Sur les abstractions et les projections des processus décisionnels de Markov de grande taille

Manel Tagorti

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Sur les abstractions et les projections des processus décisionnels de Markov de grande taille

THÈSE

présentée et soutenue publiquement le 03 Février 2015

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Manel Tagorti

Composition du jury

Président :	Kamel Smaïli	Professeur à l'Université de Lorraine
Rapporteurs :	Olivier Teytaud	Chargé de Recherche, INRIA-SACLAY
	Philippe Preux	Professeur, Université Lille 3
Examinateurs :	Rémi Munos	Directeur de recherche, INRIA Lille Grand Europe
	Raphaël Fonteneau	Chargé de Recherche, FNRS, Université de Liège
	Joerg Hoffmann	Professeur, Université de Saarland
	Bruno Scherrer	Chargé de Recherche, INRIA Nancy Grand Est

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

Je tiens tout d'abord à remercier Bruno Scherrer qui m'a encadré durant ces trois années. Son implication et ses conseils m'ont été d'une aide précieuse. Je remercie également mon directeur de thèse Joerg Hoffmann. Merci à Olivier Buffet pour sa disponibilité et son aide et à François Charpillet qui m'a accueillie dans son équipe.

Je remercie en outre Olivier Teytaud et Philippe Preux qui ont rapporté ce travail, ainsi que Raphaël Fonteneau, Kamel Smaïli et Rémi Munos qui ont accepté de faire partie du jury.

Je remercie tous mes collègues avec qui j'ai passé de très bons moments.

Sommaire

Chapitre 1

Introduction

Ces dernières décennies l'automatisation a commencé à toucher tous les secteurs de la vie quotidienne. En effet plusieurs tâches qui étaient réservées à l'homme sont actuellement traitées par des machines. Nous citerons dans ce cadre les distributeurs automatiques (distributeurs de tickets dans les gares ou des billets dans les banques), les robots industriels ou encore les caisses automatiques dans les grandes enseignes.

Ces machines s'avèrent parfois plus performantes que les humains par leur capacité de mémorisation et leur rapidité de calcul. Le succès de l'automatisation a engendré le besoin de créer des machines encore plus performantes, qui soient autonomes i.e., capables de prendre des décisions dans certaines situations sans une intervention humaine et dotées d'une capacité de raisonnement. L'intelligence artificielle (IA) est la science qui porte sur l'étude et le développement de ce genre de machines. Cette thèse s'inscrit dans ce cadre. Elle s'intéresse aux méthodes qui les rendent plus efficaces. En effet, elle s'attaque à l'un des défis qui s'oppose à plusieurs domaines de l'IA. Cette thèse revisite le problème de la dimensionnalité *"the curse of dimensionnality"* (Bellman) engendré par la taille du système considéré, étudie certaines méthodes de résolution et propose des schémas de calcul pour rendre ces méthodes plus performantes.

Intelligence artificielle

Le terme "intelligence artificielle" peut être défini de plusieurs façons ; ainsi Bellman (1978) la définit comme *l'automatisation d'activités qu'on associe à la pensée humaine, des activités telles que la prise de décision, la résolution des problèmes et l'apprentissage...* ou encore Kurzweil (1990) l'explique comme *l'art d'accomplir des tâches qui nécessitent une intelligence humaine.* Ces deux définitions se rejoignent dans le sens où le but est le même ; créer une machine qui ressemble à l'homme que ce soit par la pensée ou par les actes. L'intelligence artificielle prit effectivement naissance comme science à part entière dans les années 1940-1950 avec l'arrivée des ordinateurs. Elle fait intervenir plusieurs autres sciences telles que les mathématiques, la psychologie, les sciences cognitives ou encore les neurosciences.

La planification

La planification est un sous-domaine de l'intelligence artificielle. Elle décrit la manière dont un *agent* atteint ses objectifs. On désigne par *agent* toute entité (physique ou abstraite) capable d'agir, autonome dans ses décisions et portant une connaissance sur elle même et sur les autres. Cet agent doit décider à chaque pas de temps quelle action prendre. Ce choix dépend de ce qui

a été précédemment réalisé et aura une influence sur les décisions à prendre dans l'avenir. On y distingue deux catégories : la planification classique et la planification probabiliste.

Planification classique

La planification classique décrit les différentes étapes d'évolution d'un agent à partir de l'état initial s_0 supposé connu jusqu'à l'état final G qui est l'objectif à atteindre. L'environnement dans lequel évolue cet agent est statique (i.e., il n'existe pas de facteurs extérieurs qui pourrait influer l'environnement) et totalement observable (l'agent peut observer chaque état). Afin d'atteindre son objectif, l'agent effectue des actions *déterministes* dont il connaît les conséquences. Chaque action effectuée est une fonction *partielle* de l'espace des états. En effet, on peut associer à chaque état une action à exécuter (il existe des cas où une action donnée n'est pas associable à certains états de l'espace). Les préconditions d'une action notées $prec(a)$ décrivent les cas où l'action a peut être exécutée et l'effet d'une action notée $eff(a)$ désigne l'état résultant de a. On résout un problème de planification en le convertissant en un modèle de graphe. Ce graphe est composé d'un ensemble de nœuds qui représentent les états et d'un ensemble d'arcs partant d'un état vers un autre représentant les actions associées à l'état de départ. On peut distinguer deux types de planification :

- La planification avant qui consiste à chercher dans le graphe à partir d'un état initial s_0 l'état qui satisfait les objectifs à atteindre, en utilisant une stratégie donnée.
- La planification arrière qui consiste au contraire à partir de l'état but afin d'identifier de manière rétrograde les étapes qui mènent à cet état but.

Planification probabiliste

A l'instar de la planification classique la planification probabiliste décrit aussi la manière dont un agent se comporte pour réaliser ses objectifs. Mais contrairement à celle-ci l'agent ne peut pas prédire les conséquences des actions qu'il peut effectuer. En d'autres termes partant d'un état fixé et en choisissant une certaine action il peut basculer vers un ensemble d'états. L'agent évolue dans un environnement *stochastique*, il atteint chacun de ces états avec une certaine probabilité. On suppose généralement que l'agent a une certaine connaissance de son environnement. Il a ainsi accès aux probabilités de transition d'un état x vers un autre état y selon une action a. La probabilité d'atteindre un état à l'instant t dépend de son état à l'instant $t-1$. On dit alors que le système satisfait la propriété de *Markov*. Les problèmes de planification probabiliste peuvent être formalisés mathématiquement en utilisant les processus décisionnels de Markov (MDP) qu'on va introduire dans le chapitre 2. Les méthodes de résolution sont généralement celles utilisées dans le cadre des MDP qui sont les algorithmes issus de la programmation dynamique (cf. chapitre 2).

Apprentissage automatique

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle. Il consiste en un ensemble de méthodes d'analyse et de développement permettant d'extraire un certain nombre d'informations en partant d'un ensemble de données. Cette déduction d'informations se fait grâce à un processus d'apprentissage indépendant de toute intervention humaine. Dans l'apprentissage automatique on peut distinguer trois types d'apprentissage : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Apprentissage supervisé

On dispose en apprentissage supervisé de n couples de données $(X_i, Y_i)_{1 \leq i \leq n}$ dites *données d'entraînement*. Les variables X_i sont des variables d'entrée définies dans l'espace $\mathcal{X}$ et les variables Y_i sont des variables de sortie définies dans l'espace $\mathcal{Y}$. Les variables de sortie sont supposées indépendantes et identiquement distribuées. On cherche à construire une fonction

$$f : \cup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{F}$$

pour prédire X et Y–où les couples (X, Y) sont des *données test*–à partir des données d'entraînement. Dans le cas où l'espace $\mathcal{Y} \subseteq \mathbb{R}$, on parle de problème de régression. Sinon si l'ensemble Y est fini on parle de problème de classification. On retrouve ce type d'apprentissage dans plusieurs méthodes telles que la méthode du plus proche voisin ou encore celle des réseaux de neurones.

Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, aucune information n'a été fournie à l'algorithme, c'est à lui de trouver la structure de classification adéquate à l'ensemble des données en entrée. On distingue deux types d'approches dans l'apprentissage non supervisé. Une approche qui consiste à apprendre à l'agent non à travers des classifications explicites mais à travers un système de récompenses pour indiquer l'échec ou la réussite. L'apprentissage par renforcement est un cas de l'apprentissage non supervisé où un agent prend ses décisions en fonction des récompenses perçues. Un second type d'approche est ce qu'on appelle "le clustering" où le but n'est pas de maximiser le gain total mais d'identifer des similarités dans les données fournies. Le clustering final correspond à une certaine classification qu'on avait au départ du système.

Apprentissage par renforcement

L'apprentissage par renforcement est une des approches de l'apprentissage automatique où on vise à construire des agents capables d'apprendre et d'évoluer. On ne peut pas dire à l'agent ce qu'il doit faire mais on peut l'entraîner sur les choix possibles à faire en fonction des récompenses qu'il perçoit. Tout au long de ce processus d'entraînement l'agent doit apprendre à faire un compromis entre ce qu'on appelle l'exploitation et l'exploration : au cours de son processus de recherche l'agent doit apprendre à *exploiter* les données qu'il a acquises de par son expérience mais en même temps apprendre à *explorer* et à expérimenter de nouvelles afin de trouver une stratégie optimale ou au moins trouver une qui s'en rapproche [PDMIA, 2008]. L'apprentissage par renforcement est notamment utilisé dans les cas

— Des jeux : le jeu backgammon par exemple composé de deux joueurs où chacun doit faire avancer ses pions jusqu'à ce que l'un d'entre eux retire complètement ses pions. L'agent apprend les différentes stratégies du jeu en jouant contre un humain ou contre un autre agent.

— Des problèmes de contrôle : comme celui de l'ordonnancement d'un ascenceur. On ne sait pas quelle meilleure stratégie suivre pour avoir un ascenceur rapide. L'agent apprend à renvoyer de bonnes stratégies de contrôle, en s'adaptant aux changements de l'environnement (simulé).

Les problèmes d'apprentissage par renforcement (A/R) les plus étudiés sont ceux qui peuvent être formalisés comme un problème de type MDP. Les méthodes de résolution en (A/R) ont pour objectif de résoudre deux sortes de problèmes : un problème de prédiction, i.e. sous une stratégie donnée on cherche à connaître la somme totale de récompenses cumulées dans chaque état et un

problème de contrôle où on veut identifier la politique optimale qui garantit un gain maximum dans chaque état. Parmi ces méthodes, on cite les méthodes de Monte Carlo et les méthodes de différences temporelles incluant les algorithmes TD(0) [Sutton, 1988], *SARSA* [Watkins, 1989] et *Q-Learning* [Watkins and Dayan, 1992].

Problème

Dans les deux cas de l'apprentissage automatique et de la planification on fait face au problème de la dimensionnalité de l'espace. En effet lorsque la taille de l'espace considéré augmente il n'est plus possible d'explorer ni d'exploiter par les méthodes dont on dispose les informations que fournit le système. Ce problème a été identifié pour la première fois–expérimentalement–dans les programmes de traduction automatique. Les algorithmes proposés montraient certaines limites à résoudre les problèmes de l'explosion combinatoire (i.e. difficulté de résoudre à un problème suite à une légère modification des données de base), et ce par manque de mémoire et de puissance de calcul. Bellman l'a repéré en optimisation dynamique lorsque la taille de l'espace mathématique augmente. Dans le cadre des MDP on rencontre également ce problème. En effet les MDP énumèrent le nombre de valeurs que peut prendre un état donné ; prenons l'exemple [Boutilier et al., 1996] d'un système décrit par un ensemble de M variables boolénnes, la taille de l'espace des états correspondant est égale à 2^M. Pour un entier M assez grand, représenter un tel système n'est pas vraiment faisable. On propose dans le paragraphe suivant quelques moyens de résolution.

Approches considérées

Une réponse naturelle serait d'essayer de résoudre d'une façon approximative le problème en utilisant différentes approches :

1. Une première approche consiste à agréger les états d'un système (par exemple les nœuds dans un graphe en planification classique) en des sous-ensembles pour obtenir des modèles de taille plus réduites mais qui sont "équivalents" au modèle d'origine. Cette idée de réduction du modèle tient ses origines de l'automatique théorique [Hartmanis and Stearns, 1966] et est apparue plus récemment dans les travaux de model checking ([Burch et al., 1994], [Lee and Yannakakis, 1992]). Construire cette transformation demeure cependant un problème à étudier. Comment rassembler les différents paramètres définissant le problème ou plus précisément comment définir en termes de récompenses/transitions le nouveau modèle construit ? On abordera ces questions dans le chapitre 3.
2. La deuxième approche consiste à projeter l'espace des états sur un espace de taille plus petite. Une aggégation (ou une abstraction comme on le définira plus tard) est un cas particulier de projection. Dans ce sens on peut considérer la première approche comme un cas particulier de la deuxième. La projection renvoyée est une approximation de la solution qu'on cherche à estimer (une solution peut être par exemple la distance par rapport à un état terminal dans un graphe donné). Il existe dans ce cadre plusieurs méthodes de projection comme la méthode des moindres carrés (dans le cas où l'approximation est linéaire), on peut citer dans ce contexte les algorithmes *Least square temporal difference*LSTD(0) [Bradtke and Barto, 1996] et LSTD(λ) [Boyan, 2002] ou encore la méthode du résidu de Bellman et l'algorithme qui lui correspond *Least square Bellman residual* LSBR.

3. Les techniques de décomposition hiérarchique qui subdivisent un problème en un ensemble de sous-problèmes. Dans un contexte particulier cette subdivision peut aider à identifier les décisions ou actions les plus pertinentes et écarter celles qui le sont moins pour atteindre un objectif précis. Parmi les représentations les plus notables en décomposition hiérarchique on peut citer l'utilisation des *options* [Sutton et al., 1999], *la tâche hiérarchique* [Dietterich, 1999] ou encore *la machine abstraite* [Parr and Russell, 1997]. Dans le cadre de la première représentation utiliser des options permet de diminuer d'une manière exponentielle le problème à résoudre.

Contributions

Cette thèse ne s'est intéressée qu'aux deux premières approches. Dans la première approche on s'est particulièrement penché sur deux cas d'aggrégations : l'aggrégation moyenne– peu étudiée dans la littérature– et l'aggrégation Bounded Parameter MDP décrite dans [Givan et al., 2000].

Une propriété sur laquelle on s'est particulièrement penché est celle de la *monotonie* i.e. vérifier si une aggrégation plus fine qu'une autre induit une moindre erreur. Il semble en effet logique qu'une aggrégation qui se rapproche du modèle initial soit plus conforme à ce dernier. Afin de mettre en œuvre cette idée, nous avons considéré deux aggrégations dont l'une est plus fine que l'autre et nous les avons respectivement comparés à l'aggrégation 0 (modèle initial) dans les deux cas qu'on a décrits précédemment. Nos principales contributions à ce sujet sont les suivantes.

1. Dans le modèle moyen : on montre qu'il n'y a pas en général de monotonie de l'approximation par rapport à l'aggrégation et ce même dans un cas déterministe (un cas où les transitions sont égales à 0 ou 1). Soient α et α' deux aggrégations, considérons la relation d'ordre partiel $\alpha' \succeq \alpha$, ce qui signifie α' est plus fine que α. On peut avoir $E_{\alpha'} \geq E_\alpha$ où $E_.$ correspond à l'erreur induite par l'aggrégation.
2. Dans le BMDP : on montre que dans le cas déterministe, il y a toujours monotonie de l'approximation par rapport à l'aggrégation. Cette propriété n'est plus en général vraie dans le cas probabiliste.

Selon les situations nous présenterons des preuves ou des contre-exemples pour étayer nos arguments.

En ce qui concerne la deuxième approche on s'est particulièrement intéressé à l'algorithme LSTD(λ). On analyse la vitesse de convergence de cet algorithme grâce à des outils mathématiques qu'on introduira dans le chapitre 5. La borne d'erreur déduite dépend du nombre d'échantillons générés. Elle rend compte du rôle que joue le paramètre λ dans la convergence de l'algorithme.

Cette étude sera étendue par la suite au cas où on applique un schéma d'itérations sur les politiques stationnaires et non stationnaires à l'algorithme LSTD(λ). On propose dans le chapitre 6 une borne de performance pour les algorithmes *Least square stationary policy iteration* LSPI(λ) [Lagoudakis and Parr, 2003] et *Least square non stationary policy iteration* LS(λ)NSPI [Scherrer and Lesner, 2012].

Table des figures

2

Processus de décision Markoviens et méthodes de résolution

Notre travail se situe dans le cadre de la prise de décision dans l'incertain où un agent interagit avec un environnement stochastique. On s'intéresse à la notion de stratégie optimale afin de minimiser (maximiser) le coût total (la récompense) cumulé le long de son parcours. Une formalisation mathématique de ce problème est possible en introduisant les processus décisionnels de Markov (MDP) [Sutton and Barto, 1998] : ce processus reflète la dynamique du système en attribuant des probabilités de transition d'un état vers un autre sachant une action a. L'état du système à l'instant t ne dépendra que de son état à l'instant $t-1$, on dira alors qu'il satisfait la propriété de Markov.

Nous allons commencer par définir dans ce chapitre de manière formelle les MDP puis nous allons introduire de manière générale les méthodes de résolution de ces processus. On parlera en premier lieu des algorithmes issus de la programmation dynamique et de la programmation linéaire. Puis on présentera les principales méthodes utilisées en planification typiquement les algorithmes de la recherche heuristique. Ces algorithmes reposent sur l'utilisation des heuristiques– une préévaluation de la solution optimale (on donnera par la suite une définition plus détaillée). Il existe plusieurs manières de calculer les heuristiques, on ne considérera ceci dit que celles déduites à partir des abstractions. Ces abstractions influent sur la qualité de l'heuristique. On donnera des définitions plus précises des ces transformations ainsi que des exemples tirés de la littérature dans la deuxième partie de ce chapitre.

2.1 MDP : Définition

Le problème formalisé par les processus décisionels de Markov peut être défini par le quintuplet $\langle S, A, R, p, \gamma \rangle$:

- L'ensembe S représente l'ensemble des états possibles (supposé fini).
- L'ensemble A représente l'ensemble des actions possibles (supposé fini).
- La fonction $R : S \times A \to \mathbb{R}$ est la récompense perçue étant dans un état s donné et sachant qu'on a effectué l'action a.
- La fonction $p : S \times A \times S \to [0, 1]$ est la probabilité de transition d'un état s à un état s' sous l'action a.
- Le paramètre $\gamma \in (0, 1]$ est le facteur d'actualisation : il reflète le degré d'importance des récompenses perçues au cours du temps, pour $\gamma = 1$ l'agent ne perçoit aucune différence.

Notre objectif est d'identifier une séquence d'actions– sachant un état initial s–qui maximise l'espérance de la somme totale des gains cumulés,

$$\mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\middle|s_0=s\right].$$

Ceci consiste à trouver la stratégie optimale qui induit un coût (gain) minimum (maximum). On s'intéressera dans ce qui suit aux concepts fondamentaux et à plusieurs algorithmes pour résoudre le problème formalisé par les MDP.

2.2 Solution d'un MDP

2.2.1 Notion de politique

On appelle la séquence d'actions qu'effectue un agent une politique, que l'on note π. C'est une fonction qui dépend des espaces S, A et de l'espace temps T, on a $\pi : S \times T \times A \to [0,1]$ avec $\pi(s,t,a)$ la probabilité de choisir l'action a au temps t, sachant qu'on est dans l'état s. Il existe plusieurs types de politiques :

- Une politique π est dite déterministe si pour chaque état s on associe une seule action a : $\pi : S \times T \to A$.
- Une politique π est dite stationnaire si elle ne varie pas au cours du temps : pour deux instants t_1 et t_2 donnés on a $\pi(s,t_1,a) = \pi(s,t_2,a)$, sinon elle est dite non stationnaire.

On ne s'intéressera à ce dernier cas que dans le chapitre 6 de ce manuscrit sinon π désignera dans tout ce qui suit une politique déterministe et stationnaire. En effet on sait qu'il existe une politique (stratégie) optimale déterministe et stationnaire qui résout le MDP [Puterman, 1994].

2.2.2 La fonction de valeur

Un agent suit une certaine politique π afin d'optimiser la somme totale des récompenses perçues. On note

$$v^{\pi}(s) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\middle|s_0=s\right]$$

l'espérance de la somme totale des récompenses (moyennant le facteur γ^t) sachant un état initial s_0 et une politique π. La fonction $v^{\pi} : S \to \mathbb{R}$ est la fonction de valeur associée à l'état s sous une certaine politique π. Nous allons rappeler dans ce qui suit les différentes étapes à suivre pour estimer v^{π} [Puterman, 1994]. Dans le cas où π est une politique déterministe, on a

$$\begin{aligned} v^{\pi}(s) &= \mathbb{E}^{\pi}[r(s_0,a_0,s_1)|s_0=s] + \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\middle|s_0=s\right] \\ &= \sum_{s'}p(s,\pi(s),s')r(s,\pi(s),s') + \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\middle|s_0=s\right]. \end{aligned}$$

En utilisant la propriété de Markov il s'avère que :

$$v^{\pi}(s) = \sum_{s'}p(s,\pi(s),s')r(s,\pi(s),s') + \sum_{s'\in S}p(s,\pi(s),s')\mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty}\gamma^{t}r(s_t,a_t,s_{t+1})\middle|s_1=s'\right].$$

En posant $u = t - 1$ on trouve que

$$\begin{aligned} v^\pi(s) &= \sum_{s'} p(s,\pi(s),s')r(s,\pi(s),s') + \sum_{s'\in S} p(s,\pi(s),s')\mathbb{E}^\pi\left[\sum_{u=0}^{\infty}\gamma^{u+1}r(s_{u+1},a_{u+1},s_{u+2})\,\middle|\, s_1 = s'\right] \\ &= \sum_{s'} p(s,\pi(s),s')r(s,\pi(s),s') + \gamma\sum_{s'\in S} p(s,\pi(s),s')v^\pi(s'). \end{aligned}$$

On note $R : S \times A \to \mathbb{R}$ la fonction récompense définie par $R(s,a) = \sum_{s'} p(s,a,s')r(s,a,s')$. On a alors

$$v^\pi(s) = R(s,a) + \gamma\sum_{s'\in S} p(s,\pi(s),s')v^\pi(s').$$

C'est ce qu'on appelle l'équation de Bellman. La fonction de valeur optimale correspond à $v^*(s) = \max_\pi v^\pi(s)$. On dit qu'une politique π^* est optimale si pour tout $s \in S$, $\pi^*(s) \in \arg\max\ v^*(s)$.

Formulation matricielle :

La fonction de valeur v^π peut être également considérée comme un vecteur défini par $v^\pi : S \to \mathbb{R}^N$ où $N = |S|$. En effet pour chaque politique π fixée on peut définir la matrice de transition P^π telle que pour tout $i,j \in \{1,...,N\}$, $P^\pi_{i,j} = p(i,\pi(i),j)$. De même on peut définir R^π le vecteur récompense tel que pour tout $i \in \{1,...,n\}$, $R^\pi(i) = R(i,\pi(i))$. Le vecteur v^π vérifie alors

$$v^\pi = R^\pi + \gamma P^\pi v^\pi.$$

La matrice P est stochastique donc elle possède des valeurs propres de module inférieur à 1. Pour des valeurs de $\gamma < 1$,on a alors $(I - \gamma P)$ est inversible. Il s'en suit que

$$v^\pi = (I - \gamma P^\pi)^{-1}v^\pi.$$

2.2.3 Opérateurs de Bellman

Pour chaque politique π, on note T^π l'opérateur de Bellman : $\mathbb{R}^N \to \mathbb{R}^N$ où $N = |S|$, qui associe à chaque vecteur u de $\mathbb{R}^N$ la fonction

$$T^\pi u(s) = R(s,\pi(s)) + \gamma\sum_{s'\in S} p(s,\pi(s),s')u(s'),\ \text{pour tout } s \in S.$$

L'opérateur T^π est monotone c'est-à-dire que pour tout couple de vecteurs $(w,)u$ dans $\mathbb{R}^N \times \mathbb{R}^N$ tels que $w \le u$, $T^\pi w(s) \le T^\pi u(s)$. L'opérateur T^π est $(\gamma, \|.\|_\infty)$-contractant [Puterman, 1994]. En effet, on a pour tout couple de vecteurs (w,u) dans $\mathbb{R}^N \times \mathbb{R}^N$

$$\begin{aligned} \|T^\pi w - T^\pi u\|_\infty &\le \|\gamma\sum_{s'\in S} p(s,\pi(s),s')(w(s') - u(s'))\|_\infty \\ &\le \gamma\|w - u\|_\infty. \end{aligned}$$

On note T l'opérateur de Bellman : $\mathbb{R}^N \to \mathbb{R}^N$ vérifiant

$$Tu(s) = \max_\pi\left(R(s,\pi(s)) + \gamma\sum_{s'\in S} p(s,\pi(s),s')u(s')\right).$$

L'opérateur T a les mêmes propriétés que l'opérateur T^π. En effet T est monotone. On a pour tout couple de vecteurs (u, w) dans $\mathbb{R}^N \times \mathbb{R}^N$ tels que $w \leq u$

$$\begin{aligned} Tw(s) - Tu(s) =& \max_\pi \left(R(s, \pi(s)) + \gamma \sum_{s' \in S} p(s, \pi(s), s') w(s') \right) - \\ & \max_\pi \left(R(s, \pi(s)) + \gamma \sum_{s' \in S^\pi} p(s, \pi(s), s') u(s') \right) \\ \leq & \max_\pi \gamma \sum_{s' \in S} p(s, \pi(s), s')(w(s') - u(s)) \\ \leq & 0. \end{aligned}$$

T est $(\gamma, \|.\|_\infty)$-contractant. Pour tout couple de vecteurs (u, w) dans $\mathbb{R}^N \times \mathbb{R}^N$, on a

$$\begin{aligned} \|Tu - Tw\|_\infty &\leq \max_\pi \gamma \sum_{s' \in S} p(s, \pi(s), s') \|u - w\|_\infty \\ &= \gamma \|u - w\|_\infty. \end{aligned}$$

La fonction de valeur optimale est solution de l'équation $v = Tv$, qu'on appelle aussi l'équation de Bellman.

2.2.4 Existence d'une solution

La fonction de valeur v^π est solution de l'équation $v = T^\pi v$. En effet l'opérateur de Bellman est $(\gamma, \|.\|_\infty)$-contractant donc l'équation $v = T^\pi v$ admet une unique solution. On va rappeler dans cette partie les principaux arguments de preuve justifiant ce résultat. Soit pour $k \geq 0$ la suite de valeurs, telle que pour $k = 0$ on a v_0 une valeur arbitraire dans $\mathbb{R}$ et pour $k \geq 1$, $v_k = Tv_{k-1}$. Puisque $\|P\|_\infty = 1$ il s'en suit que

$$\begin{aligned} \|v_{k+1} - v_k\|_\infty &= \|T^\pi v_k - T^\pi v_{k-1}\|_\infty \\ &= \|\gamma P_\pi (v_k - v_{k-1})\|_\infty \\ &\leq \gamma \|v_k - v_{k-1}\|_\infty. \end{aligned}$$

En itérant ce procédé k-fois on obtient

$$\|v_{k+1} - v_k\|_\infty \leq \gamma^k \|v_1 - v_0\|_\infty.$$

Le terme $\|v_1 - v_0\|_2$ est borné. Pour $\gamma < 1$, pour tout $\epsilon > 0$ il existe un entier $N \in \mathbb{N}$ qui vérifie

$$\forall k > N,\ \forall r \in \mathbb{N},\ \|v_{k+r} - v_k\|_\infty < \epsilon.$$

La suite $(v_k)_{k \geq 0}$ est une suite de cauchy dans l'espace de Banach $(\mathbb{R}, \|.\|_\infty)$ donc elle converge vers v^π point fixe de l'équation $T^\pi v = v$. La même preuve peut être reprise pour montrer l'existence d'une unique solution de l'équation $Tv = v$.

Au lieu de considérer la fonction de valeur qui à chaque état associe la valeur associée, on peut considérer la fonction de valeur état-action qui fait correspondre à chaque couple d'état-action la valeur qui lui est associée. C'est ce qu'on va expliquer dans la suite.

2.2.5 La fonction de valeur état-action

La fonction de valeur état-action $Q^\pi : S \times A \to \mathbb{R}$ est la somme totale des récompenses cumulées en suivant une certaine politique π sachant un état intial $s_0 = s$ et une première action $a_0 = a$. Elle vérifie

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \,\middle|\, s_0 = s, a_0 = a \right] \\ &= \mathbb{E}[r_1 | s_0 = s, a_0 = a] + \mathbb{E}^\pi \left[\sum_{t=2}^{\infty} \gamma^{t-1} r_t \,\middle|\, s_0 = s, a_0 = a \right] \\ &= \sum_{s'} p(s,a,s') r(s,a,s') + \gamma \sum_{s'} p(s,a,s) \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} \gamma^t r_{t+1} \,\middle|\, s_1 = s', a_1 = \pi(s') \right] \\ &= R(s,a) + \gamma \sum_{s'} p(s,a,s') Q^\pi(s, \pi(s')). \end{aligned}$$

Si π^* appartient à l'ensemble des politiques optimales alors on a

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

Soit pour tout état s' dans S, l'action a' qui vérifie $\pi(s') = a'$, on a

$$Q^*(s,a) = R(s,a) + \gamma \max_{a'} \sum_{s'} p(s,a,s') Q^*(s,a').$$

Formulation matricielle

La fonction Q^π peut être considérée comme un matrice de taille $N \times m$, où $m = |A|$. Pour tout couple $(i,j) \in \{1,...,N\} \times \{1,...,m\}$, on a $Q^\pi_{i,j} = Q^\pi(i,j)$, noter ici qu'on associe à chaque action de A un entier dans $\{1,...,m\}$.

Une fois qu'on a introduit les principaux concepts et définitions concernant les MDP, nous allons présenter dans ce qui suit les principaux algorithmes de résolution utilisés.

2.3 Programmation Linéaire

La résolution d'un MDP peut être considérée comme celle de type programmation linéaire (PL). Un problème de programmation linéaire (PL) consiste à trouver le vecteur x qui maximise la somme $c^t x$ tel que $Ax \leq b$ et $x \geq 0$. La matrice A et les vecteurs b et c sont supposés connus. Dans le cadre des MDP ceci correspond à résoudre un système du type

$$\begin{aligned} &\text{maximiser} \;\; 1^t v \\ &\text{sous la contrainte} \;\; (I - \gamma P_a) v \leq R_a, \forall a \in A \end{aligned}$$

où $c = 1$ (1 est le vecteur unitaire), $b = R_a$ et $A = (I - \gamma P_a)$. On obtient ainsi un problème de programmation linéaire avec N variables et $N \times m$ contraintes avec $m = |A|$. Pour $\gamma < 1$, la solution à ce problème est le vecteur de fonction de valeur optimal v^*.

2.4 Programmation dynamique

Les algorithmes issus de la programmation dynamique (PD) sont des méthodes de résolution qui sont basées sur une formule récursive où à chaque itération la solution est construite à partir de celle calculée à l'étape précédente. Ils nécessitent une initialisation qui peut être arbitraire et qui n'a pas d'influence sur la convergence de l'algorithme. Dans ce qui suit on va introduire les principaux algorithmes issus de la (PD).

2.4.1 Algorithme itérations sur les valeurs

L'algorithme itérations sur les valeurs (IV) [Bellman, 1957] calcule une solution au système composé par les équations de Bellman à travers une suite d'itérations. L'idée étant d'approcher v^* par une suite v_n qui converge après un certain nombre d'itérations n vers v^*. L'algorithme est initialisé à une valeur $v = v_0$ qui est arbitraire et à chaque itération la valeur v_n est remplacée par Tv_{n-1}.

$$v_n \leftarrow \max_{\pi} T_\pi v_{n-1}.$$

L'algorithme requiert généralement un nombre infini d'itérations. Il existe ceci dit des cas où

Algorithme 1 Algorithme itérations sur les valeurs

Initialiser v arbitrairement pour chaque état $s \in S$
Répéter
  $\Delta \leftarrow 0$
  $u \leftarrow v$
  $v \leftarrow \max_\pi T_\pi u$
  $\Delta \leftarrow \max(\Delta, \|u - v\|_\infty)$
jusqu'à $\Delta < \theta$

on a besoin que d'un nombre fini (le cas des (SSP) déterministe, voir définition 1 ci-dessous). On peut choisir d'arrêter l'algorithme lorque la différence entre les valeurs successives v_n et v_{n-1} devient plus petite qu'un certain réel θ, qu'on peut prendre arbitrairement petit. On a dans ce cas [Williams and Baird, 1993]

$$\|v^* - v^\pi\|_\infty \leq \frac{2\gamma}{1-\gamma}\theta.$$

2.4.2 Algorithme itérations sur les politiques

L'algorithme itérations sur les politiques (IP) [Howard, 1960] calcule également une solution au système imposé par les équations de Bellman. Chaque itération estime cette solution en deux étapes : une étape d'évaluation de la politique et une étape de l'amélioration de la politique.

Évaluation de la politique

Cette étape consiste à évaluer la fonction de valeur v^π pour une politique fixée π. Le vecteur v^π vérifie

$$v^\pi = (I - \gamma P_\pi)^{-1} R_\pi.$$

Lorque l'espace des états S est grand, on peut de manière alternative déterminer v^π en utilisant la forme récursive $v_n = T_\pi v_{n-1}$. On a dans ce cas

$$v^\pi = \lim_{n\to\infty} T^n v_0$$

où v_0 est une valeur initiale quelconque.

Amélioration de la politique

Une fois qu'on a déterminé la fonction de valeur v^π pour une politique fixée donnée, on cherche à voir s'il existe une politique π' dite gloutonne qui satisfait

$$\pi' \leftarrow \arg\max_\pi T_\pi v^\pi.$$

Dans le cas où $\pi = \pi'$, π est la politique optimale et donc v^π correspond à v^*. Cette étape est dite étape de l'amélioration de la politique. En regroupant les deux étapes ensemble on obtient l'algorithme itérations sur les politiques décrit ci-dessous. Contrairement à l'algorithme itérations

Algorithme 2 Algorithme itérations sur les politiques

Initialiser v_0
Répéter
  $\pi_{k+1} \leftarrow glouton(v_k)$
  $v_{k+1} \leftarrow v^{\pi_{k+1}}$
  $k \leftarrow k+1$
jusqu'à $\pi_k = \pi_{k+1}$

sur les valeurs, l'algorithme itérations sur les politiques nécessite en plus un étape où on doit évaluer la politique courante π_k à chaque itération. Ceci dit (IP) converge vers la politique optimale en un nombre fini d'itérations.

Il existe des similarités entre la méthode de résolution en programmation linéaire et celle des approches itératives. En effet on peut montrer que la méthode du simplexe (en programmation linéaire) utilisant *la règle de pivot de Dantzig* est équivalente à l'algorithme itérations sur les politiques [Puterman, 1994].

Aussi tous ces algorithmes requièrent une mémoire qui est proportionnelle au nombre d'états N. Ils renvoient la solution optimale en un temps polynomial par rapport aux entiers N, m, B (qui est le nombre de bits nécessaires pour coder les paramètres du MDP en des nombres réels) [Littman et al., 1995].

Il est possible de rendre ces approches plus rapides en les initialisant avec une fonction v_0 qui se rapproche de v^*. On appelle plus communément v_0 *une heuristique*. On va donner dans ce qui suit une définition plus précise de cette notion. On va ainsi présenter les algorithmes de la recherche heuristique et introduire l'une des méthodes de calcul de ces heuristiques à savoir *les abstractions*.

2.5 Recherche heuristique

Nous nous intéresserons dans cette partie à un cas particulier de MDP qui est le *stochastic shortest path*(SSP)MDP. Nous introduisons dans ce paragraphe une définition plus formelle de ces processus.

2.5.1 Stochastic shortest path (SSP)MDP

Les (SSP)MDP ont une caractérisation assez similaire à celle des MDP présentés auparavant. Ils se distinguent toutefois des MDP par l'existence de ce qu'on appelle des états terminaux ou états but :

Définition 1. *[Mausam and Kolobov, 2012] Un (SSP)MDP est défini par le quadruplet $\langle S, A, R, P\rangle$ tel que*

- *S est l'espace des états*
- *A est l'ensemble des actions*
- *$p(s, a, s')$ est la probabilité de transition d'un état s vers un autre état s' sous l'action a*
- *$c : S \times A \in \mathbb{R}$ est la fonction coût qui indique le coût de chaque action dans un état donné*
- *$\mathcal{G} \subset \mathcal{S}$ est l'ensemble des états terminaux (but) tels que pour toute action $a \in A$ pour tout $s \in \mathcal{G}$, $s' \notin \mathcal{G}$, $p(s, a, s) = 1$, $p(s, a, s') = 0$ et $C(s, a, s) = 0$.*

Un (SSP) MDP suppose l'existence d'au moins une politique π_0 dite propre. Une politique est dite propre si on atteint sous cette politique, avec probabilité 1, l'ensemble des états $\mathcal{G}$ en un temps fini. La politique optimale π^*–qu'on cherche à évaluer– va garantir au moins un coût total inférieur à celui induit sous π_0.

On notera que résoudre un (SSP)MDP revient à minimiser le coût total cumulé et non à maximiser la somme des récompenses cumulées, comme on l'avait présenté dans le chapitre précédent mais fondamentalement il s'agit du même problème qu'on résout. Les méthodes de résolution des SSP(MDP) sont similaires à celles de résolution des MDP. Lorsque le nombre d'états est très grand il devient difficile de résoudre ces processus. Une façon de surmonter ce problème est d'utiliser ce qu'on appelle les heuristiques.

2.5.2 Heuristique : définition

Une heuristique peut être perçue comme une méthode approximative de résolution lorsqu'on dispose de peu d'informations sur le problème à traiter. On a généralement aucune garantie en utilisant cette méthode de la qualité de la solution renvoyée. Dans le contexte qu'on étudie une heuristique est une approximation du coût total cumulé sachant un état initial s_0 et un état terminal appartenant à $\mathcal{G}$.

2.5.3 Caractéristiques d'une heuristique

Les heuristiques peuvent avoir plusieurs propriétés.

Définition 2. *On dit qu'une heuristique h est <u>admissible</u> si pour tout état $s \in S$, $h(s) \leq h^*(s)$, où $h^*(s)$ est la solution optimale qu'on cherche à estimer. Une heuristique h est dite parfaite si $h = h^*$.*

Soit a une action donnée et soit $c(s, a, s')$ le coût induit du passage de s vers s', on dit que la fonction h est <u>monotone</u> si on a $h(s) \leq c(s, a, s') + h(s')$. Une heuristique monotone est forcément admissible mais la réciproque est fausse.

Etant données deux heuristiques admissibles h_1 et h_2, on dit que h_2 <u>domine</u> h_1 si $h_2(s) \geq h_1(s)$ pour tous les états s dans S.

Parmi les algorithmes les plus notables en planification dans la recherche heuristique on peut citer les algorithmes A*(cas déterministe), LAO*, *Real Time Dynamic Programming* (RTDP)

et *Labelled Real Time Dynamic Programming* (LRTDP). Une "bonne" heuristique aide ces algorithmes à opter dans chaque état pour l'action qui induit un moindre coût. C'est ce qu'on va expliquer dans ce qui suit.

2.5.4 Algorithme A^*

L'algorithme A^* recherche le meilleur chemin entre un état initial donné et l'état terminal (état but). C'est un algorithme qui converge–si l'heuristique h est admissible–vers le chemin optimal, i.e. celui qui renvoie le moindre coût.

Le principe de A^* consiste à explorer en premier lieu les nœuds dont la fonction $f(s) = c(s) + h(s)$ est minimale où $c(s)$ est le coût initial à l'état s et $h(s)$ est l'heuristique estimée à l'état s.

2.5.5 Real Time Dynamic Programming (RTDP)

L'algorithme (RTDP) [Barto et al., 1995] est un algorithme de programmation dynamique qui simule une certaine politique en construisant des trajectoires partant d'un état initial s_0 et se terminant à un état but appartenant à l'ensemble $\mathcal{G}$. Le processus de construction de cette trajectoire s'appelle un "trial". Durant chaque trial on sélectionne la meilleure action a_{best} dans l'état courant s et on met à jour la fonction v_l dans l'état s. Cette procédure (de mise à jour) permet d'améliorer à chaque étape la politique courante pour déterminer la politique optimale. L'heuristique initiale h constitue un guide initial à la recherche de la politique optimale. Une bonne heuristique est une approximation de la fonction de valeur qui nous oriente vers le bon choix des états à mettre à jour. On n'a donc pas besoin en utilisant l'algorithme RTDP de visiter tout l'espace des états S. Un inconvénient de cet algorithme est qu'il ne possède pas de condition d'arrêt, seul le temps–dont on dispose–peut déterminer l'arrêt de l'algorithme. Ceci implique qu'on ignore généralement si l'algorithme renvoie la politique ou la valeur optimale.

Algorithme 3 Real Time Dynamic Programming Algorithm [Mausam and Kolobov, 2012]

$v_l \leftarrow h$
Tant que il y a encore du temps **faire**
  Trial(s_0)
fin « Tant que »
Trial (s_0)
$s \leftarrow s_0$
Tant que $s \notin \mathcal{G}$ **faire**
  $a_{best} \leftarrow \arg\min_{a \in A} R(s,a) + \gamma \sum_{s'} p(s,a,s') v_l(s')$
  $v_l(s) \leftarrow R(s, a_{best}) + \gamma \sum_{s'} p(s, a_{best}, s') v_l(s')$
  $s \leftarrow$ exécuter a_{best} en s
fin « Tant que »
Renvoyer $\pi^*(s_0)$

2.5.6 Labeled Real Time Dynamic Programming (LRTDP)

(L)RTDP fonctionne comme l'algorithme RTDP mais il comprend en plus une méthode pour détecter les états où la fonction de valeur a convergé (à un ϵ près). Ces états sont dans ce cas labellisés comme étant résolus i.e. *solved*. Cette détection se fait grâce à l'algorithme *checksolved*.

Considérons le résidu RES^{v_l} qui vérifie

$$Res^{v_l}(s) = \left| v_l(s) - \left(\min_{a \in A} C(s,a) + \sum_{s' \in S} p(s,a,s') v_l(s') \right) \right|.$$

Pour un état s, *checksolved* vérifie si $RES^{v_l}(s) \geq \epsilon$ ou si le résidu dans l'un de ses successeurs dans le graphe $\mathcal{G}^{v_l}(s')$ est supérieur à ϵ. Si l'état s et ses successeurs ont un résidu inférieur à ϵ, ils sont labellisés dans ce cas comme *solved* sinon, les états dont le résidu est supérieur à ϵ sont mis à jour.

2.5.7 Utilité d'une heuristique

Dans des algorithmes type (L)RTDP on peut voir qu'une "bonne" heuristique peut rapidement aider à trouver la politique optimale sans avoir à visiter tous les états de l'espace et à mettre à jour les fonctions $Q(s,a)$. C'est ce qui distingue ce genre d'algorithmes de (IV) ou (IP) présentés dans le chapitre précédent où il est nécessaire d'explorer tous les états de l'espace et de mettre à jour la fonction $Q(s,a)$ à chaque itération. La question de déterminer automatiquement une heuristique est un problème difficile que nous allons considérer dans ce qui suit.

2.5.8 Comment calculer les heuristiques ?

Plusieurs méthodes de calcul d'heuristiques ont été proposées dans le cas déterministe mais très peu ont été développées dans le cas probabiliste. Il en existe certaines mais qui sont pour la plupart déduites du cadre déterministe [Mausam and Kolobov, 2012]. L'exemple le plus connu est l'heuristique h_{aodet} (all-out determinization heuristic) qui consiste à transformer un (SSP) MDP en une version déterministe. A partir du (SSP)MDP défini par la quadruplet $\langle S, A, C, P \rangle$ on définit un (SSP)MDP déterministe $\langle S, A', C', P' \rangle$, tel que pour toute action $a \in A$ verifiant pour un couple d'états s, s' donnés $p(s,a,s') > 0$, on associe l'action $a' \in A'$ qui vérifie $p'(s,a',s') = 1$ et $C(s,a,s') = C(s,a',s')$. L'heuristique h_{aodet} estime le coût minimum d'atteindre le but dans le nouveau (MDP) déterministe. Dans le cas où l'espace des actions A est grand, cette transformation va générer un espace d'actions A' encore plus grand en plus du problème de la dimension de l'espace des états S. L'estimation de h_{aodet} peut être tout aussi compliquée et peut requérir à son tour l'utilisation d'heuristiques [Mausam and Kolobov, 2012]. Dans ce qui suit nous proposons une façon de générer des heuristiques qui peuvent servir à des algorithmes type (L)RTDP.

2.6 Les abstractions

On va distinguer dans cette partie le cas des abstractions dans la planification classique de celui dans la planification probabiliste. On donnera par la suite la définition formelle de l'abstraction, qu'on convient de suivre dans ce manuscrit.

2.6.1 Planification classique

En planification classique une abstraction α est une transformation qui relie un état s dans S à un état bloc $[s]_\alpha$. Le bloc $[s]_\alpha$ désigne la classe d'équivalence $\sim_\alpha$; pour tous état s et s' dans $[s]_\alpha$ on a $\alpha(s) = \alpha(s')$. Dans le cas où $[s]_\alpha$ contient un état but de l'espace initial S alors $[s]_\alpha$ est également un état but dans le modèle abstrait. La transition entre les états bloc dépend de

celle dans les états originaux ; s'il existe une transition entre deux états s dans $[s]_\alpha$ et s' dans $[s']_\alpha$ alors il existe une transition entre les deux états blocs sinon il n'existe pas de transition. Le coût minimum $h_\alpha(s)$ d'atteindre le but dans le modèle abstrait est au plus égal au coût optimal original h^* dans le modèle original, i.e. $h^\alpha(s) \leq h^*(s)$. Les abstractions constituent un moyen efficace pour générer des heuristiques admissibles. Parmi les heuristiques déduites à partir des abstractions on peut citer l'heuristique pattern data base [Korf, 1997] (*PDB heuristic*). C'est une heuristique qui est déduite à partir des abstractions qui sont des projections d'états. Les états sont agrégés ensemble uniquement s'ils coïncident sur ces projections qu'on appelle les patterns. On peut citer également l'heuristique merge and shrink [Sabnani et al., 1989] qui est issue du domaine de model checking. Elle consiste à construire d'une manière incrémentale une abstraction (bi)similaire au modèle original (deux états sont bisimilaires s'ils concordent sur les transitions et sur le coût induit, voir la définition 4). Le terme incrémental signifie qu'on n'a pas besoin de construire ou d'expliciter l'espace d'états pour construire l'abstraction. Construire un modèle bisimilaire, même le plus grossier, s'avère ceci dit difficile dans la plupart des cas. Pour cette raison [Dräger et al., 2009] ont proposé les heuristiques *merge and shrink* basées sur ces techniques de regroupement des abstractions (merge) et de rassemblement des états en sous blocs (shrink) sans qu'il y ait de condition sur le modèle final construit (pas nécessairement bisimilaire). Ces heuristiques nécessitent le développement d'une stratégie pour rassembler les abstractions ensemble et aggréger les états dans un même état bloc. Une stratégie d'abstraction serait par exemple de ne plus exiger que l'abstraction soit bisimilaire lorsqu'on atteint un certain nombre d'états.

