixe : il s'agit de l'opérateur de Bellman $B_{\pi_{k+1}}$ dans le cas de Policy Iteration, et de l'opérateur M_k dans le cas de λPI. L'idée de LSλPI est donc de rechercher non pas le point fixe de $B_{\pi_{k+1}}$, mais celui de l'opérateur plus général M_k. Il s'agit donc de déterminer un w_{k+1} tel que

$$M_k \Phi w_{k+1} \simeq \Phi w_{k+1}.$$

Ainsi, on ne cherche plus à estimer la valeur $Q^{\pi_{k+1}}$, mais le vecteur Q_{k+1} que λPI calculerait en version exacte. LSPI devient donc un cas particulier de LSλPI pour lequel $\lambda = 1$. LSPI possède plusieurs caractéristiques intéressantes que LSλPI conserve naturellement : l'échantillonnage efficace (il s'agit d'une méthode du second ordre), l'évaluation off-policy de la fonction de valeur, qui permet de réutiliser les mêmes échantillons malgré les changements de politique, et le fait que le modèle du PDM soit optionnel mais puisse être exploité s'il est disponible. LSλPI ajoute à cela les caractéristiques de λPI discutées plus haut : le compromis biais-variance, qui peut améliorer la qualité de l'estimation, et l'évaluation optimiste de la fonction de valeur.

2.3 Méthode de projection du point fixe (PF)

Pour calculer w_{k+1}, LSPI peut utiliser deux méthodes standards, la méthode de projection du point fixe (PF) ou la méthode de minimisation du résidu quadratique (RQ), décrites par exemple dans Schoknecht (2002); Munos (2003); Lagoudakis & Parr (2003). Nous les généralisons ici dans le cas de LSλPI.

Comme $M_k \widehat{Q}_{k+1}$ n'est pas dans l'espace défini par les fonctions de base en général, le principe de la méthode du point fixe (PF) est de lui appliquer une projection orthogonale. On cherche donc la fonction de valeur approximative $\widehat{Q}_{k+1} = \Phi w_{k+1}$ qui vérifie

$$\widehat{Q}_{k+1} = \Pi M_k \widehat{Q}_{k+1} \tag{7}$$

où Π est la matrice de projection orthogonale, définie par $\Pi = \Phi(\Phi^{\mathrm{T}} D_\mu \Phi)^{-1} \Phi^{\mathrm{T}} D_\mu$. D_μ représente la matrice diagonale de taille $|\mathcal{S}||\mathcal{A}|$ dont les termes sont les poids de la projection, notés $\mu(s,a)$ où μ est une distribution de probabilités sur $S \times A$. Π correspond à la projection orthogonale selon la norme quadratique pondérée par μ, notée $\| \cdot \|_{\mu,2}$ et définie pour pour tout vecteur Q par

$$\|Q\|_{\mu,2} = \sqrt{\sum_{s,a} \mu(s,a) Q(s,a)^2}.$$

En développant l'équation (7) et en utilisant de la définition de M_k (équation (4)), on obtient

$$\begin{aligned}
\Phi w_{k+1} &= \Phi(\Phi^{\mathrm{T}} D_\mu \Phi)^{-1} \Phi^{\mathrm{T}} D_\mu (\mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}} \Phi w_k + \lambda \gamma P_{\pi_{k+1}} \Phi w_{k+1}) \\
\Phi^{\mathrm{T}} D_\mu \Phi w_{k+1} &= \Phi^{\mathrm{T}} D_\mu (\mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}} \Phi w_k + \lambda \gamma P_{\pi_{k+1}} \Phi w_{k+1}) \\
0 &= \Phi^{\mathrm{T}} D_\mu (\mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}} \Phi w_k + \lambda \gamma P_{\pi_{k+1}} \Phi w_{k+1} - \Phi w_{k+1}).
\end{aligned}$$

Ainsi, w_{k+1} est la solution du système linéaire $Aw = b$, de taille $p \times p$ (rappelons que p est le nombre de fonctions de base), avec

$$A = \Phi^{\mathrm{T}} D_\mu (\Phi - \lambda \gamma P_{\pi_{k+1}} \Phi) \qquad \text{et} \qquad b = \Phi^{\mathrm{T}} D_\mu (\mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}} \Phi w_k).$$

Lorsque le nombre d'états est élevé, A et b ne peuvent pas être calculés directement, même si un modèle du PDM est disponible. Cependant, en développant la structure de A et b, on remarque que ceux-ci peuvent

être exprimés sous la forme d'une espérance :

$$\begin{aligned} A &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu(s,a)\phi(s,a)\sum_{s'\in\mathcal{S}}P(s,a,s')\Big(\phi(s,a)-\lambda\gamma\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}} \\ &= E_{(s,a)\sim\mu,s'\sim P(s,a,\cdot)}\Big[\phi(s,a)\Big(\phi(s,a)-\lambda\gamma\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}}\Big], \\ b &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu(s,a)\phi(s,a)\sum_{s'\in\mathcal{S}}P(s,a,s')\Big(R(s,a,s')+(1-\lambda)\gamma\phi(s',\pi_{k+1}(s'))^{\mathrm{T}}w_k\Big) \\ &= E_{(s,a)\sim\mu,s'\sim P(s,a,\cdot),r'=R(s,a,s')}\Big[\phi(s,a)\Big(r'+(1-\lambda)\gamma\phi(s',\pi_{k+1}(s'))^{\mathrm{T}}w_k\Big)\Big]. \end{aligned} \tag{8}$$

On peut alors les estimer à partir d'un ensemble de L échantillons de la forme (s,a,r',s'), avec $(s,a)\sim\mu$, $s'\sim P(s,a,\cdot)$ et $r'=R(s,a,s')$. Afin de simplifier l'écriture des estimations, nous allons en fait estimer LA et Lb, ce qui ne changera pas la solution trouvée étant donné que l'on souhaite résoudre le système linéaire $Aw=b$. Notons $\widetilde{A}$ et $\widetilde{b}$ les estimations de LA et Lb basées sur les échantillons. Pour chaque échantillon (s,a,r',s') considéré, on met à jour $\widetilde{A}$ et $\widetilde{b}$ avec

$$\begin{aligned} \widetilde{A} &\leftarrow \widetilde{A}+\phi(s,a)\Big(\phi(s,a)-\lambda\gamma\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}}, \\ \widetilde{b} &\leftarrow \widetilde{b}+\phi(s,a)\Big(r'+(1-\lambda)\gamma\phi(s',\pi_{k+1}(s'))^{\mathrm{T}}w_k\Big). \end{aligned} \tag{9}$$

Si la distribution des échantillons correspond à μ, alors $\widetilde{A}$ et $\widetilde{b}$ sont bien des estimateurs non biaisés de A et b. Si un modèle du PDM est disponible, on peut exploiter cette connaissance. Les échantillons se résument alors à des couples états-actions (s,a) et la mise à jour de $\widetilde{A}$ et $\widetilde{b}$ devient

$$\begin{aligned} \widetilde{A} &\leftarrow \widetilde{A}+\phi(s,a)\Big(\phi(s,a)-\lambda\gamma\sum_{s'\in\mathcal{S}}P(s,a,s')\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}}, \\ \widetilde{b} &\leftarrow \widetilde{b}+\phi(s,a)\sum_{s'\in\mathcal{S}}P(s,a,s')\Big(R(s,a,s')+(1-\lambda)\gamma\phi(s',\pi_{k+1}(s'))^{\mathrm{T}}w_k\Big). \end{aligned} \tag{10}$$

Après avoir ainsi estimé LA et Lb à partir d'une source d'échantillons, on résout le système $\widetilde{A}w=\widetilde{b}$ pour calculer le vecteur de paramètres w_{k+1} qui caractérise fonction de valeur $\widehat{Q}_{k+1}$.

2.4 Minimisation du résidu quadratique (RQ)

Pour calculer la fonction de valeur approximative $\widehat{Q}_{k+1}=\Phi w_{k+1}$, une alternative à la méthode FP est la méthode de minimisation du résidu quadratique (RQ). Considérons l'équation (généralisée) de Bellman $Q_{k+1}=M_kQ_{k+1}$ et le *résidu de Bellman* défini par

$$\widehat{Q}_{k+1}-M_k\widehat{Q}_{k+1}.$$

On cherche à minimiser la norme quadratique de cette quantité, pondérée là aussi par une distribution μ :

$$\|\widehat{Q}_{k+1}-M_k\widehat{Q}_{k+1}\|_{\mu,2}.$$

On cherche donc un vecteur w_{k+1} qui minimise

$$\begin{aligned} &\|\Phi w_{k+1}-(1-\lambda)B_{\pi_{k+1}}\Phi w_k-\lambda B_{\pi_{k+1}}\Phi w_{k+1}\|_{\mu,2} \\ =\ &\|\Phi w_{k+1}-(1-\lambda)(\mathcal{R}+\gamma P_{\pi_{k+1}}\Phi w_k)-\lambda(\mathcal{R}+\gamma P_{\pi_{k+1}}\Phi w_{k+1})\|_{\mu,2} \\ =\ &\|\Phi w_{k+1}-\mathcal{R}-(1-\lambda)\gamma P_{\pi_{k+1}}\Phi w_k-\lambda\gamma P_{\pi_{k+1}}\Phi w_{k+1}\|_{\mu,2} \\ =\ &\|(\Phi-\lambda\gamma P_{\pi_{k+1}}\Phi)w_{k+1}-\mathcal{R}-(1-\lambda)\gamma P_{\pi_{k+1}}\Phi w_k\|_{\mu,2} \\ =\ &\|\Psi w_{k+1}-c\|_{\mu,2} \end{aligned}$$