Dans le paragraphe suivant, nous allons parler de l'abstraction dans la planification probabiliste et plus précisément dans le cadre des MDP.

2.6.2 Planification probabiliste

La technique d'abstraction est assez connue dans la planification probabiliste mais juste dans le cadre des techniques de réduction de modèles. Cette méthode consiste à agréger les états pour obtenir des modèles de taille plus réduite mais qui sont "équivalents" au modèle d'origine. L'équivalence est souvent traduite en termes de politique optimale et de fonction de valeurs. Cette idée de réduction du modèle tient ses origines de l'automatique théorique [Hartmanis and Stearns, 1966] et est parue plus récemment dans les travaux de model checking ([Burch et al., 1994], [Lee and Yannakakis, 1992]). On appelle plus communément la transformation qui relie le modèle initial à son modèle "équivalent" une abstraction. Ainsi les états ne sont agrégés que lorsqu'ils sont équivalents dans un certain sens. Par exemple deux états s_1 et s_2 seront agrégés s'ils définissent la même fonction d'action valeur i.e. pour toute politique π et pour toute action a on a $Q^\pi(s_1, a) = Q^\pi(s_2, a)$ ou s'ils partagent la même action a optimale $Q^*(s_1, a^*) = \max_a Q(s_1, a)$ et $Q^*(s_2, a^*) = \max_a Q(s_2, a)$ [Li et al., 2006]. Ces définitions supposent déjà qu'on ait une certaine connaissance du modèle : connaître Q et a^*. Ceci impose qu'on ait déjà résolu au moins partiellement le problème alors qu'on aimerait trouver une abstraction sans qu'on ait à expliciter entièrement le modèle ou à le résoudre. Une première réponse à ce problème consiste à essayer d'appliquer les techniques de merge et shrink au cas probabiliste dans le sens où on construit d'une manière incrémentale l'abstraction. Une abstraction désignera alors dans ce manuscrit n'importe quelle transformation qui associe un modèle initial à un autre de taille plus réduite. On ne s'intéressera ceci dit principalement qu'aux abstractions d'états (il existe d'autres types d'abstractions telles que les abstractions d'actions). Dans ce qui suit on donne une définition plus formelle de l'abstraction.

Définition 3. *Soit M un MDP défini par $M = \langle S, A, R, P\rangle$ et soit $M_\alpha = \langle S_\alpha, A_\alpha, R_\alpha, P_\alpha\rangle$ son image par la transformation α. On a*

- *S_α est l'ensemble des états blocs $[s]_\alpha$ tel que $|S_\alpha| < |S|$. Le bloc $[s]_\alpha$ définit une classe d'équivalence $\sim_\alpha$, pour tous $s, s' \in [s]_\alpha$ on a $\alpha(s) = \alpha(s')$.*
- *A_α est l'ensemble des actions. On suppose ici $A_\alpha = A$.*
- *P_α est la transition d'un état bloc $[s]_\alpha$ à un autre état bloc $[s']_\alpha$, pour tout a dans A, $p([s]_\alpha, a, [s']_\alpha)$ désigne la probabilité de transition de l'état bloc $[s]_\alpha$ vers l'état bloc $[s']_\alpha$ sachant l'action a.*
- *R_α désigne la fonction récompense $R_\alpha : (S_\alpha, A) \to ([s]_\alpha, a)$, on a pour toute action a dans A et pour tout état s dans S, $R([s]_\alpha, a) = \sum_{[s']_\alpha \in S_\alpha} R_\alpha([s]_\alpha, a, [s']_\alpha) p([s]_\alpha, a, [s']_\alpha)$.*

Le cas des *bisimulations* est un exemple d'abstractions.

Bisimulation

Une bisimulation est une relation de *congruence*. Une relation de congruence est définie comme étant une relation d'équivalence (satisfaisant les propriétés de réflexivité, de symétrie et de transitivité) dont les opérations algébriques (somme, produit,...) restent bien définies dans les classes d'équivalence considérées. La notion de bisimulation a été introduite pour la première fois par Milner [Milner, 1990]. Deux états s et s_1 sont dits équivalents ssi pour chaque action le successeur de l'un est équivalent au successeur de l'autre.

Définition 4. *[Milner, 1990] Soit $\mathcal{F}$ une fonction définie sur l'espace des relations binaires $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{S}$ telle qu'on ait $(s, s_1) \in \mathcal{F}(\mathcal{B})$ ssi pour toute action $a \in A$*

- *Si $s \to_a s'$ alors il existe $s'_1 \in S$ tel que $s_1 \to_a s'_1$ avec $(s', s'_1) \in \mathcal{B}$*
- *Si $s_1 \to_a s'_1$ alors il existe $s' \in S$ tel que $s \to_a s'$ avec $(s', s'_1) \in \mathcal{B}$.*

Une relation binaire $\mathcal{B}$ est une bisimulation forte si $\mathcal{B} \subseteq \mathcal{F}(\mathcal{B})$.

La définition de la bisimulation probabiliste est apparue après avec [Larsen and Skou, 1991] dans le cas des systèmes de transition probabiliste. Givan et al [Givan et al., 2003] ont après étendu cette notion au cadre des systèmes de transition probabilistes admettant des récompenses (typiquement les processus décisionnels de Markov).

Définition 5. *Une bisimulation stochastique est une relation d'équivalence $\mathcal{R}$ sur l'espace des états S qui satisfait*

$$s\mathcal{R}s' \Rightarrow \forall a \in A, R(s, a) = R(s', a) \text{ et } \forall C \in S\backslash R,\ p(s, a, C) = p(s', a, C).$$

On note $\sim$ la plus large bisimulation stochastique.

Il existe plusieurs algorithmes de résolution de MDP utilisant les abstractions. On présente dans cette partie quelques exemples.

Algorithme de raffinement de la partition (Givan et al)

Givan et al [Givan et al., 2003] ont proposé un algorithme qui pourrait éventuellement identifier les bisimulations. Cet algorithme est basé sur une méthode qui consiste à partir d'une partition initiale de l'espace des états et à décomposer à chaque étape la partition en blocs. La décomposition en blocs se fait selon certains critères qui assurent qu'on obtienne à la fin l'abstraction souhaitée, i.e. une bisimulation. A chaque raffinement on cherche à ce que les blocs

soient *stables* les uns par rapport aux autres, si c'est le cas l'algorithme retourne la bisimulation souhaitée sinon il raffine encore l'espace. Un bloc B est dit *stable* par rapport à un autre bloc C si pour chaque état s inclus dans B on a pour toute action $p(s,a,C) = p(s',a,C)$ et $R(s,a) = R(s',a)$. On note la nouvelle partition obtenue $SPLIT(B,C,P)$ telle que chaque sous bloc de B est stable par rapport à C. Cet algorithme calcule la structure d'équivalence recherchée

Algorithme 4 Algorithme de raffinement de la partition

Soit $P = P' =$ La partition induite par S
Pour tout bloc C dans P' **faire**
  Tant que P' contient un bloc B pour lequel $P' \neq SPLIT(B,C,P')$ **faire**
    $P' = SPLIT(B,C,P')$
    $I(S) =$ la relation d'équivalence représentée par P'.
  fin « Tant que »
fin « Pour »

$I(S)$ en $O(|A||S|^3)$ opérations.

Algorithme de raffinement de l'abstraction

La technique d'abstraction proposée par Kattenbelt et al [Kattenbelt et al., 2010] est fondée sur le cas de jeu aléatoire à deux joueurs. Le premier joueur correspond aux choix des paramètres caractérisant le MDP et le deuxième joueur correspond à ceux introduits par l'abstraction. Ils considèrent ainsi deux stratégies π_1 et π_2 celles associées respectivement aux joueurs 1 et 2. Ils déduisent ensuite de cette représentation une borne supérieure $\overline{v}$ et une borne inférieure $\underline{v}$ de la fonction de valeur v^* à l'instar des BMDP qu'on présentera dans le chapitre suivant. L'algorithme *Abstraction Refinement* qu'ils proposent fait intervenir deux méthodes : une méthode de raffinement basée sur la politique et une autre méthode basée sur la fonction de valeur v. Lorsque la condition $\|\overline{v} - \underline{v}\|_\infty < \epsilon$ est vérifiée pour un $\epsilon > 0$, l'algorithme s'arrête. Si une abstraction est initialement plus grossière qu'une bisimulation alors elle le restera même après un raffinement. Deux états bisimilaires ne sont pas séparés en suivant n'importe quelle stratégie. Cet algorithme peut être particulièrement utile dans le cas des MDP qui admettent un grand nombre d'états mais auxquels on peut faire correspondre une bisimulation finie.

Métrique reliée à la bisimulation

La notion de bisimulation est assez restrictive elle requiert que tous les états agrégés admettent une même récompense et les mêmes probabilités de transition selon la partition considérée. L'introduction d'une métrique dite *métrique de bisimulation* [Ferns et al., 2012] permet de rassembler deux états qui se comporteraient approximativement de la même façon. Une métrique d est dite une métrique de bisimulation d_{bis} si la relation d'équivalence $R_{d_{bis}}$ est la plus grossière bisimulation associée.

Définition 6. *Soit d_{bis} est une métrique de bisimulation, on a alors pour tous états s et s' dans S :*

$$d_{bis}(s,s') = 0 \Leftrightarrow \forall a \in A,\ R(s,a) = R(s,a') \text{ et } \forall C \in R_{d_{bis}},\ T_K(d_{bis})(P(s,a,C), P(s',a,C)) = 0.$$

Soit $\mathcal{M}$ l'ensemble des semi-métriques sur S qui assignent au plus une distance égale à 1. On a pour pour $d \in \mathcal{M}$, $T_K(d)$ la métrique de Kantorovich définie par : pour P et Q deux mesures de probabilité

$$T_K(d)(P,Q) = \max_{\nu_C} \sum_{C \in S/R_d} (P(C) - Q(C))\nu_C$$
$$\text{tel que } \forall C, D,\ (\nu_C - \nu_D) \leq \min_{s \in C, t \in D} d(s,t),$$

pour tout C, ν_C est un réel appartenant à $[0,1]$, R_d est la relation d'équivalence obtenue en regroupant les états de distance nulle sous d. De plus on a

$$T_K(d)(P,Q) = 0 \Leftrightarrow P(C) = Q(C),\ \forall C \in S/R_d.$$

Soit $F : \mathcal{M} \to \mathcal{M}$ l'opérateur défini par

$$F(d)(s,s') = \max_{a \in A}(C_R|R(s,a) - R(s',a)| + C_T(T_K(d)(P(s,a,.), P(s',a,.)))$$

avec C_R et C_T deux constantes positives vérifiant $C_R + C_T \leq 1$. Cet opérateur admet un plus petit point fixe qui est une métrique de bisimulation. Cette métrique peut être calculée en $O\left(|A||S|^4 \log(|S|)\frac{\ln(\delta)}{\ln(C_T)}\right)$ pour $\delta \in (0,1)$.

2.7 Conclusion

Nous avons présenté dans ce chapitre les principaux algorithmes de la résolution des MDP. Nous avons aussi introduit les abstractions qui sont des transformations qui associent un modèle de MDP initial à un autre de taille plus réduite. Nous allons étudier plus en détail ces fonctions dans le chapitre 3 en termes de la qualité de l'heuristique qu'elles renvoient.

3

Pathologies dans certaines abstractions des processus décisionnels de Markov

Nous avons introduit dans le chapitre précédent des méthodes de résolution des MDP qui sont les abstractions dans les cadres déterministe et probabiliste. Nous n'allons nous intéresser dans ce chapitre qu'au cadre probabiliste qui est moins évident à traiter, puisqu'il requiert une définition plus précise de l'abstraction considérée. En effet, il faut spécifier dans ce cas les transitions et récompenses associées à chaque état bloc sachant les transitions et récompenses des états qu'ils contiennent. S'il est connu qu'on obtient des heuristiques admissibles dans le cas déterministe en utilisant les abstractions, on ignore ce qu'il en est dans le cas probabiliste. Une abstraction α plus grossière qu'une abstraction α' induit une moins meilleure borne h_α du coût optimal h^* que sa concurrente $h_{\alpha'}$ dans le cas déterministe. On a pour tout s dans S, $h_\alpha(s) \leq h_{\alpha'}(s) \leq h^*(s)$. Est-ce aussi le cas dans le cadre probabiliste ? Si c'est le cas ceci nous permettrait d'avoir un contrôle monotone sur l'erreur générée par le choix de l'heuristique au fil des aggrégations–ceci malgré le caractère stochastique du modèle. Nous nous sommes intéressés à ces questions dans deux types d'abstractions : l'abstraction moyenne et l'abstraction proposée par [Givan et al., 2000]. Nous avons montré qu'il n'y avait pas en général *monotonie*, i.e. une abstraction α' plus fine que α, ne renvoie pas forcément une meilleure heuristique. On parle alors de *pathologie* puisque le résultat intuitif, vrai dans le cas deterministe, ne l'est plus dans le cas stochastique.

3.1 Abstraction Moyenne

Cette abstraction semble a priori très naturelle même si elle a été très peu étudiée dans la littérature. On la trouve dans [Ortner, 2011]. Elle consiste principalement à associer à chaque état bloc dans le modèle abstrait la moyenne des récompenses de tous les états contenus dans ce bloc et la moyenne des transitions partant de ce bloc vers un autre bloc.

Définition 7. *Soit α l'abstraction qui associe le modèle initial M à son modèle moyen $M_\alpha = \langle S_\alpha, A_\alpha, R_\alpha, P_\alpha \rangle$ qui est défini par :*

- *L'espace des états S_α*
- *L'ensemble des actions A_α (on supposera ici $A_\alpha = A$)*
- *La fonction récompense $R_\alpha : S \to \mathbb{R}$; pour tout s dans S, et pour toute action a dans A, $R_\alpha([s]_\alpha, a) = \frac{1}{|[s]_\alpha|} \sum_{s' \in [s]_\alpha} R(s, a)$ est la moyenne des récompenses associées aux états contenus dans le bloc $[s]_\alpha$*
- *La probabilité de transition $p : S_\alpha \times A \times S_\alpha \to [0, 1]$, pour tous s, s' dans S et pour toute*

action a dans A, $p([s]_\alpha, a, [s']_\alpha) = \frac{1}{|S|}\sum_{s_1\in[s]_\alpha}\sum_{s_2\in[s']_\alpha} p(s_1, a, s_2)$ est la moyenne des probabilités d'atteindre le bloc $[s']_\alpha$ sous l'action a en partant du bloc $[s]_\alpha$.

Nous donnons dans ce cadre l'exemple d'un modèle à trois états illustré ci-dessous (voir figure 3.1), où on rassemble les état 1 et 2 dans un même état bloc 12. Le modèle moyen abstrait équivalent correspond au modèle à droite où les récompenses et les transitions sont estimées selon la définition 7.

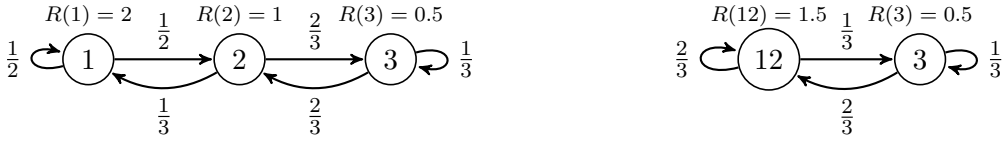

FIGURE 3.1 – De gauche à droite le MDP initial et l'abstraction moyenne correspondante $\alpha : \{1,2,3\} \to \{1,2\}, 3$.

Nous allons essayer maintenant de calculer les fonctions de valeur correspondantes au modèle initial. Ces fonctions sont solutions de l'équation de Bellman $v = Tv$, telles que

$$Tv = R + \gamma P v.$$

La solution vérifie pour $\gamma = 0.5$

$$v(1) = 3.34, \quad v(2) = 2.02, \quad v(3) = 1.4.$$

Les fonctions de valeur correspondantes au modèle moyen vérifient l'équation de Bellman $v = T^\alpha v$ avec

$$T^\alpha v = R^\alpha + \gamma P^\alpha v.$$

Il s'en suit que

$$v_\alpha(12) = 2.66, \text{ et } v_\alpha(3) = 1.66.$$

On remarque que $v_\alpha(12)$ et $v_\alpha(3)$ sont respectivement des bornes supérieures de $v(2)$ et $v(3)$ alors que $v_\alpha(12)$ est une borne inférieure de $v(1)$. La fonction de valeur v_α n'est ni ainsi une borne inférieure ni une borne supérieure de la fonction v. Ce type d'abstractions ne permet donc pas a priori d'avoir de bonnes heuristiques qui puissent assurer la convergence des algorithmes de recherche heuristique vers la solution optimale. Mais peuvent-elles au moins s'en approcher à une erreur ϵ près ? Existe-t-il des critères que doit satisfaire l'abstraction qui garantiraient une meilleure approximation de la fonction de valeur ? Nous avons pensé à la monotonie comme critère de comparaison. Ceci suggère qu'une abstraction α' qui est plus fine qu'une autre abstraction α garantirait une meilleure approximation. Cela semble naturel puisque agrandir l'espace nous rapproche du modèle initial. Cette nouvelle contrainte (contrainte de la monotonie) nous mène à introduire une définition dont on va se servir par la suite.

Définition 8. *Soient α et α' deux abstractions de S vers respectivement S_α et $S_{\alpha'}$, on dit que α' (α) est plus fine (grossière) que α (α') et on note $\alpha' \succeq \alpha$ ($\alpha' \preceq \alpha$) si pour tous états s et s' dans S, $\alpha'(s') = \alpha'(s) \Rightarrow \alpha(s') = \alpha(s)$.*

On dit de plus que α' est un raffinement direct de α s'il existe s et s' tels que $[s]_\alpha = [s']_\alpha$ mais $[s]_{\alpha'} \neq [s']_{\alpha'}$, $[s]_\alpha = [s]_{\alpha'} \cup [s']_{\alpha'}$ et $\alpha'(s) = \alpha(s)$ pour tout $s \in S \backslash [s]_\alpha$.

Nous allons juste considérer dans ce qui suit le cas du raffinement direct car il est évident que si une propriété est vraie à ce stade elle le sera automatiquement dans le cas où on considère un raffinement quelconque. Nous avons remarqué précédemment dans la figure 3.1 que l'heuristique obtenue n'est pas forcément admissible. Dans le cas de l'abstraction moyenne nous n'avons donc pas forcément $v_\alpha(s) \leq v(s)$ pour tout état s. Pour pouvoir comparer les fonctions de valeur les unes par rapport aux autres on a besoin d'introduire une mesure d'erreur. Pour cela on va introduire la distance moyenne qui estime la différence moyenne entre v et v_α dans chaque état.

Définition 9. *On définit E_α l'erreur relative à l'abstraction α telle qu'on ait*

$$E_\alpha = \frac{1}{|S|} \sum_{s \in S} |v(s) - v_\alpha([s]_\alpha)| \,.$$

Dans le modèle moyen on dira qu'une approximation $v_{\alpha'}$ est meilleure qu'une autre v_α si elle vérifie $E_{\alpha'} \leq E_\alpha$. Il reste à voir si une abstraction plus fine induit une plus petite erreur donc renvoie une meilleure heuristique.

On va montrer que même pour un MDP déterministe, il n'y a pas nécessairement monotonie de l'approximation par rapport à l'abstraction. On parlera alors de *pathologie* de l'abstraction. Dans ce qui suit on exhibe justement un cas de modèle moyen déterministe qui satisfait $E_\alpha \leq E_{\alpha'}$ pour α' un raffinement direct de α.

Modèle déficient

Proposition 1. *Il existe un MDP déterministe M, une abstraction α et un raffinement α' de α tels que $E_\alpha < E_{\alpha'}$.*

Démonstration. Soit M le MDP représenté dans la Figure 3.2 avec une seule action $\{a\}$ et un facteur d'actualisation $\gamma = 1$ (le résultat est le même pour $\gamma < 1$ mais proche de 1). Les états

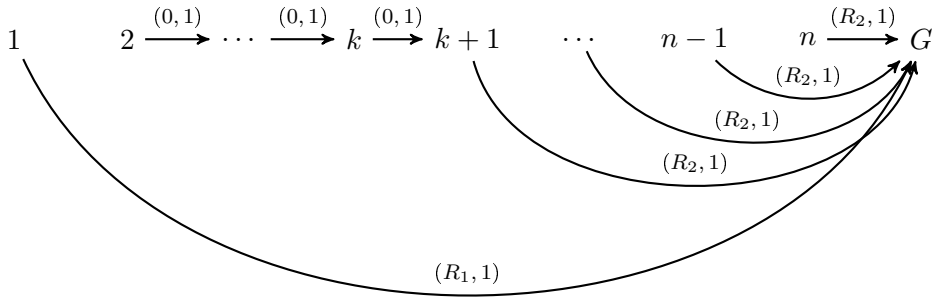

FIGURE 3.2 – Le MDP initial

dans $\{2, ..., k\}$ $(k > 2)$ sont similaires : ils admettent la même récompense $(R = 0)$ et ont la même dynamique ; ils atteignent l'état voisin avec probabilité 1. Les états dans l'ensemble $\{k+1, ..., n\}$ sont aussi similaires : ils atteignent l'état but G avec probabilité 1 et admettent un récompense R_2. L'état 1 admet une récompense $R(1) = R_1 \ll R_2$ et atteint le but avec probabilité 1.

Considérons l'abstraction $\alpha_1 : S \to \{1, ..., k\}, \{k + 1, ..., n\}, G$ (figure 3.3 ci-dessous) où les états 1, $\{2, ..., k\}$ sont dans le même bloc.

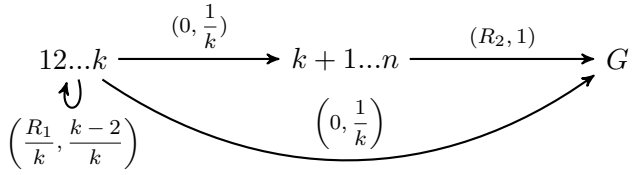

FIGURE 3.3 – L'abstraction α_1

En brisant la similarité qui existait entre les états, il en résulte une erreur E_{α_1} strictement positive. On a

$$\begin{aligned} v_{\alpha_1}(\{k+1,...,n\}) &= \frac{(n-k)R_2}{n-k} = R_2 \text{ et} \\ v_{\alpha_1}(\{1,...,k\}) &= \frac{R_1}{k} + \frac{k-2}{k} v_{\alpha_1}(\{1,...,k\}) + \frac{1}{k} v_{\alpha_1}(\{k+1,...,n\}) \\ &= \frac{R_1+R_2}{2} \neq R_2. \end{aligned}$$

En appliquant la définition de l'erreur moyenne à l'abstraction α_1 on obtient

$$E_{\alpha_1} = \frac{1}{n}\sum_{i=1}^{k} |v(i) - v_{\alpha_1}(\{1,...,k\})| \sim k\frac{R_2}{2n} \text{ for } R_1 \ll R_2.$$

Considérons maintenant l'abstraction $\alpha_2 : S \to \{1,...,n\}, G$ (figure 3.4 ci-dessous)

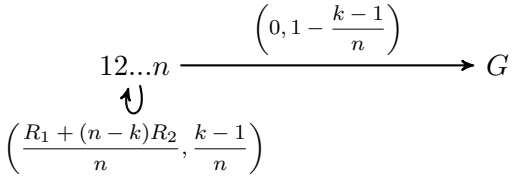

FIGURE 3.4 – L'abstraction α_2

La fonction de valeur $v_{\alpha_2}(\{1,...,n\})$ vérifie

$$\begin{aligned} v_{\alpha_2}(\{1,...,n\}) &= \frac{R_1+(n-k)R_2}{n} + \frac{k-1}{n} v_{\alpha_2}(\{1,...,n\}) \\ &\sim R_2 \text{ pour } k \ll n. \end{aligned}$$

On remarque que $v_{\alpha_2}(\{1,...,n\})$ est approximativement égale à $v_{\alpha_2}(i)$ pour tout $i \in \{2,...,n\}$ donc l'erreur induite résulte de la présence de l'état 1 dans cette agrégation. L'erreur relative à l'abstraction α_2 vérifie alors

$$E_{\alpha_2} \sim \frac{1}{n}|v(1) - v_{\alpha_2}(\{1,...,n\})| \sim \frac{1}{n}R_2.$$

Si on prend un nombre d'états k vérifiant $k \geq 2$, on peut s'apercevoir que $E_{\alpha_1} > E_{\alpha_2}$. □

En se basant sur les similarités qui existent entre les états, il est possible d'exhiber une abstraction α_0 qui vérifie $v_{\alpha_0}([s]_{\alpha_0}) = v(s)$ pour tout s. Cette abstraction vérifie $\alpha_0 : S \rightarrow 1, \{2, ..., k\}, \{k+1, ..., n\}, G$. On remarque alors que

$$\begin{aligned} v_{\alpha_0}(1) &= R_1 = v(1) \\ v_{\alpha_0}(\{k+1, ..., n\}) &= R_2 = v(k+1) = ... = v(n) \\ v_{\alpha_0}(\{2, ..., k\}) &= \frac{k-3}{k-2} v_{\alpha_0}(\{2, ..., k\}) + \frac{1}{k-2} v_{\alpha_0}(\{k+1, ..., n\}) = R_2 = v(2) = ... = v(k). \end{aligned}$$

En appliquant la définition de fonction de valeur v présentée plus haut on a $v(G) = 0$, $v(1) = R_1$, et $v(i) = R_2$ pour i dans $\{2, ..., n\}$.

Le fait que l'abstraction moyenne ne préserve pas le cadre déterministe fait d'elle une mauvaise abstraction même dans le cadre déterministe où plus une abstraction est fine plus on a des garanties sur l'heuristique obtenue. Ceci nous a menés à considérer un deuxième type d'abstraction qui fera l'objet de notre étude pour le reste de ce chapitre.

3.2 Bounded parameter MDPs

Nous allons étudier dans cette partie l'abstraction proposée par [Givan et al., 1997] qui renvoie un MDP initial à ce qu'on appelle un BMDP (bounded parameter MDP). Nous allons commencer par introduire les BMDP.

3.2.1 Définition

Les bounded parameter MDP sont une généralisation des MDP. En effet les BMDP sont un ensemble de MDP qu'on spécifie en associant à chaque couple état-action un ensemble de récompenses et un ensemble de probabilités de transition. Ils représentent un cas particulier d'une classe plus connue qui est celle des MDPIP (MDP avec paramètres imprécis). Ils peuvent être très utiles pour modéliser les variations dans certains modèles où on ne connaît pas précisément les paramètres. Ils sont également utilisés dans le cadre de la réduction des modèles (qui est notre principal intérêt ici) où on part d'un MDP avec des paramètres fixés et on obtient un autre de taille plus réduite. Le nouveau modèle est ce qu'on appelle un BMDP auquel on associe un intervalle de récompenses et de transitions.

Définition 10. *Un BMDP est défini par le quadruplet $\langle S, A, \overline{R}, \overline{P} \rangle$ tel que*

- *S est l'espace des états*
- *A est l'ensemble des actions*
- *$\overline{R} : S \times A \rightarrow \mathbb{R}$ est l'intervalle de récompenses associé au couple (s, a) pour tout s dans S et pour tout a dans A*
- *$\overline{P} : S \times A \rightarrow [0, 1]$ est l'intervalle de probabilités de transition d'un état vers un autre sachant une action donnée.*

Un BMDP $\mathcal{M} = \langle S, A, \overline{R}, \overline{P} \rangle$ peut être considéré comme la famille de MDP $\mathcal{F}$ avec $\mathcal{F} = \{M | \mathcal{M} \models M\}$ où $M = \langle S, A, R, P \rangle$ avec $R \in \overline{R}$ et $P \in \overline{P}$. On donne dans la figure 3.5 l'exemple d'un BMDP $\mathcal{M} = \langle S, A, \overline{R}, \overline{P} \rangle$ et d'un MDP $M = \langle S, A, R, P \rangle$ appartenant à ce BMDP.

La fonction de valeur liée à un BMDP est un intervalle de fonctions de valeurs–et non pas des valeurs fixes–délimité par la borne inférieure v^- et la borne supérieure v^+. Givan et al [Givan et al., 2000] proposent un algorithme qui permet de calculer ces bornes.

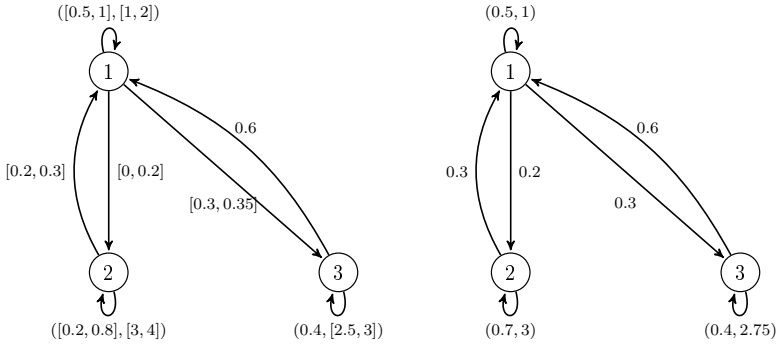

FIGURE 3.5 – Exemple d'un BMDP et d'un MDP appartenant à la famille. Les parenthèses (., .) désignent l'intervalle de transitions (à gauche) et l'intervalle de récompenses (à droite).

3.2.2 Algorithme de Dean et Givan

L'algorithme proposé dans [Givan et al., 2000] introduit une façon de calculer les extremas de la fonction de valeur dans un BMDP. Les récompenses ne posent pas a priori de problème. Pour calculer la borne inférieure (supérieure) il suffit d'assigner le minimum (maximum) de récompenses à chaque couple d'état-action. Ce sont plus les probabilités de transition qui peuvent poser problème puisqu'on doit respecter ici la contrainte pour tout s et pour tout a, $\sum_{s'} p(s,a,s') = 1$ alors que $\sum_{s'} \min p(s,a,s') \leq 1$ et $\sum_{s'} \max p(s,a,s') \geq 1$. Givan et al [Givan et al., 1997] ont proposé une méthode pour estimer v^- et v^+ qu'on développe ici. Cette méthode met en jeu un nouveau concept qui est *l'ordre maximisant* des états. L'ordre dépend de la façon dont on veut "favoriser" les transitions vers les états. Pour mieux expliquer cette idée prenons l'exemple du BMDP dans la figure 3.5. Pour des valeurs de récompenses fixées, si on choisit par exemple de privélégier 1 par rapport à 2 et 2 par rapport à 3 qu'on note : $1 > 2 > 3$, on n'obtient pas le même MDP que lorsqu'on choisit l'ordre $2 > 3 > 1$(voir figure 3.6). D'une manière plus formelle,

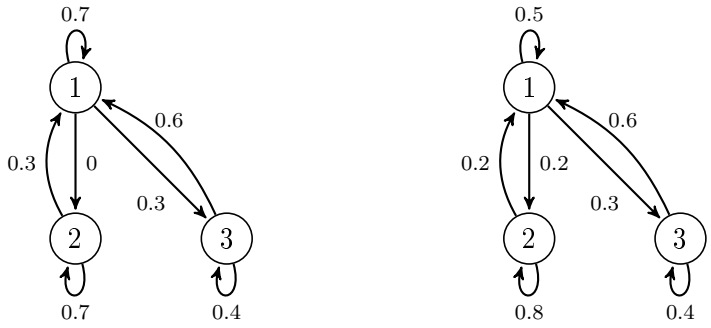

FIGURE 3.6 – Exemple d'un MDP suivant l'ordre $1 > 2 > 3$ (à gauche) et suivant l'ordre $2 > 3 > 1$ à droite.

si on note pour des entiers $q_1, q_2, ..., q_{|S|}$ un ordre donné des états ($q_1 > q_2 > ... > q_{|S|}$) et si on définit un MDP M_o défini selon cet ordre tel que $M_o \in \mathcal{M}$, alors pour chaque action a dans A

et pour chaque état s dans S on a

$$p_o(s,a,q_i) = \begin{cases} \max p_o(s,a,q_i) & \text{if } i < r \\ \min p_o(s,a,q_i) & \text{if } i > r \\ 1 - \sum_{i \neq r} p_o(s,a,q_i) & \text{if } i = r \end{cases} \tag{3.1}$$

où l'entier r vérifie $r = \inf\{l, 1 - \sum_{i=1}^{l} \max p(s,a,q_i) \leq \sum_{i=l+1}^{|S|} \min p(s,a,q_i)\}$ [1].

Définition 11. *Soit $X_{\mathcal{M}}$ l'ensemble des MDP dans $\mathcal{M}$ qui correspondent chacun à un certain ordre maximisant o des états. Si l'ensemble des états S est fini, il existe un nombre fini d'ordre des états par conséquent $\mathcal{X}_{\mathcal{M}}$ est fini.*

Grâce à l'introduction de la notion de l'ordre, on peut calculer les deux bornes extrémales délimitant les fonctions de valeur. C'est ce qui est énoncé dans le lemme suivant.

Lemme 1. *[Givan et al., 2000] Soit M un MDP dans $\mathcal{M}$, pour toute politique π, il existe des MDP $M_1 \in X_{\mathcal{M}}$ et $M_2 \in X_{\mathcal{M}}$ tels que*

$$v^{\pi}_{M_1} \leq v^{\pi}_{M} \leq v^{\pi}_{M_2}.$$

Puisque l'ensemble $X_{\mathcal{M}}$ est fini, il s'en suit qu'il existe v^- et v^+ tels que

$$v^- = \max_{\pi} \min_{M \in \mathcal{M}} v^{\pi}_{M} = \max_{\pi} \min_{M \in X_{\mathcal{M}}} v^{\pi}_{M} = \max_{\pi} (v^{\pi})^- \tag{3.2}$$

$$v^+ = \max_{\pi} \max_{M \in \mathcal{M}} v^{\pi}_{M} = \max_{\pi} \max_{M \in X_{\mathcal{M}}} v^{\pi}_{M} = \max_{\pi} (v^{\pi})^+. \tag{3.3}$$

Démonstration. On peut trouver une preuve de ce lemme dans l'Annexe A. □

Enumérer l'ensemble des MDP qui appartiennent à l'ensemble $\mathcal{M}$ n'est pas faisable, si on veut trouver algorithmiquement ces deux bornes. Introduire l'ensemble $X_{\mathcal{M}}$ nous facilite beaucoup la tâche puisqu'on sait d'après le lemme 1 qu'il existe des MDP qu'on note M^- et M^+ appartenant à $X_{\mathcal{M}}$ correspondants respectivement à v^- et v^+. On va donc se servir du concept de l'ordre qu'on a introduit plus tôt pour les estimer. Ce concept va être pris en considération lors de l'implémentation de l'algorithme de Dean et Givan (cf. algorithme 5). On converge à la fin vers l'ordre maximal (respectivement minimal) correspondant à v^+ (respectivement v^-).

Les procédures de calcul de la borne inférieure et de la borne supérieure sont similaires. On initialise la fonction de valeur $v = v_0$ où v_0 est arbitraire. A chaque itération, on ordonne pour tout $s \in S$ les valeurs $v(s)$ dans l'ordre croissant : $v(q_1) \leq v(q_2) \leq ... \leq v(q_{|S|})$ (décroissant : $v(q_1) \geq v(q_2) \geq ... \geq v(q_{|S|})$) et on associe dans l'expression de l'opérateur de Bellman à chaque valeur les probabilités de transition définies dans l'expression (3.1). Nous allons montrer dans ce qui suit la convergence de cet algorithme vers v^- et v^+.

1. Notons que l'entier r existe car sinon on aurait

$$\forall l \in \{1, ..., |S|-1\},\ 1 - \sum_{i=1}^{l} \max p(s,a,q_i) > \sum_{i=l+1}^{|S|} \min p(s,a,q_i).$$

En particulier en prenant $l = |S| - 1$ on obtient :

$$1 - \sum_{i=1}^{|S|-1} \max p(s,a,q_i) > \min p(s,a,q_{|S|}) = 1 - \max \sum_{i=1}^{|S|-1} p(s,a,q_i).$$

Ce qui n'est pas vrai donc l'entier r existe forcément.

Algorithme 5 Algorithme itérations sur les intervalles de valeur IVI

\\ Supposons que $v_{\updownarrow}$ est représenté par :
\\ $v_{\downarrow}$ vecteur représentant la borne inférieure dans chaque état
\\ $v_{\uparrow}$ vecteur représentant la borne supérieure dans chaque état
\\ Créer un vecteur o qui représente l'ordre dans chaque état :
\\ Le vecteur $o_{\uparrow}$ correspond à l'ordre croissant
\\ Le vecteur $o_{\downarrow}$ correspond à l'ordre décroissant
VI-mettre à jour $(v, o)\{$
Initialiser v et o
Créer une matrice F_a telle que pour tout a et état s
Pour $s \in S$ **faire**
  $used = \sum_{s'} minF_a(s, s')$
  $remaining = 1 - used$
  Pour $i = 1, ..., n$ **faire**
    $min = minF_a(s, o(i))$
    $desired = maxF_a(s, o(i))$
    Si $desired \leq remaining$ **alors**
      $F_a(s, o(i)) = desired$
    Sinon
      $F_a(s, o(i)) = min + remaining$
    fin « Si »
    $remaining = max(0, remaining - desired)$
  fin « Pour »
fin « Pour »
Pour $s \in S$ **faire**
  $v(s) = \max_a R(s, a) + \sum_{s'} F_a(s, s')v(s')$
fin « Pour »$\}$

3.2.3 Convergence de l'algorithme

Nous proposons dans cette partie une étude de l'algorithme suggéré par [Givan et al., 2000] à travers l'introduction des opérateurs de Bellman T_o^- et T_o^+ tels que

$$T_o^- v(s) = \max_a \min R(s, a) + \gamma \sum_{i=1}^{|S|} p_o(s, a, q_i)v(q_i) \tag{3.4}$$

$$T_o^+ v(s) = \max_a \max R(s, a) + \gamma \sum_{i=1}^{|S|} p_o(s, a, q_i)v(q_i). \tag{3.5}$$

Dans l'équation 3.4, les probabilités $p_o(s, a, q_i)$ sont égales au maximum de probabilités dans chaque intervalle pour les états q_i admettant les plus petites valeurs $v(q_i)$ et aux minimum de probabilités pour les états q_i admettant les plus grandes valeurs $v(q_i)$ (tout en respectant la somme des probabiltés égale à 1). Ces probabilités de transition changent à chaque itération en fonction des valeurs $v(q_i)$ pour tout q_i. Par contre dans l'équation 3.5, o correspond à l'ordre qui associe les plus grandes probabilités au plus grandes valeurs. L'algorithme 5 présenté est une version "corrigée" [2] de celui qui se trouve dans [Givan et al., 2000] (cf. algorithme 5). Nous allons

2. L'algorithme tel qu'il paraît dans l'article [Givan et al., 2000] calcule l'intervalle de fonction de valeur correspondant à l'ordre $\leq_{opt}$. Ce qui n'est pas le cas, il estime la borne inférieure de l'ordre $\leq_{pes}$ et la borne

montrer dans ce qui suit la convergence de cet algorithme vers v^- et v^+. Pour cela il suffit de montrer que les opérateurs T_o^- et T_o^+ sont contractants.

Proposition 2. *Les opérateurs T_o^- et T_o^+ sont $(\gamma, \|.\|_\infty)$ contractants.*

Démonstration. Considérons l'ordre o qui associe les plus grandes probabilités aux plus petites valeurs. Soient v et w les vecteurs tels que

$$u(q_1) \leq u(q_2) \leq ... \leq u(q_{|S|}),\ u \in \{v, w\}.$$

On a en suivant le même raisonnement que dans la preuve du lemme 1 (voir l'annexe A)

$$T_o^- v(s) = \max_a \min R(s,a) + \gamma \sum_{i=1}^{|S|} p_o^v(s,a,q_i) v(q_i) \leq \max_a \min R(s,a) + \gamma \sum_{i=1}^{|S|} p(s,a,q_i) v(q_i).$$

où $p_o^v(s,a,q_i)$ désigne les probabilités assignées selon l'ordre o relativement au vecteur v et $p(s,a,.)$ est n'importe quelle probabilité prise dans l'intervalle $[\min p(s,a,), \max p(s,a,.)]$. Il s'en suit que pour tout s

$$\begin{aligned} T_o^- v(s) - T_o^- w(s) &= \max_a \gamma \sum_{i=1}^{|S|} p_o^v(s,a,q_i) v(q_i) - \gamma \sum_{i=1}^{|S|} p_o^w(s,a,q_i) w(q_i) \\ &\leq \gamma \max_a \sum_{i=1}^{|S|} p_o^w(s,a,q_i)(v(q_i) - w(q_i)). \end{aligned}$$

Supposons que $T_o^- v(s) \geq T_o^- w(s)$, on a pour tout s

$$|T_o^- v(s) - T_o^- w(s)| \leq \gamma \|v - w\|_\infty.$$

On obtient le même résultat si on suppose $T_o^- v(s) \leq T_o^- w(s)$. De manière analogue on prouve que T_o^+ est $(\gamma, \|.\|_\infty)$-contractant. □

Les opérateurs T_o^- et T_o^+ sont alors γ-contractants. Ils admettent alors chacun un unique point fixe qu'on note respectivement v_o^- et v_o^+. On va montrer que $v_o^- = v^-$ et $v_o^+ = v^+$, où les bornes v^- et v^+ sont définies dans les formules (3.2) et (3.3).

Proposition 3. *Les valeurs v^- et v^+ sont respectivement les points fixes des opérateurs T_o^- et T_o^+.*

Démonstration. On a

$$T_o^- v(s) = \max_\pi \min R(s, \pi(s)) + \gamma \sum_{i=1}^{|S|} p_o(s, \pi(s), q_i) v(q_i),$$

où l'ordre o associe les plus grandes probabilités aux plus petites valeurs. Soit pour π et M donnés, le vecteur v_M^π qui vérifie pour tout s

$$v_M^\pi(s) = R(s, \pi(s)) + \gamma \sum_{i=1}^{|S|} p(s, \pi(s), q_i) v_M^\pi(q_i).$$

supérieure pour $\leq_{opt}$. Aussi on a $F_a(s, o(i))$ égale à *desired* qui correspond au maximum de transition et non à min +*desired*. Nous avons pu clarifier tous ces points grâce à une correspondance avec R.Givan.

On peut déduire de la preuve du lemme 1 (voir annexe A) que pour tout π et pour tout M

$$\min R(s,\pi(s)) + \gamma \sum_{i=1}^{|S|} p_o(s,\pi(s),q_i) v_M^\pi(q_i) \leq v_M^\pi.$$

D'où

$$T_o^- \min_{M\in\mathcal{X}_\mathcal{M}} v_M^\pi \leq \max_\pi \min_{M\in\mathcal{X}_\mathcal{M}} v_M^\pi.$$

Ce qui implique que le point fixe de $v_o^- \leq \max_\pi \min_{M\in\mathcal{X}_\mathcal{M}} v_M^\pi$ (T_o^- est monotone). Prouvons l'inégalité dans l'autre sens. Conidérons la politique π' et le MDP $M_o \in \mathcal{X}_\mathcal{M}$ qui vérifient

$$\min R(s,\pi'(s)) + \gamma \sum_{i=1}^{|S|} p_o(s,\pi'(s),q_i) v_{M_o}^{\pi'}(q_i) = v_{M_o}^{\pi'}.$$

Il s'en suit que

$$\max_\pi \min R(s,\pi(s)) + \gamma \sum_{i=1}^{|S|} p_o(s,\pi(s),q_i) v_{M_o}^{\pi'}(q_i) \geq v_{M_o}^{\pi'} \geq \min_{M_o\in\mathcal{X}_\mathcal{M}} v_{M_o}^{\pi'}.$$

On obtient alors $v_o^- \geq \min_{M_o\in\mathcal{X}_\mathcal{M}} v_{M_o}^{\pi'}$. L'inégalité est vraie pour tout π' d'où $v_o^- \geq v^-$. On prouve de manière analogue que v^+ est le point fixe de T_o^+. □

Dans toute l'analyse qui suit on va utiliser les expressions suivantes des opérateurs T_o^- et T_o^+ qui seront plus adaptées aux différentes preuves qu'on va développer. En effet, en utilisant les mêmes arguments de preuve que précédemment, on a

$$T_o^- v(s) = \max_a \min_{M\in\mathcal{M}} R_M(s,a) + \gamma \sum_{s'\in S} p_M(s,a,s') v(s')$$
$$T_o^+ v(s) = \max_a \max_{M\in\mathcal{M}} R_M(s,a) + \gamma \sum_{s'\in S} p_M(s,a,s') v(s').$$

Ces expressions nous seront utiles lorsqu'on étudiera l'abstraction-BMDP et qu'on aura à estimer les bornes v^- et v^+.

3.2.4 Abstraction-BMDP

Dans ce paragraphe, on va expliciter l'abstraction proposée par [Givan et al., 2000] qui associe un MDP donné au BMDP qui lui correspond selon un certaine aggrégation. Soit $M = \langle S, A, R, P\rangle$ le MDP initial et $M_\alpha = \langle S_\alpha, A, R_\alpha, P_\alpha\rangle$ son image par l'abstraction α. L'espace S_α est obtenu en agrégeant les états dans S en des états bloc $[s]_\alpha$. Les paramètres R_α et P_α sont obtenus via les formules :

$$\forall s \in S,\ \forall a \in A,\ R_\alpha([s]_\alpha) = \left[\min_{s\in[s]_\alpha} R(s,a), \max_{s\in[s]_\alpha} R(s,a)\right] \tag{3.6}$$

$$\forall s,s' \in S,\ \forall a \in A,\ p_\alpha([s]_\alpha,a,[s']_\alpha) = \left[\min_{s_1\in[s]_\alpha} \sum_{s_2\in[s']_\alpha} p(s_1,a,s_2), \max_{s_1\in[s]_\alpha} \sum_{s_2\in[s']_\alpha} p(s_1,a,s_2)\right]. \tag{3.7}$$

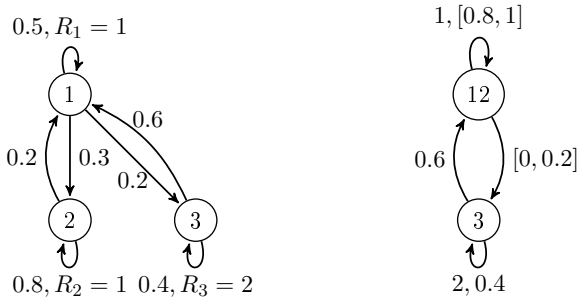

FIGURE 3.7 – Exemple d'un MDP (à gauche) et son correspondant BMDP (à droite) suivant l'abstraction $\alpha : \{1,2,3\} \to \{1,2\},\{3\}$.