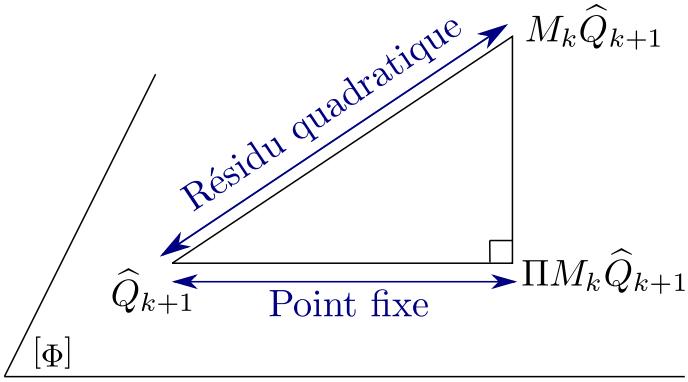

FIGURE 3 – Représentation schématique des deux méthodes. L'espace à trois dimensions représente l'espace des fonctions de valeur et le plan représente le sous-espace des fonctions de valeur approchées, qui est défini par les fonctions de base. La méthode PF cherche la fonction de valeur approchée qui est le point fixe de M_k suivi d'une projection sur l'espace des fonctions de valeur approximatives, alors que la méthode RQ cherche la fonction de valeur qui minimise la distance entre elle-même et une application de l'opérateur M_k.

où $\Psi = \Phi - \lambda\gamma P_{\pi_{k+1}}\Phi$ et $c = \mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}}\Phi w_k$. Ainsi, par une résolution standard aux moindres carrés, le vecteur de paramètres w_{k+1} qui minimise le résidu quadratique vérifie $(\Psi^{\mathrm{T}} D_\mu \Psi) w_{k+1} = \Psi^{\mathrm{T}} D_\mu c$. Notons $A = \Psi^{\mathrm{T}} D_\mu \Psi$ et $b = \Psi^{\mathrm{T}} D_\mu c$. Le problème revient alors à résoudre le système linéaire $Aw = b$, de taille $p \times p$, avec

$$\begin{aligned} A &= (\Phi - \lambda\gamma P_{\pi_{k+1}}\Phi)^{\mathrm{T}} D_\mu (\Phi - \lambda\gamma P_{\pi_{k+1}}\Phi), \\ b &= (\Phi - \lambda\gamma P_{\pi_{k+1}}\Phi)^{\mathrm{T}} D_\mu (\mathcal{R} + (1-\lambda)\gamma P_{\pi_{k+1}}\Phi w_k). \end{aligned}$$

De manière analogue à la méthode PF, la matrice A et le vecteur b peuvent s'écrire sous la forme d'une espérance :

$$\begin{aligned} A &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mu(s,a) \sum_{s'\in\mathcal{S}} P(s,a,s') \\ &\qquad \sum_{s''\in\mathcal{S}} P(s,a,s'') \Big(\phi(s,a) - \lambda\gamma\phi(s'',\pi_{k+1}(s''))\Big)\Big(\phi(s,a) - \lambda\gamma\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}} \\ &= E_{(s,a)\sim\mu, s'\sim P(s,a,\cdot)}\left[\Big(\phi(s,a) - \lambda\gamma\phi(s'',\pi_{k+1}(s''))\Big)\Big(\phi(s,a) - \lambda\gamma\phi(s',\pi_{k+1}(s'))\Big)^{\mathrm{T}}\right], \\ b &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mu(s,a) \sum_{s'\in\mathcal{S}} P(s,a,s') \sum_{s''\in\mathcal{S}} P(s,a,s'') \\ &\qquad \Big(\phi(s,a) - \lambda\gamma\phi(s'',\pi_{k+1}(s''))\Big)\Big(R(s,a,s') + (1-\lambda)\gamma\phi^{\mathrm{T}}(s',\pi_{k+1}(s'))w_k\Big) \\ &= E_{(s,a)\sim\mu, s'\sim P(s,a,\cdot), r'=R(s,a,s'), s''\sim P(s,a,\cdot)} \\ &\qquad \left[\Big(\phi(s,a) - \lambda\gamma\phi(s'',\pi_{k+1}(s''))\Big)\Big(r' + (1-\lambda)\gamma\phi^{\mathrm{T}}(s',\pi_{k+1}(s'))w_k\Big)\right]. \end{aligned}$$

On peut donc estimer la matrice A et le vecteur b à partir d'échantillons dont la distribution correspond à μ. Comme l'espérance d'un produit est en général différent du produit des espérances, on constate ici qu'il faut, de manière analogue à LSTD et LSPI (voir par exemple Sutton & Barto (1998); Munos (2003); Lagoudakis & Parr (2003)), utiliser pour chaque état s deux successeurs s' et s'' indépendants

Notons chaque échantillon (s, a, r', s', s''), où (s', r') et s'' sont les résultats de deux réalisations indépendantes de l'action a depuis l'état s (la récompense obtenue à l'état s'' n'est pas nécessaire). Là aussi, on note $\widetilde{A}$ et $\widetilde{b}$ les estimations de LA et Lb respectivement, où L désigne le nombre d'échantillons. Pour chaque

échantillon (s, a, r', s', s''), on met à jour les estimations $\widetilde{A}$ et $\widetilde{b}$ comme suit :

$$\widetilde{A} \leftarrow \widetilde{A} + \Big(\phi(s,a) - \lambda\gamma\phi(s'', \pi_{k+1}(s''))\Big)\Big(\phi(s,a) - \lambda\gamma\phi(s', \pi_{k+1}(s'))\Big)^{\mathrm{T}},$$
$$\widetilde{b} \leftarrow \widetilde{b} + \Big(\phi(s,a) - \lambda\gamma\phi(s'', \pi_{k+1}(s''))\Big)\Big(r' + (1-\lambda)\gamma\phi(s', \pi_{k+1}(s'))w_k\Big). \quad (11)$$

Enfin, si l'on dispose d'un modèle du PDM, les échantillons peuvent se limiter à des couples états-actions (s, a) et la mise à jour des estimations devient

$$\widetilde{A} \leftarrow \widetilde{A} + \Big(\phi(s,a) - \lambda\gamma \sum_{s''\in\mathcal{S}} P(s,a,s'')\phi(s'', \pi_{k+1}(s''))\Big)$$
$$\Big(\phi(s,a) - \lambda\gamma \sum_{s'\in\mathcal{S}} P(s,a,s')\phi(s', \pi_{k+1}(s'))\Big)^{\mathrm{T}},$$
$$\widetilde{b} \leftarrow \widetilde{b} + \Big(\phi(s,a) - \lambda\gamma \sum_{s''\in\mathcal{S}} P(s,a,s'')\phi(s'', \pi_{k+1}(s''))\Big)$$
$$\sum_{s'\in\mathcal{S}} P(s,a,s')\Big(R(s,a,s') + (1-\lambda)\gamma\phi(s', \pi_{k+1}(s'))w_k\Big). \quad (12)$$

On notera que lorsqu'un modèle du PDM est disponible, comme les échantillons se limitent à des couples états-actions (s, a), la contrainte de devoir générer les états et récompensenses suivants en double disparaît. Le reste de l'algorithme est identique à la méthode du point fixe : une fois LA et Lb estimés, on résout le système linéaire $\widetilde{A}w = \widetilde{b}$ afin d'obtenir la fonction de valeur $\widehat{Q}_{k+1}$.

2.5 Least-Squares λ Policy Iteration

Au final, LSλPI est résumé dans l'encart Algorithme 5, pour le choix d'une méthode (PF ou RQ) et d'une règle de mise à jour des estimations (avec ou sans modèle).

Algorithme 5 Least-Squares λ Policy Iteration

répéter
- $\pi_{k+1} \leftarrow \mathrm{glouton}(\Phi w_k)$
- Construire $\widetilde{A}$ et $\widetilde{b}$ d'après les échantillons (voir équation (9), (10), (11) ou (12))
- $w_{k+1} \leftarrow \widetilde{A}^{-1}\widetilde{b}$
- $k \leftarrow k+1$

jusqu'à $\|w_{k+1} - w_k\|_\infty < \epsilon$

LSλPI est un algorithme utilisant un paramètre λ, qui évalue les politiques avec une méthode du second ordre, et qui itère sur les politiques. Dans la littérature, les travaux aux moindres carrés qui évaluent une politique fixée, comme LSTD(λ) (Boyan, 2002) et LSPE(λ), pourraient aussi être utilisées dans un contexte d'itération sur les politiques afin de traiter des problèmes de contrôle. La principale différence de LSλPI avec l'état de l'art est qu'il s'agit un algorithme optimiste : il ne nécessite pas d'estimer la valeur de la politique gloutonne, mais seulement de suivre sa direction avec un pas ajustable selon la valeur de λ. LSTD(λ) est une version du second ordre de TD(λ) qui cherche à évaluer complètement la fonction de valeur. Il n'y a donc pas d'optimisme ici et le paramètre λ a une signification différente : il contrôle, lors de la mise à jour de la fonction de valeur, la profondeur des différences temporelles des trajectoires échantillonnées. LSPE(λ) est quant à lui l'algorithme λPI appliqué à l'évaluation d'une seule politique à l'aide d'une méthode du second ordre. LSPE(λ) applique ainsi l'opérateur T_λ (tel que défini à l'équation (5)) une infinité de fois. La politique est donc évaluée complètement. En revanche, LSλPI applique T_λ une seule fois et change de politique ensuite : l'évaluation est donc optimiste dès lors que $\lambda < 1$. Une autre différence entre LSPE(λ) et LSλPI est le fait que pour appliquer l'opérateur T_λ de façon approximative, les deux algorithmes estiment des quantités différentes. LSPE(λ) s'appuie sur une ou plusieurs trajectoires générées avec la politique courante et considère l'équation $Q_{k+1} = Q_k + \Delta_k$ (où Δ_k désigne les différences temporelles définies à

l'équation (5)), alors que LSλPI considère l'équation $Q_{k+1} = M_k Q_{k+1}$ et peut se contenter d'échantillons simples informatifs pour toutes les politiques : il est donc off-policy.

Revenons plus en détail sur les deux méthodes permettant de mettre à jour la fonction de valeur : la méthode du point fixe (PF) et la méthode de minimisation du résidu quadratique (RQ). Ces deux méthodes sont deux moyens de calculer $\widehat{Q}_{k+1}$ en cherchant à minimiser des critères différents, et les solutions qu'elles trouvent sont en général différentes. Un cas où les deux approches sont équivalentes est lorsque $\lambda = 0$: en effet, on peut voir que les estimations $\widetilde{A}$ et $\widetilde{b}$ sont construites de la même manière. L'algorithme revient dans ce cas à effectuer Fitted Value Iteration, une version approximative de Value Iteration (Szepesvári & Munos, 2005). L'espérance de l'équation (8) montre qu'il s'agit d'une régression aux moindres carrés. Dans le cas particulier où $\lambda = 1$, ce qui correspond aux évaluations faites par LSPI, la méthode PF semble donner de meilleurs résultats (Lagoudakis & Parr, 2003). De plus, sur certains exemples, on peut montrer que la méthode RQ ne calcule pas la bonne solution alors que la méthode PF le fait (Sutton *et al.*, 2009). Cependant, Schoknecht (2002) et Munos (2003) ont montré que PF est moins stable numériquement. En effet, la matrice A correspondant à PF peut être singulière, tandis que celle correspondant à BR ne l'est jamais.