Nous illustrons dans la figure qui suit un exemple de MDP et son BMDP correspondant selon une abstraction donnée :

EXEMPLE :

Soit M un MDP et son BMDP correspondant illustrés dans la figure 3.7. En appliquant les formules énoncées plus haut on obtient :

— $R_{\{12\},\{3\}}(\{12\}) = R_1 = R_2 = 1$, $R_{\{12\},\{3\}}(\{3\}) = R_3$

— $p_{\{12\},\{3\}}(\{12\},\{12\}) = [\min(p_{11}+p_{12}, p_{22}+p_{21}), \max(p_{11}+p_{12}, p_{22}+p_{21})] = [0.8, 1]$,
$p_{\{12\},\{3\}}(\{12\},\{3\}) = [\min(p_{13}, p_{23}), \max(p_{13}, p_{23})] = [0, 0.2]$

— $p_{\{12\},\{3\}}(\{3\},\{12\}) = p_{31} + p_{32} = 0.6$, $p_{\{12\},\{3\}}(\{3\},\{3\}) = 0.4$.

On peut calculer la fonction de valeur du MDP initial en utilisant un algorithme type IV présenté dans le chapitre 2 (section 2.4, algorithme 1). Le calcul de l'intervalle de fonction de valeur $[v_\alpha^-, v_\alpha^+]$ du BMDP correspondant M_α se fait via l'algorithme Interval Value Iteration IVI (algorithme 5) présenté dans la section précédente.

L'utilité d'introduire le BMDP n'est pas évidente en termes d'approximation de la fonction de valeur v^* mais elle le sera une fois qu'on aura introduit le théorème 1 qui suit. Mais avant cela on a besoin d'introduire une notion importante.

Définition 12. *Soit $M = \langle S, A, R_M, P_M\rangle$ un MDP donné et soit α une abstraction de S vers S_α. On dit qu'un MDP $N = \langle S, A, R_N, P_N\rangle$ est compatible avec M selon une abstraction donnée si*

1. *Pour tout $s \in S$ et pour tout $a \in A$, $R_N(s,a) \in [\min R_M([s]_\alpha, a), \max R_M([s]_\alpha, a)]$*
2. *Pour tous $s, s' \in S$ et pour tout $a \in A$, $p_N(s, a, [s']_\alpha) \in [\min p_M([s]_\alpha, a, [s']_\alpha)$, $\max R_M([s]_\alpha, a, [s']_\alpha)]$.*

On note l'ensemble des modèles N définis sur S qui sont compatibles avec M selon une abstraction α, $[M]_\alpha$.

Théorème 1. *[Givan et al., 1997] Soit M un MDP et M_α son BMDP correspondant par l'abstraction α on a*

$$\forall s \in S,\ v^*(s) \in [v_\alpha^-(s), v_\alpha^+(s)].$$

En effet grâce à ce type d'aggrégation on arrive à déduire une borne inférieure et une borne supérieure sur la valeur v^* qu'on cherche à calculer. Nous proposons ici une preuve de ce théorème en nous basant sur l'analyse introduite dans la section 3.2.3.

Démonstration. On sait que pour tout s dans l'espace S on a

$$v^-([s]_\alpha) = \max_{a\in A} \min_{N\in M_\alpha} R_N([s]_\alpha, a) + \gamma \sum_{s'\in S} p_N([s]_\alpha, a, [s']_\alpha) v^-([s']_\alpha).$$

En réalité l'ensemble des MDP N qui sont inclus dans le BMDP M_α peuvent être chacun identifié à l'ensemble des MDP N_1 qui sont compatibles avec $[M]_\alpha$. En d'autres termes on peut écrire que pour tout $s_1 \in [s]_\alpha$

$$\begin{aligned} v^-(s_1) = v^-([s]_\alpha) &= \max_{a\in A} \min_{N\in M_\alpha} R_N([s]_\alpha, a) + \gamma \sum_{s'\in S} p_N([s]_\alpha, a, [s']_\alpha) v^-([s']_\alpha) \\ &= \max_{a\in A} \min_{N_1\in [M]_\alpha} R_{N_1}(s_1, a) + \gamma \sum_{s'\in S} p_{N_1}(s_1, a, [s']_\alpha) v^-([s']_\alpha). \end{aligned}$$

Si on prend $\forall s \in S$, $v_0(s) = v^-([s]_\alpha)$ alors :

$$\begin{aligned} v_1(s) &= \max_{a\in A} \left(R_M(s,a) + \gamma \sum_{s'\in S} p_M(s,a,s') v_0(s') \right) \\ &= \max_{a\in A} \left(R_M(s,a) + \gamma \sum_{[s']_\alpha \in S_\alpha} \sum_{s''\in [s']_\alpha} p_M(s,a,s'') v_0(s'') \right). \end{aligned}$$

$v_0(s)$ est constante par blocs, on a alors

$$\begin{aligned} v_1(s) &= \max_{a\in A} \left(R_M(s,a) + \gamma \sum_{[s']_\alpha \in S_\alpha} p_M(s,a,[s']_\alpha) v^-([s']_\alpha) \right) \\ &\geq \max_{a\in A} \min_{N_1\in [M]_\alpha} \left(R_{N_1}(s,a) + \gamma \sum_{[s']_\alpha \in S_\alpha} p_{N_1}(s,a,[s']_\alpha) v^-([s']_\alpha) \right) \\ &= T^- v^-([s]_\alpha) \\ &= v^-([s]_\alpha) = v_0([s]_\alpha). \end{aligned}$$

L'opérateur de Bellman T est monotone (voir la partie 2.2.3), on a pour tout $s \in S$:

$$v^-([s]_\alpha) = v_0(s) \leq T v_0(s) = v_1(s) \leq ... \leq T^n v(s)$$

en prenant la limite quand $n \to \infty$ on obtient pour tout s

$$v^-([s]_\alpha) \leq v^*(s).$$

Ce même type de raisonnement peut être repris pour montrer que $v^*(s) \leq v^+([s]_\alpha)$ et on termine ainsi la preuve du théorème 1. □

3.2.5 BMDP et Jeu dynamique

On expliquera dans ce qui suit qu'un (B)MDP peut être considéré comme un cas de jeu dynamique à plusieurs joueurs.

Cadre général

Un SSP(MDP) peut être considéré comme un cas particulier d'un jeu à un joueur qui choisit à chaque étape une action a pour maximiser le gain. Un jeu à deux joueurs fait intervenir en plus un opposant qui, pour chaque choix de l'action a, choisit un autre paramètre de contrôle b dans le but au contraire de minimiser le gain. Si on définit deux politiques déterministes π_a et π_b telles que $\pi_a : S \to A$ et $\pi_b : S \to B$ (où B est un ensemble fini), la fonction de valeur associée aura cette forme pour tout $s_0 = s$

$$v^{\pi_a,\pi_b}(s) = \mathbb{E}^{\pi_a,\pi_b}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi_a(s_t), \pi_b(s_t)) \,\middle|\, s_0 = s\right].$$

Considérons maintenant les deux fonctions

$$\underline{v}(s) = \max_{\pi_a} \min_{\pi_b} v^{\pi_a,\pi_b}(s) \text{ et } \overline{v}(s) = \min_{\pi_b} \max_{\pi_a} v^{\pi_a,\pi_b}(s).$$

Chacune représente l'ordre possible dans lequel le joueur a commencé à jouer. La première fonction correspond au cas où c'est l'opposant qui joue en premier, et la deuxième fonction correspond au cas où c'est l'autre joueur qui commence en premier. Il a été prouvé d'après [Shapley, 1953] qu'on a pour tout $s \in S$, $\underline{v}(s) = \overline{v}(s)$. On notera cette valeur v^{**} pour faire la distinction avec v^*.

Cadre des BMDP

Le cas des MDPIP et donc des BMDP est un cas d'un jeu à deux joueurs où l'un choisit l'action afin de maximiser la somme des récompenses et l'autre selon le jeu peut soit l'aider en choisissant le MDP qui fait aussi maximiser le gain soit choisir un MDP qui fait minimiser le gain. On obtient dans ce cas pour tout s dans l'espace S

$$v^-(s) = \max_a \min_{N \in [M]_\alpha} v_N(s) = \min_{N \in [M]_\alpha} \max_a v_N(s) = \min_{N \in [M]_\alpha} v_N^*(s) \tag{3.8}$$

$$v^+(s) = \max_a \max_{N \in [M]_\alpha} v_N(s) = \max_{N \in [M]_\alpha} \max_a v_N(s) = \max_{N \in [M]_\alpha} v_N^*(s). \tag{3.9}$$

L'ensemble $[M]_\alpha$ ici n'est pas un ensemble fini mais on a vu dans ce qui précède que pour chaque action a les MDP $M^{\min}$ et $M^{\max}$ étaient inclus dans l'ensemble X_M donc on peut restreindre l'espace $[M]_\alpha$ à l'espace fini L_α des MDP N tels que leur image par α soit incluse dans X_M qui est fini.

3.2.6 Propriétés de l'abstraction-BMDP

Une fois qu'on a décrit les bounded parameter MDP (BMDP) on va s'intéresser au même problème que celui étudié dans le cadre du modèle moyen. On est toujours à la recherche de "bonnes heuristiques" qui nous aident à nous approcher de la fonction v^*. On va regarder si la qualité est monotone par rapport à l'abstraction (la relation d'ordre est $\alpha \preceq \alpha'$ ssi α' est un raffinement de α). A partir des définitions de v^- et de v^+ formulées dans les équations (3.8) et (3.9), nous pouvons déduire une condition suffisante qui satisfait $v^-_{\alpha'}(s) \geq v^-_\alpha(s)$ et $v^+_{\alpha'}(s) \leq v^+_\alpha(s)$, pour tout s et pour tout raffinement direct α' de α.

Proposition 4. *Soit $M = \langle S, A, R, P, \gamma\rangle$ et soient α et α' deux abstractions de M telles que α' est un raffinement direct de α. Si l'ensemble des MDP $[M]_{\alpha'}$ qui sont compatibles avec le MDP*

M selon l'abstraction α' le sont également selon l'abstraction α, dans le sens où $[M]_{\alpha'} \subseteq [M]_\alpha$, on a alors l'inclusion des fonctions de valeur

$$[v^-_{\alpha'}([s]_{\alpha'}), v^+_{\alpha'}([s]_{\alpha'})] \subseteq [v^-_\alpha([s]_\alpha), v^+_\alpha([s]_\alpha)].$$

Démonstration. La preuve découle directement des équations (3.8) et (3.9). Prendre le maximum sur un ensemble plus grand induit un plus grand maximum et idem prendre le minimum sur un ensemble plus petit induit un plus petit minimum. La condition $[M]_{\alpha'} \subseteq [M]_\alpha$ est une condition suffisante mais non nécessaire. En effet, on peut très bien imaginer des cas où $[v^-_{\alpha'}([s]_{\alpha'}), v^+_{\alpha'}([s]_{\alpha'})] \subseteq [v^-_\alpha([s]_\alpha), v^+_\alpha([s]_\alpha)]$ sans pour autant avoir l'inclusion $[M]_{\alpha'} \subseteq [M]_\alpha$. C'est ce qu'on illustre dans l'exemple qui suit :

EXEMPLE :

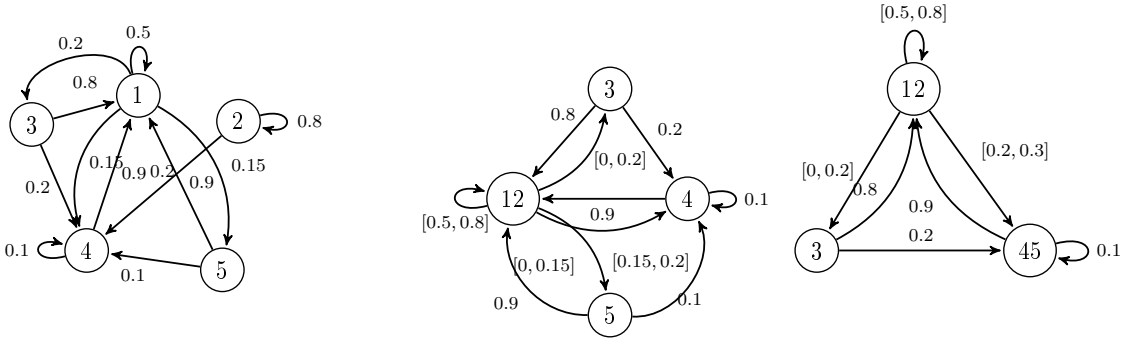

FIGURE 3.8 – De gauche à droite : le MDP M, l'abstraction α', l'abstraction α. Les flèches sont annotées avec les probabilités de transition. Les récompenses vérifient : $R(1) = R(2) = R(3) = R(4) = 0, R(5) = 1$, $\gamma = 0.99$.

On obtient en implémentant l'algorithme proposé par Givan et al :

$$v^-_{\alpha'}(12,3,4,5) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} v^+_{\alpha'}(12,3,4,5) = \begin{pmatrix} 11.158783 \\ 10.948924 \\ 11.034826 \\ 12.034826 \end{pmatrix}$$

$$v^-_{\alpha}(12,3,45) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} v^+_{\alpha}(12,3,45) = \begin{pmatrix} 29.58928 \\ 29.912131 \\ 30.370645 \end{pmatrix}.$$

On voit très bien ici que $v^-_\alpha \leq v^-_{\alpha'}$ et que $v^+_{\alpha'} \leq v^+_\alpha$. Pourtant on peut très bien expliciter un MDP N (figure 3.9 ci-dessous), tel que $N_{\alpha'} \subset N_\alpha$ dans le sens de l'inclusion des intervalles de transition mais $N_\alpha \not\subset M_\alpha$. En effet on peut remarquer que $0.35 \notin [0.2, 0.3]$. La condition de la proposition 4 est alors une condition suffisante mais pas nécessaire.

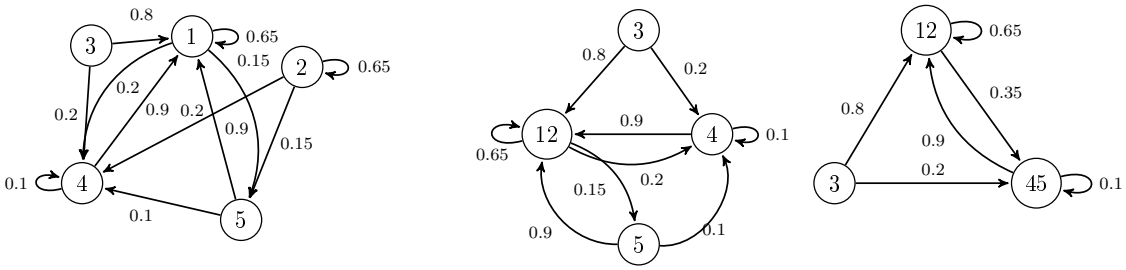

FIGURE 3.9 – De gauche à droite : le MDP N, l'abstraction α', l'abstraction α. Les flèches sont annotées avec les probabilités de transition. Les récompenses vérifient : $R(1) = R(2) = R(3) = R(4) = 0, R(5) = 1$, $\gamma = 0.99$.

□

Nous allons dans ce qui suit étudier le cadre des BMDP déterministe séparément du cadre des BMDP probabilistes, et voir si on obtient de meilleures bornes en raffinant l'abstraction dans chaque cas.

Le cas déterministe

Dans cette partie le MDP initial considéré est déterministe. Les intervalles de probabilités associés au BMDP correspondant, sachant une abstraction α donnée, sont de la forme : 0,1 et $[0, 1]$ (voir les formules (3.6) et (3.7)). Étant données deux abstractions α et α' telles que α' est un raffinement direct de α (définition 8) on va essayer de voir si pour tout $s \in S$, $v^-_{\alpha'}([s]_{\alpha'}) \geq v^-_\alpha([s]_\alpha)$ et $v^+_\alpha([s]_\alpha) \geq v^+_{\alpha'}([s]_{\alpha'})$, c'est-à-dire que α' induit de meilleures bornes que α. Nous allons montrer que oui.

Proposition 5. *Soient M un MDP déterministe, α et α' deux abstractions telles que α' est un raffinement direct de α on a alors*

$$\forall s \in S,\ [v^-_{\alpha'}([s]_{\alpha'}), v^+_{\alpha'}([s]_{\alpha'})] \subseteq [v^-_\alpha([s]_\alpha), v^+_\alpha([s]_\alpha)].$$

Démonstration. Comparer deux fonctions qui sont définies sur deux espaces différents n'est pas une chose évidente. L'astuce ici est de se ramener à l'espace initial via l'ensemble des modèles compatibles N avec l'ensemble $[M]_{\alpha'}$ et $[M]_\alpha$. Autrement dit puisqu'on sait déjà que pour toute abstraction α

$$v^-_\alpha([s]_\alpha) = \min_{N \in [M]_\alpha} v^*_N(s),$$

il suffit donc de montrer que $[M]_{\alpha'} \subseteq [M]_\alpha$ pour prouver que $v^-_\alpha([s]_\alpha) \leq v^-_{\alpha'}([s]_{\alpha'})$.

La condition $[M]_{\alpha'} \subseteq [M]_\alpha$ implique que pour tout MDP $N \in [M]_{\alpha'}$, on a $N \in [M]_\alpha$. Soit $[s_1]_\alpha$ le bloc qu'on raffine en deux autres blocs distincts $[s_2]_{\alpha'}$ et $[s_3]_{\alpha'}$ avec s_2 et s_3 deux états distincts dans $[s_1]_\alpha$. Tous les autres blocs ne changent pas après un raffinement. On note $b_1, ..., b_{|S_\alpha|-1}$ ces blocs là. L'inclusion des récompenses est évidente ici. La condition $N \in [M]_{\alpha'}$ impose que pour tout $j \in \{2, 3\}$, pour toute action a on a

$$\left[\min_{s' \in [s_j]_{\alpha'}} R_N(s', a), \max_{s' \in [s_j]_{\alpha'}} R_N(s', a)\right] \subseteq \left[\min_{s' \in [s_j]_{\alpha'}} R_M(s', a), \max_{s' \in [s_j]_{\alpha'}} R_M(s', a)\right].$$

Étant donné que pour tout MDP M on a

$$\min_{s'\in[s_1]_\alpha} R_M(s',a) = \min\left(\min_{s'\in[s_2]_{\alpha'}} R_M(s',a), \min_{s'\in[s_3]_{\alpha'}} R_M(s',a)\right)$$
$$\max_{s'\in[s_1]_\alpha} R_M(s',a) = \max\left(\max_{s'\in[s_2]_{\alpha'}} R_M(s',a), \max_{s'\in[s_3]_{\alpha'}} R_M(s',a)\right)$$

il s'en suit que

$$\left[\min_{s'\in[s_1]_\alpha} R_N(s',a), \max_{s'\in[s_1]_\alpha} R_N(s',a)\right] \subseteq \left[\min_{s'\in[s_1]_\alpha} R_M(s',a), \max_{s'\in[s_1]_\alpha} R_M(s',a)\right].$$

Il reste alors à montrer que pour tout $j,k \in \{2,3\}$ et $l \in \{1,...,|S_\alpha|-1\}$ on a

$$[\min p_N([s_1]_\alpha,a,b_l), \max p_N([s_1]_\alpha,a,b_l)] \subseteq [\min p_M([s_1]_\alpha,a,b_l), \max p_M([s_1]_\alpha,a,b_l)]$$
$$[\min p_N([s_1]_\alpha,a,[s_1]_\alpha), \max p_N([s_1]_\alpha,a,[s_1]_\alpha)] \subseteq [\min p_M([s_1]_\alpha,a,[s_1]_\alpha), \max p_M([s_1]_\alpha,a,[s_1]_\alpha)]$$
$$[\min p_N(b_l,a,[s_1]_\alpha), \max p_N(b_l,a,[s_1]_\alpha)] \subseteq [\min p_M(b_l,a,[s_1]_\alpha), \max p_M(b_l,a,[s_1]_\alpha)].$$

On va raisonner sur le min, ce raisonnement sera valable aussi pour le max. On a pour tout MDP M

$$\min p_M([s_1]_\alpha,a,b_l) = \min\left(\min p_M([s_2]_{\alpha'},a,b_l), \min p_M([s_3]_{\alpha'},a,b_l)\right).$$

On a $N \in [M]_{\alpha'}$ donc pour tout $j \in \{2,3\}$, $\min p_N([s_j]_\alpha,a,b_l) \geq \min p_M([s_j]_\alpha,a,b_l)$ par conséquent $\min p_N([s_1]_\alpha,a,b_l) \geq \min p_M([s_1]_\alpha,a,b_l)$. La première inclusion est alors prouvée.

Démontrons la deuxième inclusion. Supposons que

$$\min p_N([s_1]_\alpha,a,[s_1]_\alpha) = \min_{s\in[s_1]_\alpha} p_N(s,a,[s_2]_\alpha) + p_N(s,a,[s_3]_\alpha) = 0,$$

il existe alors un état $s_0 \in [s_1]_\alpha$ tel que $p_N(s,a,[s_2]_{\alpha'}) = 0$ et $p_N(s,a,[s_3]_{\alpha'}) = 0$. Puisque $N \in [M]_{\alpha'}$, il s'en suit que pour $j,k \in \{2,3\}$

$$[\min p_N([s_j]_{\alpha'},a,[s_k]_{\alpha'}), \max p_N([s_j]_{\alpha'},a,[s_k]_{\alpha'})] \subseteq [\min p_M([s_j]_{\alpha'},a,[s_k]_{\alpha'}), \max p_M([s_j]_{\alpha'},a,[s_k]_{\alpha'})].$$

Ceci implique que

$$\min p_M([s_1]_\alpha,a,[s_1]_\alpha) = \min_{s\in[s_1]_\alpha} p_M(s,a,[s_2]_\alpha) + p_M(s,a,[s_3]_\alpha) = 0.$$

Supposons maintenant que $\min p_N([s_1]_\alpha,a,[s_1]_\alpha) = 1$ alors il est évident que $\min p_N([s_1]_\alpha,a,[s_1]_\alpha) \geq \min p_M([s_1]_\alpha,a,[s_1]_\alpha)$. Ce qui prouve la deuxième inclusion.

Nous allons montrer maintenant la dernière inclusion. Supposons que $\min p_N(b_l,a,[s_1]_\alpha) = 0$ ceci implique qu'il existe $s_0 \in b_l$ tel que $p_N(s_0,a,[s_1]_\alpha) = 0$, il existe par conséquent un bloc ${b_l}^*$ vérifiant $p_N(s_0,a,{b_l}^*) = 1$. Puisque $N \in [M]'_\alpha$ on a

$$[\min p_N(b_l,a,{b_l}^*), \max p_N(b_l,a,{b_l}^*)] \subseteq [\min p_M(b_l,a,{b_l}^*), \max p_M(b_l,a,{b_l}^*)].$$

On a donc forcément $\max p_M(b_l,a,{b_l}^*) = 1$, il existe alors un état $s'_0 \in b_l$ tel que $p_M(s'_0,a,{b_l}^*) = 1$, d'où $p_M(s'_0,a,[s_1]_\alpha) = 0$. Ceci implique que $\min p_M(b_l,a,[s_1]_\alpha) = 0$. Si on a $\min p_N(b_l,a,[s_1]_\alpha) = 1$, il est évident que $\min p_M(b_l,a,[s_1]_\alpha) \leq \min p_N(b_l,a,[s_1]_\alpha)$. □

Le cas probabiliste

Le cas probabiliste concerne les MDP dont les probabilités de transition peuvent prendre des valeurs dans $[0, 1]$. On a vu dans la preuve précédente que le fait d'avoir deux valeurs 0 et 1 était une condition essentielle pour avoir l'inclusion des fonctions de valeurs. Ceci laisse penser qu'on n'aura probablement pas la même propriété lorsqu'on passe au cas stochastique. Faute de mieux peut-on au moins toujours garantir en partant d'une abstraction donnée α l'existence d'un raffinement α' qui garantirait l'inclusion des fonctions de valeur ? En d'autres termes peut-on trouver à chaque raffinement l'abstraction qui ferait réduire, d'une façon monotone, dans chaque état l'erreur d'approximation G_α, où pour tout $s \in S$,

$$G_\alpha([s]_\alpha) = v^+([s]_\alpha) - v^-([s]_\alpha).$$

Nous allons montrer que non.

Proposition 6. *Il existe un MDP M une abstraction α tels que pour tout raffinement direct α' de α on a*

— $\forall s \in S$, $G_{\alpha'}([s]_{\alpha'}) \geq G_\alpha([s]_\alpha)$

— $\exists s \in S$, *tel que* $G_{\alpha'}([s]_{\alpha'}) > G_\alpha([s]_\alpha)$.

Cette proposition stipule qu'il existe des cas de MDP où l'erreur d'approximation reste dans le meilleurs des cas inchangée après un raffinement. Dans la preuve qui suit, on va justement donner un exemple de ce type de MDP.

Démonstration. Considérons le MDP de la figure 3.10. L'idée derrière ce choix du modèle est

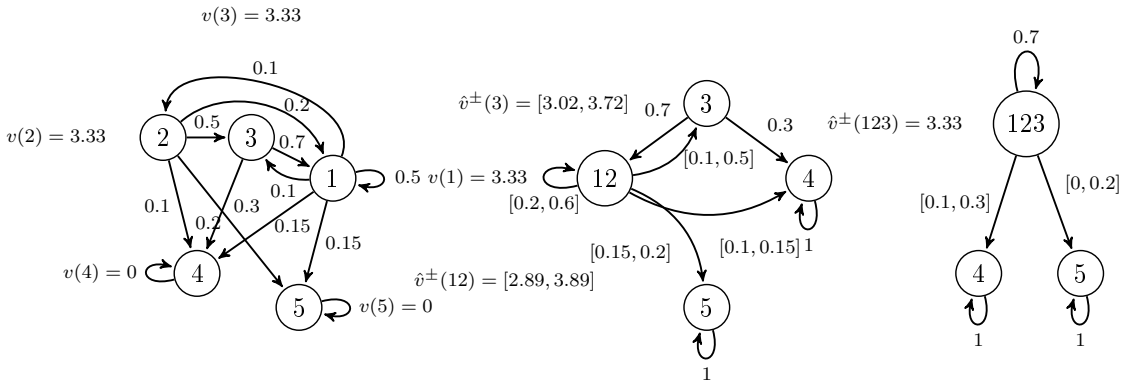

FIGURE 3.10 – De gauche à droite : le MDP initial M, l'abstraction α_1, l'abstraction α. Les flèches sont annotées des probabilités de transition. Les récompenses sont telles que : $R(1) = R(2) = R(3) = 1 (= R(\{1, 2\}) = R(\{1, 2, 3\}))$, et $R(4) = R(5) = 0$. On prend $\gamma = 1$.

de prendre des états qui se comportent de la même façon selon l'abstraction la plus grossière α. Dans ce cas on ne risque pas de commettre d'erreur puisqu'on obtient exactement les mêmes fonctions de valeur. Puis il faut observer si pour un raffinement de l'abstraction α, on va garder les mêmes fonctions de valeurs. En effet on peut voir dans la figure 3.10 que les états 1, 2 et 3 dans le modèle initial admettent la même récompense et la même probabilité de transition d'atteindre les blocs $\{1, 2, 3\}$ et $\{4, 5\}$. On a $p(1, \{1, 2, 3\}) = p(2, \{1, 2, 3\}) = p(3, \{1, 2, 3\}) = 0.7$, et $p(1, \{4, 5\}) = p(2, \{4, 5\}) = p(3, \{4, 5\}) = 0.3$). Par conséquent ils admettent la même fonction de valeur $v(1) = v(2) = v(3) = 3.33$ et $v(4) = v(5) = 0$. L'abstraction α (modèle à droite)

est une représentation similaire mais plus compacte du modèle initial qui vérifie $v(\{1,2,3\}) = v(\{1\}) = v(\{2\}) = v(\{3\}) = 3.33$. L'abstraction $\alpha_1 : 1,2,3,4,5 \to \{1,2\},\{3\},\{4\},\{5\}$ rassemble les états 1 et 2 dans un même bloc. On obtient dans chaque état des intervalles de fonction de valeur, il est clair dans ce cas que l'erreur d'approximation ne peut qu'augmenter. On a

— $G_{\alpha_1}(\{1\}) = G_{\alpha_1}(\{2\}) = 3.89 - 2.89 = 1 > G_\alpha(\{1\}) = G_\alpha(\{2\}) = 0$
— $G_{\alpha_1}(\{3\}) = 3.72 - 3.02 = 0.7 > G_\alpha(\{3\}) = 0$
— $G_{\alpha_1}(\{4\}) = G_{\alpha_1}(\{5\}) = G_\alpha(\{4\}) = G_\alpha(\{5\}) = 0$.

On obtient le même résultat si au lieu de considérer l'abstraction α_1 on considère l'abstraction $\alpha_2 : 1,2,3,4,5 \to \{1,3\},\{2\},\{4\},\{5\}$ ou l'abstraction $\alpha_3 : 1,2,3,4,5 \to \{2,3\},\{1\},\{4\},\{5\}$. En effet on obtient dans respectivement chacun des cas

$$v^-_{\alpha_2}(13,2,4,5) = \begin{pmatrix} 2.5000501 \\ 2.7500584 \\ 0 \\ 0 \end{pmatrix},\ v^+_{\alpha_2}(13,2,4,5) = \begin{pmatrix} 4.7823235 \\ 4.3475729 \\ 0 \\ 0 \end{pmatrix}$$

$$\text{et} \quad v^-_{\alpha_3}(1,23,,5) = \begin{pmatrix} 2.9033191 \\ 2.258122 \\ 0 \\ 0 \end{pmatrix},\ v^+_{\alpha_3}(1,23,4,5) = \begin{pmatrix} 4.1174752 \\ 5.2938746 \\ 0 \\ 0 \end{pmatrix}.$$

□

On pourrait à partir de ce modèle extraire un autre modèle où l'erreur augmente strictement dans tous les états $1,2,3,4,5$ telle que pour tout état s, $G_\alpha(s) > G_{\alpha'}(s)$. En effet considérons un MDP M' où on garde les même transitions dans les états 1, 2 et 3 que celles dans le MDP M mais où on change les transitions dans les états 4 et 5. On prend ainsi pour les états 4 et 5 : $p(4,4) = 0.9, p(4,1) = 0.1$ et $p(5,5) = 0.9,\ p(5,1) = 0.1$. On peut choisir un facteur γ qui soit strictement inférieur à 1 mais assez proche de 1 pour garantir l'existence d'une solution. On remarquera dans ce cas que l'erreur $G_{\alpha'}(s)$ est strictement plus grande que $G_\alpha(s)$ pour chaque état s dans $\{1,2,3,4,5\}$.

Considérons l'abstraction α du modèle M initial dans la figure 3.10, on remarque que si on agrège les états 4 et 5 on obtient une *bisimulation* (cf. chapitre 2).

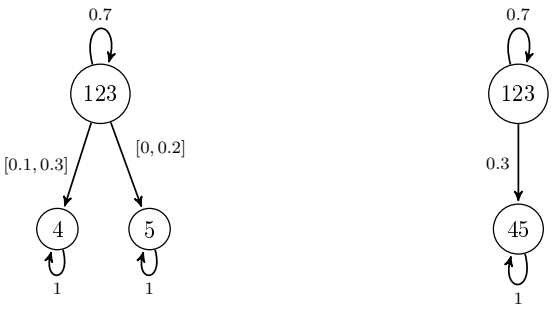

FIGURE 3.11 – De gauche à droite l'abstraction α et la bisimulation qui correspond au modèle M.

En effet on a $R(1) = R(2) = R(3)$ et $R(4) = R(5)$, et pour tout $i \in \{1,2,3\},\ p(i,123) = 0.7$ et $p(i,45) = 0.3$ (voir figure 3.11). On pourrait penser qu'un modèle type où l'erreur G_α– par

rapport à une abstraction α donnée– augmente en raffinant est un cas de *bisimulation*. Comme le montre le contre-exemple dans la figure 3.10. Une *bisimulation* est une abstraction équivalente au modèle initial i.e. admettant la même fonction de valeur et donc la même politique optimale que celle du modèle initial considéré. Si on traduit la définition 5 (chapitre 2) en termes d'abstraction ceci revient à vérifier que pour tout $s \in S$ et pour tout $s_1 \in [s]_\alpha$:

$$\forall s \in S, \forall s_1 \in [s]_\alpha,\ R(s,a) = R(s_1,a), \forall [s']_\alpha \in S_\alpha,\ p(s,a,[s']_\alpha) = p(s_1,a,[s']_\alpha).$$

Pour chaque action considérée la borne inférieure et la borne supérieure coïncident dans le cas d'une bisimulation. D'après le théorème 1, on a alors pour chaque état $s \in S$, $v(s) = v([s]_\alpha)$.

On observe des comportements différents (pathologiques dans le cas stochastique) de ceux qu'on a obtenus dans le cadre déterministe. Néanmoins on a réussi à identifier une structure qui permet de résoudre le MDP initial qui est la *bisimulation*. Dans le cas des MDP admettant un grand nombre d'états, trouver une bisimulation n'est pas une chose simple (faut-il encore qu'elle existe). Plusieurs travaux ont été proposés en ce sens afin de reconnaître ces structures (cf. chapitre 2).

3.3 Travaux analogues

Notre travail montre les limites de certaines abstractions, où raffiner une abstraction peut induire une plus grande erreur d'approximation. La structure d'équivalence exhibée qui est la bisimulation est intéressante dans le sens où elle renvoie exactement la solution recherchée. Ceci dit identifier cette structure s'avère compliqué–surtout lorsque l'espace des états est très grand– comme il a été illustré dans la métrique de [Ferns et al., 2012] ou dans l'algorithme de raffinement de partitions de [Givan et al., 2000]. Le fait qu'un raffinement induise une plus mauvaise approximation de la solution n'est pas complètement nouveau, il a déjà été observé dans le cas de l'abstraction des actions [Sandholm and Singh, 2012] et dans le cas des jeux comportant plusieurs joueurs.

Pathologies dans les jeux à plusieurs joueurs

Les pathologies dans les abstractions ont été déjà observées dans les cas de jeux à plusieurs joueurs. Dans ce cadre on peut citer l'article de [Kevin et al., 2009], où les auteurs ont étudié l'exemple du jeu Leduc Hold'em Games (type de jeu de poker). Par souci de simplicité on va considérer le cas de deux joueurs. On note pour $i \in \{1,2\}$ la fonction utilité u_i qui est la somme du gain cumulé pour avoir atteint un état terminal donné. Soit $\pi = (\pi_1, \pi_2)$ un profil de stratégie, $u_i(\pi)$ est le gain correspondant au joueur i lorsque les joueurs 1 et 2 suivent respectivement la politique π_1 et π_2 respectivement. Lorsque le joueur i connaît la stratégie π_j du joueur j sa meilleure réponse ou stratégie π_i est celle qui vérifie

$$\forall i,j \in \{1,2\} \quad u_i(\pi_i,\pi_j) = \max_{\pi'_j} u_i(\pi_i,\pi'_j).$$

Dans le cas où on ne connaît pas la stratégie de l'opposant on fait appel au concept de l'équilibre de Nash représenté par la stratégie $\pi = (\pi_1, \pi_2)$. Une stratégie d'équilibre (π_1^*, π_2^*) est une stratégie qui maximise la fonction utilité dans les pires des scénarios, i.e. dans le cas où l'opposant a employé la pire des stratégies, cela se traduit par

$$\forall i,j \in \{1,2\} \quad u_i(\pi_i^*,\pi_j^*) = \max_{\pi_i} \min_{\pi_j} u_i(\pi_i,\pi_j).$$

Pour estimer la déviation d'une stratégie donnée $\pi = (\pi_1, \pi_2)$ par rapport à la stratégie d'équilibre (π_1^*, π_2^*) on introduit une notion qui est la notion d'exploitabilité $\epsilon_i(\pi_i)$

$$\epsilon_1(\pi_1) = \max_{\pi_2'} u_2(\pi_1, \pi_2') - u_2(\pi_1^*, \pi_2^*)$$
$$\epsilon_2(\pi_2) = \max_{\pi_1'} u_1(\pi_1', \pi_2) - u_1(\pi_1^*, \pi_2^*).$$

Notons que pour tout π_i, $\epsilon_i(\pi_i) \geq 0$. Dans le cas où $\pi = (\pi_1, \pi_2) = (\pi_1^*, \pi_2^*)$ on a $\epsilon_i(\pi_i) = 0$. Pour deux abstractions α et α' telles que α' est un raffinement de α, on dit que α' est moins exploitable que α par rapport à un joueur i donné si

$$\epsilon_i(\pi_i^{\alpha'}) \leq \epsilon_i(\pi_i^{\alpha}) \tag{3.10}$$

où

$$\epsilon_i(\pi_i^{\alpha}) = \max_{\pi_j'} u_j(\pi_i^{\alpha}, \pi_j') - u_j(\pi_1^*, \pi_2^*).$$

Dans le cas du jeu de Leduc Hold'em Games, deux cas d'abstractions ont été considérées : le cas des abstractions des cartes et le cas des abstractions des mises. Il a été constaté qu'il existait des abstractions où lorsqu'on agrégeait deux cartes ensemble on obtenait une stratégie moins exploitable que dans le cas on n'agrégeait pas les cartes. Le jeu comprend 12 options de mises qui ont été réduites à 4 options de mises. Dans ces 4 options, différentes combinaisons d'actions ont été considérées. Il a été constaté qu'avoir plus d'options pouvait faire augmenter l'exploitabilité (du joueur ou de l'opposant).

Pathologies dans les abstractions d'actions

On va s'intéresser maintenant aux abstractions des actions. Une abstraction des actions signifie que les actions dont dispose un agent dans le cas du jeu abstrait diffèrent de celles dont il dispose dans le jeu de départ (il dispose généralement de moins d'actions). On va expliciter un exemple tiré de [Sandholm and Singh, 2012] qui montre qu'une abstraction plus grossière peut s'avérer moins exploitable dans le sens de l'équation (3.10).

Exemple :

Considérons deux joueurs 1 et 2. Le joueur 1 dispose des actions a et b. Le joueur 2 dispose des actions a, b et c. Si les deux joueurs choisissent la même action, c'est le joueur 2 qui gagne. Il reçoit dans ce cas une récompense $R(2) = 2$ et le joueur 1 ne reçoit rien en retour, $R(1) = 0$. Si le joueur numéro 2 choisit a ou b et que le joueur 1 choisit l'action contraire, c'est le joueur 1 qui gagne et il reçoit une récompense $R(1) = 2$ tandis que $R(2) = 0$. Si le joueur 2 choisit l'action c, les deux joueurs reçoivent la même récompense $R(1) = R(2) = 1$. Cette dernière stratégie est la stratégie d'équilibre. Le joueur 1 choisit l'action a ou b avec probabilité $\frac{1}{2}$. Le joueur 2 choisit l'action a ou b avec probabilité p et l'action c avec probabilité $1 - 2p$. Considérons l'abstraction où le joueur 1 ne dispose que de l'action a, le joueur 2 peut choisir l'action a qui est différente de la stratégie d'équilibre. Considérons une abstraction plus grossière où le joueur 2 ne dispose que de l'action c, qui correspond à la stratégie d'équilibre du jeu initial où les deux joueurs reçoivent la même récompense.

3.4 Conclusion

On a étudié dans ce chapitre deux types d'abstractions : l'abstraction moyenne et l'abstraction BMDP. On a reformalisé d'une manière originale l'approche BMDP et proposé une version

corrigée de l'algorithme [Givan et al., 2000]. On a aussi prouvé formellement le lien entre les valeurs du BMDP et celle d'un MDP sous jacent à travers la notion de compatibilité des modèles. On a montré qu'il pouvait y avoir absence de monotonie de l'approximation des fonctions de valeur par rapport aux abstractions même dans un cadre purement déterministe dans l'approche moyenne (proposition 1) et uniquement dans le cadre probabiliste (propositions 5 et 6) pour les *bounded parameter* MDP. L'analyse qu'on propose ici sur les pathologies liées à certaines abstractions est un peu plus informative que celles qui ont été déjà avancées dans la littérature. En effet contrairement à ce qui a été présenté dans [Kevin et al., 2009] et [Sandholm and Singh, 2012], les modèles qu'on exhibe ici sont des modèles admettant une seule action et un seul joueur (les contre-exemples présentés ici sont des chaînes de Markov). Cela laisse à penser que dans certains cas (les BMDP y compris), seule la stochasticité de l'environnement peut expliquer les pathologies apparaissant dans les abstractions.

4

Les méthodes de projection linéaire pour un schéma de type itérations sur les politiques

Introduction

Nous allons dresser dans ce chapitre l'état de l'art de quelques méthodes de résolution approximatives des MDP. Ceci est nécessaire par la suite pour la compréhension des chapitres 5 et 6. Ces méthodes calculent une approximation de la vraie fonction de valeur v^*. Les algorithmes associés sont une version approchée des algorithmes de la programmation dynamique introduits dans le chapitre 2. On distingue ainsi l'algorithme d'itérations sur les valeurs avec approximation (AVI) et l'algorithme d'itérations sur les politiques avec approximation (API). On présentera plus en détails ces deux algorithmes dans la première partie de ce chapitre. La seconde partie de ce chapitre sera consacrée aux méthodes de calcul d'une approximation de la fonction de valeur pour une politique π fixée qui peut servir pour un schéma de type API. On s'intéressera plus particulièrement aux méthodes dites de projection. On introduira de manière générale plusieurs de ces méthodes ainsi que les différents algorithmes correspondants.

4.1 Algorithmes de la programmation dynamique avec approximation

Afin de déterminer une approximation de la fonction optimale v^*, on va étendre les algorithmes introduits dans le premier chapitre (IV) et (IP) au cas approché.

4.1.1 Itérations sur les valeurs avec approximation AVI

Le schéma algorithmique *itérations sur les valeurs avec approximation* AVI calcule, à chaque itération k, une approximation de l'image de la fonction de valeur par l'opérateur de Bellman optimal $T = \max_\pi T_\pi$ à une erreur ϵ_k près

$$v_{k+1} \leftarrow Tv_k + \epsilon_{k+1}.$$

L'approximation v_{k+1} de Tv_k peut par exemple être calculé par un algorithme d'apprentissage supervisé. La variable ϵ_k est pour tout k l'erreur d'approximation générée. Pour $k = 0$, on

prend v_0 arbitraire. En général, l'algorithme AVI ne converge pas. Néanmoins on peut borner la différence entre la valeur optimale v^* et la valeur v_{π_k}, où v_{π_k} est la vraie valeur si on suit la politique π_k. Si on suppose les erreurs d'approximation uniformément bornées : pour tout k, $\|\epsilon_k\|_\infty \leq \epsilon$, on sait que [Bertsekas and Tsitsiklis, 1996]

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

Cette borne peut être arbitrairement grande lorsque le facteur d'actualisation γ est proche de 1, elle s'applique par conséquent aux différents cas d'application. C'est la meilleure qu'on puisse avoir, [Scherrer and Lesner, 2012] ont exhibé un modèle de MDP (déterministe) pour lequel elle est atteinte.

4.1.2 Itérations sur les politiques avec approximation API

Le schéma algorithmique *itérations sur les politiques avec approximation* (API) calcule une approximation, pour une politique π_k fixée, de la valeur v_{π_k} à une erreur ϵ_k près. On note $\mathcal{G}(v_k)$ l'ensemble des politiques gloutonnes par rapport à v_k. Le schéma de l'algorithme API peut s'écrire de la sorte

$$v_k \leftarrow v_{\pi_k} + \epsilon_k$$
$$\pi_{k+1} \in \mathcal{G}(v_k).$$

L'approximation v_k de v_{π_k} se fait par exemple à travers un algorithme d'apprentissage type *least square temporal difference* LSTD qu'on décrira dans la partie qui suit. L'algorithme *least square policy iteration* est un exemple d'algorithme d'itérations sur les politiques avec approximation. (API) admet une borne de performance identique à celle de l'algorithme (AVI). Il existe un MDP pour lequel elle est atteinte [Bertsekas and Tsitsiklis, 1996].

Une fois qu'on a introduit les principaux algorithmes d'itérations avec approximation, on va se concentrer dans ce qui suit sur l'évaluation approchée d'une politique ce qui peut servir dans un schéma type (API) [3].

4.2 Les Méthodes de projection

Considérons, pour une politique fixée π, une chaîne de Markov $\mathcal{M}$ irréductible de matrice de transition P_π et de récompense R_π prenant ses valeurs dans l'espace $\mathcal{X}$. Calculer la fonction de valeur $v^\pi = T^\pi v^\pi$ relative à cette chaîne de Markov peut s'avérer compliqué lorsque l'espace des états est très grand. On va s'intéresser ici à approcher cette fonction en estimant ses projections sur des espaces de taille plus réduite $d \ll n$. Ces espaces sont des sous-espaces vectoriels générés par des fonctions qu'on appelle les fonctions "caractéristiques" ou "features" . L'approximation $\hat{v}$ de v est une approximation *linéaire* de v (on omettra dorénavant l'exposant π). Les fonctions features $(\phi_i)_{1\leq i\leq d}$ forment la matrice des features qu'on note Φ. C'est la matrice dont les colonnes sont formées par ces fonctions, elle est de taille $|\mathcal{X}|\times d$. Il existe un vecteur $\theta \in \mathbb{R}^d$ tel que $\hat{v} = \Phi\theta$, c'est le vecteur poids associé à $\hat{v}$ la projection de v sur $span(\Phi)$, qui est l'espace généré par les fonctions $(\phi_i)_{1\leq i\leq d}$. On note Π la matrice de projection orthogonale sur l'espace des features

3. L'évaluation est plus simple pour (AVI). Ceci revient à calculer pour chaque itération k une approximation de Tv_k. On a $v_{k+1} = \mathcal{A}Tv_k$, où $\mathcal{A}$ est l'opérateur d'approximation. On peut calculer pour tout k l'approximation v_{k+1} en utilisant la méthode de régression : considérons $\mathcal{F}$ un espace fonctionnel donné, v_{k+1} est la projection du vecteur Tv_k sur l'espace $\mathcal{F}$ [Munos, 2007].

selon la norme $\|.\|_\xi$, où ξ est une distribution quelconque vérifiant $\xi(i) > 0$, pour tout i. La matrice de projection Π vérifie pour tout $u \in span(\Phi)$:

$$\Pi u = u = \Phi\alpha,$$

où α est un vecteur appartenant à $\mathbb{R}^d$. Pour tout $u \in span(\Phi)$ on a

$$\langle \Phi, u\rangle_\xi = \langle \Phi, \Phi\alpha\rangle_\xi, \tag{4.1}$$

où le produit scalaire $\langle .,.\rangle_\xi$ vérifie pour tous vecteurs u et v in $\mathbb{R}^{|S|}$, $\langle u, v\rangle_\xi = u^t D_\xi v$ où D_ξ est la matrice diagonale formée par les éléments du vecteur ξ. On a par conséquent

$$\Phi^t D_\xi u = \Phi^t D_\xi \Phi\alpha.$$

En supposant que

Hypothèse 1. *les features $(\phi_j)_{1\leq j\leq d}$ sont linéairement indépendantes,*

on a–grâce au fait que $\xi(i) > 0$ pour tout i– la matrice $\Phi^T D_\xi \Phi$ inversible (voir preuve B dans l'annexe B), il s'en suit que

$$\Pi = \Phi(\Phi^t D_\xi \Phi)^{-1}\Phi D_\xi. \tag{4.2}$$

On décrira dans ce qui suit certaines méthodes de projection qui existent dans la littérature et les différents algorithmes établis pour calculer les estimations $\hat{v}$.