Discutons maintenant de la convergence de LSλPI. Nous donnons d'abord une garantie de performance sur les politiques générées sous certaines conditions, puis nous étudions un cas où l'algorithme peut diverger selon les valeurs de λ et γ.

2.6 Bornes de performances

Nous venons d'introduire l'algorithme LSλPI, qui est une version approximative de λPI dans le cadre des moindres carrés. La question naturelle est de savoir s'il existe une garantie sur la performance des politiques générées par cet algorithme. De telles garanties sont connues dans les cas particuliers de Policy Iteration avec approximation ($\lambda = 1$) et Fitted Value Iteration ($\lambda = 0$) (Bertsekas & Tsitsiklis, 1996), mais pas dans le cas général d'un algorithme d'itération sur les politiques optimiste comme λPI avec approximation. Dans Bertsekas & Tsitsiklis (1996, section 6.4, page 320), les auteurs écrivent ainsi "This leaves us with a major theoretical question. Is there some variant of optimistic policy iteration that is guaranteed to generate policies whose performance is within $O(\epsilon/(1-\alpha))$ or even $O(\epsilon/(1-\alpha)^2)$ from the optimal ?". Le théorème que nous énonçons ici est la première borne de performance pour les algorithmes optimistes d'itération sur les politiques. Il est formulé de manière très générale en utilisant un coefficient de pondération abstrait λ_n. Selon le choix des valeurs de λ_n, de nombreuses méthodes d'itération sur les politiques avec approximation peuvent être généralisées.

Théorème 1 (Borne sur la performance de Policy Iteration approché optimiste)
Soit un ensemble de poids positifs $(\lambda_n)_{n\geq 1}$, tels que $\sum_{n\geq 1} \lambda_n = 1$. Soit une initialisation quelconque Q_0. Soit un algorithme itératif qui construit la suite $(\pi_k, Q_k)_{k\geq 1}$ de la manière suivante :

$\pi_{k+1} \leftarrow \text{glouton}(\pi_{k+1})$
$Q_{k+1} \leftarrow \sum_{n\geq 1} \lambda_n (B_{\pi_{k+1}})^n Q_k + \epsilon_{k+1}.$

ϵ_{k+1} représente l'erreur d'approximation, erreur commise en estimant la fonction de valeur de π_{k+1}. Soit ϵ une majoration uniforme de l'erreur : pour tout k, $\|\epsilon_k\|_\infty \leq \epsilon$. Alors

$$\limsup_{k\to\infty} \|Q^* - Q^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2}\epsilon.$$

λPI (et donc LSλPI) correspond au cas où $\lambda_n = (1-\lambda)\lambda^{n-1}$ pour tout n (voir l'équation (5)), mais le résultat est valable pour tout choix de coefficients λ_n dont la somme est égale à 1. Ainsi, le résultat s'applique également pour Modified Policy Iteration (Puterman, 1994), qui consiste à appliquer m fois l'opérateur de Bellman avec m fixé (c'est-à-dire de prendre $\lambda_m = 1$ et $\lambda_n = 0$ pour tout n différent de m), et pour Modified λ-Policy Iteration (Thiery & Scherrer, 2009c), où l'on prendrait $\lambda_n = (1-\lambda)\lambda^{n-1}$ pour $1 \leq n < m$, $\lambda_m = \lambda^m$, et $\lambda_n = 0$ pour $n > m$.

La preuve de ce théorème, qui se trouve en annexe, est significativement différente de celles qui ont été proposées (séparément) pour les versions approximatives de Value Iteration et Policy Iteration. Dans le cas de Policy Iteration, le raisonnement s'appuie sur la propriété de croissance des fonctions de valeurs, et dans le cas de Value Iteration, il utilise des arguments liés aux contractions. Malheureusement ces deux

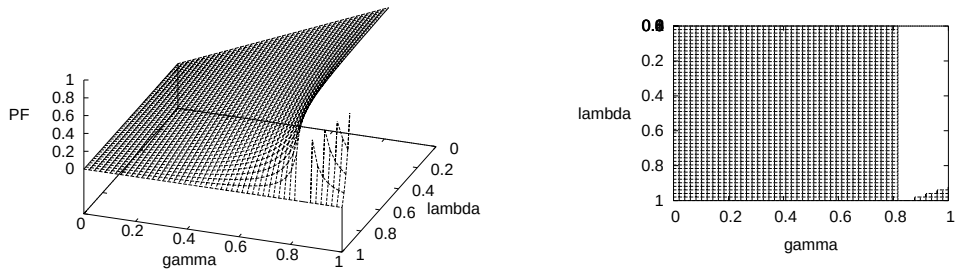

FIGURE 4 – Méthode PF. Gauche : $|\alpha_{PF}|$ en fonction de λ et γ. Droite : domaine où $|\alpha_{PF}| < 1$.

types d'arguments ne peuvent être utilisés dans le cadre du théorème ci-dessus. Nous renvoyons le lecteur intéressé à l'annexe pour plus de détails.

2.7 Cas possible d'une erreur non contrôlée

Le théorème que nous venons d'énoncer s'applique lorsque l'erreur d'approximation est bornée à chaque itération. Nous étudions ici un exemple simple tiré de Bertsekas & Tsitsiklis (1996, page 334), sur lequel les auteurs montrent, dans le cas de Fitted Value Iteration ($\lambda = 0$), que l'estimation de la fonction de valeur peut diverger pour certaines valeurs de γ, alors même que la capacité d'approximation de l'architecture linéaire permet de représenter exactement la fonction de valeur cible. Nous nous intéressons dans ce qui suit à la convergence des deux méthodes PF et RQ pour les autres valeurs de λ (rappelons que lorsque $\lambda = 0$, les deux méthodes PF et RQ sont équivalentes).

On considère un système non contrôlé avec 2 états, de sorte que les fonctions de valeurs sont définies uniquement sur l'espace d'états. La matrice de transition est donnée par

$$P = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

L'état 2 est absorbant et les récompenses sont 0. On a donc $Q(1) = Q(2) = 0$. On considère un approximateur linéaire avec $\Phi = (1\ 2)^{\mathrm{T}}$. Ici, la valeur peut être représentée exactement par l'espace choisi.

Méthode du point fixe (PF)

Dans le cas de la méthode PF, l'évolution des poids est régie par l'équation

$$w_{k+1} = (I - \lambda\gamma(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}P\Phi)^{-1}(1-\lambda)\gamma(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}P\Phi w_k.$$

On suppose ici que les échantillons sont distribués uniformément, c'est-à-dire $D_\mu = I$. On a $\Phi^{\mathrm{T}}\Phi = 5$ et donc $(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}} = (1/5\ 2/5)$. Comme $P\Phi = (2\ 2)^{\mathrm{T}}$, on en déduit que $(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}P\Phi$ vaut $6/5$. Autrement dit, on a

$$w_{k+1} = \alpha_{PF} w_k$$

avec

$$\alpha_{PF} = \frac{(1-\lambda)\frac{6}{5}\gamma}{1 - \lambda\frac{6}{5}\gamma}.$$

L'algorithme converge vers la solution si et seulement si $|\alpha_{PF}| < 1$. Ce coefficient admet des singularités : pour $\lambda\gamma$ proche de $5/6$, il tend vers $\pm\infty$. La figure 4 donne une représentation graphique du module de ce coefficient en fonction de λ et γ, ainsi que le domaine ou ce coefficient a un module inférieur à 1.

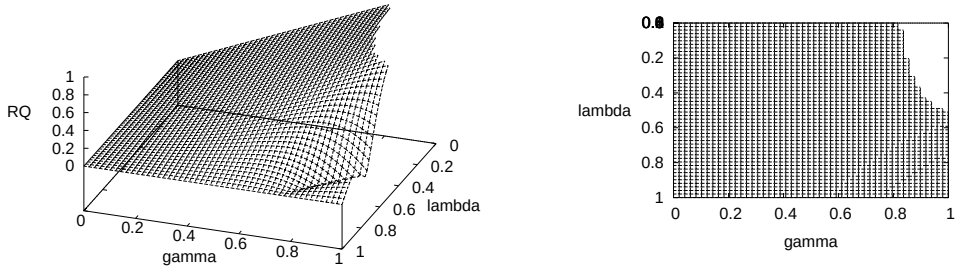

FIGURE 5 – Méthode RQ. Gauche : $|\alpha_{RQ}|$ en fonction de λ et γ. Droite : domaine où $|\alpha_{RQ}| < 1$.

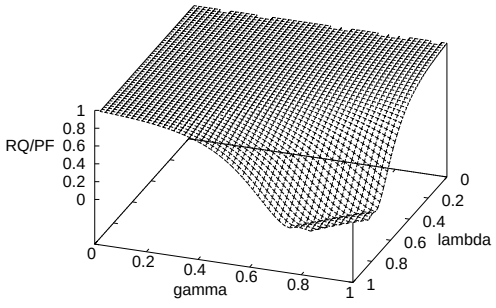

FIGURE 6 – $\frac{|\alpha_{RQ}|}{|\alpha_{PF}|}$ en fonction de λ et γ.

Méthode du résidu quadratique (RQ)

Avec la méthode RQ, l'évolution des poids est régie par l'équation

$$w_{k+1} = (\Psi^{\mathrm{T}}\Psi)^{-1}\Psi^{\mathrm{T}}(1-\lambda)\gamma P\Phi w_k$$

avec $\Psi = \Phi - \lambda\gamma P\Phi = (1-2\lambda\gamma\ 2-2\lambda\gamma)^{\mathrm{T}}$. Ainsi $\Psi\Psi^{\mathrm{T}} = (1-2\lambda\gamma)^2 + (2-2\lambda\gamma)^2$, quantité qui est toujours strictement supérieure à $1/2$. Comme $P\Phi = (2\ 2)^{\mathrm{T}}$, on en déduit que $\Psi^{\mathrm{T}}P\Phi = 6-8\lambda\gamma$. Au final, on a

$$w_{k+1} = \alpha_{RQ} w_k$$

avec

$$\alpha_{RQ} = \frac{(1-\lambda)\gamma(6-8\lambda\gamma)}{(1-2\lambda\gamma)^2 + (2-2\lambda\gamma)^2}.$$

La figure 5 donne comme précédemment une représentation graphique du module de ce coefficient en fonction de λ et γ, ainsi que le domaine ou ce coefficient a un module inférieur à 1.

Comparaison

On observe que, sur l'exemple étudié, la région où la méthode RQ converge est strictement plus grande que celle de la méthode PF. En particulier, pour toute valeur de γ, le choix parmi les valeurs de λ est plus grand pour RQ que pour PF. On notera également que pour la valeur limite $\gamma = 5/6$, la méthode PF ne converge que si l'on prend $\lambda = 1$. Enfin, dans le cas où les deux méthodes convergent, on peut voir sur le graphique de la figure 6, où nous avons tracé la courbe $\frac{|\alpha_{RQ}|}{|\alpha_{PF}|}$ en fonction de λ et γ, que la méthode RQ converge toujours plus vite que PF.

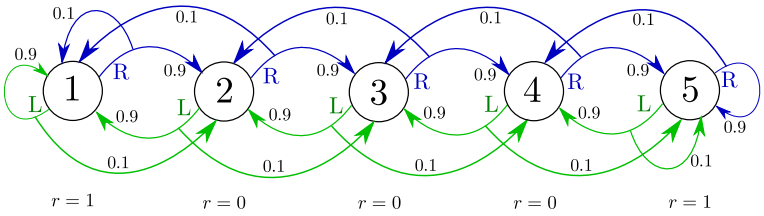

FIGURE 7 – Le PDM étudié, représenté ici avec 5 états (nos expériences comportent 20 états). Chaque action (L ou R) envoie dans la bonne direction avec une probabilité de $0,9$ et dans la direction opposée avec une probabilité de $0,1$. Les deux extrémités comportent une récompense de 1.

3 Expériences

Nous avons introduit l'algorithme LSλPI, qui ajoute à LSPI (Lagoudakis & Parr, 2003) le caractère optimiste et la possibilité de faire un compromis biais-variance de λPI (Bertsekas & Ioffe, 1996), et nous avons établi sa validité de manière théorique en donnant une garantie de convergence sous certaines conditions. Nous venons également de voir que dans le cas de l'évaluation d'une politique fixée, le choix de λ peut influer sur la convergence ou non de l'algorithme.

Nous présentons maintenant quelques expériences sur des problèmes d'itération sur les politiques afin de montrer l'intérêt de LSλPI d'un point de vue expérimental. Etant donné que LSλPI est une généralisation de LSPI, nous avons réalisé des expériences sur deux problèmes d'optimisation de politique précédemment étudiés par Lagoudakis & Parr (2003) dans le cadre de LSPI : un problème de chaîne d'états dans lequel on connaît la fonction de valeur optimale exacte, afin de pouvoir facilement évaluer les performances obtenues, et enfin le jeu de Tetris, qui est un problème plus difficile et à grand espace d'états.

3.1 Chaîne d'états

La figure 7 représente le PDM simple considéré par Lagoudakis & Parr (2003) pour illustrer le comportement de LSPI. Il s'agit d'une chaîne de 20 états avec deux actions possibles : gauche (L) ou droite (R). Chaque action envoie dans la bonne direction avec une probabilité de $0,9$, et dans la direction opposée avec une probabilité de $0,1$. Lorsque l'agent arrive à un des deux états aux extrémités de la chaîne, il obtient une récompense de 1. Dans tous les autres états, il obtient une récompense nulle. Il est clair que la politique optimale est $L \dots LR \dots R$. La fonction de valeur optimale peut être calculée facilement et de manière exacte. Ainsi, lors de nos expériences, on pourra tracer la courbe représentant la distance entre la valeur courante et la valeur optimale (pour mesurer la qualité de l'approximation), et la distance entre la valeur de la politique courante et la valeur optimale (pour mesurer la qualité de la politique obtenue).

Dans ces expériences, on n'utilisera pas la connaissance du modèle du PDM (transitions et récompenses). Comme Lagoudakis & Parr (2003), nous avons testé deux jeux de fonctions de bases pour représenter l'espace d'états. Le premier est un ensemble de fonctions de base polynômiales répétées pour chacune des deux actions :

$$\phi(s,a) = \begin{pmatrix} \mathbb{1}_{a=L} \times 1 \\ \mathbb{1}_{a=L} \times s \\ \mathbb{1}_{a=L} \times s^2 \\ \mathbb{1}_{a=R} \times 1 \\ \mathbb{1}_{a=R} \times s \\ \mathbb{1}_{a=R} \times s^2 \end{pmatrix}$$

où s est le numéro d'état (de 1 à 20), et $\mathbb{1}_{a=X} = 1$ si $a = X$ et 0 sinon. Le second jeu de fonctions est un ensemble de gaussiennes dont les moyennes sont distribuées uniformément sur l'espace d'états et dont la variance est définie par $\sigma = 4$. Pour chaque action, on a 10 gaussiennes et un terme constant, ce qui donne un total de 22 fonctions de base.

3.1.1 Influence de λ

Nous avons observé que la convergence de la fonction de valeur est plus difficile lorsque le nombre d'échantillons est faible, ou lorsque γ est élevé (c'est-à-dire lorsque l'horizon du problème est grand). Si

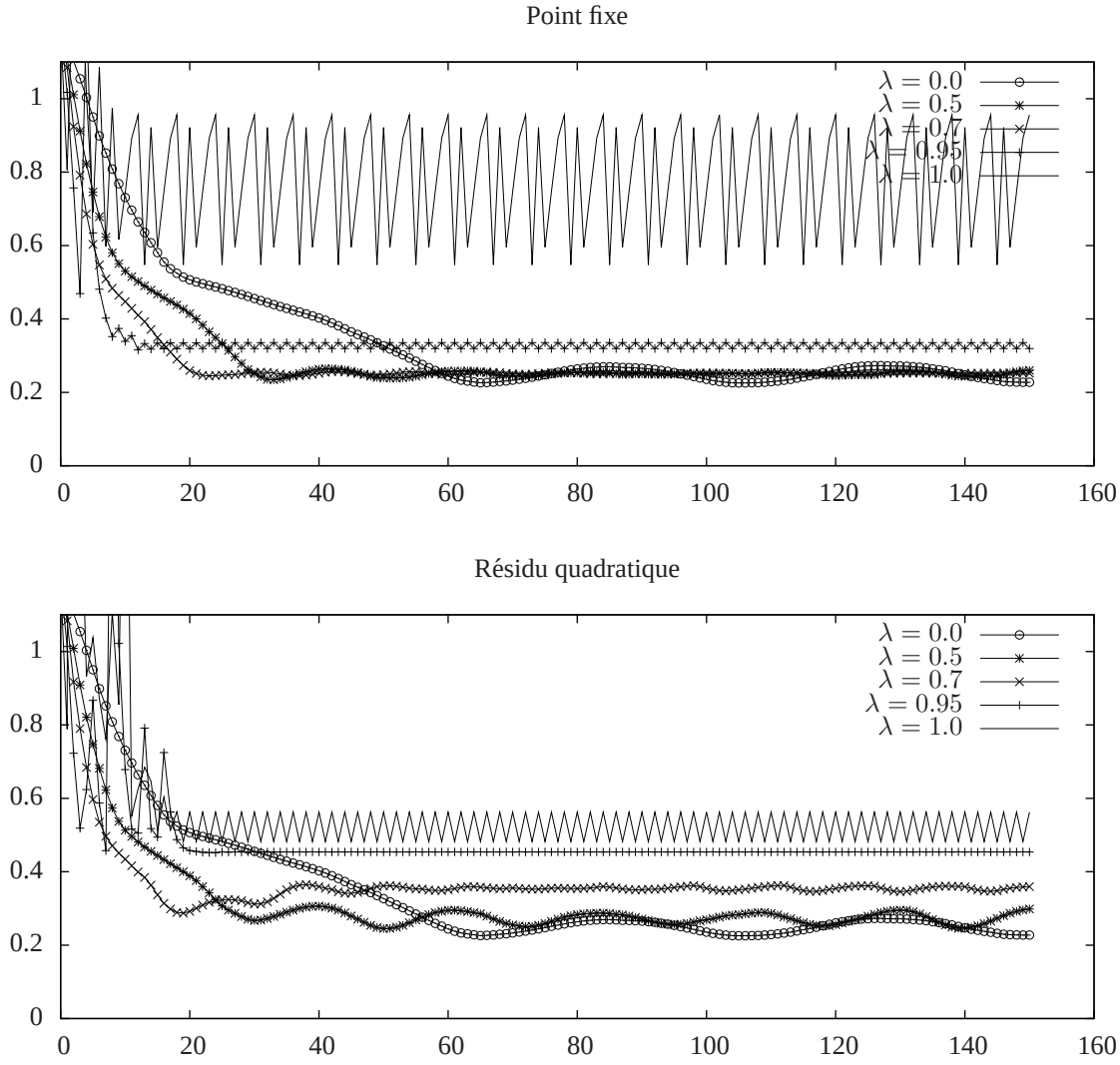

FIGURE 8 – Evolution au cours des itérations de $\|\widehat{Q}_k - Q^*\|_\infty$, distance entre la fonction de valeur approximative courante et la fonction de valeur optimale, pour plusieurs valeurs de λ. Fonctions de base gaussiennes. $\gamma = 0.95$. Moyenne de 10 exécutions, les exécutions utilisant des ensembles d'épisodes de 200 échantillons. Haut : méthode PF. Bas : méthode RQ.

le nombre d'échantillons est suffisamment important ou si γ est peu élevé, λ n'a que peu d'influence car la variance de l'estimation discutée précédemment pose moins de problèmes. Il est alors préférable d'utiliser $\lambda = 1$ afin de converger plus rapidement. Dans les cas de convergence plus difficile, on observe plus clairement une influence du paramètre λ. La figure 8 représente la distance entre la fonction de valeur à chaque itération et la fonction de valeur optimale, moyennée sur 10 exécutions ayant des ensembles d'échantillons différents, et ce pour plusieurs valeurs de λ. Les ensembles d'échantillons comportent des épisodes de 200 états visités avec la politique qui choisit une action aléatoire uniformément. La méthode utilisée est PF dans le graphique du haut, et RQ dans le graphique du bas. Pour la méthode RQ, chaque échantillon est généré avec deux transitions indépendantes afin de ne pas biaiser l'estimation (voir la discussion de la section 2.4). Dans les deux cas, on utilise un approximateur gaussien et $\gamma = 0.95$. Comme attendu, on observe que pour $\lambda < 1$, l'approximation est meilleure car la variance de l'estimation est réduite. En contrepartie, un plus grand nombre d'itérations est nécessaire pour atteindre cette bonne approximation. En effet, comme discuté précédemment, utiliser une valeur de λ inférieure à 1 introduit un biais dans la mesure où on ne cherche plus à s'approcher le plus possible de la valeur de la politique courante, mais seulement d'un certain pas dans sa direction. On remarquera que ces courbes sont similaires à celles de Kearns & Singh (2000) qui proposent une analyse théorique du compromis biais-variance de TD(λ).