4.2.1 La méthode de projection directe

La méthode de projection directe consiste à calculer la projection orthogonale de la solution de l'équation $v = Tv$ sur l'espace $span(\Phi)$. Cette méthode n'existe pas dans la littérature. On la décrit ici car elle paraît assez naturelle et simple. Elle consiste précisément à calculer la projection orthogonale $\Pi(I - \gamma P)^{-1}r$ de la solution $(I - \gamma P)^{-1}r$. En écrivant $(I - \gamma P)^{-1} = \sum_{i=0}^{\infty}(\gamma P)^i$, cette quantité équivaut à

$$\Pi(I - \gamma P)^{-1}r = \Phi(\Phi^t D_\xi \Phi)^{-1}\Phi^t D_\xi r + \gamma\Phi(\Phi^t D_\xi \Phi)^{-1}\Phi^t D_\xi Pr + \gamma^2\Phi(\Phi^t D_\xi \Phi)^{-1}\Phi^t D_\xi P^2 r + ...$$

Considérons les matrices A et b telles que

$$A = (\Phi^t D_\xi \Phi),\ b = \Phi^t D_\xi (I + \gamma P + \gamma^2 P^2 + ...)r.$$

Il s'en suit que

$$\Pi(I - \gamma P)^{-1}r = \Phi A^{-1}b.$$

Pour estimer les matrices A et b on procède comme suit : soient n trajectoires indépendantes $(X_i^j)_{j\geq 1}$ pour $i \in \{1, 2, ..., n\}$ telles que pour tous i, j, $X_i^0 \sim \xi$ et $X_i^{j+1} \sim P_\pi(.|X_i^j)$, avec $i \in \{1, ..., n\}$. Soient $\hat{A}$ et $\hat{b}$ les matrices qui sont construites à partir des échantillons X_i^j, on a

$$\hat{A} = \frac{1}{n}\sum_{i=1}^{n}\phi(X_i^0)\phi^t(X_i^0)$$
$$\hat{b} = \frac{1}{n}\sum_{i=1}^{n}\phi(X_i^0)\sum_{j=0}^{\infty}\gamma^j r(X_i^j).$$

Puisque les variables $(X_i)_{i\geq 1}$ sont iid, on a pour tout i

$$\mathbb{E}_\xi[\phi(X_i^0)\phi^t(X_i^0)] = \Phi^t D_\xi \Phi = A$$
$$\mathbb{E}_\xi\left[\phi(X_i^0)\sum_{j=0}^{\infty}\gamma^j r(X_i^j)\right] = \Phi^t D_\xi \sum_{j=0}^{\infty}\gamma^j P^j r = b,$$

où $\mathbb{E}_\xi[.]$ est l'espérance de la variable [.] sachant que le premier échantillon a été tiré selon la loi ξ. Les matrices $\hat{A}$ et $\hat{b}$ sont alors des estimateurs non biaisés de A et b. La méthode de projection directe renvoie asymptotiquement la meilleure approximation $\hat{v} = \Phi\hat{A}^{-1}\hat{b}$ de v sur $span(\Phi)$. Cependant elle requiert généralement un nombre de trajectoires n assez longues [4]$(X_i^0, X_i^1, ...)$, ce qui la rend très peu utilisable. On lui préfère ainsi d'autres méthodes de projection telle que la méthode du point fixe projeté qu'on développe dans ce qui suit.

4.2.2 Les Méthodes du point fixe

On distingue dans ce qui suit deux méthodes du point fixe projeté : la méthode du résidu de Bellman et la méthode des différences temporelles.

Méthode du résidu de Bellman

La méthode du résidu de Bellman a été proposée par [Schweitzer and Seidman, 1985] afin d'approcher la fonction de valeur dans un modèle donné. Elle consiste à minimiser l'erreur entre le vecteur $\hat{v} = \Phi\theta$ et $T\hat{v}$ selon la norme $\|.\|_\xi$. Elle estime le vecteur θ_{res} tel que

$$\begin{aligned}\theta_{res} &= \arg\min_{\theta\in\mathbb{R}^d} \|\Phi\theta - T\Phi\theta\|_\xi \\ &= \arg\min_{\theta\in\mathbb{R}^d} \|(I-\gamma P)\Phi\theta - r\|_\xi.\end{aligned}$$

Le vecteur θ_{res} vérifie $\nabla_{\theta_{res}} g(\theta) = 0$ avec

$$g(\theta) = \sqrt{\theta^t\Phi^t(I-\gamma P)^t D_\xi(I-\gamma P)\Phi\theta - \theta^t\Phi^t(I-\gamma P)^t D_\xi r - r^t D_\xi(I-\gamma P)\Phi\theta + r^t D_\xi r}.$$

Déterminer le vecteur θ_{res} équivaut à résoudre une équation de type $A\theta_{res} = b$ telle que

$$A = \Phi^t(I-\gamma P)^t D_\xi(I-\gamma P)\Phi,\ b = \Phi^t(I-\gamma P)^t D_\xi r.$$

La matrice A est une matrice symétrique. En effet elle peut s'écrire de la sorte

$$A = G^t G, \text{ avec } G = \sqrt{D}_\xi(I-\gamma P)\Phi$$

donc elle est forcément semi-définie positive. Si de plus on a les hypothèses 1 et 2 qui sont vérifiées, on sait d'après [Schoknecht, 2002] que la matrice A est définie positive donc inversible. D'où

$$\theta_{res} = A^{-1}b.$$

Pour calculer une estimation de θ_{res} on procède comme suit ; on considère les échantillons $(X_i, r(X_i), Y_i, Y_i')$ tels que les variables $(X_i)_{i\geq 1}$ sont des variables iid de loi ξ, les variables Y_i et

4. On peut certes toujours tronquer la trajectoire et obtenir une erreur qui dépend de $\gamma^L V_{\max}$, où L est la longueur de la trajectoire.

Algorithme 6 Least Square Bellman Residual Algorithm

Soient $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$.
Pour $i = 0, 1, 2, ..., n$ **faire**
$\hat{A} \leftarrow \hat{A} + (\phi(X_i) - \gamma\phi(Y_i))(\phi^t(X_i) - \gamma\phi^t(Y_i'))$.
$\hat{b} \leftarrow \hat{b} + (\phi(X_i) - \gamma\phi(Y_i))r(X_i)$.
fin « Pour »
$\hat{\theta} = \hat{A}^{-1}\hat{b}$.

Y_{i+1} sont deux variables indépendantes générées selon la loi $P_\pi(.|X_i)$. Soient $\hat{A}$ et $\hat{b}$ les matrices qui vérifient respectivement

$$\hat{A} = \frac{1}{n}\sum_{i=1}^{n}(\phi(X_i) - \gamma\phi(Y_i))(\phi^t(X_i) - \gamma\phi^t(Y_i'))$$
$$\hat{b} = \frac{1}{n}\sum_{i=1}^{n}(\phi(X_i) - \gamma\phi(Y_i))r(X_i).$$

Notons $\sigma(X_1, ..., X_i)$ la sigma-algèbre générée par les variables $(X_j)_{j\leq i}$ on a

$$\begin{aligned}&\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_i))(\phi^t(X_i) - \gamma\phi^t(Y_i'))]\\ &= \mathbb{E}_\xi[\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_i'))(\phi^t(X_i) - \gamma\phi^t(Y_i'))|\sigma(X_1, ..., X_i)]].\end{aligned}$$

Puisque les variables Y_i et Y_i' sont indépendantes on a

$$\begin{aligned}&\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_i))(\phi^t(X_i) - \gamma\phi^t(Y_i'))]\\ &= \mathbb{E}_\xi[\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_i'))|\sigma(X_1, ..., X_i)]\mathbb{E}[(\phi^t(X_i) - \gamma\phi^t(Y_i'))|\sigma(X_1, ..., X_i)]]\\ &= \mathbb{E}_\xi[(\phi(X_i) - \gamma P\phi(X_i))(\phi^t(X_i) - \gamma P\phi^t(X_i))].\end{aligned}$$

Les variables $(X_i)_{i\geq 1}$ sont iid, il s'en suit que pour tout $i \geq 1$, $\mathbb{E}_\xi[(\phi(X_i) - \gamma P\phi(X_i))(\phi^t(X_i) - \gamma P\phi^t(X_i))] = A$. De même on a pour tout i

$$\begin{aligned}\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_i))r(X_i)] &= \mathbb{E}[\mathbb{E}_\xi[(\phi(X_i) - \gamma\phi(Y_{i+1}))r(X_i)|\sigma(X_1, ..., X_i)]]\\ &= \mathbb{E}[\mathbb{E}_\xi[(\phi(X_i) - \gamma P\phi(X_i))r(X_i)]] = b.\end{aligned}$$

Les matrices $\hat{A}$ et $\hat{b}$ sont donc des estimateurs non biaisés de A et b. Une analyse en échantillons finis de cet algorithme a été établie dans [Maillard et al., 2010]. L'algorithme *Least square Bellman residual* LSBR (cf. algorithme 6) repose sur la méthode du résidu de Bellman. Il calcule les estimations $\hat{A}$ et $\hat{b}$ en utilisant les échantillons $(X_i, Y_i, Y_i')_{i\geq 1}$. LSBR renvoie asymptotiquement le vecteur θ défini par $\theta_{res} = A^{-1}b$.

Méthode des différences temporelles

La méthode des différences temporelles consiste à minimiser la distance entre $\hat{v}$ qui appartient à l'espace des features et $T\hat{v}$ qui a priori n'y appartient pas. La plus petite distance est atteinte lorsque $\hat{v} - T\hat{v}$ est orthogonal à $span(\Phi)$ et donc orthogonal à $\hat{v}$ qui appartient à $span(\Phi)$. Notons $v_{\theta_{TD}}$ cette valeur, on a

$$\langle v_{\theta_{TD}}, v_{\theta_{TD}} - Tv_{\theta_{TD}}\rangle_\xi = 0.$$

Considérons Π l'opérateur de projection orthogonale. En appliquant Π à l'équation précédente, on obtient

$$\langle \Pi v_{\theta_{TD}}, \Pi(v_{\theta_{TD}} - T v_{\theta_{TD}}) \rangle_\xi = 0 \Rightarrow \langle v_{\theta_{TD}}, \Pi(v_{\theta_{TD}} - T v_{\theta_{TD}}) \rangle_\xi = 0.$$

Il suffit donc que $\Pi(v_{\theta_{TD}} - T v_{\theta_{TD}}) = 0$ pour annuler le produit scalaire défini plus haut. Le vecteur θ_{TD} vérifie

$$\Pi v_{\theta_{TD}} = \Pi T v_\theta \Leftrightarrow v_{\theta_{TD}} = \Pi T v_{\theta_{TD}}.$$

On conclut donc que θ_{TD} est une solution de l'équation $\Phi\theta = \Pi T \Phi\theta$. En remplaçant les variables par leurs expressions on obtient

$$\Phi\theta = \Phi(\Phi^t D_\xi \Phi)^{-1} \Phi^t D_\xi (r + \gamma P \Phi\theta).$$

Ceci est équivalent à résoudre une équation du type $A\theta = b$ avec $A = \Phi^t D_\xi (I - \gamma P)\Phi$ et $b = \Phi^t D_\xi R$. En général, on n'a aucune garantie sur l'existence d'une solution à l'équation $A\theta = b$. Pour cela, il est usuel de faire une hypothèse de plus sur la chaîne de Markov $\mathcal{M}$.

Hypothèse 2. *$\mathcal{M}$ admet une mesure invariante μ telle que $\mu = \mu P_\pi$. Cette mesure vérifie $\mu(i) > 0$ pour tout $i \in \{1, ..., n\}$.*

Sous cette hypothèse, on a $\|\Pi\|_\mu = 1$ [Tsitsiklis and Roy, 1997] d'où ΠT est un opérateur $(\|.\|_\mu, \gamma)$-contractant. Par le théorème du point fixe de Banach l'équation $v = \Pi T v$ admet une unique solution $v_{\theta_{TD}} = \Phi\theta_{TD}$. Ceci prouve l'existence d'une solution à l'équation $A\theta = b$. Sous les hypothèses 1 et 2, on a $\Phi^t D_\mu \Phi$ inversible et donc θ_{TD} est unique et vérifie

$$\theta_{TD} = (\Phi^t D_\mu \Phi)^{-1} \Phi^t D_\mu v_{\theta_{TD}}.$$

Pour calculer une estimation de θ_{TD} on procède comme suit : on considère les échantillons $(X_i)_{i \geq 1}$ générés à partir de la chaîne de Markov $\mathrm{CM}(\xi, P)$ (cette notation signifie de mesure initiale ξ et de matrice de transition P) et les matrices $\hat{A}$ et $\hat{b}$ telles que

$$\hat{A} = \frac{1}{n-1} \sum_{i=1}^{n-1} \phi(X_i)(\phi^t(X_i) - \gamma \phi^t(X_{i+1}))$$
$$\hat{b} = \frac{1}{n-1} \sum_{i=1}^{n-1} \phi(X_i) r(X_i).$$

En utilisant une version de la loi des grands nombres [Nedic and Bertsekas, 2002] (voir le théorème 13 et la proposition 9 dans l'annexe B) on peut montrer que

$$\hat{A} \overset{n \to \infty}{\longrightarrow} \mathbb{E}_\mu[\phi(X_i)(\phi^t(X_i) - \gamma \phi^t(X_{i+1}))] = A$$
$$\hat{b} \overset{n \to \infty}{\longrightarrow} \mathbb{E}_\mu[\phi(X_i) r(X_i)] = b.$$

L'algorithme *Least square temporal difference* (cf. algorithme 7), introduit par [Bradtke and Barto, 1996], utilise la méthode des différences temporelles. Il construit à chaque pas de temps une estimation $\hat{A}$ et $\hat{b}$ de A et b et renvoie asymptotiquement le vecteur $\theta_{TD} = A^{-1} b$. Il existe une version *récursive* de cet algorithme qu'on décrira une fois qu'on introduit LSTD(λ) qui est plus général.

Algorithme 7 Least Square Temporal Difference Algorithm LSTD(0)

Entrée : π la politique à évaluer
Générer :la trajectoire $(X_0, a_0, r_1, X_1, a_1, r_2, ...)$ sous π
Initialiser : $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$.
Pour $t = 1, 2, ...$ **faire**
  $\hat{A} \leftarrow \hat{A} + \phi(X_{t-1})(\phi(X_{t-1}) - \gamma\phi(X_t))^t$.
  $\hat{b} \leftarrow \hat{b} + \phi(X_{t-1})r(X_{t-1})$.
  $t \leftarrow t + 1$.
fin « Pour »
$\hat{\theta} = \hat{A}^{-1}\hat{b}$.

4.2.3 La méthode du résidu de Bellman et la méthode des différences temporelles : une vision unifiée

[Scherrer, 2010] a proposé une vision unifiée des méthodes des différences temporelles (TD) et du résidu de Bellman (BR). Soit Π n'importe quel opérateur de projection sur $span(\Phi)$ vérifiant $\Pi^2 = \Pi$, les estimations $\hat{v}_{TD}$ et $\hat{v}_{BR}$ induites respectivement par les méthodes de différences temporelles et la méthode du résidu de Bellman sont des projections *obliques* de v sur $span(\Phi)$. On a

$$\hat{v}_{TD} = \Pi_{L^t X_{TD}} v, \ \hat{v}_{BR} = \Pi_{L^t X_{BR}} v$$

où la notation $\Pi_{L^t X}$ signifie orthogonalement à l'espace $span(L^t X)$, avec $L = (I - \gamma P)$, $X_{TD} = D_\xi \Phi$ et $X_{BR} = D_\xi L \Phi$. On a en particulier $\hat{v}_{TD}$ et $\hat{v}_{BR}$ solutions des équations de Bellman

$$\hat{v}_{TD} = \Pi_{X_{TD}} T\hat{v}_{TD}, \text{ et } \hat{v}_{BR} = \Pi_{X_{BR}} T\hat{v}_{BR}.$$

De cette vision on peut déduire une borne sur l'erreur d'approximation $\|v - \hat{v}_X\|_\xi$, X désigne la nature de la méthode utilisée. Puisque $\Pi_{L^t X}$ et $\Pi_{D_\xi \Phi}$ sont deux projections sur l'espace $span(\Phi)$, on peut écrire que $\Pi_{L^t X}\Pi_{D_\xi \Phi} = \Pi_{D_\xi \Phi}$. Par conséquent

$$\begin{aligned} \|v - \hat{v}_X\|_\xi = \|v - \Pi_{L^t X} v\|_\xi &= \|(I - \Pi_{L^t X})(I - \Pi_{D_\xi \Phi})v\|_\xi \\ &\leq \|\Pi_{L^t X}\|_\xi \|v - \hat{v}_{best}\|_\xi, \end{aligned}$$

où $\hat{v}_{best}$ est la projection *orthogonale* de v sur $span(\Phi)$ (noter que pour tout Π opérateur de projection on a l'égalité $\|(I - \Pi_{L^t X})\|_\xi = \|\Pi_{L^t X}\|_\xi$). [Yu and Bertsekas, 2008] ont également dérivé ces bornes mais l'analyse proposée par [Scherrer, 2010] à travers la notion de projection oblique permet de simplifier d'une manière significative le calcul.

L'algorithme LSTD(λ)

L'algorithme *Least square temporal difference* avec traces d'éligibilité LSTD(λ) (cf. algorithme 8) a été introduit par Boyan [Boyan, 2002]. Considérons T^λ l'opérateur défini par

$$T^\lambda = (1 - \lambda) \sum_{i=0}^{\infty} \lambda^i T^{i+1}.$$

LSTD(λ) renvoie une approximation de la fonction v qui vérifie $v = T^\lambda v$ (notons ici que pour tout λ, v est solution de $v = Tv$ si et seulement si elle est solution de $v = T^\lambda v$). Il calcule ainsi

Algorithme 8 Least Square Temporal Difference Algorithm LSTD(λ)

Entrée : π la politique à évaluer
Générer : la trajectoire $(X_0, a_0, r_1, X_1, a_1, r_2, ...)$ sous π
Initialiser : $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$, $z_0 := 0$.
Pour $t = 1, 2, ...$ **faire**
  $\hat{A} \leftarrow \hat{A} + z_{t-1}(\phi(X_{t-1}) - \gamma\phi(X_t))^t$.
  $\hat{b} \leftarrow \hat{b} + z_{t-1}r(X_{t-1})$.
  $z_t \leftarrow \lambda\gamma z_{t-1} + \phi(X_t)$.
  $t \leftarrow t + 1$.
fin « Pour »
Renvoyer : $\hat{\theta} = \hat{A}^{-1}\hat{b}$.

une solution à l'équation $v = T^\lambda v$ projetée sur l'espace des features, $v = \Pi T^\lambda v$. Cet algorithme fait intervenir en plus le paramètre $\lambda \in [0, 1]$. Pour $\lambda = 0$, on retrouve l'algorithme LSTD(0) (cf. algorithme 7) introduit par [Bradtke and Barto, 1996]. L'algorithme LSTD(1) renvoie la projection orthogonale Πv de v qui est la meilleure approximation $\hat{v}$ de v. LSTD(λ) fait ainsi le pont entre LSTD(0) et la méthode des projection directe introduite dans le paragraphe précédent. Le paramètre λ permet d'avoir une meilleure approximation de la fonction de valeur v. En effet on peut s'apercevoir que l'opérateur T^λ est $(\gamma_\lambda, \|.\|_\mu)$-contractant avec $\gamma_\lambda = \frac{(1-\lambda)\gamma}{1-\gamma\lambda}$ [Tsitsiklis and Roy, 1997]. On a pour $u, w \in \mathbb{R}^N$

$$\begin{aligned}\|T^\lambda u - T^\lambda w\|_\mu &= \|(1-\lambda)\gamma \sum_{i=0}^{\infty} (\lambda\gamma)^i P^{i+1}(u-w)\|_\mu \\ &\leq \frac{(1-\lambda)\gamma}{1-\gamma\lambda}\|(u-w)\|_\mu.\end{aligned}$$

Il est clair que prendre de grandes valeurs de λ rend l'opérateur T^λ encore plus contractant [5]. L'équation $v = \Pi T^\lambda v$ est équivalente à une équation de type $A\theta = b$ avec

$$A = \Phi^t D_\mu (I - \gamma P)(I - \lambda\gamma P)^{-1}\Phi, \; b = \Phi^t D_\mu (I - \gamma P)^{-1} r. \tag{4.3}$$

LSTD(λ) calcule à chaque pas de temps les estimations $\hat{A}$ et $\hat{b}$ de respectivement A et b,

$$\hat{A} = \frac{1}{n-1}\sum_{i=1}^{n-1} z_i(\phi^t(X_i) - \gamma\phi^t(X_{i+1})) \tag{4.4}$$

$$\hat{b} = \frac{1}{n-1}\sum_{i=1}^{n-1} z_i r(X_i). \tag{4.5}$$

Le vecteur z_i est le vecteur trace d'éligibilité qui vérifie

$$z_i = \sum_{k=1}^{i} (\lambda\gamma)^{i-k}\phi(X_k). \tag{4.6}$$

5. Même si théoriquement faire augmenter λ devrait améliorer la qualité de l'approximation θ, il s'avère expérimentalement que ce sont les valeurs intérmédiaires du paramètre λ qui l'emportent [Nedic and Bertsekas, 2002].

L'introduction de ce vecteur revient à prendre en compte des échantillons précédents. Pour $\lambda = 0$, seul l'échantillon courant est considéré. Dès que $\lambda > 0$ c'est toute la trajectoire à partir de l'échantillon X_1 qui est considérée. Quand le nombre d'échantillons tend vers l'infini LSTD(λ) converge vers le vecteur θ qui vérifie $\theta = A^{-1}b$ [Nedic and Bertsekas, 2002].

Le calcul du vecteur $\hat{\theta} = \hat{A}^{-1}\hat{b}$ requiert l'inversibilité de la matrice $\hat{A}$. A partir d'un certain nombre d'échantillons cette matrice est inversible puisque A l'est. Ceci dit pour s'assurer de l'inversibilité de $\hat{A}$ à chaque pas de temps on peut utiliser la variation proposée par [Nedic and Bertsekas, 2002] en initialisant la matrice $\hat{A}$ à $\hat{A} = \delta I$ où $\delta > 0$. L'inversion de la matrice $\hat{A}$ nécessite un temps de l'ordre de $O(d^3)$. Une implémentation plus efficace de l'algorithme LSTD(λ) utilise une version récursive pour calculer l'inverse de $\hat{A}$. Cette version repose sur la formule de Shermann-Morrison :

$$\begin{aligned} B_t &= \left(A_{t-1} + z_t(\phi(X_t) - \gamma\phi(X_{t+1}))^t\right)^{-1} \\ &= B_{t-1} - \frac{B_{t-1}z_t(\phi(X_t) - \gamma\phi(X_{t+1}))^t B_{t-1}}{1 + z_t(\phi(X_t) - \gamma\phi(X_{t+1}))^t B_{t-1} z_t}. \end{aligned}$$

On obtient la version *récursive* ou *en ligne* de LSTD(λ) (cf. algorithme 9).

Algorithme 9 LSTD(λ)-REC

Entrée : π la politique à évaluer
Générer : la trajectoire $(X_0, a_0, r_1, X_1, a_1, r_2, ...)$ sous π
Initialiser : $B := \frac{1}{\delta}I$, $\hat{b} := 0$, $t := 0$, $z_0 := 0$.
Pour $t = 1, 2, ...$ **faire**
  $B \leftarrow B - \frac{Bz_{t-1}(\phi(X_{t-1}) - \gamma\phi(X_t))^t B}{1 + z_{t-1}(\phi(X_{t-1}) - \gamma\phi(X_t))^t B z_{t-1}}$.
  $\hat{b} \leftarrow \hat{b} + z_{t-1}r(X_{t-1})$.
  $z_t \leftarrow \lambda\gamma z_{t-1} + \phi(X_t)$.
  $t \leftarrow t + 1$.
fin « Pour »
Renvoyer : $\hat{\theta} = B\hat{b}$.

Grâce à cette dernière version on passe à $O(d^2)$ et on gagne $O(d)$ lors de l'exécution de LSTD(λ).

Off policy LSTD(λ)

On a considéré une version "on policy" de l'algorithme LSTD(λ) où la trajectoire infinie est générée par la politique qu'on souhaite évaluer. Il existe une version "off policy" pour évaluer une politique π_2 à partir de la trajectoire générée par la politique π_1, dite d'exploration, qui est l'algorithme : off policy LSTD(λ). En exploitant les observations générées par la politique *d'exploration* on peut approcher la valeur de n'importe quelle politique π_2. La politique d'exploration induit une chaîne de Markov de matrice de transition P_{π_1} et la politique à évaluer induit une chaîne de Markov de matrice transition P_{π_2}. Dans le cas où ces deux politiques coïncident on retrouve la version on-line de l'algorithme LSTD(λ) introduite dans le paragraphe précédent. Soit $(X_i)_{i\geq 1}$ une chaîne de Markov de matrice de transition P_{π_1}, l'algorithme off policy LSTD(λ) calcule les estimations $\hat{A}$ et $\hat{b}$ correspondantes

$$\hat{A} = \frac{1}{n}\sum_{i=1}^{n} z_i \left(\gamma \frac{\pi_2(a_i|X_i)}{\pi_1(a_i|X_i)}\phi(X_i) - \phi(X_{i+1})\right) \text{ et } \hat{b} = \frac{1}{n}\sum_{i=1}^{n} z_i r(X_i).$$

Le vecteur z_i est le vecteur trace d'éligibilité qui vérifie

$$z_i = \sum_{j=1}^{i} \left(\lambda\gamma \frac{\pi_2(a_{j-1}|X_{j-1})}{\pi_1(a_{j-1}|X_{j-1})} \right)^j \phi(X_{j-1}).$$

Une preuve de la convergence de cet algorithme a été établie premièrement dans le cas où $\lambda\gamma\frac{\pi_2(a_{j-1}|X_{j-1})}{\pi_1(a_{j-1}|X_{j-1})} < 1$ (ce cas comprend la version on-line) dans [Bertsekas and Yu, 2009]. Puis elle a été étendue dans le cas général dans [Yu, 2010]. En effet l'auteure a prouvé que si la politique d'exploration induisait une chaîne de Markov irréductible et que si elle choisissait les mêmes actions que celle de la politique à évaluer avec probabilité non nulle (i.e., pour tout s et pour tout a, $\pi_1(a|s) \prec \pi_2(a|s)$) alors l'algorithme off policy LSTD(λ) converge vers le vecteur θ défini par $\theta = A^{-1}b$, on a $\hat{A} \to A$ et $\hat{b} \to b$ avec

$$A = \Phi^T D_{\pi_1}(I - \gamma P_{\pi_2})(I - \lambda\gamma P_{\pi_2})^{-1}\Phi,\ b = \Phi^T D_{\pi_1}(I - \lambda\gamma P_{\pi_2})^{-1} r.$$

Le vecteur $\Phi A^{-1}b$ est une approximation du coût induit selon la politique π_2. Cette approximation est solution de l'équation $v = \Pi T^{\lambda} v$, où Π est la matrice de projection sur l'espace $span(\Phi)$ selon la norme pondérée par la mesure invariante induite par la politique d'exploration π_1.

Algorithme 10 Off-policy LSTD(λ)

Entrées : la politique π_1 d'exploration, la politique π_2 à évaluer
Générer : la trajectoire $(X_0, a_0, r_1, X_1, a_1, r_2, ...)$ sous π_1
Initialiser : $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$, $z_0 := 0$.
Pour $t = 1, 2, ...$ **faire**
  $\hat{A} \leftarrow \hat{A} + z_{t-1}\left(\gamma\frac{\pi_2(a_{t-1}|X_{t-1})}{\pi_1(a_{t-1}|X_{t-1})}\phi(X_t) - \phi(X_{t-1})\right)^t$.
  $\hat{b} \leftarrow \hat{b} + z_{t-1} r(X_{t-1})$.
  $z_{t+1} \leftarrow \lambda\gamma\frac{\pi_2(a_{t-1}|X_{t-1})}{\pi_1(a_{t-1}|X_{t-1})} z_{t-1} + \phi(X_t)$.
fin « Pour »
Renvoyer : $\hat{\theta} = \hat{A}^{-1}\hat{b}$.

4.3 Description de l'algorithme LSPI

Lorsqu'on utilise un algorithme de type LSTD(0) dans un schéma de type itérations sur les politiques on obtient l'algorithme *Least square policy iteration* LSPI. L'algorithme 11 est celui qui a été décrit dans [Lazaric et al., 2012] et qui a été fortement inspiré de [Lagoudakis and Parr, 2003]. Il utilise une version de LSTD qui fait intervenir les fonctions d'actions états Q, l'algorithme LSTDQ. Les fonctions features ϕ dépendent dans ce cas non plus des états mais des couples états-actions (s, a). La matrice des features correspondante est une matrice de taille $Nm \times d$ (N désignant la taille de l'espace des états, m celle de l'espace des actions). A l'instar de LSTD, cet algorithme construit à chaque itération k, les matrices $\hat{A}$ et $\hat{b}$ à partir de la trajecoire $(X_0, a_0, r_0, X_1, a_1, r_1, ..., X_N)$ telle que $a_i = \mathcal{G}(\theta_k, X_i)$ avec

$$\forall s, \theta,\ \mathcal{G}(\theta, s) = \arg\max_{a \in \mathcal{A}} \phi(s, a)^t \theta.$$

La matrice $\hat{A}$ est inversible à partir d'un certain nombre d'échantillons, on peut utiliser les mêmes méthodes que celles développées pour LSTD pour garantir l'inversibilité de $\hat{A}$ à chaque mise à jour.

Algorithme 11 Least Square Policy Iteration Algorithm LSPI

Entrée : $\nu, N, \theta_1, \gamma, \phi$
Pour $k = 1, 2, ...$ **faire**
  Générer : $X_0 \sim \nu$
  Générer : la trajectoire de taille N, $(X_0, a_0, r_0, X_1, a_1, r_1, ..., X_N)$ telle que $a_i = \mathcal{G}(\theta_k, X_i)$
  $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$, $z_0 := 0$
  Pour $t = 1, 2, ..., N$ **faire**
    $\hat{A} \leftarrow \hat{A} + \phi(X_{t-1}, a_{t-1})(\phi(X_{t-1}, a_{t-1}) - \gamma\phi(X_t, a_t))^t$
    $\hat{b} \leftarrow \hat{b} + \phi(X_{t-1}, a_{t-1})r(X_{t-1}, a_{t-1})$
  fin « Pour »
  Renvoyer : $\theta_{k+1} \leftarrow \hat{A}^{-1}\hat{b}$
fin « Pour »

4.3.1 Comparaison des différentes méthodes de projection

Une comparaison des méthodes des différences temporelles et du résidu de Bellman existe dans [Munos, 2003]. La méthode des différences temporelles paraît moins stable et moins prévisible que la méthode du résidu de Bellman. En effet le système linéaire à résoudre imposé par cette méthode n'admet pas de solution pour certaines valeurs de γ, on peut se référer à [Scherrer, 2010] pour des exemples. Elle exige certains critères sur la distribution initiale ξ pour garantir l'existence d'une solution. Ce qui n'était pas le cas pour l'autre méthode où on échantillonnait par rapport à n'importe quelle mesure initiale ξ. Ceci dit la méthode des différences temporelles est préférable (dans le cas où il existe une solution). En effet, on a l'erreur $E_{TD(0)}$ qui est bornée par l'erreur E_{BR} [Scherrer, 2010]. Expérimentalement, la méthode des différences temporelles présente certains avantages puisqu'elle utilise un échantillonnage simple à comparer de la méthode (BR) qui nécessite un double échantillonnage. Il est cependant difficile de trancher entre les deux méthodes. Bien que la méthode des différences temporelles (pour $\lambda = 0$) soit souvent meilleure que la méthode du résidu de Bellman (en termes d'expériences effectuées), il s'avère que l'erreur en moyenne induite soit plus grande, ceci à cause des problèmes de stabilité dont on vient de parler. Une façon de surmonter ce problème est de considérer des valeurs strictement positives de λ. En effet on peut montrer que les problèmes de stabilité disparaissent pour des valeurs de λ proches de 1 ($\lambda = 1$ renvoie la projection orthogonale de v). Une comparaison de ces méthodes pour tout $\lambda \in [0, 1]$ demeure une question à investiguer.

4.3.2 Conclusion

On s'est intéressé dans ce chapitre aux différentes méthodes d'évaluation approchées d'une politique. On a ainsi introduit les méthodes des moindres carrés. Nous avons décrit la méthode de projection directe, celle du résidu de Bellman ainsi que celle des différences temporelles. Nous avons aussi présenté les algorithmes les plus connus qui sont basés sur les deux dernières approches. Suite à cette présentation, nous avons proposé une analyse comparative des méthodes de projection introduites, inspirée des travaux de [Scherrer, 2010]. Les chapitres qui suivent portent principalement sur la méthode des différences temporelles et plus précisément sur l'algorithme LSTD(λ). On analyse dans le chapitre suivant cet algorithme en estimant une borne d'erreur sur sa vitesse de convergence en fonction du paramètre λ–qui le caractérise.

5

Vitesse de convergence et calcul d'une borne d'erreur de LSTD(λ)

Dans ce chapitre nous allons étudier l'algorithme LSTD(λ) introduit dans le chapitre 4. Cet algorithme renvoie une meilleure approximation que LSTD(0) sans être beaucoup plus difficile à implémenter (si on considère la version récursive). En effet, le paramètre λ intervient dans la qualité de la borne d'erreur entre la valeur asymptotique calculée par l'algorithme et la valeur effective v [Tsitsiklis and Roy, 1997]. En faisant varier λ de 0 à 1, on passe d'une façon continue de la projection oblique de v à sa projection orthogonale [Scherrer, 2010]. Nous allons plus précisément calculer une borne sur la vitesse de convergence de cet algorithme en fonction du nombre d'échantillons générés et du paramètre λ. Cette partie constitue une preuve détaillée du théorème 5 (page 65) qui estime la borne d'erreur de LSTD(λ). Nous allons dans ce cadre introduire les outils mathématiques nécessaires, avant d'entrer par la suite dans les détails plus techniques.

5.1 Problématique

L'algorithme LSTD(λ) (voir chapitre 4) calcule pour un nombre d'échantillons n une estimation $\hat{\theta}_{LSTD(\lambda)}$ de $\theta_{LSTD(\lambda)} = A^{-1}b$, où $A = \Phi^t D_\mu (I - \gamma P)(I - \lambda\gamma P)^{-1}\Phi$ et $b = \Phi^t D_\mu (I - \gamma P)^{-1} R$. Lorsque la matrice $\hat{A}$ est inversible on a $\hat{\theta}_{LSTD(\lambda)} = \hat{A}^{-1}\hat{b}$, avec

$$\hat{A} = \frac{1}{n-1}\sum_{i=1}^{n-1} z_i(\phi^t(X_i) - \gamma\phi^t(X_{i+1})), \text{ et } \hat{b} = \frac{1}{n-1}\sum_{i=1}^{n-1} z_i r(X_i).$$

Étant donnée que la matrice A est inversible et que $\hat{A} \to A$, la matrice $\hat{A}$ va l'être également avec grande probabilité à partir d'un nombre d'échantillons $n \geq n_0$. Notons $\hat{v}_{LSTD(\lambda)}$ l'estimation de $v_{LSTD(\lambda)}$ pour un nombre d'échantillons n, on a d'après [Nedic and Bertsekas, 2002]

$$\hat{v}_{LSTD(\lambda)} = \Phi\hat{A}^{-1}\hat{b}, \quad \hat{v}_{LSTD(\lambda)} \overset{n\to\infty}{\longrightarrow} v_{LSTD(\lambda)}.$$

Notre objectif ici est de déterminer la vitesse à laquelle converge $\hat{v}_{LSTD(\lambda)}$ vers $v_{LSTD(\lambda)}$ en fonction du nombre d'échantillons n et du paramètre λ. Ceci est possible une fois qu'on a contrôlé la vitesse à laquelle converge $\hat{A}$ vers A et $\hat{b}$ vers b. Les méthodes classiques auxquelles on a recours pour contrôler les différences $\|\hat{A} - A\|_2$ et $\|\hat{b} - b\|_2$ font intervenir les inégalités de concentration qu'on développe dans le paragraphe suivant.

5.2 Outils Mathématiques

Nous présentons dans cette partie les principaux outils mathématiques dont on va se servir dans la preuve de notre résultat principal.

5.2.1 Les inégalités de concentration

Une variable est dite *concentrée* si elle reste assez proche d'une quantité déterministe telle que la moyenne ou la médiane avec une forte probabilité [Chafai, 2012]. Les inégalités de *concentration* renvoient une borne sur la probabilité avec laquelle ces variables restent proches. Soit Y une variable qui peut être décrite comme la somme de variables indépendantes. L'inégalité de *Bernstein* fournit une borne–avec une certaine probabilité– sur la déviation de la variable Y par rapport à son espérance $\mathbb{E}[Y]$.

Théorème 2. *Soient $X_1, X_2, ..., X_n$ des variables indépendantes qui satisfont $|X_k - \mathbb{E}[X_k]| \leq B$ pour tout $k \in \{1, ..., n\}$. Soit $Y = \sum_{k=1}^{n} X_k$ et σ la variance de Y telle que $\sigma = \sqrt{\mathbb{E}[(Y - \mathbb{E}[Y])^2]}$. On a pour $\epsilon \geq 0$*

$$\mathbb{P}(|Y - \mathbb{E}[Y]| \geq \epsilon) \leq 2 \exp\left(-\frac{\epsilon^2}{\sigma^2 + B\frac{\epsilon}{3}}\right).$$

Cette inégalité a été prouvée par *Sergei Bernstein* [Bernstein, 1927], elle concerne les variables aléatoires scalaires dont la déviation est bornée et qui possèdent une variance finie. Il existe d'autres inégalités de concentration qui sont des dérivées de l'inégalité de Bernstein dont l'inégalité de Hoeffding [Hoeffding, 1963] qui concerne les variables aléatoires bornées et indépendantes, ainsi que l'inégalité de Azuma-Hoeffding [Azuma, 1967] qui porte sur les martingales à accroissements bornés. On peut étendre l'inégalité de *Bernstein* valable pour les variables aléatoires scalaires au cas des matrices aléatoires. En d'autres termes il est possible de contrôler la déviation d'une matrice Y de dimension $d \times k$ par rapport à son espérance $\mathbb{E}[Y]$ (voir annexe B définition 20) selon la norme euclidienne $\|.\|_2$ vérifiant pour $M \in \mathbb{R}^{d \times k}$, $\|M\|_2 = \sup_{v \in \mathbb{R}^k} \frac{\|Mv\|_2}{\|v\|_2}$. Similairement à ce qui a été introduit précédemment, si on considère la matrice Y comme la somme de matrices indépendantes (voir annexe B définition 21), on a le résultat suivant.

Théorème 3. *Soient $X_1, ..., X_n$ des matrices aléatoires indépendantes de taille $d \times k$, telles qu'on ait :*

$$\|X_k - \mathbb{E}[X_k]\|_2 \leq B, \text{ pour tout } k \in \{1, ..., n\}.$$

Considérons $Y = \sum_{k=1}^{n} X_k$ et σ le paramètre de variance qui vérifie

$$\sigma^2 = \max\left\{\|\mathbb{E}[(Y - \mathbb{E}[Y])(Y - \mathbb{E}[Y])^t]\|_2, \|\mathbb{E}[(Y - \mathbb{E}[Y])^t(Y - \mathbb{E}[Y])]\|_2\right\}.$$

On a pour $\epsilon \geq 0$

$$\mathbb{P}(\|Y - \mathbb{E}[Y]\|_2 \geq \epsilon) \leq (d \times k) \exp\left(-\frac{\epsilon^2}{\sigma^2 + B\frac{\epsilon}{3}}\right).$$

L'énoncé de ce théorème ressemble à celui du théorème précédent. Nous remarquons néamoins que la borne dans le théorème 3 dépend de l'entier $d \times k$ qui est la taille de la matrice considérée. Ce facteur est réduit à l'entier 2 dans le cas du théorème 2. La borne peut être dans ce cas relativement grande dans le cas où la matrice Y est de taille plus ou moins grande. Il est possible de se défaire de cette dépendance en utilisant un résultat de Hayes.

Théorème 4. *Soit $Y_0, ..., Y_n$ une martingale*[6] *à temps discret prenant ses valeurs dans un espace euclidien telle que $Y_0 = 0$ et $\|Y_k - Y_{k-1}\|_2 \leq B$, pour tout $k \geq 1$. On a alors pour tout $\epsilon \geq 0$*

$$\mathbb{P}(\|Y_n\|_2 \geq \epsilon) \leq 2e^2 \exp\left(-\frac{\epsilon^2}{2nB^2}\right).$$

Le théorème de Hayes est une extension de l'inégalité d'Azuma-Hoeffding dans le cas des variables aléatoires vectorielles. Considérons maintenant le lemme qui découle du théorème 4.

Lemme 2. *Soient $X_0, ...X_n$ un ensemble de variables aléatoires indépendantes de $\mathbb{R}^d$, vérifiant $X_0 = 0$, $\|X_k - \mathbb{E}[X_k]\|_2 \leq B$ pour tout $k \in \{1, ..., n\}$. On a alors pour tout $\epsilon \geq 0$*

$$\mathbb{P}\left(\left\|\sum_{i=0}^{n} X_i - \mathbb{E}[X_i]\right\|_2 \geq \epsilon\right) \leq 2e^2 \exp\left(-\frac{\epsilon^2}{2nB^2}\right).$$

Démonstration. Considérons $Y_n = \sum_{i=0}^{n} X_i - \mathbb{E}[X_i]$, on peut montrer que Y_n est une $\sigma(Y_0, ..., Y_n)$ martingale. En effet Y_n est adaptée par rapport à la filtration $\sigma(Y_0, ..., Y_n)$. De plus on a

$$\mathbb{E}[Y_n|Y_0, ...Y_{n-1}] = \sum_{i=0}^{n-1} X_i - \mathbb{E}[X_i] + \mathbb{E}[X_n] - \mathbb{E}[X_n] = Y_{n-1}.$$

On peut remarquer que pour tout $k \in \{1, ..., n\}$ on a

$$\|Y_k - Y_{k-1}\|_2 = \|X_k - X_{k-1}\|_2 \leq B.$$

On peut alors appliquer le théorème 4 à Y_n et conclure. □

On remarque que la borne dépend du réel e^2 et non plus du facteur $d \times k$ qui peut être arbitrairement grand. Le terme nB^2 apparaissant dans l'exponentielle "remplace" la variance σ^2 dans le théorème 3. En effet on peut s'apercevoir que $\sigma^2 \leq nB^2$. On va se servir tout au long de ce chapitre du lemme 2 pour dériver les bornes d'erreur sur $\hat{A}$, $\hat{b}$ et $\hat{v}$.

5.2.2 Processus β-mélangeants

Les inégalités de concentration qu'on vient d'introduire nécessitent l'indépendance des variables $X_1, ..., X_n$. L'algorithme LSTD(λ) fait intervenir les estimations

$$\hat{A} = \frac{1}{n}\sum_{i=1}^{n} z_i \left(\phi(X_i) - \gamma\phi(X_{i+1})\right)^t \text{ et } \hat{b} = \frac{1}{n}\sum_{i=1}^{n} z_i r(X_i)$$

avec $z_i = \sum_{j=1}^{i} (\lambda\gamma)^{i-j}\phi(X_j)$, qui sont déduites à partir de la chaîne de Markov stationnaire CM(μ, P), μ étant la mesure invariante de la chaîne considérée. Les échantillons $X_1, ..., X_n$ sont donc dépendants. Il n'est pas possible dans ce cas d'appliquer directement les inégalités de concentration. Il faudrait faire des hypothèses en plus sur la chaîne de Markov CM(μ, P). Pour cela on supposera que les échantillons sont "faiblement dépendants" ou mélangeants. On dit qu'un processus est *mélangeant* ou satisfait la propriété de mélange si on a

6. voir définition 31 dans l'annexe B.

Définition 13.

$$\alpha(n) = \frac{1}{2}\sup\left\{\mathbb{E}\left[|\mathbb{P}(B|\sigma_1^l) - \mathbb{P}(B)|\right], B \in \sigma_{l+n}^{\infty}, l \geq 1\right\}.$$

où $\sigma_t^u = \sigma(X_t, ..., X_u)$ *pour* $t \leq u$, $\sigma(X_t, ..., X_u)$ *est la sigma algèbre engendrée par les variables* $(X_i)_{t \leq i \leq u}$.

La notion de mélange a été introduite pour la première fois par [Rosenblatt, 1956] pour prouver le théorème central limite dans le cas des variables faiblement dépendantes. Il existe plusieurs processus qui satisfont la propriété de mélange. Chanda [Chanda, 1974] a montré qu'une certaine classe de processus stochastiques linéaires (voir définition 26 dans l'annexe B)–dont la fonction caractéristique est lebesgue intégrable–étaient mélangeants. Withers [Withers, 1981] a établi certaines conditions qui rendent certains processus linéaires mélangeants. La définition 13 énoncée ci-dessus concerne un certain type de mélange qui est le α-mélange. Il existe d'autres types de mélange tel que le β-mélange.

Définition 14. *[Volonskii and Rozanov, 1959] Un processus* $(X_n)_{n \geq 1}$ *est dit* β*-mélangeant si*

$$\beta(n) = \sup_{l \geq 1} \mathbb{E}\left[\sup_{B \in \sigma_{l+n}^{\infty}} \left\{|\mathbb{P}(B|\sigma_1^l) - \mathbb{P}(B)|\right\}\right] \to 0, n \to \infty$$

où, $\sigma(X_t, ..., X_u) = \sigma(X_t^u)$ *pour* $t \leq u$.