On remarque que pour $\lambda = 1$, au bout de quelques itérations seulement, la fonction de valeur cesse de s'améliorer et se met à osciller en restant relativement loin de l'optimal par rapport aux valeurs de λ inférieures. Les valeurs intermédiaires de λ offrent le meilleur compromis en donnant une bonne approximation et avec un nombre d'itérations raisonnable. Sur la plupart des expériences que nous avons effectuées, la méthode PF et la méthode RQ donnent des performances similaires, avec un léger avantage pour la méthode PF. On observe ainsi sur la figure 8 qu'avec la méthode RQ, il y a plus de valeurs de λ pour lesquelles la fonction de valeur se stabilise trop rapidement, avant d'avoir atteint une bonne approximation. En pratique, la méthode PF semble donc un peu plus performante que la méthode RQ étant donné qu'elle donne de bons résultats pour un plus grand intervalle de valeurs de λ. Cependant, elle peut en théorie poser des problèmes de stabilité numérique dans certains cas (voir section 2.5), bien que nous n'ayons pas rencontré de tels problèmes au cours de nos expériences.

On observe également, surtout dans le cas de la méthode RQ, qu'il serait intéressant d'utiliser une valeur décroissante de λ. En effet, dans les premières itérations, une valeur de λ proche de 1 permet de s'approcher rapidement de la fonction de valeur optimale, puis au fur et à mesure des itérations, les valeurs de λ plus petites conduisent à une meilleure approximation.

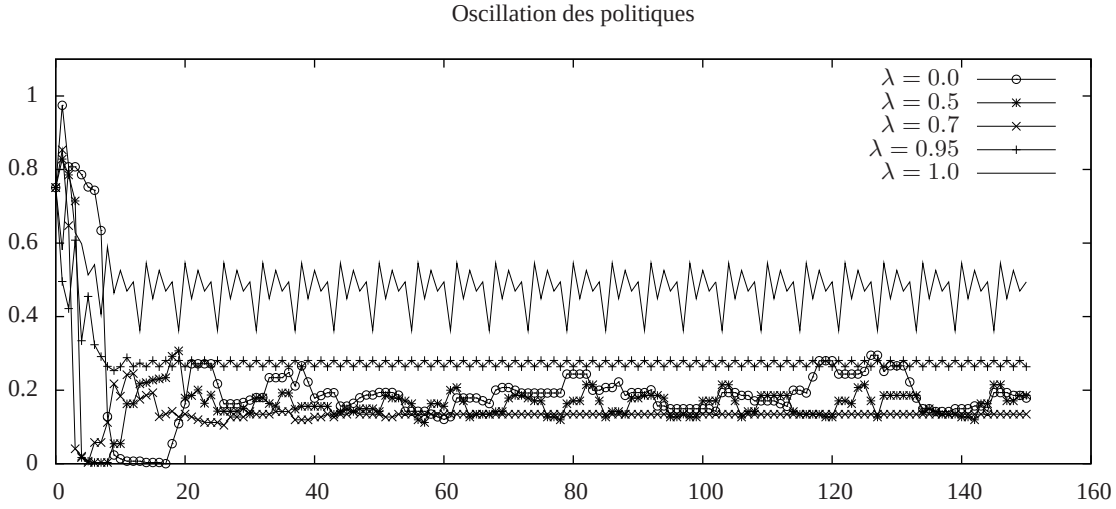

FIGURE 9 – Chaîne d'états. Évolution de $\|Q^{\pi_k} - Q^*\|_\infty$, distance entre la valeur de la politique courante et la valeur optimale, pour l'expérience de la figure 8 avec la méthode PF.

Par ailleurs, on observe que lorsque la fonction de valeur ne converge pas, la politique oscille avec une fréquence qui augmente avec λ. Cela se produit lorsqu'il y a un cycle dans la séquence des politiques. On peut observer ce phénomène sur la figure 9, qui représente $\|Q^{\pi_k} - Q^*\|_\infty$ pour l'expérience de la figure 8

avec la méthode PF. Pour les petites valeurs de λ, la politique oscille lentement car LSλPI réalise des petits pas. Lorsque λ augmente, les oscillations sont plus rapides puisque les pas sont plus importants (voir la vue intuitive de la figure 2). Il est intéressant de constater qu'il y a ensuite valeurs intermédiaires de λ pour lesquelles la politique converge (par exemple $\lambda = 0,7$). Enfin, pour les grandes valeurs de λ, la politique ne converge plus et oscille à nouveau, avec une fréquence plus importante. La possibilité d'utiliser λ pour stabiliser la politique est d'autant plus intéressante car on peut montrer [4] que lorsque la politique a convergé, le coefficient $\frac{2\gamma}{(1-\gamma)^2}$ du Théorème 1 est réduit d'un facteur $(1 - \lambda\gamma)$.

3.1.2 Cas d'une politique fixée

La figure 10 représente une expérience où la convergence est moins difficile que sur l'exemple précédent car $\gamma = 0,9$ (les autres paramètres sont inchangés et les échantillons sont les mêmes). On s'intéresse ici à la méthode PF uniquement, où l'on observe un phénomène qui n'apparaît pas avec la méthode RQ. Le graphique du haut indique la qualité de la politique courante : il représente, comme sur la figure 9, la distance entre la valeur de la politique courante et la valeur optimale. On remarque que la politique converge, excepté dans le cas où $\lambda = 1$. Le graphique du bas représente, comme précédemment, la distance entre la fonction de valeur approximative courante et la fonction de valeur optimale. A partir de l'itération 40 environ, la politique à évaluer devient la même pour toutes les valeurs de λ pour lesquelles la politique a convergé. Il semble alors, d'après le graphique, que l'estimation de la fonction de valeur converge vers la même quantité quelle que soit la valeur de λ.

Nous pouvons en effet vérifier analytiquement que, lorsque la politique est fixée et que la méthode PF converge, elle converge vers une valeur qui ne dépend pas de λ. Ce n'est pas le cas de la méthode RQ en général. En utilisant la définition de M_k (équation 4), le fait que $\widehat{Q}_{k+1} = \widehat{Q}_k$ et que $\pi_{k+1} = \pi_k$, on a

$$\widehat{Q}_{k+1} = \Pi M_k \widehat{Q}_{k+1} = \Pi((1-\lambda)B_{\pi_{k+1}}\widehat{Q}_{k+1} + \lambda B_{\pi_{k+1}}\widehat{Q}_{k+1}) = \Pi B_{\pi_{k+1}}\widehat{Q}_{k+1}.$$

Ainsi, $\widehat{Q}_{k+1}$ converge vers le point fixe de $\Pi B_{\pi_{k+1}}$, qui ne dépend pas de λ. On pourrait alors penser que λ est inutile dans la méthode PF, mais rappelons que ce n'est qu'en cas de convergence de la politique et de la fonction de valeur que cette dernière cesse de dépendre de λ. Or, nous avons vu que c'est justement le réglage de λ qui peut permettre d'obtenir la convergence ou non. Cette propriété suggère que le choix de λ est plus difficile dans le cas de la méthode RQ étant donné qu'il influe non seulement sur la convergence, mais aussi sur la fonction de valeur obtenue après convergence de la politique.

3.2 Tetris

Tetris est un célèbre jeu vidéo qui consiste à déplacer et tourner des pièces de différentes formes qui tombent les unes après les autres dans une grille de 10 colonnes et 20 lignes. Lorsqu'une ligne est pleine, celle-ci est supprimée et toutes les cellules au-dessus d'elle descendent d'une ligne. L'objectif est de supprimer un maximum de lignes avant qu'il n'y ait plus assez d'espace libre en haut de la pile. On peut trouver une spécification détaillée de Tetris sur le site de Fahey (2003). Tetris a fait l'objet de nombreux travaux de recherche (voir la revue de Thiery & Scherrer (2009a)). Résoudre Tetris est un problème difficile : il contient un grand nombre de configurations (de l'ordre de $2^{200} \simeq 10^{60}$). De plus, trouver une séquence de coups qui maximise le nombre de lignes est un problème NP-complet, même dans le cas où la séquence de pièces est connue à l'avance (Demaine *et al.*, 2003).

Nous avons reproduit le protocole expérimental de Lagoudakis *et al.* (2002). Nous avons ainsi lancé des expériences avec les mêmes fonctions de base et en utilisant la connaissance du modèle du PDM. Les fonctions de base, définies sur l'espace d'états-actions, sont la hauteur maximale de la pile, le nombre de trous, la somme des différences de hauteur entre colonnes adjacentes (en valeur absolue), la hauteur moyenne des colonnes, le changement de ces quantités dans l'état suivant (afin de capturer l'effet du choix d'une action depuis l'état courant), le nombre de lignes réalisées en effectuant l'action et enfin un terme constant. Bien que notre politique initiale soit la même que celle de Lagoudakis *et al.* (2002) (communication personnelle), les scores peuvent difficilement être comparés. La politique initiale réalise environ 250 lignes de moyenne par partie sur notre implémentation, tandis qu'ils reportent un score initial moyen de 600 lignes. Ceci est vraisemblablement dû à des différences d'implémentation qui peuvent avoir un impact significatif sur le score (voir Thiery & Scherrer (2009b)).