Il existe une autre définition du β*-mélange [Bradley, 2005] qui peut être plus utile lorsqu'on a à traiter les prolèmes de type mesurabilité d'un processus :*

$$\beta(n) = \sup_{j \in \mathbb{Z}} \beta(\sigma_1^j, \sigma_{j+n}^{\infty}) \to 0, \ n \to \infty$$

$$où, \ \beta(\sigma_1^j, \sigma_{j+n}^{\infty}) = \frac{1}{2}\sup \sum_{l=1}^{J}\sum_{i=1}^{I} |\mathbb{P}(A_i \cap B_l) - \mathbb{P}(A_i)\mathbb{P}(B_l)|$$

sachant que $(A_i)_{1 \leq i \leq I}$ *est une partition de* σ_1^j *et* $(B_l)_{1 \leq l \leq J}$ *est une partition de* σ_{j+n}^{∞}. *En particulier on dit qu'un processus est exponentiellement* β*-mélangeant si le coefficient de mélange satisfait*

$$\beta(n) \leq \overline{\beta}\exp(-bn^{\kappa}), \ \textit{pour tout } n \geq 1 \tag{5.1}$$

tels que $b > 0$, $\overline{\beta} > 0$ *et* $\kappa > 0$.

La propriété (5.1) est satisfaite par une large classe de processus, par exemple les processus linéaires (voir définition 26 dans l'annexe B), dont les processus ARMA (voir définition 29) qui satisfont l'hypothèse avec $\kappa = 1$. Les chaînes de Markov Harris récurrentes (voir définitions 22 et 23 et dans l'annexe) et apériodiques (voir définition 24 dans l'annexe B) satisfont également cette propriété. Les variables iid satisfont cette propriété pour $\kappa = \infty$. La notion de β-mélange est plus forte que celle du α-mélange, dans le sens où la première notion implique la deuxième. Il existe une notion encore plus forte que celle du β-mélange qui est le ϕ-mélange. Un processus est dit ϕ-mélangeant si

Définition 15.

$$\phi(n) = \sup\left\{|\mathbb{P}(B|A) - \mathbb{P}(B)| : A \in \sigma_l, B \in \sigma_{l+n}^{\infty}\right\}$$

Les chaînes de Markov uniformément (géométriquement) ergodiques (voir définition 28 dans l'annexe B) sont ϕ-mélangeantes. Les chaînes de Markov stationnaires et β-exponentiellement mélangeantes sont ϕ-mélangeantes [Bradley, 2005]. Tous ces coefficients de mélange estiment implicitement le degré de dépendance entre les variables X_k et X_{n+k} pour un entier k fixé. Le processus est mélangeant si ces variables sont indépendantes lorsque n est très grand. Il existe d'autres types de mélange, tels que le ρ-mélange ou encore le ψ-mélange on peut se référer à [Bradley, 2005] pour plus de détails. On s'intéressera ceci dit plus particulièrement au β-mélange qui est également l'hypothèse faite par [Lazaric et al., 2012] [7]. On va établir dans ce qui suit l'inégalité de concentration équivalente au théorème 4 dans le cas des variables exponentiellement β-mélangeantes. Cette inégalité nous servira par la suite au calcul de la vitesse de convergence de l'algorithme LSTD(λ).

5.3 Vitesse de convergence et calcul de borne d'erreurs de l'algorithme LSTD(λ)

Dans cette section, nous allons étudier la vitesse de convergence de l'algorithme LSTD(λ) en fonction du nombre d'échantillons n générés et du paramètre λ. Dans le cas $\lambda = 0$ et pour un nombre d'échantillons fini n, [Lazaric et al., 2012] ont obtenu une borne d'erreur de l'ordre $\tilde{O}(\frac{1}{\sqrt{n}})$ qui est valide avec forte probabilité [8]. Une analyse similaire dans le cas $\lambda > 0$ n'a, à notre connaissance, pas encore été proposée dans la littérature. Étudier le cas $\lambda \neq 0$ peut être intéressant car ce paramètre intervient dans la qualité de la borne d'erreur. C'est ce que nous allons plus détailler dans ce qui suit. On supposera pour cela $\phi \in \mathcal{B}(\mathcal{X}, \mathcal{L})$ où l'ensemble $\mathcal{B}(\mathcal{X}, \mathcal{L})$ désigne l'ensemble des fonctions définies sur l'espace [9] $\mathcal{X}$ et majorées par L.

5.4 Résultat principal

Cette section contient notre résultat principal. Ce résultat repose sur l'hypothèse qui suit.

Hypothèse 3. *On supposera la chaîne de Markov $\mathcal{M}$ exponentiellement β-mélangeante.*

On va maintenant énoncer le théorème principal qui fournit la vitesse de convergence de l'algorithme LSTD(λ).

Théorème 5. *On fait les hypothèses 1 et 3 et on suppose que $X_1 \sim \mu$. Pour tout $n \geq 1$ et $\delta \in (0, 1)$, on définit les fonctions :*

$$I(n, \delta) = 32\Lambda(n, \delta) \max\left\{\frac{\Lambda(n, \delta)}{b}, 1\right\}^{\frac{1}{\kappa}}$$

$$où \quad \Lambda(n, \delta) = 2\left(\log\left(\frac{8n^2}{\delta}\right) + \log(\max\{4e^2, n\overline{\beta}\})\right).$$

7. On n'aurait pas pu travailler sous l'hypothèse du α-mélange car la technique de blocs de Yu qu'on va introduire par la suite nécessite la convergence uniforme. Cette dernière hypothèse ne remplit pas cette condition. Notons toutefois que dans le cas où l'espace S est au plus dénombrable, pour une chaîne de Markov ergodique et stationnaire, le α-mélange est équivalent au β-mélange [Bradley, 2005].

8. La notation $f(n) = \tilde{O}(g(n))$ signifie $f(n) = O(g(n)\log^k g(n))$ pour $k \geq 0$.

9. $\mathcal{X}$ est supposé au plus dénombrable.

Soit $n_0(\delta)$ le plus petit entier tel que pour tout $n \geq n_0(\delta)$,

$$\frac{2dL^2}{(1-\gamma)\nu}\left[\frac{2}{\sqrt{n-1}}\sqrt{\left(\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil+1\right)I(n-1,\delta)+\frac{1}{(n-1)(1-\lambda\gamma)}}+\frac{2}{n-1}\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil\right]<1 \tag{5.2}$$

où ν est la plus petite valeur propre de la matrice de Gram $\Phi^t D_\mu \Phi$. Alors, pour tout δ, avec une probabilité supérieure ou égale à $1-\delta$, on a pour tout $n \geq n_0(\delta)$, $\hat{A}$ inversible et :

$$\|v_{LSTD(\lambda)} - \hat{v}_{LSTD(\lambda)}\|_\mu \leq \frac{4V_{max}dL^2}{\sqrt{n-1}(1-\gamma)\nu}\sqrt{\left(1+\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil\right)I(n-1,\delta)+h(n,\delta)}$$

avec $h(n,\delta) = \tilde{O}(\frac{1}{n})$.

La constante ν est strictement positive sous l'hypothèse 1. Pour tout δ, il est clair que l'entier $n_0(\delta)$ existe et est fini puisque le terme à gauche tend vers 0 quand n tend vers l'infini. Comme la fonction $\left(1+\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil\right)I(n-1,\frac{\delta}{n^2})$ vérifie $\left(1+\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil\right)I(n-1,\frac{\delta}{n^2}) = \tilde{O}(1)$, on peut déduire que LSTD(λ) estime $v_{LSTD(\lambda)}$ avec une vitesse de l'ordre $\tilde{O}\left(\frac{1}{\sqrt{n}}\right)$. Comme la fonction $\lambda \mapsto \frac{1}{\log\left(\frac{1}{\lambda\gamma}\right)}$ est croissante, notre borne sur la vitesse de convergence a tendance à se détériorer lorsque λ augmente. D'autre part, la garantie de qualité de $v_{LSTD(\lambda)}$ s'améliore lorsqu'on augmente le paramètre λ, comme le montre le résultat suivant de la littérature.

Théorème 6 ([Tsitsiklis and Roy, 1997]). *L'erreur d'approximation vérifie*[10] *:*

$$\|v - v_{LSTD(\lambda)}\|_\mu \leq \frac{1-\lambda\gamma}{1-\gamma}\|v - \Pi v\|_\mu.$$

Lorsque $\lambda = 1$, on a–comme mentionné dans la section 4.2.2, chapitre 4– LSTD(1) qui renvoie la projection orthogonale Πv de v. En utilisant l'inégalité triangulaire, on déduit des théorèmes 5 et 6 une borne sur l'erreur globale.

Corollaire 7. *Sous les mêmes hypothèses et les mêmes les notations du théorème 5, pour tout δ, avec une probabilité $1-\delta$, pour tout $n \geq n_0(\delta)$, l'erreur globale de LSTD(λ) vérifie :*

$$\begin{aligned}\|v - \hat{v}_{LSTD(\lambda)}\|_\mu \leq & \frac{1-\lambda\gamma}{1-\gamma}\|v - \Pi v\|_\mu + \frac{4V_{max}dL^2}{\nu\sqrt{n-1}(1-\gamma)}\left(\left(1+\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil\right)I(n-1,\delta)\right)^{\frac{1}{2}} \\ & + h(n,\delta).\end{aligned}$$

10. Cette borne peut être améliorée, comme l'a suggéré V. Papavassilou [Tsitsiklis and Roy, 1997] ; en utilisant le théorème de Pythagore, on obtient

$$\|v - v_{LSTD(\lambda)}\|_\mu \leq \frac{1-\lambda\gamma}{\sqrt{(1-\gamma)(1+\gamma-2\lambda\gamma)}}\|v - \Pi v\|_\mu.$$

Par souci de simplicité, nous garderons la forme du théoreme 6.

Remarque 1. *Le résultat énoncé dans le corollaire 7 est légèrement plus fort que celui de [Lazaric et al., 2012] dans le cas $\lambda = 0$: Pour une propriété $P(n)$, notre résultat est de la forme "$\forall \delta$, $\exists n_0(\delta)$ tq. $\forall n \geq n_0(\delta)$, $P(n)$ est vraie avec probabilité $1-\delta$" alors que le leur est de la forme "$\forall n$, $\forall \delta$, $P(n)$ est vraie avec probabilité $1-\delta$". De plus, sous les mêmes hypothèses, l'erreur globale obtenue par [Lazaric et al., 2012], est bornée par*

$$\|\tilde{v}_{LSTD(0)} - v\|_\mu \leq \frac{4\sqrt{2}}{1-\gamma}\|v - \Pi v\|_\mu + \tilde{O}\left(\frac{1}{\sqrt{n}}\right),$$

où $\tilde{v}_{LSTD(0)}$ est la solution tronquée (avec V_{max}) de l'algorithme pathwise LSTD[11]*, alors qu'avec notre analyse nous obtenons*

$$\|\hat{v}_{LSTD(0)} - v\|_\mu \leq \frac{1}{1-\gamma}\|v - \Pi v\|_\mu + \tilde{O}\left(\frac{1}{\sqrt{n}}\right).$$

Le terme correspondant à l'erreur d'approximation est plus fin d'un facteur $4\sqrt{2}$ avec notre analyse. Aussi, contrairement à ce que l'on présente ici, l'analyse proposée par [Lazaric et al., 2012] ne nous renseigne pas sur la vitesse de convergence de LSTD(0) (ils ne proposent pas une borne sur l'erreur $\|v_{LSTD(0)} - \hat{v}_{LSTD(0)}\|_\mu$). Leur analyse est basée sur un modèle de régression Markovien qui consiste à borner directement *l'erreur globale selon la norme μ. Notre argumentation, qui consiste à borner l'erreur d'approximation puis l'erreur d'estimation, permet d'avoir un résultat plus fin.*

Comme il a déjà été mentionné précédemment, la valeur $\lambda = 1$ minimise la borne sur l'erreur d'approximation $\|v - v_{LSTD(\lambda)}\|_\mu$ (le premier terme à droite dans le corollaire 7) alors que la valeur $\lambda = 0$ minimise la borne sur l'erreur d'estimation $\|v_{LSTD(\lambda)} - \hat{v}_{LSTD(\lambda)}\|_\mu$ (le second terme). Il existe pour tout n et pour tout δ une valeur optimale λ^* du paramètre qui minimise l'erreur globale, créant ainsi un compromis entre ces deux erreurs. La figure 5.1 illustre la relation qui existe entre λ et n. Le choix optimal λ^* dépend aussi bien des paramètres du processus β-mélangeant (b, κ et $\overline{\beta}$) que de la qualité de l'espace de projection $\|v - \Pi v\|_\mu$, qui sont généralement des quantités inconnues en pratique. Il est clair cependant que lorsque n tend vers l'infini, λ^* tend vers 1.

La section suivante contient une preuve détaillée du théorème 5.

5.5 Preuve du théorème 5

Dans cette section, on développe les arguments justifiant les résultats de la section précédente. La preuve est organisée en deux parties. Dans la première partie, on prouve une inégalité de concentration pour des processus vectoriels avec traces d'éligibilité infiniment longues. Ensuite dans la seconde partie, on prouve le théorème 5 : on applique l'inégalité à l'erreur induite en estimant A et b, puis on relie ces erreurs à celle sur $v_{LSTD(\lambda)}$.

5.5.1 Inégalité de concentration pour les estimations avec des traces d'éligiblité infiniment longues

L'une des premières difficultés de l'algorithme LSTD(λ) est que les variables $A_i = z_i(\phi(X_i) - \gamma\phi(X_{i+1}))^t$ (respectivement $b_i = z_i r(X_i)$) ne sont pas indépendantes. On ne peut plus dès lors appliquer les résultats standards de concentration afin d'estimer la vitesse de convergence des

11. Il s'agit d'une version similaire de LSTD, voir [Lazaric et al., 2012] pour plus de details.

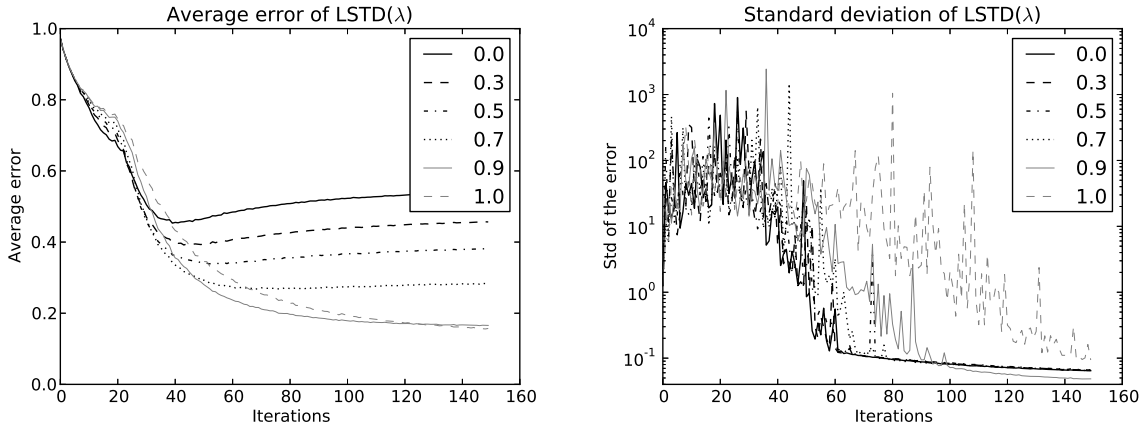

FIGURE 5.1 – **Courbes d'apprentissage pour différentes valeurs de** λ. On génère 1000 MDPs Garnet aléatoires [Archibald et al., 1995] (les Garnet MDP sont des processus décisionnels de Markov (N, m, b, γ) qui se caractérisent par leur nombre d'états N, le nombre d'actions m, le facteur de branchement b (b représente le nombre d'états atteints à partir de chaque couple état-action) et le facteur de discontinuité γ. On a pris $N = 100$, $\gamma = 0.99$. Les récompenses sont uniformes et aléatoires. On a aussi généré 1000 features d'espace aléatoires de dimension 20 (en prenant des matrices aléatoires avec des entrées uniformes et aléatoires). Pour toutes ces valeurs de $\lambda \in \{0.0, 0.3, 0.5, 0.7, 0.9, 1.0\}$, on montre (à gauche) la moyenne de l'erreur *réelle* et (à droite) la déviation standard en fonction du nombre d'échantillons. Empiriquement, la meilleure valeur de λ a l'air d'être une fontion monotone du nombre d'échantillons n, qui tend vers 1 asymptotiquement. Ceci concorde avec le résultat énoncé dans le Corollaire 7.

estimations vers leur limites. Puisque les variables A et b admettent la même structure, on notera G la matrice qui a cette forme générale

$$\hat{G} = \frac{1}{n-1} \sum_{i=1}^{n-1} G_i \tag{5.3}$$

$$\text{avec} \quad G_i = z_i (\tau(X_i, X_{i+1}))^t \tag{5.4}$$

où z_i, défini dans l'équation (4.6), satisfait $z_i = \sum_{l=1}^{i} (\lambda\gamma)^{i-l} \phi(X_l)$, et $\tau : \mathcal{X}^2 \mapsto \mathbb{R}^k$ est tel que, pour tout $1 \leq i \leq k$, τ_i appartient à $\mathcal{B}(\mathcal{X}^2, L')$ pour une constante L' finie [12]. Les variables G_i sont calculées à partir d'une même trajectoire, et sont par conséquent significativement dépendantes. Néanmoins, grâce à l'hypothèse 3, nous allons être en mesure d'utiliser la technique de blocs de [Yu, 1994] qui nous ramène au cas indépendant. Ce passage du cas β-mélangeant au cas indépendant nécessite l'hypothèse de stationnarité (lemme 5). Malheureusement, les variables G_i ne définissent pas un processus stationnaire puisqu'elles sont une fonction $\sigma(\mathcal{X}^{i+1})$-mesurable du vecteur *non-stationnaire* $(X_1, \ldots, X_{i+1})$. Afin de résoudre ce problème, on va approcher G_i par sa version tronquée stationnaire G_i^m. Ceci est possible si on approche la trace z_i par sa version m-tronquée :

$$z_i^m = \sum_{l=\max(i-m+1,1)}^{i} (\lambda\gamma)^{i-l} \phi(X_l).$$

12. On note $\mathcal{X}^i = \underbrace{X \times \mathcal{X} \ldots \times \mathcal{X}}_{\text{i fois}}$ pour $i \geq 1$.

Puisque la fonction ϕ est bornée par une certaine constante L et que l'influence des anciens événements est bornée par une puissance de $\lambda\gamma < 1$, on peut montrer que $\|z_i - z_i^m\|_\infty \leq \frac{L}{1-\lambda\gamma}(\lambda\gamma)^m$. Pour m vérifiant $m > \frac{\log(n-1)}{\log\frac{1}{\lambda\gamma}}$, on obtient $\|z_i - z_i^m\|_2 = O\left(\frac{1}{n}\right)$. Il paraît alors raisonnable d'approcher G par $\hat{G}^m$ satisfaisant

$$\hat{G}^m = \frac{1}{n-1}\sum_{i=1}^{n-1} G_i^m, \tag{5.5}$$

$$\text{avec} \quad G_i^m = z_i^m(\tau(X_i, X_{i+1}))^t. \tag{5.6}$$

Pour tout $i \geq m$, G_m^i est une fonction $\sigma(\mathcal{X}^{m+1})$-mesurable du vecteur stationnaire $Z_i = (X_{i-m+1}, X_{i-m+2}, \ldots, X_{i+1})$. On peut alors appliquer la technique de [Yu, 1994] à G_i^m, mais pour cela il faut d'abord vérifier que les variables G_i^m définissent bien un processus β-mélangeant. On sait d'après [Yu, 1994] que toute fonction d'un processus β-mélangeant est un processus β-mélangeant de coefficient $\beta^f \leq \beta$, donc il nous suffit de prouver que le processus Z_i est β-mélangeant. Pour cela on va essayer de relier les coefficients β_Z du processus Z_i à ceux du processus X_n supposé β-mélangeant d'après l'hypothèse 3. C'est ce que l'on fait dans le lemme suivant.

Lemme 3. *Soit $(X_n)_{n\geq 1}$ un processus β-mélangeant alors $(Z_n)_{n\geq m} = (X_{n-m+1}, X_{n-m+2}, \ldots, X_{n+1})_{n\geq m}$ est un processus β-mélangeant dont le i-ème coefficient β-mélangeant β_i^Z vérifie $\beta_i^Z \leq \beta_{i-m}^X$ pour $i \geq m$.*

Démonstration. Soit $\Gamma = \sigma(Z_m, ..., Z_t)$ pour $t \geq m$, par définition on a

$$\Gamma = \sigma(Z_j^{-1}(B) : j \in \{m, ..., t\}, B \in \sigma(\mathcal{X}^{m+1})).$$

Pour tout $j \in \{m, ..., t\}$ on a

$$Z_j^{-1}(B) = \{\omega \in \Omega, Z_j(\omega) \in B\}.$$

Pour $B = B_0 \times ... \times B_m$, on peut observer que

$$Z_j^{-1}(B) = \{\omega \in \Omega, X_{j-m+1}(\omega) \in B_0, ..., X_{j+1}(\omega) \in B_m\}.$$

D'où, on peut s'apercevoir que

$$\Gamma = \sigma(X_j^{-1}(B) : j \in \{m, ..., t\}, B \in \sigma(\mathcal{X})) = \sigma(X_1, ..., X_{t+1}).$$

De même on peut montrer que $\sigma(Z_{t+i}^\infty) = \sigma(X_{t+i-m+1}^\infty)$. Soit β_i^X le $i^{\text{ème}}$ coefficient β-mélangeant du processus $(X_n)_{n\geq 1}$. On a

$$\beta_i^X = \sup_{t\geq 1} \mathbb{E}\left[\sup_{B\in\sigma(X_{t+i}^\infty)} |P(B|\sigma(X_1, ..., X_t)) - P(B)|\right].$$

D'une façon similaire, on a

$$\beta_i^Z = \sup_{t\geq m} \mathbb{E}\left[\sup_{B\in\sigma(Z_{t+i}^\infty)} |P(B|\sigma(Z_m, ..., Z_t)) - P(B)|\right].$$

En appliquant les identités qu'on a montrées juste avant on peut voir que

$$\beta_i^Z = \sup_{t\geq m} \mathbb{E}\left[\sup_{B\in\sigma(X_{t+i-m+1}^{\infty})} |P(B|\sigma(X_1,...,X_{t+1})) - P(B)|\right].$$

En posant $u = t+1$ on a

$$\beta_i^Z = \sup_{u\geq m+1} \mathbb{E}\left[\sup_{B\in\sigma(X_{u+i-m}^{\infty})} |P(B|\sigma(X_1,...,X_u)) - P(B)|\right].$$

Par conséquent, on a pour $i \geq m$

$$\beta_i^Z \leq \beta_{i-m}^X. \qquad \square$$

Soit $\|.\|_F$ la norme de Frobenius : pour $M \in \mathbb{R}^{d\times k}$, $\|M\|_F^2 = \sum_{l=1}^d \sum_{j=1}^k (M_{l,j})^2$. On peut maintenant énoncer l'inégalité de concentration pour le processus β-mélangeant $\hat{G}$ avec des traces infiniment longues.

Lemme 4. *Soit la matrice G_i de taille $d \times k$ définie par*

$$G_i = \sum_{l=1}^{i} (\lambda\gamma)^{i-l} \phi(X_l)(\tau(X_i, X_{i+1}))^t. \tag{5.7}$$

Rappelons que $\phi = (\phi_1, ..., \phi_d)$ est tel que pour tout j, $\phi_j \in \mathcal{B}(\mathcal{X}, \mathcal{L})$ et que $\tau \in \mathcal{B}(\mathcal{X}^2, L')$ (voir page 68). Sous les hypothèses et notations du théorème 5, on a pour tout $\delta \in (0,1)$, avec probabilité $1-\delta$,

$$\left\| \frac{1}{n-1}\sum_{i=1}^{n-1} G_i - \frac{1}{n-1}\sum_{i=1}^{n-1} \mathbb{E}[G_i] \right\|_2 \leq \frac{B_2}{\sqrt{n-1}} \sqrt{(m+1)J(n-1,\delta)} + \epsilon(n),$$

où

$$J(n,\delta) = 32\Gamma(n,\delta) \max\left\{\frac{\Gamma(n,\delta)}{b}, 1\right\}^{\frac{1}{\kappa}},$$

$$\Gamma(n,\delta) = \log\left(\frac{2}{\delta}\right) + \log(\max\{4e^2, n\overline{\beta}\}),$$

$$\epsilon(n) = 2\left\lceil \frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)} \right\rceil \frac{\sqrt{d\times k} L L'}{(n-1)(1-\lambda\gamma)}.$$

Par rapport aux quantités I et Λ introduites dans l'énoncé du théorème 5, les quantités introduites ici sont telles que $J(n,\delta) = I(n, 4n^2\delta)$ et $\Gamma(n,\delta) = \Lambda(n, 4n^2\delta)$.

Démonstration. La preuve consiste i) à montrer que l'erreur induite en considérant $\hat{G}^m$, la version avec la trace tronquée, au lieu de $\hat{G}$ est bornée par $\epsilon(n)$, et ii) à appliquer la technique de blocs de [Yu, 1994] similaire à ce que [Lazaric et al., 2012] ont employée pour LSTD(0). Soient

$$\epsilon_1 = \frac{1}{n-1}\sum_{i=1}^{m-1} G_i - \mathbb{E}[G_i]$$

$$\text{et} \quad \epsilon_2 = \frac{1}{n-1}\sum_{i=m}^{n-1} (z_i - z_i^m)\tau(X_i, X_{i+1})^t - \mathbb{E}[(z_i - z_i^m)\tau(X_i, X_{i+1})^t].$$

On a

$$\begin{aligned}\frac{1}{n-1}\sum_{i=1}^{n-1} G_i - \mathbb{E}[G_i] &= \frac{1}{n-1}\sum_{i=m}^{n-1} G_i - \mathbb{E}[G_i] + \epsilon_1 \\ &= \frac{1}{n-1}\sum_{i=m}^{n-1} z_i(\tau(X_i, X_{i+1})^t - \mathbb{E}[z_i(\tau(X_i, X_{i+1})^t] + \epsilon_1 \\ &= \frac{1}{n-1}\sum_{i=m}^{n-1} z_i^m\tau(X_i, X_{i+1})^t - \mathbb{E}[z_i^m\tau(X_i, X_{i+1})^t] + \epsilon_1 + \epsilon_2 \\ &= \frac{1}{n-1}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m]) + \epsilon_1 + \epsilon_2. \end{aligned} \tag{5.8}$$

Pour tout i, on a $\|z_i\|_\infty \leq \frac{L}{1-\lambda\gamma}$, $\|G_i\|_\infty \leq \frac{LL'}{1-\lambda\gamma}$, et $\|z_i - z_i^m\|_\infty \leq \frac{(\lambda\gamma)^m L}{1-\lambda\gamma}$. Par conséquent—en utilisant $\|M\|_2 \leq \|M\|_F = \sqrt{d\times k}\|x\|_\infty$ pour $M \in \mathbb{R}^{d\times k}$ avec x le vecteur obtenu en concaténant les colonnes de M—, on obtient

$$\|\epsilon_1 + \epsilon_2\|_2 \leq \frac{2(m-1)\sqrt{d\times k}LL'}{(n-1)(1-\lambda\gamma)} + \frac{2(\lambda\gamma)^m\sqrt{d\times k}LL'}{(1-\lambda\gamma)}. \tag{5.9}$$

En concaténant les colonnes de la matrice G_i^m, la matrice peut être considérée comme un vecteur U_i^m de taille dk. On a alors pour tout $\epsilon > 0$,

$$\begin{aligned}\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_2 \geq \epsilon\right) &\leq \mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_F \geq \epsilon\right) \\ &= \mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(U_i^m - \mathbb{E}[U_i^m])\right\|_2 \geq \epsilon\right). \end{aligned} \tag{5.10}$$

Les variables U_i^m définissant un processus β-mélangeant stationnaire (lemme 3), on utilise la technique de décomposition proposée par [Yu, 1994] qui consiste à regrouper les variables $U_m^m, \ldots, U_{n-1}^m$ en $2\mu_{n-m}$ blocs de taille a_{n-m} (on suppose $n-m = 2a_{n-m}\mu_{n-m}$). Les blocs sont de deux sortes : ceux qui contiennent les indices pairs $E = \cup_{l=1}^{\mu_{n-m}} E_l$ et ceux qui contiennent les indices impairs $H = \cup_{l=1}^{\mu_{n-m}} H_l$. En regroupant les variables dans des blocs on obtient

$$\begin{aligned}\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) \leq &\mathbb{P}\left(\left\|\sum_{i\in H} U_i^m - \mathbb{E}[U_i^m]\right\|_2 + \left\|\sum_{i\in E} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \right. \\ &\left. (n-m)\frac{\epsilon}{2}\right) \qquad (5.11) \\ \leq &\mathbb{P}\left(\left\|\sum_{i\in H} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right) + \\ &\mathbb{P}\left(\left\|\sum_{i\in E} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right) \qquad (5.12) \\ = &2\mathbb{P}\left(\left\|\sum_{i\in H} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right). \qquad (5.13) \end{aligned}$$

L'équation (5.11) découle de l'inégalité triangulaire. L'équation (5.12) découle du fait que $\{X + Y \geq a\}$ implique $\{X \geq \frac{a}{2}\}$ ou $\{Y \geq \frac{a}{2}\}$. La stationnarité du processus implique l'équation (5.13). Puisque $H = \cup_{l=1}^{\mu_{n-m}} H_l$ on a

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) \leq 2\mathbb{P}\left(\left\|\sum_{l=1}^{\mu_{n-m}}\sum_{i\in H_l} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right)$$
$$= 2\mathbb{P}\left(\left\|\sum_{l=1}^{\mu_{n-m}} U(H_l) - \mathbb{E}[U(H_l)]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right) \quad (5.14)$$

où on définit $U(H_l) = \sum_{i\in H_l} U_i^m$. Considérons maintenant la séquence de blocs $(U'(H_l))_{l=1,\ldots,\mu_{n-m}}$ indépendants tel que chaque bloc $U'(H_l)$ admet la même distribution que $U(H_l)$. On va utiliser le lemme suivant.

Lemme 5. *[Yu, 1994] Soit $X_1, \ldots, X_n$ une séquence d'échantillons générés à partir d'un processus β-mélangeant de coefficient $\{\beta_i\}$. Soit $X(H) = (X(H_1), \ldots, X(H_{\mu_{n-m}}))$ tel que pour tout j, $X(H_j) = (X_i)_{i\in H_j}$. On définit $X'(H) = (X'(H_1), \ldots, X'(H_{\mu_{n-m}}))$ tel que les variables $X'(H_j)$ sont indépendantes et tel que pour tout j, $X'(H_j)$ a la même distribution que $X(H_j)$. Soient Q et Q' les distributions de $X(H)$ et $X'(H)$ respectivement. Pour toute fonction mesurable $h : \mathcal{X}^{a_n\mu_n} \to \mathbb{R}$ bornée par B, on a*

$$|\mathbb{E}_Q[h(X(H))] - \mathbb{E}_{Q'}[h(X'(H))]| \leq B\mu_n\beta_{a_n}.$$

En appliquant le Lemme 5, l'équation (5.14) implique que :

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) \leq 2\mathbb{P}\left(\left\|\sum_{l=1}^{\mu_{n-m}} U'(H_l) - \mathbb{E}[U'(H_l)]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right) +$$
$$2\mu_{n-m}\beta_{a_{n-m}}. \quad (5.15)$$

Les variables $U'(H_l)$ sont indépendantes, de plus le processus $\sum_{l=1}^{\mu_{n-m}} U'(H_l) - \mathbb{E}[U'(H_l)]$ est une $\sigma(U'(H_1), \ldots, U'(H_{\mu_{n-m}}))$ martingale :

$$\mathbb{E}\left[\sum_{l=1}^{\mu_{n-m}} U'(H_l) - \mathbb{E}[U'(H_l)] \,\middle|\, U'(H_1), \ldots, U'(H_{\mu_{n-m}-1})\right]$$
$$= \sum_{l=1}^{\mu_{n-m}-1} U'(H_l) - \mathbb{E}[U'(H_l)] + \mathbb{E}[U'_{H_{\mu_{n-m}}} - \mathbb{E}[U'_{H_{\mu_{n-m}}}]]$$
$$= \sum_{l=1}^{\mu_{n-m}-1} U'(H_l) - \mathbb{E}[U'(H_l)].$$

On peut alors appliquer le lemme 2. En effet en prenant $X_{\mu_{n-m}} = \sum_{l=}^{\mu_{n-m}} U'(H_l) - \mathbb{E}[U'(H_l)]$, et en observant que $\|X_i - X_{i-1}\|_2 = \|U'(H_l) - \mathbb{E}[U'(H_l)]\|_2 \leq a_{n-m}C$ avec $C = \frac{2\sqrt{dk}LL'}{1-\lambda\gamma}$, le lemme 2 implique

$$\mathbb{P}\left(\left\|\sum_{l=1}^{\mu_{n-m}} U'(H_l) - \mathbb{E}[U'(H_l)]\right\|_2 \geq \frac{(n-m)\epsilon}{4}\right) \leq 2e^2 e^{-\frac{(n-m)^2\epsilon^2}{32\mu_{n-m}(a_{n-m}C)^2}}$$
$$= 2e^2 e^{-\frac{(n-m)\epsilon^2}{16a_{n-m}C^2}}$$

où la deuxième inégalité est obtenue en utilisant l'égalité $2a_{n-m}\mu_{n-m} = n-m$. En combinant les équations (5.14) et (5.15), on obtient finalement

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) \leq 4e^2 e^{-\frac{(n-m)\epsilon^2}{16a_{n-m}C^2}} + 2(n-m)\beta^U_{a_{n-m}}.$$

Le vecteur U_i^m est une fonction du vecteur $Z_i = (X_{i-m+1}, \ldots, X_{i+1})$. D'après le lemme 3 on sait que pour tout $j > m$,

$$\beta_j^U \leq \beta_j^Z \leq \beta_{j-m}^X \leq \overline{\beta}e^{-b(j-m)^\kappa}.$$

On obtient alors

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) \leq 4e^2 e^{-\frac{(n-m)\epsilon^2}{16a_{n-m}C^2}} + 2(n-m)\overline{\beta}e^{-b(a_{n-m}-m)^\kappa} = \delta'. \quad (5.16)$$

Pour avoir le même exposant dans les deux exponentielles, on va suivre le même raisonnement que [Lazaric et al., 2012] ; En prenant $a_{n-m} - m = \left\lceil \frac{C_2(n-m)\epsilon^2}{b} \right\rceil^{\frac{1}{\kappa+1}}$ avec $C_2 = (16C^2\zeta)^{-1}$, et $\zeta = \frac{a_{n-m}}{a_{n-m}-m}$, on obtient

$$\delta' \leq (4e^2 + (n-m)\overline{\beta}) \exp\left(-\min\left\{\left(\frac{b}{(n-m)\epsilon^2 C_2}\right), 1\right\}^{\frac{1}{k+1}} \frac{1}{2}(n-m)C_2\epsilon^2\right). \quad (5.17)$$

En écrivant

$$\Lambda(n,\delta) = \log\left(\frac{2}{\delta}\right) + \log(\max\{4e^2, n\overline{\beta}\})$$

et

$$\epsilon(\delta) = \sqrt{2\frac{\Lambda(n-m,\delta)}{C_2(n-m)}\max\left\{\frac{\Lambda(n-m,\delta)}{b}, 1\right\}^{\frac{1}{\kappa}}},$$

on peut montrer que

$$\exp\left(-\min\left\{\left(\frac{b}{(n-m)(\epsilon(\delta))^2 C_2}\right), 1\right\}^{\frac{1}{k+1}} \frac{1}{2}(n-m)C_2(\epsilon(\delta))^2\right) \leq \exp\left(-\Lambda(n-m,\delta)\right). \quad (5.18)$$

En effet [13], il y a deux cas :

1. Supposons $\min\left\{\left(\frac{b}{(n-m)(\epsilon(\delta))^2 C_2}\right), 1\right\} = 1$. On a

$$\begin{aligned}
&\exp\left(-\min\left\{\left(\frac{b}{(n-m)(\epsilon(\delta))^2 C_2}\right), 1\right\}^{\frac{1}{k+1}} \frac{1}{2}(n-m)C_2(\epsilon(\delta))^2\right) \\
&= \exp\left(-\Lambda(n-m,\delta)\max\left\{\frac{\Lambda(n-m,\delta)}{b}, 1\right\}^{\frac{1}{k}}\right) \\
&\leq \exp\left(-\Lambda(n-m,\delta)\right).
\end{aligned}$$

13. Cette inégalité existe dans [Lazaric et al., 2012], et est développée ici pour être complet.

2. Supposons maintenant que $\min\left\{\left(\frac{b}{(n-m)(\epsilon(\delta))^2 C_2}\right), 1\right\} = \left(\frac{b}{(n-m)(\epsilon(\delta))^2 C_2}\right)$. On a alors

$$\begin{aligned}
&\exp\left(-\frac{1}{2} b^{\frac{1}{k+1}} ((n-m) C_2 (\epsilon(\delta))^2)^{\frac{k}{k+1}}\right) = \\
&\exp\left(-\frac{1}{2} b^{\frac{1}{k+1}} (\Lambda(n-m,\delta)^{\frac{k}{k+1}} \max\left\{\frac{\Lambda(n-m,\delta)}{b}, 1\right\}^{\frac{1}{k+1}}\right) = \\
&\exp\left(-\frac{1}{2} \Lambda(n-m,\delta)^{\frac{k}{k+1}} \max\left\{\Lambda(n-m,\delta), b\right\}^{\frac{1}{k+1}}\right) \leq \\
&\exp\left(-\Lambda(n-m,\delta)\right).
\end{aligned}$$

En combinant les équations (5.17) et (5.18), on trouve

$$\delta' \leq (4e^2 + (n-m)\overline{\beta}) \exp\left(-\Lambda(n-m,\delta)\right).$$

En remplaçant $\Lambda(n-m,\delta)$ par son expression, on obtient

$$\exp\left(-\Lambda(n-m,\delta)\right) = \frac{\delta}{2} \max\{4e^2, (n-m)\overline{\beta}\}^{-1}.$$

Puisque $4e^2 \max\{4e^2, (n-m)\overline{\beta}\}^{-1} \leq 1$ et $(n-m)\overline{\beta} \max\{4e^2, (n-m)\overline{\beta}\}^{-1} \leq 1$, on a

$$\delta' \leq 2\frac{\delta}{2} \leq \delta.$$

Puisque $a_{n-m} - m \geq 1$, on a

$$\zeta = \frac{a_{n-m}}{a_{n-m} - m} = \frac{a_{n-m} - m + m}{a_{n-m} - m} \leq 1 + m.$$

Soit $J(n,\delta) = 32\Lambda(n,\delta) \max\left\{\frac{\Lambda(n,\delta)}{b}, 1\right\}^{\frac{1}{\kappa}}$. L'équation (5.16) est réduite à

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(U_i^m - \mathbb{E}[U_i^m])\right\|_2 \geq \frac{C}{\sqrt{n-m}}\left((m+1)J(n-m,\delta)\right)^{\frac{1}{2}}\right) \leq \delta. \tag{5.19}$$

$J(n,\delta)$ est une fonction croissante de n, et $\frac{n-1}{\sqrt{n-1}(n-m)} = \frac{1}{\sqrt{n-m}}\sqrt{\frac{n-1}{n-m}} \geq \frac{1}{\sqrt{n-m}}$, on a alors

$$\begin{aligned}
&\mathbb{P}\left(\left\|\frac{1}{n-1}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_2 \geq \frac{C}{\sqrt{n-1}}\left((m+1)J(n-1,\delta)\right)^{\frac{1}{2}}\right) \\
&\leq \mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_2 \geq \frac{C}{\sqrt{n-1}}\frac{n-1}{n-m}\left((m+1)J(n-1,\delta)\right)^{\frac{1}{2}}\right) \\
&\leq \mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_2 \geq \frac{C}{\sqrt{n-m}}\left((m+1)J(n-m,\delta)\right)^{\frac{1}{2}}\right).
\end{aligned}$$

En utilisant les équations (5.10) et (5.19), on peut déduire que

$$\mathbb{P}\left(\left\|\frac{1}{n-1}\sum_{i=m}^{n-1}(G_i^m - \mathbb{E}[G_i^m])\right\|_2 \geq \frac{C}{\sqrt{n-1}}\left((m+1)J(n-1,\delta)\right)^{\frac{1}{2}}\right) \leq \delta. \tag{5.20}$$

En combinant les équations (5.8), (5.9), (5.20), en utilisant $C = \frac{2\sqrt{dk}LL'}{1-\lambda\gamma}$, et en choisissant $m = \left\lceil \frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)} \right\rceil$, on obtient le résultat.

□

En utilisant une preuve fortement similaire, on peut déduire l'inégalité de concentration générale suivante pour les processus β-mélangeants.

Lemme 6. *Soient $Y = (Y_1, \ldots, Y_n)$ des variables aléatoires prenant leurs valeurs dans l'espace $\mathbb{R}^d$, générées par un processus exponentiellement β-mélangeant stationnaire avec paramètres $\overline{\beta}$, b et κ, tel que pour tout i, $\|Y_i - \mathbb{E}[Y_i]\|_2 \leq B_2$ presque sûrement. Alors pour tout $\delta > 0$,*

$$\mathbb{P}\left\{ \left\| \frac{1}{n}\sum_{i=1}^{n} Y_i - \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[Y_i] \right\|_2 \leq \frac{B_2}{\sqrt{n}} \sqrt{J(n,\delta)} \right\} > 1 - \delta.$$

où $J(n,\delta)$ est défini dans le Lemme 4.

Remarque 2. *Si les variables Y_i étaient indépendantes, on aurait $\beta_i = 0$ pour tout i, c'est-à-dire que, en prenant $\overline{\beta} = 0$ et $b = \infty$, $J(n,\delta)$ est dans ce cas égale à $32 \log \frac{8e^2}{\delta} = O(1)$, et on retrouve bien le résultat standard décrit dans le lemme 2. Le prix à payer pour le fait d'avoir des échantillons β-mélangeants au lieu d'échantillons indépendants est le terme $J(n,\delta) = \tilde{O}(1)$; en d'autres termes, ce prix est raisonnable.*

5.5.2 Preuve du Théorème 5

Une fois qu'on a introduit le résultat de concentration, on peut prouver le théorème 5. Une étape importante avant cela consiste à dériver le lemme suivant.

Lemme 7. *Soient $\epsilon_A = \hat{A} - A$ et $\epsilon_b = \hat{b} - b$. Soit ν la plus petite valeur propre de la matrice $\Phi^T D_\mu \Phi$. Pour tout $\lambda \in (0,1)$, l'estimation $\hat{v}_{LSTD(\lambda)}$ vérifie*[14] *:*

$$\|v_{LSTD(\lambda)} - \hat{v}_{LSTD(\lambda)}\|_\mu \leq \frac{1-\lambda\gamma}{(1-\gamma)\sqrt{\nu}} \|(I + \epsilon_A A^{-1})^{-1}\|_2 \|\epsilon_A \theta - \epsilon_b\|_2.$$

De plus, si les constantes ϵ et C sont telles que $\|\epsilon_A\|_2 \leq \epsilon < C \leq \frac{1}{\|A^{-1}\|}$, alors $\hat{A}$ est inversible et

$$\|(I + \epsilon_A A^{-1})^{-1}\|_2 \leq \frac{1}{1 - \frac{\epsilon}{C}}.$$

Démonstration. A partir des définitions de $v_{LSTD(\lambda)}$ et $\hat{v}_{LSTD(\lambda)}$, on a

$$\begin{aligned} \hat{v}_{LSTD(\lambda)} - v_{LSTD(\lambda)} &= \Phi\hat{\theta} - \Phi\theta \\ &= \Phi A^{-1}(A\hat{\theta} - b). \end{aligned} \tag{5.21}$$

D'une part, on peut observer en partant de l'expression de A dans l'équation (4.3) et en écrivant $M = (1-\lambda)\gamma P(I - \lambda\gamma P)^{-1}$ et $M_\mu = \Phi^t D_\mu \Phi$ que

$$\begin{aligned} \Phi A^{-1} &= \Phi \left[\Phi^t D_\mu (I - \gamma P)(I - \lambda\gamma P)^{-1}\Phi\right]^{-1} \\ &= \Phi \left[\Phi^t D_\mu (I - \lambda\gamma P - (1-\lambda)\gamma P)(I - \lambda\gamma P)^{-1}\Phi\right]^{-1} \\ &= \Phi(M_\mu - \Phi^t D_\mu M \Phi)^{-1}. \end{aligned}$$

14. Quand $\hat{A}$ n'est pas inversible, on prend $\hat{v}_{LSTD(\lambda)} = \infty$ et l'inégalité est toujours vraie puisque, comme on va le voir, l'inversibilité de $\hat{A}$ est équivalente à celle de $(I + \epsilon_A A^{-1})$.