4. Nous omettons ces détails par manque de place.

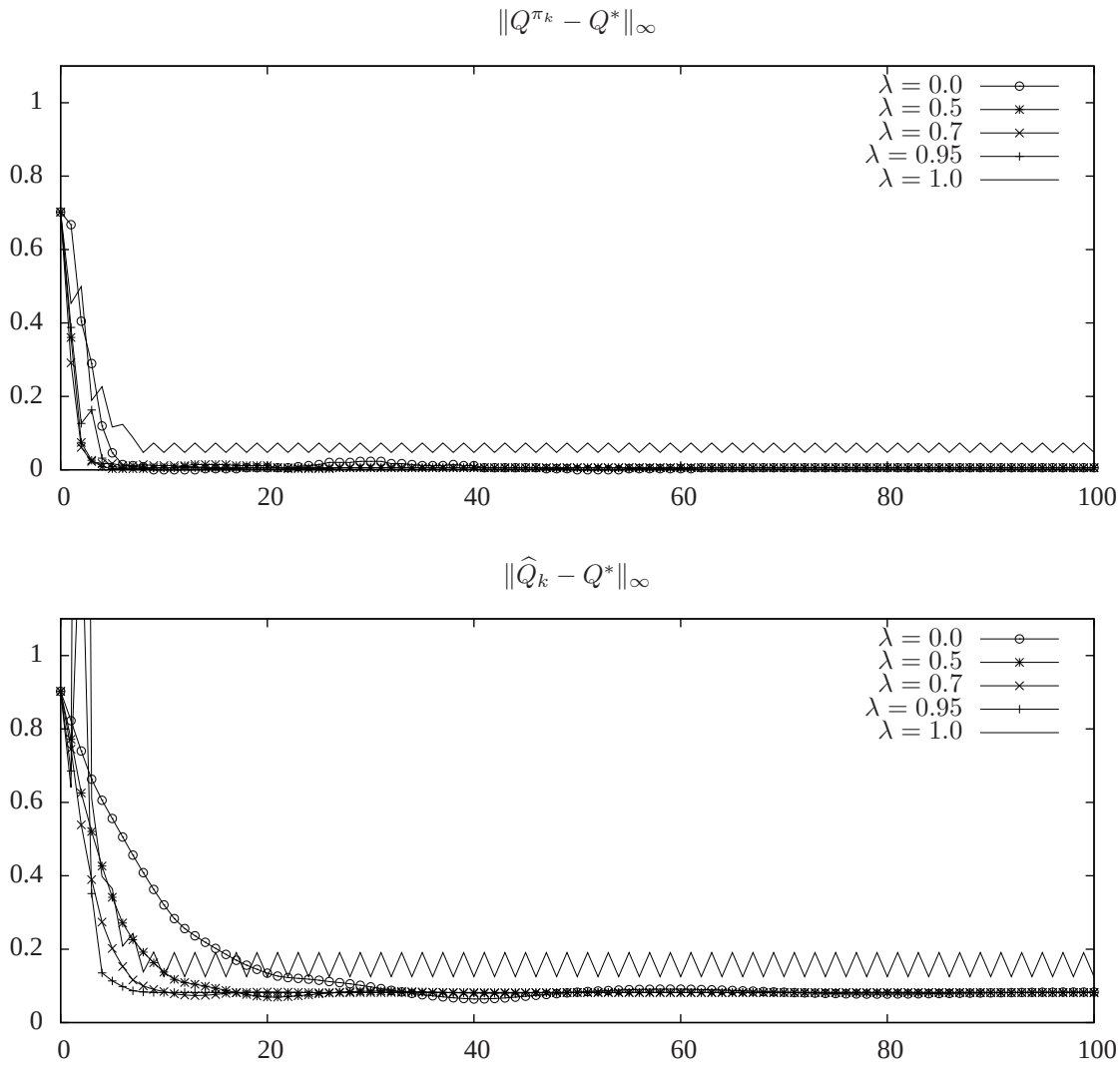

FIGURE 10 – Résultat de la méthode PF appliquée avec $\gamma = 0.9$ et des fonctions de base gaussiennes. Haut : Evolution de $\|\widehat{Q}^{\pi_k} - Q^*\|_\infty$, distance entre la fonction de valeur exacte de la politique courante, pour plusieurs valeurs de λ. Bas : Evolution de $\|\widehat{Q}_k - Q^*\|_\infty$, distance entre la fonction de valeur approximative courante et la fonction de valeur optimale, pour plusieurs valeurs de λ. Fonctions de base gaussiennes. $\gamma = 0.9$. Moyenne de 10 exécutions, chaque exécution utilisant une ensemble différent d'épisodes de 200 échantillons. On observe que lorsque la politique est la même pour différentes valeurs de λ, la fonction de valeur semble converger vers une limite qui ne dépend pas de λ. Nous avons vérifié cette propriété analytiquement, propriété qui ne s'applique pas à la méthode RQ.

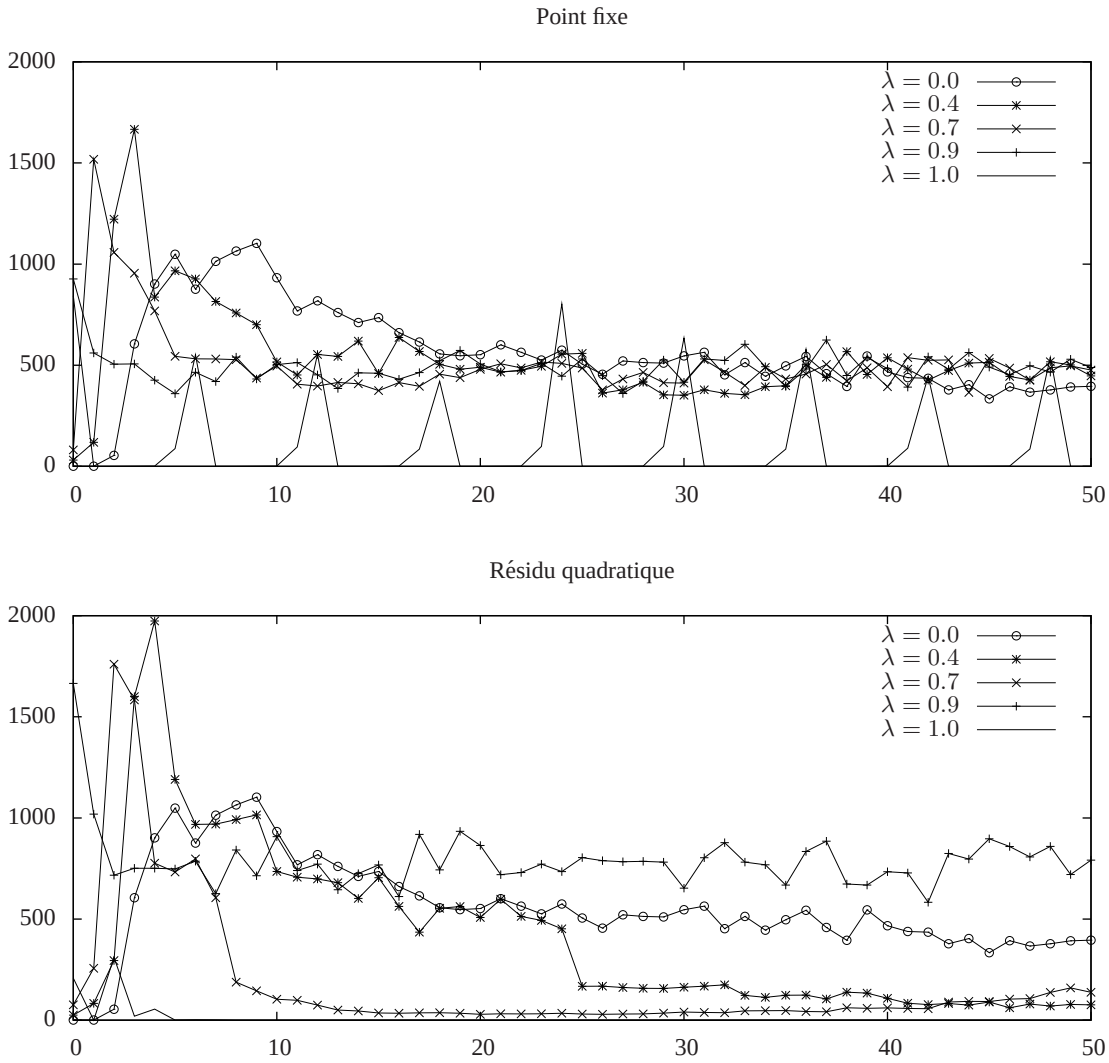

FIGURE 11 – Score moyen de 100 parties de Tetris pour différentes valeurs de λ à chaque itération de LSλPI. A cause du faible nombre d'échantillons (1000), l'algorithme diverge lorsque $\lambda = 1$ pour les deux méthodes. Lorsque la convergence est obtenue, c'est avec la méthode RQ que la meilleure performance est atteinte (800 lignes de moyenne), pour $\lambda = 0,9$.

Nous avons d'abord lancé LSλPI sur un ensemble de 10 000 échantillons, comme Lagoudakis *et al.* (2002) l'ont fait pour LSPI (c'est-à-dire $\lambda = 1$). Nous avons observé que diminuer λ n'améliorait pas la performance (cela ne faisait que ralentir la convergence). On peut supposer que l'ensemble d'échantillons était trop grand pour que λ soit utile. Nous avons donc ensuite employé un ensemble d'échantillons plus réduit (1 000 échantillons au lieu de 10 000) afin de rendre la convergence plus difficile. La figure 11 représente la performance des politiques apprises pour différentes valeurs de λ. Lorsque $\lambda = 1$, l'algorithme est très instable et génère de mauvaises politiques car le nombre d'échantillons est faible, ce qui rend la variance de l'estimation importante. Le score oscille entre 0 et 600 lignes par partie avec la méthode PF, et tombe à 0 avec la méthode RQ. De meilleures performances sont atteintes pour d'autres valeurs de λ. Comme pour le problème de la chaîne d'états, on remarque que λ a plus d'influence dans le cas de la méthode RQ. Après convergence, la meilleure valeur de λ semble être $0,9$ et avec la méthode RQ. La politique correspondante réalise environ 800 lignes par partie (rappelons ici que la politique initiale faisait environ 250 lignes par partie).