Les matrices A et M_μ sont inversibles donc la matrice $I - M_\mu^{-1}\Phi^t D_\mu M\Phi$ l'est également et on a

$$\Phi A^{-1} = \Phi(I - M_\mu^{-1}\Phi^t D_\mu M\Phi)^{-1}M_\mu^{-1}.$$

On sait d'après [Tsitsiklis and Roy, 1997] que $\|\Pi\|_\mu = 1$—on rappelle que Π est la projection définie à l'équation (4.2)—et $\|P\|_\mu = 1$. Il s'en suit que $\|\Pi M\|_\mu = \frac{(1-\lambda)\gamma}{1-\lambda\gamma} < 1$ et que la matrice $(I - \Pi M)$ est inversible. En utilisant l'identité $X(I - YX)^{-1} = (I - XY)^{-1}X$ avec $X = \Phi$ et $Y = M_\mu^{-1}\Phi^t D_\mu M$, on obtient

$$\Phi A^{-1} = (I - \Pi M)^{-1}\Phi M_\mu^{-1}. \tag{5.22}$$

D'autre part, en utilisant les identités $A\theta = b$ et $\hat{A}\hat{\theta} = \hat{b}$ on peut voir que pour $\epsilon_A = \hat{A} - A$ on a :

$$\begin{aligned}
A\hat{\theta} - b &= A\hat{\theta} - b - (\hat{A}\hat{\theta} - \hat{b}) \\
&= \hat{b} - b - \epsilon_A\theta + \epsilon_A\theta - \epsilon_A\hat{\theta} \\
&= \hat{b} - b - (\hat{A} - A)\theta + \epsilon_A(\theta - \hat{\theta}) \\
&= \hat{b} - \hat{A}\theta - (b - A\theta) + \epsilon_A A^{-1}(A\theta - A\hat{\theta}) \\
&= \hat{b} - \hat{A}\theta + \epsilon_A A^{-1}(b - A\hat{\theta}).
\end{aligned}$$

D'où

$$A\hat{\theta} - b = \hat{b} - \hat{A}\theta - \epsilon_A A^{-1}(A\hat{\theta} - b).$$

Ceci implique que

$$\begin{aligned}
A\hat{\theta} - b &= (I + \epsilon_A A^{-1})^{-1}(\hat{b} - \hat{A}\theta) \\
&= (I + \epsilon_A A^{-1})^{-1}(\epsilon_b - \epsilon_A\theta).
\end{aligned} \tag{5.23}$$

où la deuxième égalité vient de l'identité $A\theta = b$. En utilisant les équations (5.22) et (5.23), l'équation (5.21) peut être réécrite de la manière suivante :

$$\hat{v}_{LSTD(\lambda)} - v_{LSTD(\lambda)} = (I - \Pi M)^{-1}\Phi M_\mu^{-1}(I + \epsilon_A A^{-1})^{-1}(\epsilon_b - \epsilon_A\theta). \tag{5.24}$$

On va maintenant essayer de borner le terme $\|\Phi M_\mu^{-1}(I + \epsilon_A A^{-1})^{-1}(\epsilon_b - \epsilon_A\theta)\|_\mu$. Pour tout x, on a

$$\|\Phi M_\mu^{-1}x\|_\mu = \sqrt{x^t M_\mu^{-1}\Phi^t D_\mu \Phi M_\mu^{-1}x} = \sqrt{x^t M_\mu^{-1}x} \leq \frac{1}{\sqrt{\nu}}\|x\|_2, \tag{5.25}$$

où ν est plus petite valeur propre réelle de la matrice de Gram M_μ. En prenant la norme $\|.\|_\mu$ dans l'équation (5.24) et en utilisant l'inégalité précédente, on obtient :

$$\begin{aligned}
\|\hat{v}_{LSTD(\lambda)} - v_{LSTD(\lambda)}\|_\mu &\leq \|(I - \Pi M)^{-1}\|_\mu \|\Phi M_\mu^{-1}(I + \epsilon_A A^{-1})^{-1}(\epsilon_b - \epsilon_A\theta)\|_\mu \\
&\leq \|(I - \Pi M)^{-1}\|_\mu \frac{1}{\sqrt{\nu}}\|(I + \epsilon_A A^{-1})^{-1}(\epsilon_b - \epsilon_A\theta)\|_2 \\
&\leq \|(I - \Pi M)^{-1}\|_\mu \frac{1}{\sqrt{\nu}}\|(I + \epsilon_A A^{-1})^{-1}\|_2\|(\epsilon_b - \epsilon_A\theta)\|_2.
\end{aligned}$$

La première partie du lemme est obtenue en utilisant à nouveau l'égalité $\|\Pi M\|_\mu = \frac{(1-\lambda)\gamma}{1-\lambda\gamma} < 1$, qui implique que

$$\|(I - \Pi M)^{-1}\|_\mu = \left\| \sum_{i=0}^{\infty} (\Pi M)^i \right\|_\mu \leq \sum_{i=0}^{\infty} \|\Pi M\|_\mu^i \leq \frac{1}{1 - \frac{(1-\lambda)\gamma}{1-\lambda\gamma}} = \frac{1-\lambda\gamma}{1-\gamma}. \tag{5.26}$$

On va maintenant prouver la seconde partie du lemme. La matrice $\hat{A}$ est inversible si et seulement si la matrice $\hat{A}A^{-1} = (A + \epsilon_A)A^{-1} = I + \epsilon_A A^{-1}$ l'est également. On note $\rho(\epsilon_A A^{-1})$ le rayon spectral (voir définition 32 dans l'annexe B) de la matrice $\epsilon_A A^{-1}$. Une condition suffisante pour l'inversibilité de $\hat{A}A^{-1}$ est d'imposer que $\rho(\epsilon_A A^{-1}) < 1$. On a pour toute matrice M réelle carrée $\rho(M) \leq \|M\|_2$. Ainsi, pour ϵ et C tels que $\|\epsilon_A\|_2 \leq \epsilon < C < \frac{1}{\|A^{-1}\|_2}$, on déduit

$$\rho(\epsilon_A A^{-1}) \leq \|\epsilon_A A^{-1}\|_2 \leq \|\epsilon_A\|_2 \|A^{-1}\|_2 \leq \frac{\epsilon}{C} < 1.$$

La matrice $\hat{A}$ est alors inversible et on a :

$$\|(I + \epsilon_A A^{-1})^{-1}\|_2 = \left\| \sum_{i=0}^{\infty} (\epsilon_A A^{-1})^i \right\|_2 \leq \sum_{i=0}^{\infty} \left(\frac{\epsilon}{C}\right)^i = \frac{1}{1 - \frac{\epsilon}{C}}.$$

Ceci conclut la preuve du lemme 7. □

Il suffit d'après le lemme 7, pour conclure la preuve du théorème 5, de contrôler les termes $\|\epsilon_A\|_2$ et $\|\epsilon_A\theta - \epsilon_b\|_2$. C'est ce que l'on fait maintenant.

Contrôle du terme $\|\epsilon_A\|_2$. En utilisant l'inégalité triangulaire, on peut voir que

$$\|\epsilon_A\|_2 \leq \|\mathbb{E}[\epsilon_A]\|_2 + \|\epsilon_A - \mathbb{E}[\epsilon_A]\|_2. \tag{5.27}$$

Soit $\hat{A}_{n,k} = \phi(X_k)(\phi(X_n) - \gamma\phi(X_{n+1}))^t$. On sait d'après [Tsitsiklis and Roy, 1997] que

$$A = \mathbb{E}_\mu \left[\frac{1}{n-1} \sum_{i=1}^{n-1} \sum_{k=-\infty}^{i} (\lambda\gamma)^{i-k} \hat{A}_{i,k} \right].$$

Pour tout n et k, on a $\|\hat{A}_{n,k}\|_2 \leq 2dL^2$. On peut borner le premier terme à droite de l'inégalité dans l'équation (5.27) comme suit :

$$\begin{aligned}
\|\mathbb{E}[\epsilon_A]\|_2 &= \left\| A - \mathbb{E}\left[\frac{1}{n-1} \sum_{i=1}^{n-1} \sum_{k=1}^{i} (\lambda\gamma)^{i-k} \hat{A}_{i,k} \right] \right\|_2 \\
&= \left\| \mathbb{E}\left[\frac{1}{n-1} \sum_{i=1}^{n-1} \left(\sum_{k=-\infty}^{i} (\lambda\gamma)^{i-k} \hat{A}_{i,k} - \sum_{k=1}^{i} (\lambda\gamma)^{i-k} \hat{A}_{i,k} \right) \right] \right\|_2 \\
&= \left\| \mathbb{E}\left[\frac{1}{n-1} \sum_{i=1}^{n-1} (\lambda\gamma)^i \sum_{k=-\infty}^{0} (\lambda\gamma)^{-k} \hat{A}_{i,k} \right] \right\|_2 \\
&\leq \frac{1}{n-1} \sum_{i=1}^{n-1} (\lambda\gamma)^i \frac{2dL^2}{1-\lambda\gamma} \\
&\leq \frac{1}{n-1} \frac{2dL^2}{(1-\lambda\gamma)^2} = \epsilon_0(n).
\end{aligned}$$

Soit δ_n un paramètre dans $(0,1)$, qui dépend de n et qu'on fixera par la suite. A partir de l'équation (5.27) et de la borne déduite, on obtient :

$$\begin{aligned}\mathbb{P}(\|\epsilon_A\|_2 \geq \epsilon_1(n,\delta_n)) &\leq \mathbb{P}(\|\epsilon_A - \mathbb{E}[\epsilon_A]\|_2 \geq \epsilon_1(n,\delta_n) - \epsilon_0(n)) \\ &\leq \delta_n\end{aligned}$$

pour $\epsilon_1(n,\delta_n)$ vérifiant—cf. lemme 4—$\epsilon_1(n,\delta_n) - \epsilon_0(n) = \frac{4dL^2}{(1-\lambda\gamma)\sqrt{n-1}}\sqrt{(m+1)J(n-1,\delta_n)} + \epsilon(n)$ avec $\epsilon(n) = \frac{4mdL^2}{(n-1)(1-\lambda\gamma)}$, c'est-à-dire

$$\epsilon_1(n,\delta_n) = \frac{4dL^2}{(1-\lambda\gamma)\sqrt{n-1}}\sqrt{(m+1)J(n-1,\delta_n)} + \epsilon(n) + \epsilon_0(n). \tag{5.28}$$

Contrôle du terme $\|\epsilon_A\theta - \epsilon_b\|_2$. En utilisant le fait que $A\theta = b$, les définitions de $\hat{A}$ et $\hat{b}$, et le fait que $\phi(x)^t\theta = [\phi\theta](x)$, on a

$$\begin{aligned}\epsilon_A\theta - \epsilon_b &= \hat{A}\theta - \hat{b} \\ &= \frac{1}{n-1}\sum_{i=1}^{n-1} z_i(\phi(X_i) - \gamma\phi(X_{i+1}))^t\theta - \frac{1}{n-1}\sum_{i=1}^{n-1} z_i r(X_i) \\ &= \frac{1}{n-1}\sum_{i=1}^{n-1} z_i([\phi\theta](X_i)^t - \gamma[\phi\theta](X_{i+1})^t - r(X_i)) \\ &= \frac{1}{n-1}\sum_{i=1}^{n-1} z_i\Delta_i\end{aligned}$$

où, comme $v_{LSTD(\lambda)} = \Phi\theta$, Δ_i est un nombre égal à

$$\Delta_i = v_{LSTD(\lambda)}(X_i) - \gamma v_{LSTD(\lambda)}(X_{i+1}) - r(X_i).$$

On peut ainsi contrôler $\|\epsilon_A\theta - \epsilon_b\|_2$ en suivant les mêmes étapes de preuve que précédemment. En effet, on a

$$\begin{aligned}&\|\epsilon_A\theta - \epsilon_b\|_2 \leq \|\epsilon_A\theta - \epsilon_b - \mathbb{E}[\epsilon_A\theta - \epsilon_b]\|_2 + \|\mathbb{E}[\epsilon_A\theta - \epsilon_b]\|_2 \qquad (5.29)\\ \text{et}\quad &\|\mathbb{E}[\epsilon_A\theta - \epsilon_b]\|_2 \leq \|\mathbb{E}[\epsilon_A]\|_2\|\theta\|_2 + \|\mathbb{E}[\epsilon_b]\|_2.\end{aligned}$$

On a $\|\mathbb{E}[\epsilon_A]\|_2 \leq \epsilon_0(n) = \frac{1}{n-1}\frac{2dL^2}{(1-\lambda\gamma)^2}$. On peut montrer similairement que $\|\mathbb{E}[\epsilon_b]\|_2 \leq \frac{1}{n-1}\frac{\sqrt{d}LR_{\max}}{(1-\lambda\gamma)^2}$. On peut donc conclure que

$$\|\mathbb{E}[\epsilon_A\theta - \epsilon_b]\|_2 \leq \frac{1}{n-1}\frac{2dL^2}{(1-\lambda\gamma)^2}\|\theta\|_2 + \frac{1}{n-1}\frac{\sqrt{d}LR_{\max}}{(1-\lambda\gamma)^2} = \epsilon'_0(n).$$

A partir de l'équation (5.29) et de la borne déduite on obtient

$$\mathbb{P}(\|\epsilon_A\theta - \epsilon_b\|_2 \geq \epsilon_2(\delta_n)) \leq \mathbb{P}(\|\epsilon_A\theta - \epsilon_b - \mathbb{E}[\epsilon_A\theta - \epsilon_b]\|_2 \geq \epsilon_2(\delta_n) - \epsilon'_0(n)) \leq \delta_n$$

pour $\epsilon_2(\delta_n)$ vérifiant—cf. lemme 4—

$$\begin{aligned}\epsilon_2(\delta_n) =& \frac{2\sqrt{d}L\|\Delta_i\|_\infty}{(1-\lambda\gamma)\sqrt{n-1}}\sqrt{\left(\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil + 1\right)J(n-1,\delta_n)} + \frac{2\sqrt{d}L\|\Delta_i\|_\infty}{(n-1)(1-\lambda\gamma)}\left\lceil\frac{\log(n-1)}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil + \\ &\epsilon'_0(n). \qquad (5.30)\end{aligned}$$

Il nous reste à borner le terme $\|\Delta_i\|_\infty$. Pour cela, il suffit de borner $v_{LSTD(\lambda)}$. Pour tout $x \in \mathcal{X}$, on a

$$|v_{LSTD(\lambda)}(x)| = |\phi^t(x)\theta| \leq \|\phi^t(x)\|_2 \|\theta\|_2 \leq \sqrt{d}L\|\theta\|_2,$$

où on obtient la première inégalité par l'inégalité de Cauchy-Schwarz. On peut borner $\|\theta\|_2$. En effet, on observe d'une part que

$$\|v_{LSTD(\lambda)}\|_\mu = \|\Phi\theta\|_\mu \geq \sqrt{\theta^t M_\mu \theta} \geq \sqrt{\nu}\|\theta\|_2.$$

D'autre part, on a

$$\|v_{LSTD(\lambda)}\|_\mu = \|(I - \Pi M)^{-1}\Pi(I - \lambda\gamma P)^{-1}r\|_\mu \leq \frac{R_{\max}}{1-\gamma} = V_{\max}.$$

Par conséquent

$$\|\theta\|_2 \leq \frac{V_{\max}}{\sqrt{\nu}}.$$

Il s'en suit que

$$\forall x \in \mathcal{X},\ |v_{LSTD(\lambda)}(x)| \leq \frac{\sqrt{d}LV_{\max}}{\sqrt{\nu}}.$$

Ainsi, pour tout i on a

$$\begin{aligned}|\Delta_i| &= |v_{LSTD(\lambda)}(X_i) - \gamma v_{LSTD(\lambda)}(X_{i+1}) - r(X_i)| \\ &\leq \frac{\sqrt{d}LV_{\max}}{\sqrt{\nu}} + \gamma\frac{\sqrt{d}LV_{\max}}{\sqrt{\nu}} + (1-\gamma)V_{\max}.\end{aligned}$$

Puisque $\Phi^t D_\mu \Phi$ est une matrice symétrique, on a $\nu \leq \|\Phi^t D_\mu \Phi\|_2$. D'un autre côté on peut voir que

$$\|\Phi^T D_\mu \Phi\|_2 \leq d \max_{j,k} |\phi_k^t D_\mu \phi_j| = d \max_{j,k} |\phi_k^t D_\mu^{\frac{1}{2}} D_\mu^{\frac{1}{2}} \phi_j| \leq d \max_{j,k} \|\phi_k^t\|_\mu \|\phi_j\|_\mu \leq dL^2,$$

de sorte que $\nu \leq dL^2$. Il s'en suit, que pour tout i,

$$|\Delta_i| \leq \frac{\sqrt{d}LV_{\max}}{\sqrt{\nu}} + \gamma\frac{\sqrt{d}LV_{\max}}{\sqrt{\nu}} + \frac{\sqrt{d}L}{\sqrt{\nu}}(1-\gamma)V_{\max} = 2\frac{\sqrt{d}L}{\sqrt{\nu}}V_{\max}.$$

Conclusion de la preuve. Nous allons maintenant conclure la preuve du théorème 5. Une fois qu'on a contrôlé les termes $\|\epsilon_A\|_2$ et $\|\epsilon_A\theta - \epsilon_b\|_2$, on peut déduire que

$$\begin{aligned}&\mathbb{P}\left\{\exists n \geq 1,\ \{\|\epsilon_A\|_2 \geq \epsilon_1(n,\delta_n)\} \cup \{\|\epsilon_A\theta - \epsilon_b)\|_2 \geq \epsilon_2(n,\delta_n)\}\right\} \\ \leq &\sum_{n=1}^{\infty} \mathbb{P}\left\{\|\epsilon_A\|_2 \geq \epsilon_1(n,\delta_n)\right\} + P\left\{\|\epsilon_A\theta - \epsilon_b)\|_2 \geq \epsilon_2(n,\delta_n)\right\} \\ \leq &2\sum_{n=1}^{\infty}\delta_n = \frac{1}{2}\frac{\pi^2}{6}\delta < \delta\end{aligned}$$

si on choisit $\delta_n = \frac{1}{4n^2}\delta$. D'après la seconde partie du lemme 7, pour tout δ, avec probabilité $1-\delta$, pour tout n tel que $\epsilon_1(n,\delta_n) < C$, on a $\hat{A}$ inversible et

$$\begin{aligned}\|v_{LSTD(\lambda)} - \hat{v}_{LSTD(\lambda)}\|_\mu &\leq \frac{1-\lambda\gamma}{(1-\gamma)\sqrt{\nu}} \frac{\epsilon_2(n,\delta_n)}{1-\frac{\epsilon_1(n,\delta_n)}{C}} \\ &= \frac{1-\lambda\gamma}{(1-\gamma)\sqrt{\nu}} \left[\epsilon_2(n,\delta_n) + \frac{\epsilon_1(n,\delta_n)\,\epsilon_2(n,\delta_n)}{C-\epsilon_1(n,\delta_n)}\right].\end{aligned}$$

On obtiendra la borne dans le théorème 5 en remplaçant $\epsilon_1(n,\delta_n)$ et $\epsilon_2(n,\delta_n)$ par leurs définitions respectives dans les équations (5.28) et (5.30).

Pour compléter la preuve du théorème 5, il reste à montrer comment choisir C, ce qui nous permettra de montrer que la condition $\epsilon_1(n,\delta_n) < C < \frac{1}{\|A^{-1}\|_2}$ est équivalente à celle qui caractérise l'entier $n_0(\delta)$ définie dans le théorème 5. On a

$$\forall v \in \mathbb{R}^d, \|\Phi A^{-1} v\|_\mu = \sqrt{(A^{-1}v)^t M_\mu A^{-1} v} \geq \sqrt{\nu}\|A^{-1}v\|_2.$$

On peut observer que

$$\|\Phi A^{-1} v\|_\mu = \|(I - \Pi M)\Phi M_\mu^{-1} v\|_\mu \leq \frac{1-\lambda\gamma}{1-\gamma}\|\Phi M_\mu^{-1} v\|_\mu \leq \frac{1-\lambda\gamma}{(1-\gamma)\sqrt{\nu}}\|v\|_2$$

où la dernière inégalité est obtenue à partir de l'équation (5.25). Par conséquent on a

$$\|A^{-1}\|_2 \leq \frac{1-\lambda\gamma}{(1-\gamma)\nu},$$

on peut prendre alors $C = \frac{(1-\gamma)\nu}{1-\lambda\gamma}$. Ceci conclut la preuve du théorème 5.

5.6 Quelques Remarques

On aurait pu également concevoir la preuve qu'on a proposée d'une autre manière non moins intéressante. En effet, soient $U_m^m, ..., U_{n-m-1}^m$ des variables aléatoires exponentiellement β-mélangeantes selon la définition donnée plus haut. Supposons de plus qu'il existe un entier l qui vérifie $n - m = l \times m$. On peut regrouper ces variables en blocs de la sorte

$$\begin{aligned}\left\{U_m^m, ..., U_{n-m-1}^m\right\} &= \underbrace{\left\{U_m^m, ..., U_{2m-1}^m\right\}}_{bloc\ 1}\underbrace{\left\{U_{2m}^m, ..., U_{3m-1}^m\right\}}_{bloc\ 2} ... \underbrace{\left\{U_{n-2m}^m, ..., U_{n-m-1}^m\right\}}_{bloc\ l} \\ &= \underbrace{\left\{U_m^m, U_{2m}^m, ..., U_{n-2m}^m\right\}}_{bloc\ 1'}\underbrace{\left\{U_{m+1}^m, U_{2m+1}^m, ..., U_{n-2m+1}^m\right\}}_{bloc\ 2'} ... \\ &\quad \underbrace{\left\{U_{2m-1}^m, U_{3m-1}^m, ..., U_{n-m-1}^m\right\}}_{bloc\ l'}\end{aligned}$$

On peut alors écrire que

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1} U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon\right) = \mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{2m-1}\sum_{j=0}^{l-2} U_{i+mj}^m - \mathbb{E}[U_{i+mj}^m]\right\|_2 \geq \epsilon\right).$$

Les blocs 1′, 2′′,...,l' admettent tous la même loi, celle du bloc $\left\{U_m^m, U_{2m}^m, ..., U_{n-2m}^m\right\}$. Par conséquent

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{2m-1}\sum_{j=0}^{l-2}U_{i+mj}^m - \mathbb{E}[U_{i+mj}^m]\right\|_2 \geq \epsilon\right) = \mathbb{P}\left(\left\|\frac{m}{n-m}\sum_{j=0}^{l-2}U_{m+mj}^m - \mathbb{E}[U_{m+mj}^m]\right\|_2 \geq \epsilon\right)$$

En appliquant le lemme 5 aux variables $(Y_j)_{j\geq 0} = (U_{m+mj}^m)_{j\geq 0}$, on a en posant $l = 2\mu_l a_l$ avec a_l la taille des blocs et μ_l le nombre de blocs considérés

$$\mathbb{P}\left(\left\|\frac{m}{n-m}\sum_{j=0}^{l-1}U_{m+mj}^m - \mathbb{E}[U_{m+mj}^m]\right\|_2 \geq \epsilon\right) \leq 2\mathbb{P}\left(\left\|\sum_{k=1}^{\mu_l}U'(H_k) - \mathbb{E}[U'(H_k)]\right\|_2 \geq \frac{(n-m)\epsilon}{4m}\right) + 2\mu_l\beta_{a_l}^Y.$$

Les variables $U'(H_k)_{1\leq k\leq \mu_l}$ sont des variables iid donc le processus $\sum_{k=1}^{\mu_l}U'(H_k) - \mathbb{E}[U'(H_k)]$ définit une martingale. Par ailleurs les variables $(U_{m+mj}^m)_{j\geq 0}$ sont des variables β-mélangeantes dont le coefficient β_i^Y est égal à celui du processus défini par les variables $(X_i)_{i\geq 0}$. En effet les variables $(U_{m+mj}^m)_{j\geq 0}$ sont construites à partir des variables $(X_n, ..., X_{n+m+1})$, $(X_{n+m+1}, ..., X_{n+2m+2})$, ..., dont le coefficient de mélange est identique à celui du processus $(X_n)_{n\geq 1}$. Il s'en suit en suivant le même raisonnement d'en haut que

$$\begin{aligned}\mathbb{P}\left(\left\|\frac{1}{l}\sum_{j=0}^{l-1}U_{m+mj}^m - \mathbb{E}[U_{m+mj}^m]\right\|_2 \geq \epsilon\right) &\leq 4e^2e^{-\frac{l\epsilon^2}{16a_lC^2}} + 2l\beta_{a_l} \\ &\leq 4e^2e^{-\frac{l\epsilon^2}{16a_lC^2}} + 2le^{-ba_l^\kappa}.\end{aligned}$$

Maintenant en reprenant la même preuve que celle proposée par [Lazaric et al., 2012] avec $a_l = \lceil\frac{Cl\epsilon^2}{b}\rceil^{\frac{1}{\kappa}}$, on obtient pour $\delta \in (0,1)$ avec probabilité $1-\delta$

$$\mathbb{P}\left(\left\|\frac{1}{n-m}\sum_{i=m}^{n-1}U_i^m - \mathbb{E}[U_i^m]\right\|_2 \geq \epsilon(\delta)\right) \leq \delta$$

avec

$$\epsilon(\delta) = 2B_2\sqrt{2\frac{\Lambda(l,\delta)}{l}\max\left\{\frac{\Lambda(l,\delta)}{b}, 1\right\}}.$$

On retombe sur un résultat qui ressemble à celui qui a été proposé précédemment. La fonction $\Lambda(.,\delta)$ est une fonction croissante en l et le facteur m correspond à $m = 1 + \left\lceil\frac{\log n}{\log\left(\frac{1}{\lambda\gamma}\right)}\right\rceil$, puisqu'on a supposé $m \geq 1$. Cette perspective est intéressante car si on supposait les variables $(U_i^m)_{i\geq 1}$ m-dépendantes (voir définition 30 dans l'annexe B), les variables $(U_{m+mj}^m)_{j\geq 0}$ seraient indépendantes et le coefficient β-mélangeant correspondant aurait été égal à $\beta = 0$. Dans ce cas là déterminer la loi de $\sum_{i=m}^{n-1}U_i^m - \mathbb{E}[U_i^m]$ revient à déterminer la loi de la somme de l variables indépendantes. L'hypothèse de β-mélange équivaut à celle de m-dépendance mais pour $m = \infty$, le fait qu'on arrive à mesurer la vitesse avec laquelle les variables deviennent indépendantes–vitesse exponentielle–facilite le calcul et est à l'origine du terme logarithmique qui apparaît en plus.

Ce type de raisonnement a été inspiré de l'article [Modha and Masry, 1996] où des inégalités de concentration propres au cas des variables m-dépendantes et aux variables β-mélangeantes ont été établies. Ces inégalités ont été dérivées de l'inégalité de Bernstein dans le cas de variables définies dans $\mathbb{R}$. Les auteurs utilisent pour cela un lemme proposé par D. Bosq [Bosq, 1975] et non la technique de décomposition en blocs de Yu introduite auparavant. Notons que l'inégalité de concentration obtenue dans [Modha and Masry, 1996] pour des variables aléatoires $(U_i)_{i\geq 1}$ stationnaires exponentiellement β-mélangeantes correspond à

$$\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^{n} U_i - \mathbb{E}[U_i]\right| \geq \epsilon\right) \leq 2(1+4e^{-2}\overline{\beta})\exp\left(-\frac{\epsilon^2 N^\beta}{2(\mathbb{E}[|U_1|^2]+\epsilon\frac{B_2}{3})}\right)$$

avec

$$N^\beta = \left\lfloor n\left\lceil\left\{\frac{8n}{b}\right\}^{\frac{1}{\kappa}}\right\rceil^{-1}\right\rfloor.$$

Cette inégalité fait intervenir les paramètres du processus $\overline{\beta}$, b et κ. En posant $\delta = (1+4e^{-2}\overline{\beta})\exp\left(-\frac{\epsilon^2 N^\beta}{2(\mathbb{E}[|U_1|^2]+\epsilon\frac{B_2}{3})}\right)$, on obtient $\epsilon(\delta) = O\left(\sqrt{\frac{\log N^\beta}{N^\beta}}\right)$ avec $N^\beta \sim n^{\frac{\kappa}{\kappa+1}}$. On remarque que l'inégalité de Bernstein induit une borne sur la vitesse de convergence qui est plus lente que celle qu'on propose dans ce manuscrit. Néanmoins pour $\kappa = \infty$, on retrouve la borne estimée pour le cas iid. Cette différence est dûe au fait qu'on ne part pas de la même inégalité de concentration mais aussi du fait qu'ils n'utilisent pas la technique de Yu qui suppose $n = 2\mu_n a_n$.

5.7 Conclusion et Discussion

Dans ce chapitre on a étudié la vitesse de convergence de l'algorithme LSTD(λ) en termes du nombre d'échantillons n et du paramètre λ. On a montré, sous l'hypothèse β-mélangeante, que la vitesse de convergence était de l'ordre $\tilde{O}(\frac{1}{\sqrt{n}})$. Pour cela, nous avons introduit une inégalité de concentration vectorielle pour les estimations basées sur des traces infiniment longues (lemme 4). Une version plus simple de cette inégalité de concentration, qui concerne de manière générale les processus exponentiellement β-mélangeants stationnaires (énoncée au lemme 2) pourrait être utile dans d'autres contextes où on aurait besoin de relâcher l'hypothèse iid sur les échantillons.

La borne de performance que nous déduisons de notre analyse est plus précise que celle obtenue par [Lazaric et al., 2012] dans le cas $\lambda = 0$. L'analyse qu'ils ont employée utilise un modèle de régression Markovien :

$$Y_t = v(X_t) + \xi_t$$

où v est la fonction de valeur, X_t la chaîne de Markov considérée et ξ_t est le terme d'erreur induit. La variable ξ_t est adaptée à la filtration $(X_1, ..., X_{t+1})$, est bornée $|\xi_t| \leq C$ et vérifie $\mathbb{E}[\xi_t | X_1, ..., X_t] = 0$. On pourrait considérer un modèle qui est équivalent au leur dans le cas $\lambda \neq 0$ mais l'analyse n'aurait pas pu être étendue et c'est ce qu'on va expliquer dans ce qui suit. Considérons le modèle de type

$$Y_t = v(X_t) + \Xi_t$$

dans le cas $\lambda \neq 0$, où Ξ_t est le terme d'erreur correspondant. La variable Ξ_t vérifie pour $Y_t =$

$T^{\lambda}v(X_t)$

$$\begin{aligned}\Xi_t &= T^{\lambda}v(X_t) - v(X_t)\\ &= (1-\lambda)\sum_{i=0}^{\infty}\lambda^i(T^{i+1}v(X_t) - v(X_t))\\ &= (1-\lambda)\sum_{i=0}^{\infty}\lambda^i\sum_{k=0}^{i}(T^{k+1} - T^k)v(X_t).\end{aligned}$$

Rappelons dans ce cadre que

$$T^k v(X_t) = \sum_{i=0}^{k-1}\gamma^i r(X_{t+i}) + \gamma^k v(X_{t+k}).$$

En remplaçant $T^k v(X_t)$ par son expression il s'en suit que

$$\begin{aligned}\Xi_t &= (1-\lambda)\sum_{i=0}^{\infty}\lambda^i\sum_{k=0}^{i}(\gamma^k r(X_t) + \gamma^{k+1}v(X_{t+k+1}) - \gamma^k v(X_{t+k}))\\ &= (1-\lambda)\sum_{i=0}^{\infty}\lambda^i\sum_{k=0}^{i}\gamma^k\xi_{t+k}\\ &= \sum_{k=0}^{\infty}(\lambda\gamma)^k\xi_{t+k}.\end{aligned}$$

Pour $\lambda = 0$, on retrouve bien $\Xi_t = \xi_t$. L'erreur Ξ_t correspond à la somme de toutes les erreurs qui se sont propagées au cours du temps facteur le terme $(\lambda\gamma)^k$. Si on suppose ξ_{t+k} bornée pour tout k alors la variable Ξ_t serait également bornée. Par ailleurs elle n'est pas adaptée à la filtration $X_1, ..., X_{t+1}$ mais à celle définie par les variables $X_1, ..., X_\infty$. Nous pouvons tronquer l'erreur Ξ_t avec un nombre L par exemple de sorte que Ξ_t soit $X_1, ..., X_{L+t+1}$ adaptée. Ceci dit on aura

$$\mathbb{E}[\Xi_t | X_1, ..., X_{L+t}] = \sum_{k=0}^{L-1}\xi_{t+k} \neq 0.$$

L'une des conditions nécessaires au raisonnement dans les travaux de [Lazaric et al., 2012] (indispensable pour pouvoir appliquer l'inégalité d'Azuma) n'est pas satisfaite dans le cas de $\lambda \neq 0$. Nous ne pouvons donc pas reprendre l'analyse proposée dans le cas $\lambda = 0$. Nous pensons néanmoins qu'il est possible, en utilisant la technique de troncature des traces que nous avons suivie ici, d'étendre leur analyse au cas $\lambda \neq 0$. Cependant, ce faisant, on gardera toujours le facteur $4\sqrt{2}$ dans la borne finale. Une question un peu plus difficile serait d'envisager une telle analyse pour l'algorithme off-policy LSTD(λ) introduit dans le chapitre précédent (cf algorithme 10, chapitre 4). Nous gardons cette perspective pour des travaux futurs.

6

Calcul d'une borne de performance pour l'algorithme LS(λ)NSPI

6.1 Introduction

On va étendre dans ce chapitre l'algorithme Least Square Policy Iteration (LSPI) décrit dans le chapitre 4 aux cas des politiques non stationnaires. On va s'intéresser au cas du schéma algorithmique itérations sur les politiques non stationnaires de période fixe p, NSPI(p) [Scherrer, 2014]. L'algorithme LS(λ)NSPI obtenu est une généralisation de LSPI au cas de $p > 1$ et de $\lambda > 0$. On propose dans ce chapitre une description de cet algorithme. On introduit une borne de performance pour LS(λ)NSPI en fonction du nombre d'échantillons n, du paramètre λ, de la période p ainsi que des coefficients de concentration présentés dans [Scherrer, 2014]. On obtient asymptotiquement une meilleure borne que celle obtenue par [Lazaric et al., 2012] qui ont uniquement considéré le cas $p = 1$, $\lambda = 0$. On va montrer qu'augmenter la période p, pour un nombre d'échantillons n fixé, améliore en général la qualité de la borne. C'est ce qu'on déduit de la borne dérivée et de l'ensemble de simulations effectuées sur une certaine classe de MDP. On va également étudier les variations du paramètre optimal λ^*–minimisant la borne–en fonction du nombre d'échantillons et de la période p.

6.2 Description de l'algorithme LS(λ)NSPI

L'algorithme *Least square non stationary policy iteration de paramètre* λ, LS(λ)NSPI utilise LSTD(λ) dans un schéma de type itérations politiques *non stationnaires* au lieu d'un schéma de politiques *stationnaires* comme c'était le cas pour LSPI. Considérer des politiques non stationnaires peut être intéressant puisqu'il s'avère que certains algorithmes de la programmation dynamique avec approximation tels que API et AVI possèdent une meilleure borne de performance dans le cas où on considère des politiques non stationnaires. C'est ce qu'on va développer dans ce qui suit.

6.2.1 Cadre général

On considère π la *politique périodique non stationnaire* composée des i politiques stationnaires $\pi_1, \pi_2, ..., \pi_i$ pour $i \geq 1$. Cette politique choisit la première action selon la politique π_i, la deuxième action selon la politique π_{i-1}...et la i ème selon la politique π_1, l'action $i+1$ est choisie selon la

politique π_i et ainsi de suite...Elle s'exprime formellement de la manière suivante :

$$\pi = \pi_i \pi_{i-1} ... \pi_1 \pi_i \pi_{i-1} ... \pi_1 ...$$

La fonction de valeur v_π associée à cette politique est l'unique solution de l'équation

$$v = T_\pi v, \text{ avec } T_\pi = T_\pi = T_{\pi_i} T_{\pi_{i-1}} ... T_{\pi_1}. \tag{6.1}$$

L'opérateur de Bellman T_π introduit dans cette équation satisfait pour tout $v \in \mathbb{R}^N$

$$T_\pi v = r_\pi + \Gamma_\pi v$$

où

$$r_\pi = r_{\pi_i} + \sum_{l=1}^{i-1} \gamma^l \prod_{j=0}^{l-1} P_{\pi_{i-j}} r_{\pi_{i-l}}$$

et $\Gamma_\pi = (\gamma P_{\pi_i})(\gamma P_{\pi_{i-1}})...(\gamma P_{\pi_1})$. L'opérateur T_π est $(\gamma^i, \|.\|_\infty)$ contractant, en effet on a pour tous vecteurs $v, w \in \mathbb{R}^{|S|}$

$$\begin{aligned} \|T_\pi u - T_\pi w\|_\infty &\leq \|\Gamma_\pi (u - w)\|_\infty \\ &\leq \gamma^i \|u - w\|_\infty. \end{aligned}$$

Par le théorème du point fixe de Banach, il existe une unique solution v_π à l'équation (6.1).

6.2.2 AVI utilisant des politiques non stationnaires

Le schéma algorithmique itérations sur les valeurs avec approximation AVI (cf. section 4.1.1, chapitre 4) calcule implicitement au bout de k itérations les politiques π_0,π_1,...,π_k qui sont les politiques gloutonnes par rapport aux valeurs v_0,...,v_k. Cet algorithme peut être décrit comme suit

$$v_{k+1} \leftarrow T v_k + \epsilon_{k+1}$$

où T est l'opérateur de Bellman optimal.

Au lieu de considérer la dernière politique gloutonne π_k, on prend en compte les p dernières politiques gloutonnes π_k,π_{k-1},...,π_{k-p+1}. [Scherrer and Lesner, 2012] ont calculé une borne de performance de cet algorithme. Supposons que $\|\epsilon_k\|_\infty \leq \epsilon$ pour tout k, on a le résultat suivant.

Théorème 8. *Pour chaque itération k et chaque période p vérifiant $1 \leq p \leq k$, l'erreur entre la valeur v^* et la valeur $v_{\pi_{k,p}}$ vérifie*

$$\|v^* - v_{\pi_{k,p}}\|_\infty \leq \frac{2}{1 - \gamma^p} \left(\frac{\gamma - \gamma^k}{1 - \gamma} \epsilon + \gamma^k \|v_k - v_0\|_\infty \right),$$

où $\pi_{k,p} = \pi_k \pi_{k-1} ... \pi_{k-p+1} \pi_k \pi_{k-1} ... \pi_{k-p+1} ....$ Quand $k \to \infty$ on remarque qu'on obtient une borne de performance meilleure que dans le cas de AVI introduit dans le chapitre 4. En effet l'erreur globale est majorée dans ce cas par $\frac{2\gamma}{(1-\gamma^p)(1-\gamma)}\epsilon$. Pour de grandes valeurs de p, on peut significativement améliorer le résultat, surtout lorsque le paramètre d'actualisation γ est proche de 1.

Algorithme 12 AVI

Initialiser v_0
Pour $i = 0, 1, 2, ...$ **faire**
  $v_{k+1} \leftarrow Tv_k + \epsilon_{k+1}$
fin « Pour »

6.2.3 API utilisant des politiques non stationnaires de période croissante

Le schéma algorithmique API utilisant les politiques non stationnaires considère à chaque itération k les k dernières politiques gourmandes. La politique $\pi_{k,k}$ générée est construite à chaque fois à partir d'un nombre croissant de politiques stationnaires, d'où l'appellation "politique de période croissante". Cet algorithme peut être décrit comme suit

$$\begin{aligned} v_k &\leftarrow v_{\pi_{k,k}} + \epsilon_k \\ \pi_{k+1} &\in \mathcal{G}(v_k). \end{aligned} \tag{6.2}$$

Une borne de performance a été calculée dans [Scherrer and Lesner, 2012]. On a le résultat suivant.

Théorème 9. *Au bout de k itérations, l'erreur entre la valeur v^* et la valeur $v_{\pi_{k,k}}$ vérifie*

$$\|v^* - v_{\pi_{k,k}}\|_\infty \leq \frac{2(\gamma - \gamma^k)}{1-\gamma}\epsilon + \gamma^{k-1}\|v^* - v_{\pi_{1,1}}\|_\infty + 2(k-1)\gamma^k V_{\max}.$$

Lorsque le nombre d'itérations k devient infiniment grand cet algorithme renvoie une borne de performance meilleure d'un facteur $\frac{1}{1-\gamma}$ que celle de l'algorithme API introduit dans le chapitre 4.

6.2.4 API utilisant des politiques non stationnaires de période fixe p

L'algorithme *itérations sur les politiques non stationnaires de période p*, NSPI(p) introduit dans [Scherrer and Lesner, 2012], [Scherrer, 2014] est une variante de l'algorithme *approximate policy iteration* (API) utilisant des politiques *non stationnaires*. Cet algorithme peut être décrit comme suit

$$\begin{aligned} v_k &\leftarrow v_{\pi_{k,p}} + \epsilon_k \\ \pi_{k+1} &\in \mathcal{G}(v_k), \end{aligned} \tag{6.3}$$

où $\mathcal{G}_{\epsilon_{k+1}}(v_k)$ est l'ensemble des politiques gourmandes par rapport à v_k. La politique $\pi_{k,p}$ est telle que $\pi_{k,p} = \pi_k\pi_{k-1}...\pi_{k-p+1}\pi_k\pi_{k-1}...\pi_{k-p+1}...$ pour p un entier fixé vérifiant $p \leq k$. La valeur $v_{\pi_{k,p}}$ associée est solution de l'équation (6.1). La valeur v_k est une approximation de $v_{\pi_{k,p}}$. On note l'erreur d'approximation ϵ_k. La politique optimale π_{k+1} est déduite à partir de v_k. Celle-ci appartient à l'ensemble des politiques gourmandes par rapport à v_k, noté $\mathcal{G}(v_k)$. A l'itération $k+1$ la politique $\pi_{k+1,p}$ est la concaténation de π_{k+1} et de $\pi_k...\pi_{k-p}$, elle vérifie donc $\pi_{k+1,p} = \pi_{k+1}...\pi_{k-p+2}\pi_{k+1}...\pi_{k-p+2}....$

Théorème 10. *Pour tout p tel que $k \geq p$, l'erreur entre la valeur v^* et $v_{\pi_{k,p}}$ vérifie*

$$\|v^* - v_{\pi_{k,p}}\|_\infty \leq \gamma^{k-p}\|v^* - v_{\pi_{p,p}}\|_\infty + \frac{2(\gamma - \gamma^{k+1-p})}{(1-\gamma)(1-\gamma^p)}\epsilon.$$

Lorsque la période p devient infiniment grande (et par conséquent k aussi), on obtient la même borne de performance que celle de l'algorithme API avec une période croissante. Cela requiert ceci dit une mémoire infinie, ce qui est restrictif même si cela permet d'obtenir une borne de meilleure qualité.

On propose dans ce qui suit une analyse de l'algorithme LSTD(λ) dans le cas d'une politique non stationnaire π. Cette analyse nous permettra par la suite de déduire une borne de performance pour l'algorithme NSPI(p) dans le cas où on utilise LSTD(λ) pour estimer v_π.

Algorithme 13 API non stationnaire

Initialiser $\pi_{0,p} = \pi_p \pi_{p-1} ... \pi_0$
Pour $i = 0, 1, 2, ...$ **faire**
  $v_k \leftarrow v_{\pi_{k,p}} + \epsilon_k$
  $\pi_{k+1} \leftarrow \mathcal{G}(v_k)$
  $\pi_{k+1,p} \leftarrow \pi_{k+1} \pi_{k,p-1}$
fin « Pour »

6.2.5 L'algorithme Least square temporal difference LSTD(λ) dans le cas non stationnaire

L'algorithme Least square temporal difference LSTD(λ) dans le cas non stationnaire est l'algorithme qui renvoie une estimation de v_π, où π est la politique non stationnaire de période fixe p, $\pi = \pi_{p-1}\pi_{p-2}...\pi_0\pi_{p-1}\pi_{p-2}...\pi_0...$ Par analogie au cas stationnaire on va faire les hypothèses qui suivent.

Hypothèse 4. *Les fonctions features $(\phi_j)_{1 \leq j \leq d}$ sont linéairement indépendantes.*

Hypothèse 5. *On va supposer la matrice Γ_π irréductible et apériodique (voir définition 25 dans l'annexe B) de sorte qu'il existe une mesure invariante μ qui satisfait $\mu\Gamma_\pi = \mu$* [15].

Le but de l'algorithme Least square temporal difference LSTD(λ) dans le cas non stationnaire est de calculer une solution à l'équation

$$v_{LSTD(\lambda)} = \Pi T_\pi^\lambda v_{LSTD(\lambda)}, \text{ où } T_\pi^\lambda = (1-\lambda)\sum_{i=0}^{\infty} \lambda^i T_\pi^{i+1}, \tag{6.4}$$

et où la matrice Π est la matrice de projection orthogonale sur l'espace des features selon la norme $\|.\|_\mu$. L'opérateur T_π est l'opérateur de Bellman associé à π.

La solution $v_{LSTD(\lambda)}$ appartient à l'espace défini par les fonctions features de sorte qu'il existe $\theta \in \mathbb{R}^d$ tel que $v_{LSTD(\lambda)} = \Phi\theta$. L'équation (6.4) se réécrit alors de la sorte

$$A\theta = b$$

telles que les matrices A_π et b_π vérifient

$$A = \Phi^t D_\mu (I - \Gamma_\pi)(I - \lambda\Gamma_\pi)^{-1}\Phi, \quad b = \Phi^t D_\mu (I - \lambda\Gamma_\pi)^{-1} r_\pi.$$

Grâce à l'hypothèse 5 on a l'opérateur de projection orthogonal Π non expansif par rapport à $\|.\|_\mu$ donc il existe une unique solution à l'équation $v = \Pi T_\pi v$. L'hypothèse 4 implique de plus

15. Noter que dans le cas d'un espace au plus dénombrable l'existence d'une mesure invariante μ finie implique $\mu(i) > 0$ pour tout i.

que la matrice $\Phi^t D_\mu \Phi$ est inversible d'où l'unicité du vecteur θ vérifiant $\theta = (\Phi^t D_\mu \Phi)^{-1}\Phi v$. L'hypothèse 5 est difficile à remplir en général car le produit de deux matrices irréductibles et apériodiques n'est pas irréductible et apériodique. En effet une matrice P irréductible et apériodique est une matrice *primitive*[16] et le produit de deux matrices primitives n'est pas une matrice primitive.

Définition 16. *Une matrice A est dite primitive ssi il existe un entier m tel que $A^m > 0$.*

On va illustrer cela par un exemple.

Exemple 1. *Si on considère deux politiques π_0 et π_1 telles que les matrices P^{π_0} et P^{π_1} sont primitives, on peut montrer que $P^{\pi_0}P^{\pi_1}$ n'est pas en général une matrice primitive [Schwarz, 1965]. Soient*

$$P^{\pi_0} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}, \; P^{\pi_1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}.$$

Considérons le produit $P^{\pi_0}P^{\pi_1}$ on a

$$P^{\pi_0}P^{\pi_1} = \begin{pmatrix} 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 1 \end{pmatrix}.$$

La matrice P^{π_0} est irréductible car $1 \to 2 \to 3 \to 1$ et $3 \to 2$. Elle est apériodique car l'état 2 est apériodique $2 \to 3 \to 2$ et $2 \to 3 \to 1 \to 2$. De même on peut s'apercevoir que la matrice P^{π_1} est irréductible et apériodique. Par contre dans la matrice $P^{\pi_0}P^{\pi_1}$ on peut remarquer que $1 \nrightarrow 1$ donc la matrice $P^{\pi_0}P^{\pi_1}$ n'est pas irréductible.

A priori nous ne pouvons donc rien conclure sur la matrice Γ_π pour π une politique non stationnaire, on a besoin de faire des hypothèses en plus pour s'assurer que la matrice Γ_π est irréductible et apériodique. Il existe des conditions dans [Schwarz, 1965] qui assurent que la matrice produit est primitive mais elles sont difficiles à satisfaire vu que dans notre cas, on a p matrices. On peut notamment supposer que pour tout $i \in \{0, ..., p-1\}$, les matrices P_{π_i} composant la matrice Γ_π sont strictement positives c'est-à-dire que tous les éléments sont strictement positifs ($m = 1$) mais c'est une hypothèse qui est également assez forte. Nous nous restreindrons à l'hypothèse 5 qui suppose que Γ_π est une matrice primitive.