Il faut noter que jusqu'à présent, on ne peut pas comparer directement les résultats de LSPI ou LSλPI sur Tetris avec d'autres approches d'apprentissage par renforcement (Tsitsiklis & van Roy, 1996; Bertsekas & Ioffe, 1996; Kakade, 2001; Farias & van Roy, 2006; Ramon & Driessens, 2004) étant donné que les fonctions de bases proposées par Lagoudakis & Parr (2003) dans LSPI sont assez différentes, et que de plus, elles sont définies sur l'espace d'états-actions. Il serait cependant intéressant de redéfinir sur cet espace les fonctions de base les plus utilisées de la littérature des travaux sur Tetris, afin d'avoir une idée plus précise du succès de LSλPI sur cette application par rapport aux autres approches.

Conclusion

Nous avons proposé l'algorithme LSλPI, une implémentation de λPI (Bertsekas & Ioffe, 1996) dans le contexte des moindres carrés. LSλPI généralise LSPI (Lagoudakis & Parr, 2003) en y ajoutant le paramètre λ de λPI. Il s'agit à notre connaissance du premier algorithme qui optimise une politique en ayant les caractéristiques suivantes : compromis biais-variance, optimisme, méthode du second ordre et off-policy (c'est-à-dire sans nécessité de regénérer des échantillons à chaque changement de politique).

Nous avons présenté un résultat analytique original qui montre que les algorithmes de type Policy Iteration optimiste avec approximation, tels que LSλPI, ont une garantie de performance sous réserve que l'erreur d'approximation soit bornée à chaque itération. Même si tous les algorithmes d'évaluation de politique peuvent être utilisés dans un contexte d'itération sur les politiques *de façon optimiste*, il n'y avait jusqu'à maintenant pas de garantie de performance à notre connaissance, sauf dans le cas particulier de l'optimisme le plus extrême ($\lambda = 0$) qui correspond à Fitted Value Iteration.

Enfin, nous présentons des résultats expérimentaux qui confirment l'influence de λ sur la qualité de l'approximation et la performance des politiques générées. Nos résultats empiriques sur deux problèmes de contrôle optimal, une chaîne d'états et le jeu Tetris, montrent que les valeurs de λ intermédiaires (différentes de 0 et 1) peuvent donner de meilleurs résultats en pratique lorsque le nombre d'échantillons est limité. Cela peut s'avérer intéressant dans des applications d'apprentissage on-line et off-policy.

Perspectives

Avec LSλPI, comme avec les autres algorithmes du second ordre et ceux utilisant un paramètre λ, un problème qui se pose est de savoir quelle méthode choisir (FP ou RQ) et comment fixer la valeur de λ. λ peut en effet avoir une influence cruciale sur la convergence ou non de l'algorithme et sur les performances obtenues. Les expériences suggèrent qu'avec la méthode PF, il y a une plus grande plage de valeurs de λ qui permettent d'obtenir de bonnes performances. Cela confirme la tendance selon laquelle la méthode PF donnerait des résultats légèrement meilleurs en pratique. Cependant, les expériences sur Tetris montrent qu'une fois que λ est correctement fixé, la méthode RQ peut faire mieux que la méthode PF. En outre, une valeur décroissante de λ peut offrir le meilleur compromis entre la vitesse de convergence et la qualité de l'estimation. Kearns & Singh (2000) proposent une méthode analytique pour déterminer la valeur optimale de λ à chaque itération dans le cas de TD(λ). Il serait intéressant d'étudier une approche similaire pour λPI.

Références

BELLMAN R. E. (1957). *Dynamic Programming*. Princeton, NJ : Princeton University Press.

BERTSEKAS D. & IOFFE S. (1996). *Temporal differences-based policy iteration and applications in neuro-dynamic programming*. Rapport interne, MIT.

BERTSEKAS D. & TSITSIKLIS J. (1996). *Neurodynamic Programming*. Athena Scientific.

BOYAN J. A. (2002). Technical update : Least-squares temporal difference learning. *Machine Learning*, **49**, 233–246.

BRADTKE S. J. & BARTO A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, **22**, 33–57.

DEMAINE E. D., HOHENBERGER S. & LIBEN-NOWELL D. (2003). Tetris is hard, even to approximate. In *Proc. 9th International Computing and Combinatorics Conference (COCOON 2003)*, p. 351–363.

FAHEY C. P. (2003). Tetris AI, Computer plays Tetris. `http://colinfahey.com/tetris/tetris.html`.

FARIAS V. & VAN ROY B. (2006). *Tetris : A study of randomized constraint sampling*. Springer-Verlag.

KAKADE S. (2001). A natural policy gradient. In *Advances in Neural Information Processing Systems (NIPS 14)*, p. 1531–1538.

KEARNS M. & SINGH S. (2000). Bias-variance error bounds for temporal difference updates. In *In Proceedings of the 13th Annual Conference on Computational Learning Theory*, p. 142–147.

LAGOUDAKIS M. G. & PARR R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, **4**, 1107–1149.

LAGOUDAKIS M. G., PARR R. & LITTMAN M. L. (2002). Least-squares methods in reinforcement learning for control. In *In SETN'02 : Proceedings of the Second Hellenic Conference on AI*, p. 249–260 : Springer-Verlag.

MUNOS R. (2003). Error bounds for approximate policy iteration. In *ICML'03*, p. 560–567.

MUNOS R. (2007). Performance Bounds in Lp norm for Approximate Value Iteration. *SIAM Journal on Control and Optimization*.

NEDIĆ A. & BERTSEKAS D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, **13**(1-2), 79–110.

PUTERMAN M. (1994). *Markov Decision Processes*. Wiley, New York.

RAMON J. & DRIESSENS K. (2004). On the numeric stability of gaussian processes regression for relational reinforcement learning. In *ICML-2004 Workshop on Relational Reinforcement Learning*, p. 10–14.

SCHOKNECHT R. (2002). Optimality of reinforcement learning algorithms with linear function approximation. In *NIPS*, p. 1555–1562.

SUTTON R. & BARTO A. (1998). *Reinforcement Learning, An introduction*. BradFord Book. The MIT Press.

SUTTON R. S., MAEI H. R., PRECUP D., BHATNAGAR S., SILVER D., SZEPESVÁRI C. & WIEWIORA E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML'09 : Proceedings of the 26th Annual International Conference on Machine Learning*, p. 993–1000.

SZEPESVÁRI C. & MUNOS R. (2005). Finite time bounds for sampling based fitted value iteration. In *ICML'05 : Proceedings of the 22nd international conference on Machine learning*, p. 880–887 : ACM.

THIERY C. & SCHERRER B. (2009a). Building Controllers for Tetris. *International Computer Games Association Journal*, **32**, 3–11.

THIERY C. & SCHERRER B. (2009b). Construction d'un joueur artificiel pour Tetris. *Revue d'Intelligence Artificielle*, **23**, 387–407.

THIERY C. & SCHERRER B. (2009c). Une approche modifiée de Lambda-Policy Iteration. In *Journées Francophones Planification Décision Apprentissage*, Paris France : UPMC-Paris 6.

TSITSIKLIS J. N. & VAN ROY B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning*, **22**, 59–94.

YU H. & BERTSEKAS D. P. (2009). Convergence Results for Some Temporal Difference Methods Based on Least Squares. *IEEE Trans. Automatic Control*, **54**, 1515–1531.

Annexes

A Preuve du théorème

Nous donnons ici la preuve du Théroème 1, dont nous rappelons d'abord l'énoncé.

Théorème 1 (Borne sur la performance de Policy Iteration approché optimiste)

Soit un ensemble de poids positifs $(\lambda_n)_{n\geq 1}$*, tels que* $\sum_{n\geq 1}\lambda_n = 1$*. Soit une initialisation quelconque* Q_0*. Soit un algorithme itératif qui construit la suite* $(\pi_k, Q_k)_{k\geq 1}$ *de la manière suivante :*

$\pi_{k+1} \leftarrow \text{glouton}(\pi_{k+1})$
$Q_{k+1} \leftarrow \sum_{n\geq 1}\lambda_n (B_{\pi_{k+1}})^n Q_k + \epsilon_{k+1}.$

ϵ_{k+1} *représente l'erreur d'approximation, erreur commise en estimant la fonction de valeur de* π_{k+1}*. Soit* ϵ *une majoration uniforme de l'erreur : pour tout* k*,* $\|\epsilon_k\|_\infty \leq \epsilon$*. Alors*

$$\limsup_{k\to\infty} \|Q^* - Q^{\pi_k}\|_\infty \leq \frac{2\gamma}{(1-\gamma)^2}\epsilon.$$

Preuve

Notations et idée générale de la preuve

Nous noterons
– $b_k = Q_k - B_{\pi_{k+1}}Q_k$ l'erreur de Bellman,
– $d_k = Q^* - (Q_k - \epsilon_k)$ la différence entre la fonction de valeur optimale et l'itéré Q_k (avant erreur),
– $s_k = Q_k - \epsilon_k - Q^{\pi_k}$ la différence entre l'itéré Q_k (avant erreur) et la (vraie) valeur de la politique π_k,
– $\beta = \sum_{n\geq 1}\lambda_n\gamma^n$ (on pourra remarquer que $0 \leq \beta \leq \gamma$).

La distance entre la valeur de la politique optimale et la valeur de la politique courante peut s'écrire de la manière suivante :

$$\begin{aligned}
\|Q^* - Q^{\pi_k}\|_\infty &= \max(Q^* - Q^{\pi_k}) \\
&= \max(Q^* - Q_k + \epsilon_k + Q_k - \epsilon_k - Q^{\pi_k}) \\
&= \max(d_k + s_k) \\
&\leq \max d_k + \max s_k \qquad (13)
\end{aligned}$$

L'idée de la preuve est de calculer des majorations de d_k et de s_k. Comme nous allons le voir dans le détail, les majorations que nous obtiendrons dépendront toutes deux d'une majoration de l'erreur de Bellman b_k, que nous commençons par calculer.