Description de l'algorithme Dans cette section on va décrire l'algorithme LSTD(λ) dans le cas d'une politique non stationnaire. Par analogie au cas stationnaire on approche les matrices A et b par respectivement les matrices $\hat{A}$ et $\hat{b}$ définies comme suit :

$$\hat{A} = \frac{1}{l-1}\sum_{i=1}^{l-1} z_i(\phi^t(X_{pi})' - \gamma^p \phi^t(X_{p(i+1)})') \tag{6.5}$$

$$\hat{b} = \frac{1}{l-1}\sum_{i=1}^{l-1} z_i r_\pi(X_{pi}), \quad z_i = \sum_{j=1}^{i} (\gamma^p\lambda)^{i-j}\phi(X_{pj}), \tag{6.6}$$

$$\text{où } r_\pi(X_{pi}) = \sum_{j=0}^{p-1} \gamma^j r_{\pi_{p-1-j}}(X_{p(i-1)+j}).$$

16. Dans le cas d'une chaîne de Markov finie ces deux notions sont équivalentes.

Les variables $(X_{pi})_{i\geq 1}$ vérifient, $X_{p(i+1)} \sim \Gamma_\pi(.|X_{pi})$, avec $\Gamma_\pi = P_{\pi_{p-1}} P_{\pi_{p-2}} ... P_{\pi_0}$. Pour tout $i \geq 1$ on a, $X_{p(i-1)+j+1} \sim P_{\pi_{p-1-j}}(.|X_{p(i-1)+j})$ avec $j \in \{0, ..., p-1\}$.

Proposition 7. *Considérons le processus $(Y_i)_{i\geq 1} = (X_{pi})_{i\geq 1}$ tel que $Y_1 \sim \mu$ alors $(Y_i)_{i\geq 1}$ est une chaîne de Markov CM (μ, Γ_π).*

Démonstration.

$$
\begin{aligned}
&\mathbb{P}(Y_i = y_i | Y_{(i-1)} = y_{(i-1)} ...., Y_0 = y_0) = \mathbb{P}(X_{pi} = y_i | X_{p(i-1)} = y_{(i-1)} ....X_0 = y_0) \\
&= \frac{\mathbb{P}(X_{pi} = y_i, X_{p(i-1)} = y_{(i-1)} ....Y_0 = y_0)}{\mathbb{P}(X_{p(i-1)} = y_{i-1} ....X_0 = y_0)} \\
&= \sum_{x_{pi-1}, ..., x_{pi-1+p}} \frac{\mathbb{P}(X_{pi} = y_i, X_{pi-1} = x_{pi-1}, X_{pi-2} = x_{pi-2}, ..., X_{p(i-1)} = y_{(i-1)} ....X_0 = y_0)}{\mathbb{P}(X_{p(i-1)} = y_{(i-1)} ....X_0 = y_0)} \\
&= \sum_{x_{pi-1}, ..., x_{pi-1+p}} p_{\pi_{p-1}}(y_{(i-1)}, x_{pi-(p-1)}) ... p_{\pi_0}(x_{pi-1}, y_i) \\
&= \Gamma_\pi(y_{i-1}, y_i) \\
&= \mathbb{P}(Y_i = y_i | Y_{(i-1)} = y_{(i-1)}).
\end{aligned}
$$

□

En utilisant le théorème 13 et la proposition 9 dans l'annexe B [Nedic and Bertsekas, 2002] appliqués au processus $(Y_i)_{i\geq 1}$, on a $\hat{A} \to A$, $\hat{b} \to b$ *presque sûrement.*

Vitesse de convergence de l'algorithme LSTD(λ) étant donnée une politique non stationnaire π

La chaîne de Markov $(Y_i)_{i\geq 1}$ est stationnaire et ergodique donc β-mélangeante [Bradley, 2005]. On supposera qu'elle est exponentiellement β-mélangeante [17](voir définition 14 page 64). Notons $\overline{\beta}$, b et κ ses paramètres. On peut appliquer une analyse similaire à celle du cas stationnaire en remplaçant le paramètre γ par la paramètre γ^p et le nombre d'échantillons n (correspondant aux échantillons X_i) par $\frac{n}{p}$ (correspondant aux échantillons Y_i). Un équivalent du théorème 1 dans le cas non stationnaire peut s'écrire alors de la manière suivante.

Théorème 11. *On fait les hypothèses 4 et 5 et on suppose $Y_1 \sim \mu$. Pour tout $n \geq 1$ et $\delta \in (0,1)$, on définit les fonctions :*

$$
I(n,\delta) = 32\Lambda(n,\delta) \max\left\{\frac{\Lambda(n,\delta)}{b}, 1\right\}^{\frac{1}{\kappa}}
$$

$$
et \;\; \Lambda(n,\delta) = \log\left(\frac{8n^2}{\delta}\right) + \log(\max\{4e^2, n\overline{\beta}\}).
$$

17. Si S est un espace continu ou dénombrable alors la chaîne est β-mélangeante. Si par contre l'espace S est fini alors elle sera exponentiellement β-mélangeante.

Soit $n_0(\delta)$ le plus petit entier tel que

$$\forall n \geq n_0(\delta),\ \frac{2dL^2}{(1-\gamma^p)\nu}\left[\frac{2\sqrt{p}}{\sqrt{n-p}}\sqrt{\left(\left\lceil\frac{\log(\frac{n}{p}-1)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil+1\right)I(\frac{n}{p}-1,\delta)+\frac{p}{(n-p)(1-\lambda\gamma^p)}+\frac{2p}{(n-p)}\left\lceil\frac{\log(\frac{n}{p}-1)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil}\right]<1$$

où ν est la plus petite valeur de propre de la matrice $\Phi^t D_\mu \Phi$. Alors, pour tout δ, avec probabilité $1-\delta$, pour tout $n \geq n_0(\delta)$, la matrice $\hat{A}$ est inversible et on a :

$$\|v_{LSTD(\lambda)} - \hat{v}_{LSTD(\lambda)}\|_\mu \leq \frac{4V_{max}dL^2\sqrt{p}}{\sqrt{n-p}(1-\gamma^p)\nu}\sqrt{\left(1+\left\lceil\frac{\log\left(\frac{n}{p}-1\right)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil\right)I\left(\frac{n}{p}-1,\delta\right)} + h\left(\frac{n}{p},\delta\right),$$

où $h(n,\delta) = \tilde{O}(\frac{1}{n})$ et $V_{\max} = \frac{R_{\max}}{(1-\gamma)}$.

Corollaire 12. *L'erreur globale vérifie alors*

$$\|\hat{v}_{LSTD(\lambda)} - v_\pi\|_\mu \leq \frac{1-\lambda\gamma^p}{1-\gamma^p}\|v_\pi - \Pi v_\pi\|_\mu + \frac{4V_{max}dL^2\sqrt{p}}{\sqrt{n-p}(1-\gamma^p)\nu}\sqrt{\left(1+\left\lceil\frac{\log\left(\frac{n}{p}-1\right)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil\right)I\left(\frac{n}{p}-1,\delta\right)} + h\left(\frac{n}{p},\delta\right). \tag{6.7}$$

A l'instar de LSTD(λ) stationnaire, on peut s'apercevoir que la valeur $\lambda = 1$ minimise la borne sur l'erreur d'approximation et que $\lambda = 0$ minimise la borne sur l'erreur d'estimation. Il existe une valeur optimale $\lambda^* \in [0,1]$ qui minimise la borne globale.

6.2.6 Borne de performance de l'algorithme LS(λ)NSPI

Par souci de simplicité, l'algorithme LS(λ)NSPI qu'on considère dans ce qui suit fait intervenir l'algorithme LSTD(λ) et non LSTDQ(λ) (cf. chapitre 4). Autrement dit on aura besoin lors de l'estimation de la politique gourmande d'un modèle de MDP tel que

$$\forall s \in S,\ \pi(s) \in \mathcal{G}(v(s)) = \arg\max_{a\in A}\left[r(s,a) + \gamma\sum_{s'\in S} p(s,a,s')v(s')\right].$$

L'algorithme LS(λ)NSPI utilisant les fonctions d'actions valeurs Q peut nécessiter, comme il a été mentionné dans [Lazaric et al., 2012] pour LSPI, l'introduction de politiques gourmandes stochastiques. Ces politiques permettent d'explorer tout l'espace d'actions même celles qui ne vérifient pas $a = \pi(.)$.

On ne peut pas utiliser pour estimer une borne de performance pour l'algorithme LS(λ)NSPI (cf. algorithme 14) le résultat du théorème 10, puisqu'il faut avoir un contrôle uniforme sur les erreurs $\|v_k - v_{\pi_{k,p}}\|_\infty$ pour tout k. Le théorème 11 introduit une borne d'erreur en norme L_2 pondérée par la mesure $\mu_{\pi_{k,p}}$. On va alors calculer la borne sur l'erreur $v^* - v_K$ en norme L_q, $q \in \{1,2\}$ pondérée par une certaine mesure σ. On a besoin dans ce cas d'introduire les coefficients de concentration qui relient la distribution σ à la distribution ρ défnie dans l'hypothèse 6.

Algorithme 14 Least Square Non Stationary Policy Iteration Algorithm LS(λ)NSPI

Entrée : $\lambda, N, \nu, \theta_0, \theta_1, ..., \theta_{p-1}, \gamma, \phi$
Pour $k = p-1, p, ...$ **faire**
  Générer : $X_0 \sim \nu$
  Générer : la trajectoire de taille N, $(X_0, a_0, r_0, X_1, a_1, r_1, ..., X_N)$ telle que $a_i = \mathcal{G}(\phi^t \theta_{k-i+\left[\frac{i}{p}\right]p}(X_i))$
  $\hat{A} := 0$, $\hat{b} := 0$, $t := 0$, $z_0 := 0$
  Pour $t = 1, 2, ..., N$ **faire**
    $\hat{A} \leftarrow \hat{A} + z_{t-1}(\phi(X_{t-1}) - \gamma^p \phi(X_{t+p-2}))^t$
    $\hat{b} \leftarrow \hat{b} + z_{t-1}(r_{\pi_k}(X_{t-1}) + \sum_{j=1}^{p-1} \gamma^j r_{\pi_{k-j}}(X_{t-1+j}))$
    $z_t \leftarrow \lambda \gamma^p z_{t-1} + \phi(X_{t+p-2})$
  fin « Pour »
  Renvoyer : $\theta_{k+1} \leftarrow \hat{A}^{-1}\hat{b}$
fin « Pour »

Définition 17. *[Scherrer, 2014] Soient σ et ρ deux distributions données, pour tout i et q on note $c_q(i)$ la derivée de Radon Nikodym suivante*[18] *(voir définition 33 dans l'annexe C) pour tout $i \geq 1$*

$$c_q^{\sigma,\rho}(i) = \sup_{\pi_1,...,\pi_i} \left\| \frac{d\sigma P_{\pi_1} P_{\pi_2} ... P_{\pi_i}}{d\rho} \right\|_{q,\rho}$$

où pour tout $u \in \mathbb{R}^N$ on a $\|u\|_{q,\rho} = \left(\sum_{i=1}^N u^q(i)\rho(i)\right)^{\frac{1}{q}}$. On note pour $p, q \in \mathbb{N}^* \cup \{\infty\}$ et $u \in \mathbb{N}$, la constante $C_{q,p,u}^{\sigma,\rho}$ qui vérifie

$$C_{q,p,u}^{\sigma,\rho} = (1-\gamma^p)(1-\gamma) \sum_{j=0}^{\infty} \sum_{t=0}^{\infty} \gamma^{j+tp} c_q^{\sigma,\rho}(j+tp+u),$$

où pour $i = 0$, $c_\infty^{\sigma,\rho}(0) = \left\| \frac{d\sigma}{d\rho} \right\|_{q,\rho}$.

Définition 18. *[Scherrer, 2014] Soient $c_{\pi^*}(1), c_{\pi^*}(2), ...$ les plus petits coefficients dans $[1, \infty]$ telles que pour tout i et pour toutes les politiques $\pi_1, \pi_2, ..., \pi_i$, $\sigma P_{\pi^*}^i \leq c_{\pi^*}(i)\rho$. On note $C_{\pi^*}^{(1)}$ le coefficient dans $[1, \infty]$ qui vérifie*

$$C_{\pi^*}^{(1)} = (1-\gamma) \sum_{i=0}^{\infty} \gamma^i c_{\pi^*}(i).$$

Le résultat qu'on va énoncer reste valide sous certaines hypothèses analogues à celles faites par [Lazaric et al., 2012] :

Hypothèse 6. *Il existe une distribution ρ telle que pour toute politique non stationnaire π, on a $\rho \leq C\mu_\pi$, avec C une constante vérifiant $0 < C < \infty$ et μ_π est la mesure invariante définie dans l'hypothèse 5.*

18. Si la mesure $\sigma P_{\pi_1} P_{\pi_2} ... P_{\pi_i}$ n'est pas absolument continue par rapport à la mesure ρ alors $c_q(i) = \infty$.

Hypothèse 7. *On supposera qu'il existe un processus exponentiellement β-mélangeant de paramètres $\overline{\beta}$, b et κ qui est plus lent que tous les processus β mélangeants définis par les politiques non stationnaires π. En d'autres termes il existe des paramètres $\overline{\beta}$, κ et b tels que*

$$\overline{\beta} \leq \overline{\beta}_\pi,\ b \leq b_\pi,\ \kappa \leq \kappa_\pi.$$

Hypothèse 8. *On suppose qu'il existe une constante $\tilde{\nu} \leq \nu_\pi$ pour toute politique non stationnaire π où ν_π est la petite valeur propre de la matrice $\Phi^t D_{\mu_\pi} \Phi$.*

Proposition 8. *On fait les hypothèses 4, 5, 6 et 7, on suppose que pour toute itération $k \in \{0, ..., K-1\}$, $Y_1 \sim \mu_{\pi_{k,p}}$. Pour tout $n \geq 1$ et $\delta \in (0,1)$, on définit les fonctions :*

$$I_{\overline{\beta},b,\kappa}(n,\delta) = 32\Lambda_{\overline{\beta},b,\kappa}(n,\delta)\max\left\{\frac{\Lambda_{\overline{\beta},b,\kappa}(n,\delta)}{b}, 1\right\}^{\frac{1}{\kappa}}$$

$$et\ \ \Lambda_{\overline{\beta},b,\kappa}(n,\delta) = \log\left(\frac{8n^2}{\delta}\right) + \log(\max\{4e^2, n\overline{\beta}\}).$$

Soit $n_0(\delta)$ le plus petit entier tel que

$$\forall n \geq n_0(\delta),\ \frac{2dL^2}{(1-\gamma^p)\tilde{\nu}}\left[\frac{2\sqrt{p}}{\sqrt{n-p}}\sqrt{\left(\left\lceil\frac{\log(\frac{n}{p}-1)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil+1\right)I_{\overline{\beta},b,\kappa}\left(\frac{n}{p}-1,\frac{\delta}{K}\right)}+\right.$$
$$\left.\frac{p}{(n-p)(1-\lambda\gamma^p)}+\frac{2p}{(n-p)}\left\lceil\frac{\log(\frac{n}{p}-1)}{\log\left(\frac{1}{\lambda\gamma^p}\right)}\right\rceil\right] < 1$$

où $\tilde{\nu}$ est définie dans l'hypothèse 8. Alors pour tout δ, avec probabilité $1-\delta$, pour tout $n \geq n_0(\delta)$, pour tout $k = 0, ..., K-1$, $\hat{A}_{\pi_{k,p}}$ est inversible et on a :

$$\|v^* - v_{\pi_{K,p}}\|_{1,\sigma} \leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[\sqrt{C}C^{\sigma,\rho}_{2,p,1}\left[\frac{(1-\lambda\gamma^p)}{1-\gamma^p}\max_{0\leq k<K}\|v_{\pi_{k,p}} - \Pi_{\mu_{\pi_{k,p}}}v_{\pi_{k,p}}\|_{\mu_{\pi_{k,p}}}\right.\right.$$
$$\left.\left.+\overline{g}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right)\right] + \gamma^{K-1}R_{\max}\right] \tag{6.8}$$

et

$$\|v^* - v_{\pi_{K,p}}\|_{2,\sigma} \leq \frac{2\gamma}{(1-\gamma^p)(1-\gamma)}\left[\sqrt{CC^{\sigma,\rho}_{\infty,p,1}}\left[\frac{(1-\lambda\gamma^p)}{1-\gamma^p}\max_{0\leq k<K}\|v_{\pi_{k,p}} - \Pi_{\mu_{\pi_{k,p}}}v_{\pi_{k,p}}\|_{\mu_{\pi_{k,p}}}+\right.\right.$$
$$\left.\left.+\overline{g}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right)\right] + \gamma^{K-1}R_{\max}\right],\ où\ \overline{g}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right) = \tilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{6.9}$$

Remarque 3. *La première borne obtenue est en norme L_1 pondérée par la mesure σ et la deuxième est en norme L_2 pondérée par la même mesure. Dans le cas $p = 1$ et $\lambda = 0$, la borne*

obtenue par [Lazaric et al., 2012] vérifie pour $n \geq \tilde{n}_0$, pour $\delta \in (0,1)$ avec probabilité $1-\delta$,

$$\|v^* - v_{\pi_K}\|_{2,\sigma} \leq \frac{4\gamma}{(1-\gamma)^2}\left((1+\gamma)\sqrt{CC^{\sigma,\rho}_{\infty,1,1}}\left[\frac{4\sqrt{2}}{\sqrt{1-\gamma^2}}\max_{0\leq k<K}\|v_{\pi_{k,p}} - \Pi_{\mu_{\pi_{k,p}}} v_{\pi_{k,p}}\|_{\mu_{j,p}} + \overline{h}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right)\right] + \gamma^{K-1}R_{\max}\right),$$

où $\overline{h}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right) = \tilde{O}\left(\frac{1}{\sqrt{n}}\right)$.

On obtient alors une meilleure borne, lorsque $n \to \infty$, que celle obtenue par [Lazaric et al., 2012] dans le cas $p = 1$ et $\lambda = 0$. On pourrait appliquer l'inégalité de Pythagore et avoir le terme $\frac{1}{\sqrt{1-\gamma^2}}$ qui apparaît dans la borne de [Lazaric et al., 2012] (voir la remarque 10 en bas de la page 66) au lieu de $\frac{1}{1-\gamma}$. Considérer une politique non stationnaire de période p grande améliore asymptotiquement la qualité de la borne de performance mais nécessite un nombre d'échantillons n plus grand. Il existe un paramètre λ^* compris dans l'intervalle $[0,1]$ qui minimise la borne globale. Ce paramètre dépend comme dans les cas précédents de n, p, $\beta_{\pi_{k,p}}$, $\kappa_{\pi_{k,p}}$, $b_{\pi_{k,p}}$ pour tout $k \in \{0,...,K-1\}$. Lorsque le nombre d'échantillons n tend vers l'infini, pour p fixé, on a λ^* qui tend vers 1. Un choix judicieux de la période serait de prendre p de l'ordre de $\left\lfloor \frac{\log(n)}{\log\left(\frac{1}{\gamma}\right)} \right\rfloor$ de sorte que $\gamma^p \sim \frac{1}{n}$. On aura alors lorsque p tend vers l'infini, n qui tend vers l'infini ($p \leq n$) et donc λ^* qui tend vers 0.

Remarque 4. *On peut utiliser les mêmes arguments que ceux de [Scherrer, 2014] pour montrer que les coefficients de concentration $C^{\sigma,\rho}_{\infty,p,1}$ et $C^{\sigma,\rho}_{\infty,1,1}$ sont équivalents, dans le sens où si l'un est fini (ou infini) alors l'autre l'est également (voir la proposition 12 dans l'annexe C).*

Remarque 5. *Comme on le montre dans le corps de la preuve de la proposition 8 qui suit, on peut dériver une autre borne en fonction des coefficients de concentration $C^{(1)}_{\pi^*}$ et $C^{\sigma,\rho}_{\infty,p,p}$,*

$$\|v^* - v_{\pi_{K,p}}\|_{2,\sigma} \leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[\sqrt{c^{\rho,\rho}_{\infty}(1)C\left(C^{(1)}_{\pi^*} + \gamma^p C^{\sigma,\rho}_{\infty,p,p}\right)}\left[\frac{(1-\lambda\gamma^p)}{1-\gamma^p}\max_{0\leq k<K}\|v_{\pi_{k,p}} - \Pi v_{\pi_{k,p}}\|_{\mu_{\pi_{k,p}}} + \overline{g}_{\overline{\beta},b,\kappa}\left(\frac{n}{p},\frac{\delta}{K}\right)\right] + \gamma^{K-1}R_{\max}\right]. \tag{6.10}$$

[Scherrer, 2014] propose une classification hiérarchique des coefficients de concentration. Les coefficients de concentration $C^{(1)}_{\pi^*}$ et $C^{\sigma,\rho}_{\infty,p,p}$ sont meilleurs que $C^{\sigma,\rho}_{\infty,p,1}$, dans le sens où :

1. *On a $C^{\sigma,\rho}_{\infty,p,1} \geq \frac{(1-\gamma^p)}{\gamma(p+1)}\left[C^{(1)}_{\pi^*} - c_{\pi^*}(0)\right]$ (voir proposition 12 dans l'annexe C). Si σ et ρ sont absolument continues, on a $c_{\pi^*}(0) < \infty$. On peut facilement avoir $C^{(1)}_{\pi^*} < \infty$ alors que la constante $C^{\sigma,\rho}_{\infty,p,1} = \infty$, puisque la constante $C^{(1)}_{\pi^*}$ dépend d'une seule politique alors que $C^{\sigma,\rho}_{\infty,p,1}$ dépend de toutes les politiques. Si σ et ρ ne sont pas absolument continues on ne peut pas conclure.*

2. *On a pour tout* $u \in \mathbb{N}^*$

$$\begin{aligned}
C^{\sigma,\rho}_{\infty,p,u} &= (1-\gamma^p)(1-\gamma)\sum_{j=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{j+tp}c_\infty(j+tp+u) \\
&= \frac{(1-\gamma^p)(1-\gamma)}{\gamma^{u-1}}\sum_{j=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{j+tp+u-1}c_\infty(j+tp+u) \\
&= \frac{(1-\gamma^p)(1-\gamma)}{\gamma^{u-1}}\sum_{z\geq u-1}\sum_{t=0}^{\infty}\gamma^{z+tp}c_\infty(z+tp+1) \\
&\leq \frac{1}{\gamma^{u-1}}C^{\sigma,\rho}_{\infty,p,1}.
\end{aligned}$$

Il est aussi facile d'avoir $C^{\sigma,\rho}_{\infty,p,u} < \infty$ *alors que* $C^{\sigma,\rho}_{\infty,p,1} = \infty$*, il suffit pour cela d'avoir* $c_q(z) = \infty$ *pour un certain* $z \leq u$*.*

Dans ce qui suit on propose une preuve de la proposition 8.

Démonstration. On obtient la condition sur le nombre d'échantillons n_0 en remplaçant l'équation page 79 par

$$\mathbb{P}\Big\{\exists\ n \geq 1, \exists k, 0 \leq k \leq K-1,\ \{\|\epsilon_{A_{\pi_{k,p}}}\|_2 \geq \epsilon_1(n,\delta_n)\} \cup \{\|\epsilon_{A_{\pi_{k,p}}}\theta - \epsilon_{b_{\pi_{k,p}}})\|_2 \geq \epsilon_2(n,\delta_n)\}\Big\}$$

ce qui nous mène à considérer le paramètre $\delta' = K\delta$. La condition *"pour tout* n*,* $\epsilon(n,\delta_n) < C$*, où* $C = \frac{(1-\gamma)\nu}{(1-\lambda\gamma)}$ *"* est remplacée par la condition *"pour tout* n*, pour tout* $0 \leq k \leq K-1$*,* $\epsilon(n, \frac{\delta_n}{K}) < C_k$*, où* $C_k = \frac{(1-\gamma)\nu_{\pi_{k,p}}}{(1-\lambda\gamma)}$ *"*. Puisque $\tilde{\nu}$ est une borne inférieure des $\nu_{\pi_{k,p}}$ pour $0 \leq k \leq K-1$, imposer *"pour tout* n*, pour tout* $0 \leq k \leq K-1$*,* $\epsilon(n, \frac{\delta_n}{K}) < \frac{(1-\gamma)\tilde{\nu}}{(1-\lambda\gamma)}$ *"* implique que *"pour tout* n*, pour tout* $0 \leq k \leq K-1$*,* $\hat{A}_{k,p}$ *est inversible"*. La condition sur le nombre d'échantillons est obtenue en écrivant $\frac{(1-\lambda\gamma)}{(1-\gamma)\tilde{\nu}}\epsilon(n, \frac{\delta_n}{K}) < 1$, $\tilde{\nu}$ est strictement positive puisque les valeurs propres $\nu_{\pi_{k,p}}$ le sont pour tout k sous les hypothèses 4 et 5.

On va prouver maintenant l'inégalité énoncée dans la proposition. Soient v^* la valeur qu'on cherche à estimer et $v_{\pi_{k+1,p}}$ la valeur étant donnée une politique $\pi_{k+1,p}$ non stationnaire donnée. On va suivre les mêmes étapes de preuve que dans [Scherrer and Lesner, 2012]. On rappelle que v^* est la valeur optimale, π^* la politique associée.

On a $\pi_{k,p}$ la politique non stationnaire, $T_{\pi_{k,p}} = T_{\pi_k}T_{\pi_{k-1}}...T_{\pi_{k-p+1}}$ est l'opérateur de Bellman associé qui vérifie pour tout v, $T_{\pi_{k,p}}v = r_{k,p} + \Gamma_{k,p}v$ avec

$$r_{k,p} = r_{\pi_k} + \sum_{i=1}^{p-1}\gamma^i\prod_{j=0}^{i-1}P_{\pi_{k-j}}r_{\pi_{k-i}},\ \text{et}\ \Gamma_{k,p} = (\gamma P_{\pi_k})(\gamma P_{\pi_{k-1}})...(\gamma P_{\pi_{k-p+1}}).$$

On a

$$\begin{aligned}
v_{\pi^*} - v_{\pi_{k+1,p}} &= T_{\pi^*}v^* - T_{\pi_{k+1,p}}v_{\pi_{k+1,p}} \\
&= T_{\pi^*}v^* - T_{\pi^*}v_{\pi_{k,p}} + T_{\pi^*}v_{\pi_{k,p}} - T_{\pi_{k+1,p+1}}v_{\pi_{k,p}} + T_{\pi_{k+1,p+1}}v_{\pi_{k,p}} - T_{\pi_{k+1,p}}v_{\pi_{k+1,p}} \\
&= \gamma P_{\pi^*}(v^* - v_{\pi_{k,p}}) + T_{\pi^*}v_{\pi_{k,p}} - T_{\pi_{k+1}}T_{\pi_{k,p}}v_{\pi_{k,p}} + T_{\pi_{k+1}}T_{\pi_{k,p}}v_{\pi_{k,p}} - T_{\pi_{k+1,p}}v_{\pi_{k+1,p}} \\
&= \gamma P_{\pi^*}(v^* - v_{\pi_{k,p}}) + T_{\pi^*}v_{\pi_{k,p}} - T_{\pi_{k+1}}v_{\pi_{k,p}} + T_{\pi_{k+1,p}}T_{\pi_{k-p+1}}v_{\pi_{k,p}} - T_{\pi_{k+1,p}}v_{\pi_{k+1,p}} \\
&= \gamma P_{\pi^*}(v^* - v_{\pi_{k,p}}) + T_{\pi^*}v_{\pi_{k,p}} - T_{\pi_{k+1}}v_{\pi_{k,p}} + \Gamma_{k+1,p}(T_{\pi_{k-p+1}}v_{\pi_{k,p}} - v_{\pi_{k+1,p}}).
\end{aligned} \tag{6.11}$$

Définissons pour tout $k \geq 0$

$$e_k = \max_{\pi'} T_{\pi'} v_{\pi_{k,p}} - T_{\pi_{k+1}} v_{\pi_{k,p}},$$

le maximum étant calculé ici composante par composante. On va essayer de borner le terme $(T_{\pi_{k-p+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}})$ en fonction de e_k

$$\begin{aligned} T_{\pi_{k-p+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}} &= T_{\pi_{k-p+1}} v_{\pi_{k,p}} - T_{\pi_{k+1}} v_{\pi_{k,p}} + T_{\pi_{k+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}} \\ &= T_{\pi_{k-p+1}} v_{\pi_{k,p}} - T_{\pi_{k+1}} v_{\pi_{k,p}} + T_{\pi_{k+1}} T_{\pi_{k,p}} v_{\pi_{k,p}} - T_{\pi_{k+1,p}} v_{\pi_{k+1,p}} \\ &= T_{\pi_{k-p+1}} v_{\pi_{k,p}} - T_{\pi_{k+1}} v_{\pi_{k,p}} + T_{\pi_{k+1,p}} T_{\pi_{k-p+1}} v_{\pi_{k,p}} - T_{\pi_{k+1,p}} v_{\pi_{k+1,p}} \\ &\leq e_k + \Gamma_{k+1,p}(T_{\pi_{k-p+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}}). \end{aligned}$$

On conclut alors que

$$T_{\pi_{k-p+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}} \leq (I - \Gamma_{k+1,p})^{-1} e_k.$$

En bornant $T_{\pi_{k-p+1}} v_{\pi_{k,p}} - v_{\pi_{k+1,p}}$ dans l'équation (6.11) il s'en suit que

$$v_{\pi^*} - v_{\pi_{k+1,p}} \leq \gamma P_{\pi^*}(v^* - v_{\pi_{k,p}}) + e_k + \Gamma_{k+1,p}(I - \Gamma_{k+1,p})^{-1} e_k.$$

Par induction, on obtient l'équation qui suit

$$v_{\pi^*} - v_{\pi_{K,p}} \leq \sum_{i=0}^{K-1} (\gamma P_{\pi^*})^i (I + \Gamma_{K-1-i,p}(I - \Gamma_{K-1-i,p})^{-1}) e_{K-1-i} + (\gamma P_{\pi^*})^K (v_{\pi^*} - v_{\pi_{0,p}}). \tag{6.12}$$

Puisque $\pi_{k+1} \in \mathcal{G}(v_k)$ on a $T_{\pi'} v_k \leq T_{\pi_{k+1}} v_k$ d'où

$$e_k \leq \max_{\pi'} \gamma(P_{\pi_k} - P_{\pi'})\epsilon_k = \gamma(P_{\pi_k} - P_{\pi'(k)})\epsilon_k, \tag{6.13}$$

où $\pi'(k)$ désigne n'importe quelle politique appartenant à $\arg\max_\pi \gamma(P_{\pi_k} - P_\pi)$. En bornant e_k dans l'équation (6.12), on a

$$\begin{aligned} |v_{\pi^*} - v_{\pi_{K,p}}| &\leq \sum_{i=0}^{K-1} (\gamma P_{\pi^*})^i (I + \Gamma_{K-1-i,p}(I - \Gamma_{K-1-i,p})^{-1})(\gamma P_{\pi_{K-1-i}} + \gamma P_{\pi'(K-1-i)})|\epsilon_{K-1-i}| \\ &\quad + \gamma^K P_{\pi^*}^K V_{\max} 1 \\ &= \sum_{i=0}^{K-1} (\gamma P_{\pi^*})^i \sum_{t \geq 0} \Gamma_{K-1-i,p}^t (\gamma P_{\pi_{K-1-i}} + \gamma P_{\pi'(K-1-i)})|\epsilon_{K-1-i}| + \gamma^K V_{\max} 1, \end{aligned} \tag{6.14}$$

le vecteur $|\epsilon_{K-1-i}|$ est le vecteur formé des valeurs absolues de chaque comosante et 1 est le vecteur $(1, 1, ..., 1)^t$. Dorénavant similairement à [Scherrer, 2014] on va introduire l'opérateur Γ défini de la manière suivante.

Définition 19. *Pour tout $n \in \mathbb{N}$, on définit $\mathbb{P}_n$ le plus petit ensemble des γ-noyaux de tranistion qui sont définis comme suit*

- *pour tout ensemble de n politiques, $\{\pi_1, \pi_2, ..., \pi_n\}$, $\gamma P_{\pi_1}, \gamma P_{\pi_2}, ..., \gamma P_{\pi_n} \in \mathbb{P}_n$,*
- *pour tout $\alpha \in (0, 1)$, $(P_1, P_2) \in P_n \times P_n$, $\alpha P_1 + (1 - \alpha) P_2 \in \mathbb{P}_n$.*

On note Γ^n n'importe quel élément de $\mathbb{P}_n$. Par exemple si on écrit un noyau de transition $P = \alpha_1\Gamma^i + \alpha_2\Gamma^j\Gamma^k = \alpha_1\Gamma^i + \alpha_2\Gamma^{j+k}$, cela signifie qu'il existe $P_1 \in \mathbb{P}_i$, $P_2 \in \mathbb{P}_j$, $P_3 \in P_k$, $P_4 \in \mathbb{P}_{k+j}$ tels que $P = \alpha_1 P_1 + \alpha_2 P_2 P_3 = \alpha_1 P_1 + \alpha_2 P_4$.

Cette définition permet de réécrire l'équation (6.14) de la manière suivante

$$|v_{\pi^*} - v_{\pi_{K,p}}| \leq 2\sum_{i=0}^{K-1}\sum_{t\geq 0}\Gamma^{i+pt+1}|\epsilon_{K-i-1}| + \gamma^K 1 V_{\max}.$$

Afin de finir la preuve de la proposition 8 on va suivre les mêmes étapes de preuve que celles du lemme 6 proposé dans [Scherrer, 2014]. On a par l'inégalité de Hölder (voir le lemme 8 et la proposition 11 dans l'annexe C)

$$\sigma\Gamma^t|z| = \|\Gamma^t|z|\|_{1,\sigma} \leq \gamma^t c_q^{\sigma,\rho}(t)\|z\|_{q',\rho} = \gamma^t c_q^{\sigma,\rho}(t)(\rho|z|^{q'})^{\frac{1}{q'}}. \tag{6.15}$$

Soit $L = 2\xi_1\sum_{i=0}^{K-1}\sum_{t\geq 0}\gamma^{i+pt+1} + \xi_2\gamma^K$, où ξ_1 et ξ_2 sont des réels positifs qu'on définira par la suite. On a

$$\begin{aligned}
\|v_{\pi^*} - v_{\pi_{K,p}}\|_{1,\sigma} &\leq L\frac{2\xi_1\sum_{i=0}^{K-1}\sum_{t\geq 0}\sigma\Gamma^{i+pt+1}\frac{|\epsilon_{K-1-i}|}{\xi_1} + \xi_2\gamma^K\sigma 1\frac{V_{\max}}{\xi_2}}{L} \\
&\leq L\frac{2\xi_1\sum_{i=0}^{K-1}\sum_{t\geq 0}\gamma^{i+pt+1}c_2^{\sigma,\rho}(i+pt+1)\left(\rho\frac{|\epsilon_{K-1-i}|^2}{\xi_1^2}\right)^{\frac{1}{2}} + \xi_2\gamma^K\frac{V_{\max}}{\xi_2}}{L} \\
&\leq L\frac{2\xi_1\sum_{i=0}^{K-1}\sum_{t\geq 0}\gamma^{i+pt+1}c_2^{\sigma,\rho}(i+pt+1)\frac{\|\epsilon_{K-1-i}\|_{\rho,2}}{\xi_1} + \xi_2\gamma^K\frac{V_{\max}}{\xi_2}}{L} \\
&\leq L\frac{2\xi_1\gamma\frac{C_{2,p,1}^{\sigma,\rho}}{(1-\gamma)^p(1-\gamma)}\frac{\max_{0\leq i\leq K-1}\|\epsilon_{K-1-i}\|_{\rho,2}}{\xi_1} + \xi_2\gamma^K\frac{V_{\max}}{\xi_2}}{L}.
\end{aligned}$$

On a $\sum_{i=0}^{K-1}\sum_{t\geq 0}\gamma^{i+pt+1} = \frac{\gamma(1-\gamma^K)}{(1-\gamma)(1-\gamma^p)}$. Si on prend $\xi_1 = \frac{C_{2,p,1}^{\sigma,\rho}}{1-\gamma^K}\max_{0\leq i\leq K-1}\|\epsilon_{K-i}\|_{\rho,2}$ et $\xi_2 = V_{\max}$ on obtient

$$\begin{aligned}
\|v_{\pi^*} - v_{\pi_{K,p}}\|_{1,\sigma} \leq L &= \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}C_{2,p,1}^{\sigma,\rho}\max_{0\leq i\leq K-1}\|\epsilon_{K-i-1}\|_{2,\rho} + V_{\max}\gamma^K \\
&\leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[C_{2,p,1}^{\sigma,\rho}\max_{0\leq i\leq K-1}\|\epsilon_{K-i-1}\|_{2,\rho} + R_{\max}\gamma^{K-1}\right].
\end{aligned}$$

D'après l'hypothèse 6 on sait que $\|.\|_\rho \leq \sqrt{C}\|.\|_{\mu_{\pi_{k,p}}}$ pour tout $0 \leq k \leq K-1$ d'où

$$\|v_{\pi^*} - v_{\pi_{K,p}}\|_{1,\sigma} \leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[C_{2,p,1}^{\sigma,\rho}\sqrt{C}\max_{0\leq i\leq K-1}\|\epsilon_{K-i-1}\|_{2,\mu_{\pi_{K-i,p}}} + R_{\max}\gamma^{K-1}\right]. \tag{6.16}$$

En combinant les équations (6.7) (page 91) et l'équation (6.16) on obtient l'équation (6.8). Notons ici que

$$\mathbb{P}\left(\max_{0\leq i\leq K-1}\|\epsilon_{K-i-1}\|_{2,\mu_{\pi_{K-i-1,p}}} \leq \epsilon(\delta')\right) = 1 - \mathbb{P}\left(\max_{0\leq i\leq K-1}\|\epsilon_{K-1-i}\|_{2,\mu_{\pi_{K-1-i,p}}} \geq \epsilon(\delta')\right).$$

L'événement $\{\max_{0\leq i\leq K-1}\|\epsilon_{K-1-i}\|_{2,\rho} \leq \epsilon(\delta')\}$ implique qu'il existe $j \in \{0,1,...,K-1\}$ tel que

$\|\epsilon_{K-1-j}\|_{2,\rho} \leq \epsilon(\delta')$ d'où

$$\mathbb{P}\left(\max_{0\leq i\leq K-1} \|\epsilon_{K-1-i}\|_{2,\mu_{\pi_{K-1-i,p}}} \geq \epsilon(\delta')\right) \leq \sum_{i=0}^{K-1} \mathbb{P}\left(\|\epsilon_{K-1-i}\|_{2,\mu_{\pi_{K-1-i,p}}} \geq \epsilon(\delta')\right) = K\delta'.$$

Si on pose $\delta = K\delta'$ on obtient l'inégalité avec probabilité $1-\delta$, telle qu'elle apparaît dans l'équation (6.8). On peut montrer la deuxième inégalité de la proposition en appliquant le même raisonnement mais pour la norme L_2. On obtient

$$\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma} \leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[\sqrt{CC_{\infty,p,1}^{\sigma,\rho}} \max_{0\leq i\leq K-1} \|\epsilon_{K-1-i}\|_{2,\mu_{\pi_{K-1-i,p}}} + R_{\max}\gamma^{K-1}\right]. \tag{6.17}$$

On prouve de la même façon l'inégalité (6.10) dans la remarque 5. En prenant $L = \xi_1 \sum_{i=0}^{K-1}\sum_{t\geq 0}\gamma^{i+pt} + \xi_2\gamma^K$. On a si on part de l'équation (6.12)

$$\begin{aligned}
\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma}^2 &\leq L^2\sigma\frac{\left(\sum_{i=0}^{K-1}(\gamma P_{\pi^*})^i(I+\Gamma^p(I-\Gamma^p)^{-1})|e_{K-1-i}| + \gamma^K 1 V_{\max}\right)^2}{L^2} \\
&\leq L^2\sigma\left(\frac{\xi_1\sum_{i=0}^{K-1}(\gamma P_{\pi^*})^i(I+\Gamma^p(I-\Gamma^p)^{-1})\frac{|e_{K-1-i}|}{\xi_1} + \xi_2\gamma^K\frac{1V_{\max}}{\xi_2}}{L}\right)^2 \\
&= L^2\sigma\left(\frac{\xi_1\sum_{i=0}^{K-1}(\gamma P_{\pi^*})^i\sum_{t\geq 0}\Gamma^{pt}\frac{|e_{K-1-i}|}{\xi_1} + \xi_2\gamma^K\frac{1V_{\max}}{\xi_2}}{L}\right)^2.
\end{aligned}$$

En appliquant l'inégalité de convexité (voir le théorème 14 dans l'annexe C, les coefficients λ_i correspondent à $\frac{\xi_1}{L}\sum_{t\geq 0}\gamma^{i+pt}$, pour $0\leq i\leq K-1$ et $\lambda_K = \frac{\xi_2\gamma^K}{L}$) et l'inégalité de Jensen (voir le théorème 15 dans l'annexe C) à l'opérateur $\sum_{t\geq 0} P_{\pi^*}^i\left(\frac{\Gamma^p}{\gamma^p}\right)^t$

$$\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma}^2 \leq L^2\frac{\xi_1\sum_{i=0}^{K-1}\gamma^i\sigma P_{\pi^*}^i(I+\sum_{t\geq 1}\Gamma^{pt})\left(\frac{|e_{K-1-i}|}{\xi_1}\right)^2 + \xi_2\gamma^K\left(\frac{V_{\max}}{\xi_2}\right)^2}{L}.$$

En utilisant l'équation 6.15, il s'en suit que

$$\begin{aligned}
\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma}^2 &\leq L^2\left[\frac{\xi_1\sum_{i=0}^{K-1}\left(\gamma^i c_{\pi^*}(i) + \sum_{t\geq 1}\gamma^{i+pt}c_\infty^{\sigma,\rho}(i+pt)\right)\rho\left(\frac{|e_{K-1-i}|}{\xi_1}\right)^2}{L} + \right. \\
&\qquad \left.\frac{\xi_2\gamma^K\left(\frac{V_{\max}}{\xi_2}\right)^2}{L}\right] \\
&\leq L^2\left[\frac{\xi_1\left(\frac{C_{\pi^*}^{(1)}}{1-\gamma} + \frac{\gamma^p C_{\infty,p,p}^{\sigma,\rho}}{(1-\gamma)(1-\gamma^p)}\right)\frac{\max_{0\leq i\leq K-1}\|e_{K-i}\|_{2,\rho}^2}{\xi_1^2}}{L} + \frac{\xi_2\gamma^K\left(\frac{V_{\max}}{\xi_2}\right)^2}{L}\right] \\
&\leq L^2\frac{\xi_1\frac{(C_{\pi^*}^{(1)}+\gamma^p C_{\infty,p,p}^{\sigma,\rho})(1-\gamma^K)}{(1-\gamma)(1-\gamma^p)(1-\gamma^K)}\frac{\max_{0\leq i\leq K-1}\|e_{K-1-i}\|_{2,\rho}^2}{\xi_1^2} + \xi_2\gamma^K\left(\frac{V_{\max}}{\xi_2}\right)^2}{L}.
\end{aligned}$$

En posant $\xi_1 = \sqrt{\frac{C^{(1)}_{\pi^*}+\gamma^p C^{\sigma,\rho}_{\infty,p,p}}{(1-\gamma^K)}}\max_{0\leq i\leq K-1}\|e_{K-1-i}\|_{2,\rho}$ et $\xi_2 = V_{\max}$, on obtient

$$\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma} \leq L \leq \frac{(1-\gamma^K)}{(1-\gamma^p)(1-\gamma)}\sqrt{\frac{C^{(1)}_{\pi^*}+\gamma^p C^{\sigma,\rho}_{\infty,p,p}}{(1-\gamma^K)}}\max_{0\leq i\leq K-1}\|e_{K-1-i}\|_{2,\rho} + \gamma^K V_{\max}.$$

L'équation (6.13) implique que

$$\|e_{K-1-i}\|_{2,\rho} \leq 2\gamma\sqrt{c^{\rho,\rho}_{\infty}(1)}\|\epsilon_{K-1-i}\|_{2,\rho},$$

où la constante $c^{\rho,\rho}_{\infty}(1)$ est introduite dans la définition 17. En utilisant l'hypothèse 6, on a

$$\|v_{\pi^*} - v_{\pi_{K,p}}\|_{2,\sigma} \leq \frac{2\gamma}{(1-\gamma)(1-\gamma^p)}\left[\sqrt{Cc^{\rho,\rho}_{\infty}(1)\left(C^{(1)}_{\pi^*}+\gamma^p C^{\sigma,\rho}_{\infty,p,p}\right)}\max_{0\leq i\leq K-1}\|\epsilon_{K-1-i}\|_{2,\mu_{\pi_{K-i,p}}} + \gamma^{K-1}R_{\max}\right]. \tag{6.18}$$

En combinant les équations (6.7) (page 91) et (6.18) on obtient l'inégalité (6.10).

□

6.3 Simulations

On a effectué un certain nombre d'expériences sur une classe de MDP qui sont les Garnet MDP (N, m, b, γ) (voir page 67), avec $\gamma = 0.99$. Les récompenses sont uniformes et aléatoires. La base de features considérée est celle de Fourier [19] engendrée par les fonctions cosinus telles que

$$\forall j \in \{1, ..., d\}, \forall i \in \{1, ..., N\}, \phi_j(i) = \cos\left(\frac{2j \times i}{N}\right).$$

Deux types d'expériences ont été menées : un premier type d'expériences qui rend compte de la variation de l'erreur globale en fonction du paramètre λ et pour une période $p \in \{1, 5, 10, 15\}$. Un deuxième type d'expériences qui rend compte de la variation de l'erreur globale en fonction de la période p pour une valeur fixe de λ (dans toutes ces expériences on a considéré le cas $\lambda = 0$, car c'est la meilleure au vue des expériences menées).

Lors de l'implémentation de l'algorithme LSNSPI(0) on a dû utiliser des politiques gourmandes stochastiques (on a mélangé la politique déterministe gourmande avec la politique uniforme π_u vérifiant $\pi_u(s,a) = \frac{1}{m}$ pour tout $s \in S$ et pour tout $a \in A$, on a pris $\pi = 0.85 * \pi_{gourmande} + 0.15 * \pi_u$). En effet la classe des MDP qu'on a considérée a un faible facteur de branchement b ne permettant pas de visiter tous les états $s \in S$. Si de plus on considère des politiques gourmandes déterministes certaines actions ne seront jamais choisies et par conséquent plusieurs états seront rarement visités. Uniformiser la politique gourmande permet de choisir toutes les actions dans tous les états avec une probabilité strictement positive.

On pourrait par ailleurs augmenter le facteur de branchement b mais on ne distinguera plus dans ce cas de grandes différences entre les périodes p. En effet, les coefficients de concentration introduits dans les définitions 17 et 18 peuvent être très grands lorsque la dynamique de transition

19. Si on considère à la place une base de features aléatoires on n'obtient pas le résultat escompté. On n'a pas encore bien compris l'influence de ce choix. On compte l'étudier à l'avenir.

est déterministe, ce qui correspond au cas $b = 1$. Augmenter la période p permet de contrôler ces termes, c'est ce qu'on peut voir dans la remarque 5 où la borne d'erreur fait intervenir la quantité $\sqrt{Cc_{\infty}^{\rho,\rho}(1)(C_{\pi^*}^{(1)} + \gamma^p C_{\infty,p,p}^{\sigma,\rho})}$. Si par contre la dynamique de transition est très aléatoire, les coefficients de concentration ont tendance à admettre de petites valeurs. On remarquera moins l'influence de p, qu'on a envie de mettre en valeur à travers les simulations effectuées .