Une borne supérieure sur l'erreur de Bellman b_k

Comme π_{k+1} est la politique gloutonne par rapport à Q_k, on a $B_{\pi_k}Q_k \leq B_{\pi_{k+1}}Q_k$, ce qui nous permet de dire que

$$\begin{aligned}
b_k &= Q_k - B_{\pi_{k+1}}Q_k \\
&= Q_k - B_{\pi_k}Q_k + B_{\pi_k}Q_k - B_{\pi_{k+1}}Q_k \\
&\leq Q_k - B_{\pi_k}Q_k \\
&= (Q_k - \epsilon_k + \epsilon_k) - B_{\pi_k}(Q_k - \epsilon_k + \epsilon_k) \\
&= (Q_k - \epsilon_k) - B_{\pi_k}(Q_k - \epsilon_k) + \epsilon_k - \gamma P_{\pi_k}\epsilon_k \\
&= \sum_{n\geq 1}\lambda_n\left[(B_{\pi_k})^n Q_{k-1}\right] - \sum_{n\geq 1}\lambda_n\left[(B_{\pi_k})^{n+1}Q_{k-1}\right] + (I - \gamma P_{\pi_k})\epsilon_k \\
&= \sum_{n\geq 1}\lambda_n\left[(B_{\pi_k})^n Q_{k-1}\right] - (B_{\pi_k})^{n+1}Q_{k-1}\right] + (I - \gamma P_{\pi_k})\epsilon_k \\
&= \sum_{n\geq 1}\lambda_n(\gamma P_{\pi_k})^n(Q_{k-1} - B_{\pi_k}Q_{k-1}) + (I - \gamma P_{\pi_k})\epsilon_k \\
&= \sum_{n\geq 1}\lambda_n(\gamma P_{\pi_k})^n b_{k-1} + (I - \gamma P_{\pi_k})\epsilon_k.
\end{aligned}$$

En utilisant le fait que P_{π_k} est une matrice stochastique, on en déduit :

$$\max b_k \leq \sum_{n\geq 1}\lambda_n\gamma^n \max b_{k-1} + (1+\gamma)\epsilon = \beta \max b_{k-1} + (1+\gamma)\epsilon.$$

On en déduit par récurrence que

$$\max b_k \leq \sum_{j=0}^{k-1} \beta^j (1+\gamma)\epsilon + \beta^k \max b_0 = \frac{1+\gamma}{1-\beta}\epsilon + O(\gamma^k). \tag{14}$$

Une borne supérieure sur d_k

Etudions à présent le terme d_k et son évolution.

$$\begin{aligned} d_{k+1} &= Q^* - (Q_{k+1} - \epsilon_{k+1}) \\ &= Q^* - \sum_{n\geq 1} \lambda_n (B_{\pi_{k+1}})^n Q_k \\ &= \sum_{n\geq 1} \lambda_n \left[Q^* - (B_{\pi_{k+1}})^n Q_k\right]. \end{aligned} \tag{15}$$

Comme π_{k+1} est la politique gloutonne par rapport à Q_k, on a $B_{\pi^*}Q_k \leq B_{\pi_{k+1}}Q_k$, et donc

$$\begin{aligned} Q^* - (B_{\pi_{k+1}})^n Q_k &= B_{\pi^*}Q^* - B_{\pi^*}Q_k + B_{\pi^*}Q_k - B_{\pi_{k+1}}Q_k + B_{\pi_{k+1}}Q_k - \\ &\quad -(B_{\pi_{k+1}})^2 Q_k + (B_{\pi_{k+1}})^2 Q_k - \ldots + (B_{\pi_{k+1}})^{n-1} Q_k - (B_{\pi_{k+1}})^n Q_k \\ &\leq B_{\pi^*}Q^* - B_{\pi^*}Q_k + \gamma P_{\pi_{k+1}}(Q_k - B_{\pi_{k+1}}Q_k) + \\ &\quad +(\gamma P_{\pi_{k+1}})^2 (Q_k - B_{\pi_{k+1}}Q_k) + \ldots + (\gamma P_{\pi_{k+1}})^{n-1}(Q_k - B_{\pi_{k+1}}Q_k) \\ &= \gamma P_{\pi^*}(Q^* - Q_k) + \\ &\quad + \left[\gamma P_{\pi_{k+1}} + (\gamma P_{\pi_{k+1}})^2 + \ldots + (\gamma P_{\pi_{k+1}})^{n-1}\right](Q_k - B_{\pi_{k+1}}Q_k) \\ &= \gamma P_{\pi^*}(Q^* - (Q_k - \epsilon_k)) - \gamma P_{\pi^*}\epsilon_k + \\ &\quad + \left[\gamma P_{\pi_{k+1}} + (\gamma P_{\pi_{k+1}})^2 + \ldots + (\gamma P_{\pi_{k+1}})^{n-1}\right](Q_k - B_{\pi_{k+1}}Q_k) \\ &= \gamma P_{\pi^*} d_k - \gamma P_{\pi^*}\epsilon_k + \left[\gamma P_{\pi_{k+1}} + (\gamma P_{\pi_{k+1}})^2 + \ldots + (\gamma P_{\pi_{k+1}})^{n-1}\right] b_k. \end{aligned}$$

Comme P_{π^*} et $P_{\pi_{k+1}}$ sont des matrices stochastiques, on en déduit

$$\begin{aligned} \max[Q^* - (B_{\pi_{k+1}})^n Q_k] &\leq \gamma \max d_k + \gamma\epsilon + (\gamma + \gamma^2 + \ldots + \gamma^{n-1}) \max b_k \\ &= \gamma \max d_k + \gamma\epsilon + \frac{\gamma - \gamma^n}{1-\gamma} \max b_k. \end{aligned}$$

En utilisant l'équation (15), on obtient la récurrence suivante sur $\max d_k$:

$$\max d_{k+1} \leq \gamma \max d_k + \gamma\epsilon + \sum_{n\geq 1} \lambda_n \left[\frac{\gamma - \gamma^n}{1-\gamma} \max b_k\right] = \gamma \max d_k + \gamma\epsilon + \frac{\gamma - \beta}{1-\gamma} \max b_k.$$

A l'aide de la majoration de l'erreur de Bellman obtenue précédemment (équation (14)) on en déduit :

$$\max d_{k+1} \leq \gamma \max d_k + \gamma\epsilon + \frac{\gamma - \beta}{(1-\gamma)(1-\beta)}(1+\gamma)\epsilon + O(\gamma^k),$$

ce qui donne, en prenant la limite supérieure,

$$\limsup_{k\to\infty} \max d_k \leq \frac{\gamma}{1-\gamma}\epsilon + \left[\frac{\gamma - \beta}{(1-\gamma)^2(1-\beta)}\right](1+\gamma)\epsilon. \tag{16}$$

Une borne supérieure sur s_k

Considérons maintenant le terme s_k de l'équation (13) :

$$\begin{aligned} s_{k+1} &= Q_{k+1} - \epsilon_{k+1} - Q^{\pi_{k+1}} \\ &= \sum_{n\geq 1} \lambda_n \left[(B_{\pi_{k+1}})^n Q_k\right] - (B_{\pi_{k+1}})^\infty Q_k \\ &= \sum_{n\geq 1} \lambda_n \left[(B_{\pi_{k+1}})^n Q_k - (B_{\pi_{k+1}})^\infty Q_k\right]. \end{aligned} \tag{17}$$

On peut observer que

$$\begin{aligned}(B_{\pi_{k+1}})^n Q_k - (B_{\pi_{k+1}})^\infty Q_k &= (B_{\pi_{k+1}})^n Q_k - (B_{\pi_{k+1}})^{n+1} Q_k + (B_{\pi_{k+1}})^{n+1} Q_k - (B_{\pi_{k+1}})^{n+2} Q_k + \dots \\ &= (\gamma P_{\pi_{k+1}})^n (Q_k - B_{\pi_{k+1}} Q_k) + (\gamma P_{\pi_{k+1}})^{n+1} (Q_k - B_{\pi_{k+1}} Q_k) + \dots \\ &= (\gamma P_{\pi_{k+1}})^n [I + \gamma P_{\pi_{k+1}} + (\gamma P_{\pi_{k+1}})^2 + \dots] b_k.\end{aligned}$$

Comme précédemment, en utilisant le fait que $P_{\pi_{k+1}}$ est une matrice stochastique, on obtient :

$$\max[(B_{\pi_{k+1}})^n Q_k - (B_{\pi_{k+1}})^\infty Q_k] \leq \gamma^n (1 + \gamma + \gamma^2 + \dots) \max b_k = \frac{\gamma^n}{1-\gamma} \max b_k.$$

En utilisant l'équation (17), on en déduit une majoration de $\max s_{k+1}$:

$$\max s_{k+1} \leq \frac{1}{1-\gamma} \left[\sum_{n \geq 1} \lambda_n \gamma^n \max b_k \right] = \frac{\beta}{1-\gamma} \max b_k.$$

A l'aide de la majoration de l'erreur de Bellman (équation (14)) et en prenant la limite supérieure, on a

$$\limsup_{k \to \infty} \max s_k = \frac{\beta}{(1-\gamma)(1-\beta)} (1+\gamma)\epsilon. \tag{18}$$

Conclusion de la preuve

Finalement, revenons à l'équation (13) et utilisons les majorations que nous venons d'obtenir pour d_k (équation (16)) et s_k (équation (18)) :

$$\begin{aligned}\limsup_{k \to \infty} \|Q^* - Q^{\pi_k}\|_\infty &\leq \limsup_{k \to \infty} \max d_k + \limsup_{k \to \infty} \max s_k \\ &= \frac{\gamma}{1-\gamma}\epsilon + \left[\frac{\gamma - \beta}{(1-\gamma)^2(1-\beta)} + \frac{\beta}{(1-\gamma)(1-\beta)} \right] (1+\gamma)\epsilon. \\ &= \frac{\gamma}{1-\gamma}\epsilon + \left[\frac{\gamma - \beta + (1-\gamma)\beta}{(1-\gamma)^2(1-\beta)} \right] (1+\gamma)\epsilon. \\ &= \frac{\gamma}{1-\gamma}\epsilon + \left[\frac{\gamma}{(1-\gamma)^2} \right] (1+\gamma)\epsilon. \\ &= \frac{\gamma(1-\gamma) + \gamma(1+\gamma)}{(1-\gamma)^2}\epsilon \\ &= \frac{2\gamma}{(1-\gamma)^2}\epsilon. \quad \blacksquare\end{aligned}$$