6.3.1 Variation de l'erreur en fonction du paramètre λ

On a considéré un ensemble de Garnet $\text{MDP}(20, 2, 2, 0.99)$ et la base de Fourier telle qu'elle est définie plus haut avec $d = 5$. On a calculé pour chaque MDP l'erreur égale à $\|v^* - v_{\pi_k}\|_2$, pour chaque itération $k \in \{1, ..., 100\}$. Les courbes représentent la moyenne et la déviation standard de toutes ces erreurs pour différentes valeurs du paramètre λ en fonction du nombre d'itérations. Les quatre expériences illustrent la variation du paramètre λ pour différentes valeurs de la période $p \in \{1, 5, 10, 15\}$. On remarque que pour $p = 1$ ce sont les valeurs intermédiaires de λ qui garantissent une erreur minimale. Plus on augmente la période p plus ce sont les valeurs minimales de λ qui l'emportent. Pour p assez grand ($p = 15$), on s'aperçoit que c'est $\lambda^* = 0$ qui garantit la plus petite erreur.

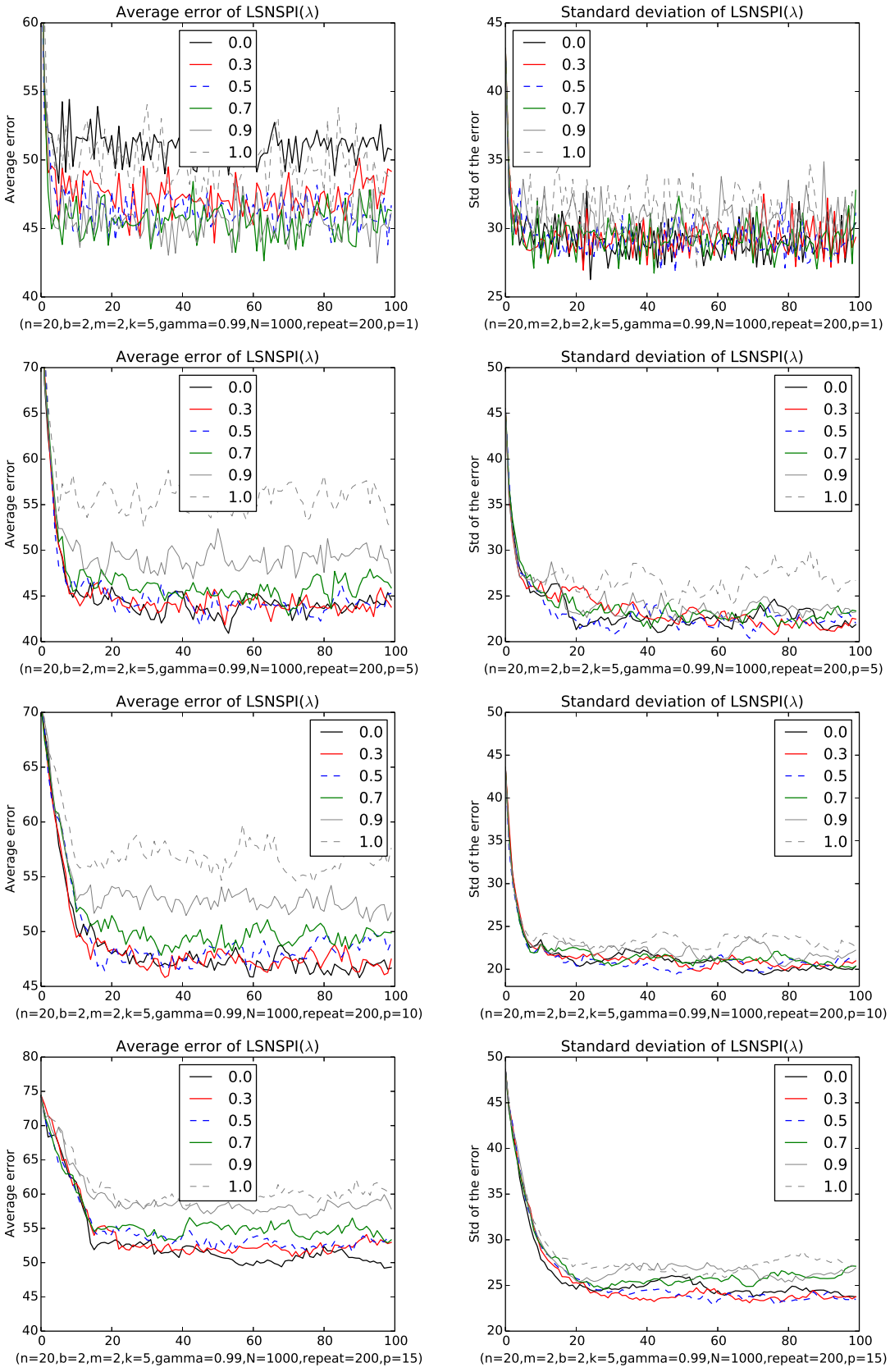

FIGURE 6.1 – De haut en bas : variation de l'erreur moyenne et de la déviation standard en fonction des valeurs du paramètre λ étant données des valeurs de la période $p = 1, 5, 10, 15$ (sur les figures n désigne le nombre des états et N le nombre d'échantillons).

On aurait pu observer cela en considérant la borne d'erreur introduite dans la proposition 8. En effet plus le paramètre p est grand plus le réel $\frac{n}{p}$ est petit. Le terme $\frac{n}{p} = l$ correspond au nombre d'échantillons effectif étant donnée une politique stationnaire $\pi_{k,p}$ (voir la remarque 3 page 94). Augmenter la période revient à diminuer l ce qui implique l'augmentation de l'erreur d'estimation. La valeur $\lambda = 0$ est celle qui minimise la borne sur l'erreur d'estimation. En outre pour des valeurs grandes de p, le terme $\frac{1-\lambda\gamma^p}{1-\gamma^p}$ tend vers 1 pour n'importe quelle valeur du paramètre λ.

6.3.2 Variation de l'erreur en fonction de la période p

Dans les différentes expériences qui suivent, on illustre la variation de l'erreur $\|v^* - v_{\pi_k}\|_2$ en fonction de la période $p \in \{1, 2, 3, 4\}$ pour une valeur fixée du paramètre λ. A la vue des expériences précédentes, on a choisi $\lambda = 0$. On a effectué nos expériences sur un ensemble de Garnet MDP (N, m, b, γ). L'erreur représentée correspond à la moyenne et à la déviation standard de toutes ces erreurs en fonction des itérations $k \in \{1, ..., 100\}$. La première figure correspond à une classe de Garnet MDP(N, m, b, γ) pour qui on fait varier le nombre des états $N \in \{20, 50, 100\}$. La deuxième figure correspond à celle pour qui on fait varier le nombre d'actions $m \in \{2, 5, 10\}$. Enfin la troisième figure représente celle pour qui on fait varier le facteur de branchement $b \in \{1, 2, 4\}$. Cela permet d'une part de mesurer l'influence de ces paramètres sur l'erreur globale et de mesurer d'autre part l'influence de la période p pour différentes valeurs de ces paramètres.

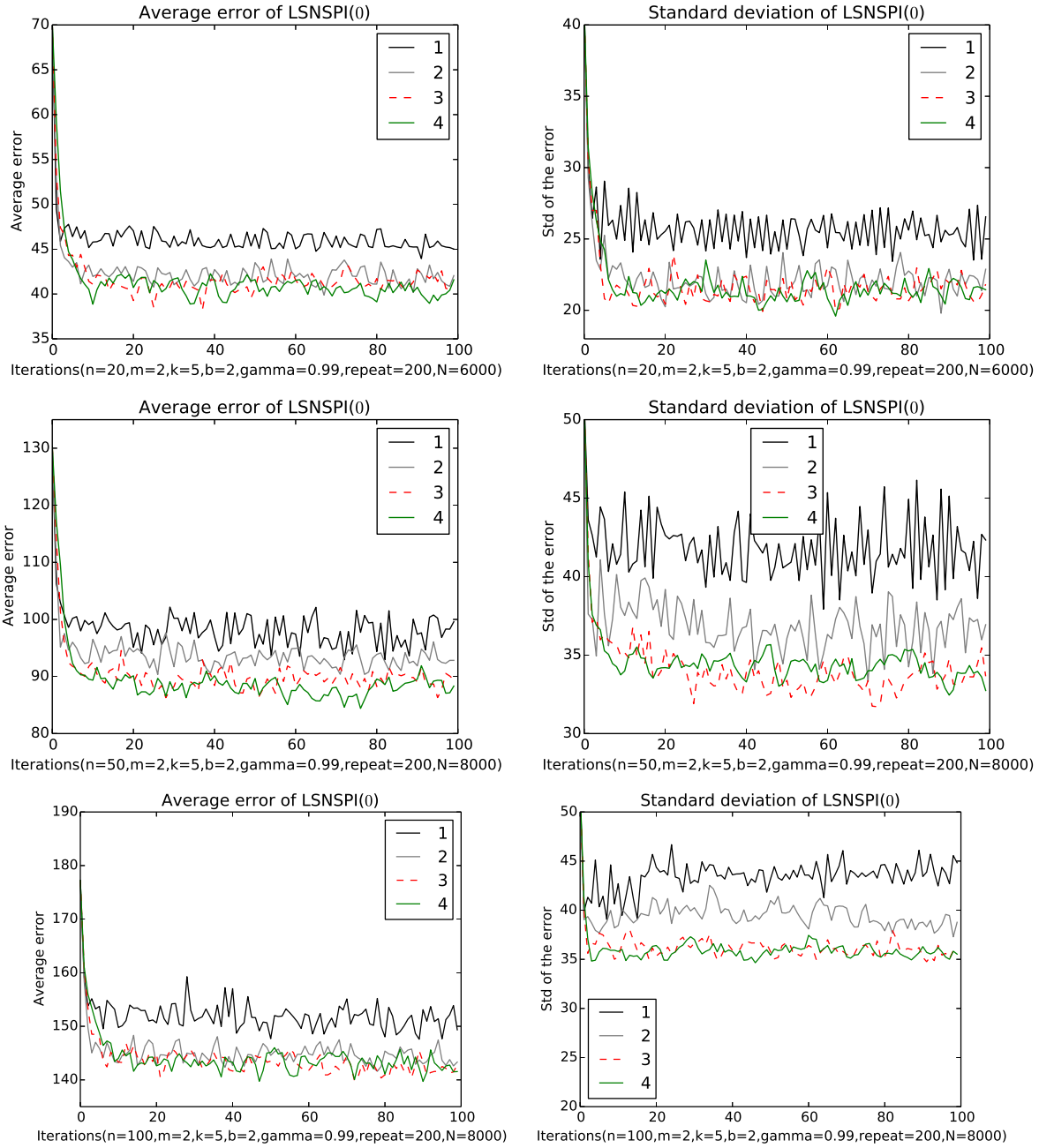

FIGURE 6.2 – De haut en bas : variation de l'erreur (l'erreur moyenne et la déviation standard) en fonction de la période p, pour un nombre d'états $N = 20, 50, 100$.

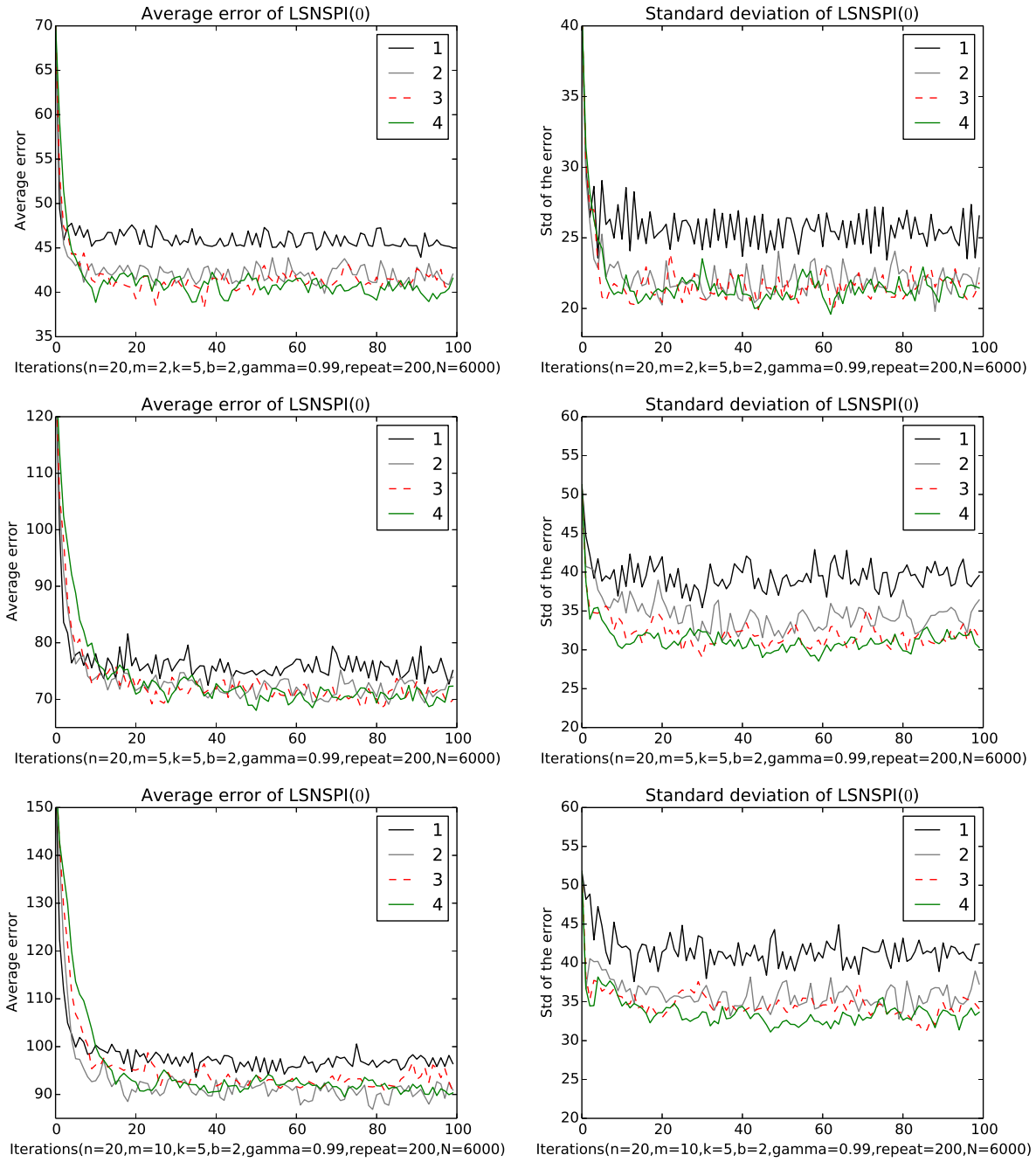

FIGURE 6.3 – Variation de l'erreur (l'erreur moyenne et la déviation standard) en fonction de la période p, pour un nombre d'actions $m = 2, 5, 10$.

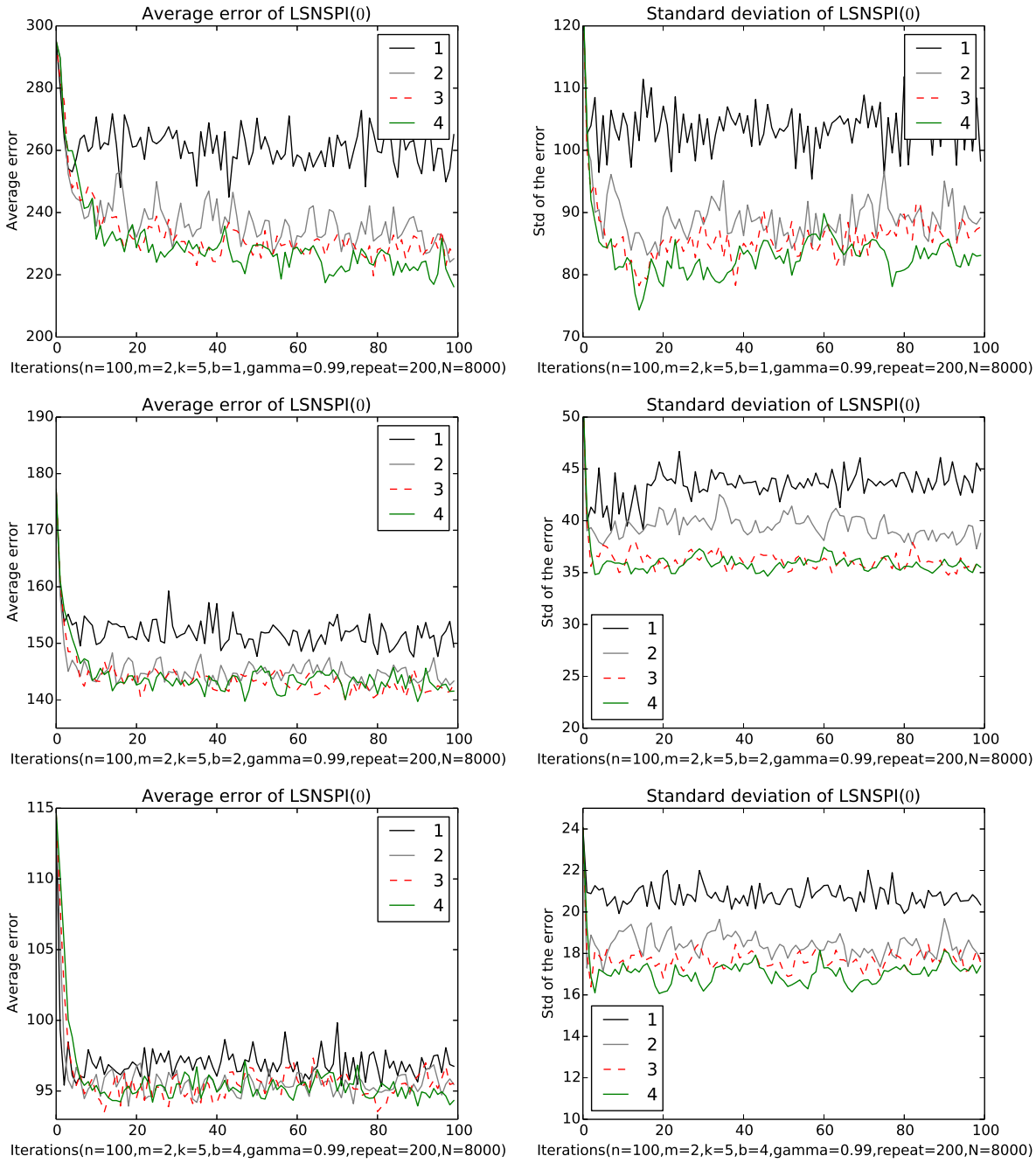

FIGURE 6.4 – Variation de l'erreur (l'erreur moyenne et la déviation standard) en fonction de la période p, pour un facteur de branchement $b = 1, 2, 4$.

On remarque que pour les différents types d'expériences menées, considérer des politiques non stationnaires diminue globalement l'erreur. En effet, outre le fait qu'augmenter la période p rend l'opérateur de Bellman plus contractant, ceci a également l'avantage de rendre la chaîne de Markov générée plus ergodique (à partir d'un état s donné on peut visiter un autre état avec une probabilité non nulle). Ceci explique le fait qu'on ait des erreurs qui sont plus petites pour $p > 1$. On s'est restreint au cours de nos expériences à des valeurs de p qui sont plus ou moins petites (la plus grande période choisie est égale à 4) car considérer de grandes périodes nécessite un nombre plus grand d'échantillons. Ce qui requiert à son tour un temps de calcul beaucoup plus lent.

Les paramètres N, m, b ont une influence sur l'erreur globale. En effet on peut s'apercevoir qu'augmenter le nombre des états N et des actions m fait augmenter l'erreur. Ce qui est logique puisque les problèmes deviennent plus durs à résoudre. Le nombre de politiques possibles est élevé et il est difficile d'identifier la politique optimale. D'un autre côté, augmenter le facteur de branchement b fait diminuer l'erreur globale, les dynamiques étant plus aléatoires. Les différentes constantes de concentration se valent dans ce cas comme on l'a expliqué auparavant.

6.4 Conclusion

On a présenté dans ce chapitre une analyse de l'algorithme LS(λ)NSPI. Cette analyse étend celle qu'on a présentée dans le chapitre précédent pour LSTD(λ). Elle rend compte de la propagation de l'erreur dûe à LSTD(λ) au fil des itérations quand on utilise cet algorithme d'évaluation de politique dans un schéma pour optimiser la politique (NSPI). On a tout d'abord calculé la borne de performance de LSTD(λ) dans le cas d'une politique non stationnaire et sous l'hypothèse que la chaîne de Markov considérée est exponentiellement β-mélangeante. Ensuite on a estimé cette borne au cas de LS(λ)NSPI en supposant qu'à chaque itération k, et pour chaque politique π_k la chaîne de Markov est exponentiellement β-mélangeante. La borne déduite utilise les mêmes hypothèses que celles employées dans [Lazaric et al., 2012] pour l'analyse de LSPI. Elle fait cependant intervenir des coefficients de concentration introduits dans [Scherrer, 2014] qui sont un peu plus fins. L'utilisation des politiques non stationnaires permet d'avoir une meilleure borne de performance que dans le cas stationnaire (correspondant à $p = 1$). C'est ce que reflètent à la fois l'analyse théorique et les expérimentations effectuées sur une classe de Garnet MDP pour qui on a fait varier les différents paramètres. On a utilisé lors de l'implémentation de LS(λ)NSPI la base formée par les fonctions cosinus. Outre la régularité des fonctions qui entre en jeu, il serait intéressant d'étudier les différents facteurs qui peuvent avoir un impact sur la qualité de l'algorithme.

7

Conclusion générale

Cette thèse s'est intéressée aux différentes méthodes de résolution des processus décisionnels de Markov de grande taille. On a commencé par introduire dans le premier chapitre le formalisme des processus décisionnels de Markov (MDP) ainsi que les principaux algorithmes de résolution issus de la programmation dynamique. On a également présenté les différentes méthodes de résolution issues de la planification. On s'est plus spécifiquement intéressé aux abstractions d'états. Ainsi le deuxième chapitre a été conscacré à l'étude de deux types d'abstractions : l'abstraction moyenne et l'abstraction BMDP. L'abstraction moyenne est une approche de résolution qui semble assez naturelle et simple. En effet, pour estimer les fonctions de valeur il suffit de résoudre l'équation optimale de Bellman avec les nouveaux paramètres (transitions+récompenses). L'approche des BMDP est un peu plus délicate puisqu'on a non plus des valeurs exactes mais de bornes sur les fonction de valeur. On a eu recours pour les estimer à l'algorithme proposé par [Givan et al., 2000]. On a dans ce cadre analysé l'algorithme en introduisant les opérateurs T_o^- et T_o^+ relatifs respectivement aux bornes inférieure et supérieure. Cette analyse nous a permis par la suite de comparer les erreurs d'approximation $E_{\alpha'}$ et E_α pour deux abstractions α et α' telles que $\alpha \preceq \alpha'$, i.e. α' plus fine que α. Ce qui nous permettait de voir si on avait monotonie de l'erreur par rapport à l'abstraction. On a montré les résultats suivants.

- Dans le cas de l'abstraction moyenne, il peut y avoir absence de monotonie de l'approximation par rapport à l'abstraction même dans un cadre déterministe. En d'autres termes pour deux abstractions α et α' telles que $\alpha \preceq \alpha'$, l'abstraction α' n'induit pas toujours une moindre erreur. Pour illuster cela on exhibé un exemple de MDP déterminste où on voit ce phénomène.
- Dans le cas l'abstraction BDMP, on a monotonie de l'approximation par rapport à l'abstraction dans un cadre déterministe. Considérons une abstraction α telles que v_α^- et v_α^+ sont respectivement la borne inférieure et la borne supérieure. Pour α' un raffinement de α, on a $v_{\alpha'}^-$ qui est plus grande que v_α^- et $v_{\alpha'}^+$ plus petite que v_α^+. Cette propriété n'est plus en général vraie dans le cas stochastique. On a pris comme exemple le modèle d'une bisimulation où les états se comportent de la même façon au sein des blocs. On avait l'erreur qui augmentait strictement dans tous les état-blocs suite à un raffinement. Une abstraction de type bisimulation peut s'avérer alors meilleure que n'importe quelle autre abstraction plus fine.

Le reste de la thèse s'est porté sur les méthodes de projection sur un espace linéaire. Le chapitre 4 constitue un état de l'art les introduisant. Ces méthodes incluent celle des moindres carrés dont celle du point fixe projeté. On y distingue deux types de méthodes : celle du résidu de Bellman et celle des différences temporelles. On a décrit ces deux méthodes et introduit les algorithmes

associés à chacune d'elles. On a également dressé une analyse comparative de ces deux approches basée sur les travaux de [Scherrer, 2010]. L'algorithme *Least square temporal difference algorithm* LSTD(λ) repose sur la méthode des différences temporelles. On a proposé dans le chapitre 5 une analyse de sa vitesse de convergence en fonction du nombre d'échantillons et du paramètre λ. Cette analyse suppose certaines hypothèses entre autres celle du β-mélange. Cette hypothèse a constitué l'argument clé nous permettant de dériver la borne d'erreur. Ainsi, on a réussi à dériver une inégalité de concentration vectorielle pour les estimations basées sur des traces infiniment longues (lemme 4, page 70). En tronquant la trace d'éligibilité, on s'est servi de cette inégalité pour déduire une borne de convergence des matrices $\hat{A}$ et $\hat{b}$ vers A et b. On a montré les résultats suivants.

— La vitesse de convergence est une fonction de $\tilde{O}(\frac{1}{\sqrt{n}})$, comme c'est également le cas dans les travaux de [Lazaric et al., 2012] pour $\lambda = 0$.
— La borne d'erreur dépend du paramètre λ. Il existe un paramètre $\lambda^* \in [0, 1]$ optimal qui minimise la borne d'erreur. On peut expliciter ce paramètre. Il dépend du nombre d'échantillons n des paramètres du processus β-mélangeant et de l'erreur de projection $\|v - \Pi v\|_\mu$, où μ est la mesure stationnaire de la chaîne considérée.

Le chapitre 6 considère le cas où on utilise LSTD(λ) dans un schéma type itérations sur les politiques non stationnaires de période fixe p, NSPI(p) [Scherrer, 2014]. On obtient alors l'algorithme LSNSPI(λ) qu'on décrit page 92 de ce manuscrit. En nous basant sur l'analyse faite pour LSTD(λ), on a réussi à déduire une borne de performance de cet algorithme. On a montré les points qui suivent.

— L'erreur globale vérifie lorsque le nombre d'échantillons tend vers l'infini :

$$\lim_{k\to\infty} \|v^* - v_{\pi_k}\|_\sigma \leq \lim_{k\to\infty} \frac{2\gamma\sqrt{CC^{\sigma,\rho}_{\infty,p,1}}(1-\lambda\gamma^p)}{(1-\gamma)(1-\gamma^p)^2} \max_{0\leq k\leq K-1} \|v - \Pi_{\mu_{\pi_{k,p}}} v\|_{\mu_{\pi_{k,p}}}.$$

— Cette borne est légèrement meilleure que celle proposée par [Lazaric et al., 2012] dans le cas de l'algorithme LSPI qui correspond à l'algorithme LSNSPI(λ), pour $p = 1$ et $\lambda = 0$.
— Augmenter la période p améliore la qualité de la borne si on considère le nombre d'échantillons adéquat, i.e. le quotient $\frac{n}{p}$ doit rester relativement grand.
— Il existe un paramètre optimal λ^* qui minimise la borne. Il dépend du nombre d'échantillons, des paramètres des processus β-mélangeants et de la période p.

Les expérimentations qu'on a effectuées confirment les deux derniers points cités ci-dessus. En effet, on a constaté la diminution de l'erreur lorsque p augmente. Aussi on a le paramètre λ^* qui tend vers 0 lorsque la période devient très grande. Les expérimentations illustrent également l'influence de la dynamique de transitions sur l'erreur, plus la chaîne est ergodique plus l'erreur globale est petite. Ce qu'on peut aussi deviner à partir des coefficients de concentration introduits.

Perspectives

L'approche qu'on a considérée basée sur les abstractions pour résoudre un MDP s'avère un peu trop *exigeante*. Elle nous a permis néanmoins d'identifier certains types d'abstractions qui sont les bisimulations qu'on pourrait exploiter comme l'ont fait [Givan et al., 2003] ou [Ferns et al., 2012]. Il serait notamment intéressant d'envisager certains points dans l'avenir.

— Considérer des raffinements quelconques et non plus des raffinements directs comme on l'a fait dans notre cas. En d'autres termes il serait intéressant de voir si à partir d'une abstraction donnée on peut toujours trouver un raffinement quelconque (en dehors de celui qui mène au modèle 0) tel que l'erreur d'approximation diminue.

— Etudier le problème de proposer une méthode de raffinement qui soit monotone, c'est-à-dire qui fournisse de meilleures approximations à chaque raffinement.
— Vérifier si les structures de bisimulation peuvent expliquer de manière générale les pathologies identifiées dans certaines abstractions.

Les algorithmes de résolution qu'on a analysés renvoient une solution approximative–sous certaines hypothèses sur le processus décisionnel de Markov considéré. On pourrait envisager d'utiliser ces mêmes hypothèses pour étudier les questions qui suivent.

— Estimer une borne sur la vitesse de convergence de l'algorithme off policy LSTD(λ) en adaptant et généralisant les arguments avancés par [Yu, 2010].
— (Re)déduire la borne de convergence de LSBR obtenue par [Maillard et al., 2010], en utilisant l'inégalité de concentration concernant les variables iid ce qui est a priori plus simple.
— Déduire une borne de performance lorsqu'on utilise l'algorithme off policy LSTD(λ) dans un schéma type approximation sur les politiques (non) stationnaires.

On peut notamment considérer une perspective qui se distingue de celles citées plus haut. On a déjà mentionné lors de l'implémentation de l'algorithme LS(λ)NSPI qu'on n'obtient pas le même résultat lorsqu'on utilise une base de features autre que celle de Fourier. Il serait intéressant d'étudier l'influence de ces fonctions sur l'erreur globale renvoyée.

A

Preuve du lemme 1

Soit M un MDP appartenant à $\mathcal{M}$, on a v_M^π solution de l'équation de Bellman $v_M^\pi = T^\pi v_M^\pi$. Considérons $o = q_1, q_2, ..., q_N$ l'ordre des états tel que

$$v_M^\pi(q_1) \leq v_M^\pi(q_2) \leq ... \leq v_M^\pi(q_{|S|}).$$

Notons M_o le MDP qui correspond à associer au plus petite valeur la plus grande des probabilités. On a

$$\sum_{i=1}^{N} p_{M_o}(s, \pi(s), q_i) v_M^\pi(q_i) - \sum_{i=1}^{N} p_M(s, \pi(s), q_i) v_M^\pi(q_i) = \sum_{i=1}^{N} (p_{M_o}(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)) v_M^\pi(q_i).$$

Soit r l'indice défini dans l'équation (3.1) (page 33), on a

$$v_M^\pi(q_1) \leq v_M^\pi(q_2) \leq ... \leq v_M^\pi(q_r) \leq v_M^\pi(q_{|S|}).$$

Il s'en suit que

$$\begin{aligned}
&\sum_{i=1}^{N} \left(p_{M_o}(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_i) = \sum_{i=1}^{r-1} \left(\max p_M(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_i) \\
&+ \left(p_{M_o}(s, \pi(s), q_r) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_r) + \sum_{i=r+1}^{N} \left(\min p_M(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_i) \\
&\leq \sum_{i=1}^{r-1} \left(\max p_M(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_r) \\
&+ \left(p_{M_o}(s, \pi(s), q_r) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_r) + \sum_{i=r+1}^{N} \left(\min p_M(s, \pi(s), q_i) - p_M(s, \pi(s), q_i)\right) v_M^\pi(q_r) = 0.
\end{aligned}$$

On a alors pour tout s

$$T_{M_o}^\pi v_M^\pi(s) = r^\pi(s) + \sum_{i=1}^{N} (p_{M_o}(s, \pi(s), q_i) v_M^\pi(q_i) \leq r^\pi(s) + \sum_{i=1}^{N} (p_M(s, \pi(s), q_i) v_M^\pi(q_i) = v_M^\pi(s).$$

En appliquant l'opérateur $T_{M_o}^\pi$ infiniment de fois on obtient $v_{M_o}^\pi \leq v_M^\pi$ ($T_{M_o}^\pi$ est monotone). Si on considère maintenant l'ordre qui associe les plus grandes probabilités aux plus grandes valeurs on obtient $v_M^\pi \leq v_{M_o}^\pi$.

B

Compléments sur les chaînes de Markov

Définition 20. *Soit M une matrice aléatoire de taille $K \times L$, on note $H = \mathbb{E}[M]$ la matrice définie dans $\mathbb{R}^{K\times L}$ telle que*

$$\forall i, j \in \{1, ..., K\} \times \{1, ..., L\},\ H_{ij} = \mathbb{E}[M_{ij}].$$

Définition 21. *On dit que deux matrices aléatoires M et N sont indépendantes si les collections de variables aléatoires constituant chaque matrice sont indépendantes.*

Définition 22. *[Meyn and Tweedie, 1993] Soit $(X_n)_{n\geq 1}$ une chaîne de Markov définie sur un espace euclidien $\mathcal{X}$. On définit pour tout A ensemble mesurable de $\mathcal{B}(\mathcal{X})$ (tribu générée par l'espace $\mathcal{X}$) le temps d'entrée de la chaîne en A*

$$T_A = \inf\{n \geq 1, X_n \in A\}$$

et on définit

$$L(x, A) = \mathbb{P}_x(T_A < \infty),$$

i.e., la probabilité en partant de x que la chaîne soit dans l'ensemble A en un temps fini. Une chaîne de Markov est dite ψ irréductible s'il existe une mesure $\psi \in \mathcal{B}(\mathcal{X})$, telle que si $\psi(A) > 0$ alors $L(x, A) > 0$ pour $x \in \mathcal{X}$.

Définition 23. *[Meyn and Tweedie, 1993] On définit n_A le nombre de visites de la chaîne à l'ensemble A après le temps 0. On a*

$$n_A = \sum_{i=1}^{\infty} 1_{X_n \in A}.$$

Un ensemble A est dit Harris récurrent si

$$Q(x, A) = \mathbb{P}_x(n_A = \infty) = 1,\ \textit{pour } x \in A.$$

On dit qu'une chaîne de Markov $(X_n)_{n\geq 1}$ est Harris récurrente si elle est ψ irréductible et chaque ensemble de $\mathcal{B}^+(\mathcal{X})$ est Harris récurrent. On a

$$\mathcal{B}^+(\mathcal{X}) = \{A \in \mathcal{B}(\mathcal{X}), \psi(A) > 0\}.$$

C'est l'ensemble des mesures positives.

Définition 24. *Soit $(X_n)_{n\geq 1}$ une chaîne de Markov de noyau de transition p. Soit $q = pgcd\{n > 0 : p^n(i,i) > 0\}$ (pgcd=le plus grand commun dénominateur), q est appelé la période de i. Si $q = 1$ alors la chaîne est dite apériodique.*

Définition 25. *On dit qu'une chaîne de Markov est irréductible et apériodique s'il existe un entier k tel que pour tout couple d'états $(s, s') \in \mathcal{X} \times \mathcal{X}$ on a $p^k(s, s') > 0$.*

Théorème 13. *[Nedic and Bertsekas, 2002] Supposons les hypothèses suivantes remplies :*

1. *$\mathcal{M}$ une chaîne de Markov ergodique admettant une mesure invariante ν strictement positive*
2. *les fonctions de features $(\phi_j)_{1\leq j\leq d}$ sont linéairement indépendantes*

alors pour tout $\lambda \in [0, 1]$, le vecteur $\hat{\theta} = \hat{A}^{-1}\hat{b}$ converge presque sûrement vers $\theta = A^{-1}b$.

Démonstration. On prouve ce théorème en montrant que $\hat{A} \to A$ et $\hat{b} \to b$. Ceci grâce à la proposition suivante.

Proposition 9. *Soit $\{X_k\}$ une séquence de variables aléatoires d'espérance nulle et de variance uniformément bornée, soit Z_t tel que*

$$Z_t = \frac{1}{t+1}\sum_{k=0}^{t+1} X_k, \forall t.$$

S'il existe des réels positifs $\overline{C}$ et q tels que

$$|\mathbb{E}[X_t Z_t]| \leq \frac{\overline{C}}{(t+1)^q}, \forall t,$$

alors Z_t converge vers 0 avec probabilité 1.

□

Définition 26. *On dit qu'une famille $X = (X_\phi)_{\phi\in E}$ de variables aléatoires réelles indexées par E est un processus linéaire si pour tout élément ω de Ω, l'application de E dans $\mathbb{R}$ définie par $\phi \to X_\phi(\omega)$ est une application linéaire.*

Définition 27. *La distance en variation totale entre deux mesures μ et ν definies sur $\sigma(\Omega)$ est définie par*

$$d_{TV} = \sup_{A\in\sigma(\Omega)} |\mu(A) - \nu(A)|.$$

Définition 28. *Etant donnée une chaîne de Markov de noyau de transition ergodique p sur un ensemble fini ou dénombrable de loi invariante ν. On dit qu'elle est uniformément ergodique s'il existe $r < 1$ et $A \geq 0$ tels que pour tout $n \geq 0$,*

$$\sup_{x\in\mathcal{X}} d_{TV}(p^n(x, .), \nu) \leq Ar^n;$$

où d_{TV} est la distance en variation totale.

Définition 29. *Un processus X_t est un ARMA (p, q) processus si*

— *il est stationnaire*

— *s'il (ou $X_t - \mathbb{E}[X_t]$) satisfait la différence linéaire qui suit*

$$X_t - \sum_{i=1}^{p} \phi_i X_{t-i} = w_t + \sum_{i=1}^{q} \theta_i w_{t-i}$$

où $w_t \sim WN(0, \sigma^2)$ (WN désigne bruit blanc de moyenne nulle et de variance σ) et les paramètres ϕ_i et θ_i sont des constantes dans $\mathbb{R}$.

Définition 30. *[Hunter, 2002] On dit que des variables $X_1, X_2, ...$ sont m-dépendantes pour un entier m donné si le vecteur $(X_1, X_2, ..., X_i)$ est indépendant du vecteur $(X_{i+j}, X_{i+1+j}, ...)$ pour $j > m$.*

Définition 31. *Soit $(\Omega, \mathcal{F}, (\mathcal{F}_n)_n, \mathbb{P})$ un espace de probabilité filtré. Un processus réel $(X_n)_n$ est une martingale s'il est adapté à $(\mathcal{F}_n)_n$ i.e. pour tout n, X_n est $\mathcal{F}_n$-mesurable et pour tout n, X_n est intégrable vérifiant*

$$\mathbb{E}[X_{n+1}|\mathcal{F}_n] = X_n \ p.s.$$

Lemme 8. *(Inégalité de Hölder) Soient q, q' des nombres de $[1, \infty)$ vérifiant $1 = \frac{1}{q} + \frac{1}{q'}$ (avec la convention $\frac{1}{\infty} = 0$). Soient f et g deux fonctions mesurables de $\mathcal{X}$ dans $\mathbb{R}$ et $L_p(\mathcal{X})$ l'espace des fonctions L_p dans $\mathcal{X}$. Si $f \in L^q(\mathcal{X})$ et $g \in L^{q'}(\mathcal{X})$, alors on a $fg \in L^1(\mathcal{X})$ et*

$$\|fg\|_1 \leq \|f\|_{q'} \|f\|_q.$$

Définition 32. *En dimension finie le rayon spectral ρ d'une matrice A (diagonalisable) correspond au module de sa plus grande valeur propre $\rho(A) = \max_i |\lambda_i|$.*

Proposition 10. *Supposons qu'on a*
— *la mesure μ strictement positive pour tout état x dans $\mathcal{X}$*
— *les features $(\phi_j)_{1 \leq j \leq d}$ linéairement indépendantes*
alors la matrice $\Phi^t D_\mu \Phi$ est inversible, où Φ est la matrice des features et D_μ est la matrice diagonale composée des éléments de μ.

Démonstration. Soit u un vecteur de $\mathbb{R}^d$, on a

$$u^t \Phi^t D_\mu \Phi u = \sum_{j=1}^{|\mathcal{X}|} \mu_j \left(\sum_{i=1}^{d} \phi_i(x_j) u_i \right)^2.$$

Par conséquent $u^t \Phi^t D_\mu \Phi u = 0$ implique que $\mu_j (\sum_{i=1}^d \phi_i(x_j) u_i)^2 = 0$ pour tout j puisqu'il s'agit de quantités positives. La première condition de la proposition implique que $\sum_{i=1}^d \phi_i(x_j) u_i = 0$. Puisque les features sont linéairement indépendantes il s'en suit que $u_i = 0$, pour tout i. D'où la matrice $\Phi^t D_\mu \Phi$ est inversible. □

C

Compléments sur les coefficients de concentration

Définition 33. *Soient ν une mesure σ-finie positive sur $(\mathcal{X}, \mathcal{A})$ et μ une mesure σ-finie positive sur $(\mathcal{X}, \mathcal{A})$ telles que ν est absolument continue par rapport à ρ, dans le sens où pour tout ensemble $A \in \mathcal{A}$ on a $\rho(A) = 0 \Rightarrow \nu(A) = 0$ alors ν possède une dérivée de Radon Nikodym f par rapport à ρ. La fonction f est positive et mesurable vérifiant*

$$\forall A \in \mathcal{A},\ \nu(A) = \int_A f d\rho.$$

Proposition 11. *Soient σ et ρ deux mesures de probabilités. Soit Γ l'opérateur défini dans la page 96 (définition 19) alors*

$$\sigma\Gamma^t|z| = \|\Gamma^t|z|\|_{1,\sigma} \leq \gamma^t c_q(t)\|z\|_{q',\rho} = \gamma^t c_q(t)(\rho|z|^{q'})^{\frac{1}{q'}},$$

où c_q est le coefficient de concentration introduit dans la page 92 (définition 17).

Démonstration. On a

$$\begin{aligned}\sigma\Gamma^t|z| &= \sum_{i=1}^N \sigma(i) \sum_{j=1}^N \Gamma^t_{ij}|z|(j) \\ &= \sum_{j=1}^N \frac{\sum_{i=1}^N \sigma(i)\Gamma^t_{ij}}{\rho(j)}|z|(j)\rho(j).\end{aligned}$$

En appliquant l'inégalité de Hölder, on obtient

$$\begin{aligned}\sigma\Gamma^t|z| &\leq \left(\sum_{j=1}^N \left(\frac{\sum_{i=1}^N \sigma(i)\Gamma^t_{ij}}{\rho(j)}\right)^q \rho(j)\right)^{\frac{1}{q}} \left(\sum_{j=1}^N |z|^{q'}(j)\rho(j)\right)^{\frac{1}{q'}} \\ &= \gamma^t c_q(t)\|z\|_{q',\rho}.\end{aligned}$$

□

Théorème 14. *Soient $(\lambda_1, ..., \lambda_n)$, n-uplets de réels positifs vérifiant*

$$\sum_{i=1}^n \lambda_i = 1.$$

Considérons f une fonction convexe et $(x_1, ...x_n)$ n-uplets appartenant au domaine de défintion de f. On a

$$f\left(\sum_{i=1}^{n}\lambda_i x_i\right) \leq \sum_{i=1}^{n}\lambda_i f(x_i).$$

Théorème 15. *Soit A un opérateur positif borné tel que $A.1 = 1$, où 1 est le vecteur $1 = (1, ..., 1)^t$ et g une fonction convexe, on a alors pour tout tel que $Q \geq 0$*

$$g(AQ) \leq A(g \circ Q).$$

Proposition 12. *Les coefficients de concentration $C^{\sigma,\rho}_{\infty,p,u}$ et $C^{\sigma,\rho}_{\infty,1,u}$ sont équivalents dans le sens où si l'un est fini (infini) alors l'autre l'est également.*

Démonstration. Cette preuve est analogue à celle qui se trouve dans [Scherrer, 2014]. On a d'un côté

$$\begin{aligned} C^{\sigma,\rho}_{\infty,p,u} &= (1-\gamma)(1-\gamma^p)\sum_{j=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{j+tp}c(j+tp+u) \\ &\leq (1-\gamma)(1-\gamma^p)\sum_{j=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{j+t}c(j+t+u) \\ &\leq \frac{1-\gamma^p}{1-\gamma}C^{\sigma,\rho}_{\infty,1,u}. \end{aligned}$$

Et d'un autre côté

$$C^{\sigma,\rho}_{\infty,p,u} = (1-\gamma)(1-\gamma^p)\sum_{j=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{j+tp}c(j+tp+u).$$

En posant $v = j + tp$, on obtient

$$\begin{aligned} C^{\sigma,\rho}_{\infty,p,u} &= (1-\gamma)(1-\gamma^p)\sum_{t=0}^{\infty}\sum_{v\geq tp}^{\infty}\gamma^v c(v+u) \\ &= (1-\gamma)(1-\gamma^p)\sum_{v=0}^{\infty}\left(\left\lfloor\frac{v}{p}\right\rfloor+1\right)\gamma^v c(v+u) \\ &\geq (1-\gamma)(1-\gamma^p)\sum_{v=0}^{\infty}\max\left(\frac{v}{p},1\right)\gamma^v c(v+u). \end{aligned}$$

On a pour tout $v \geq p$, $\frac{v}{p} \geq \frac{v+1}{p+1}$

$$\begin{aligned} C^{\sigma,\rho}_{\infty,p,u} &\geq (1-\gamma)(1-\gamma^p)\left[\sum_{v=0}^{p-1}\gamma^v c(v+u) + \sum_{v=p}^{\infty}\frac{v+1}{p+1}\gamma^v c(v+u)\right] \\ &= (1-\gamma)(1-\gamma^p)\left[\sum_{v=0}^{p-1}\gamma^v c(v+u) + \left(\sum_{v=0}^{\infty}\frac{v+1}{p+1}\gamma^v c(v+u) - \sum_{v=0}^{p-1}\frac{v+1}{p+1}\gamma^v c(v+u)\right)\right]. \end{aligned}$$

Puisque $v + 1 \leq p + 1$ pour $0 \leq v \leq p - 1$, il s'en suit que

$$C^{\sigma,\rho}_{\infty,p,u} \geq \frac{(1-\gamma^p)C^{\sigma,\rho}_{\infty,1,u}}{(1-\gamma)(p+1)}.$$

□

Bibliographie

[Archibald et al., 1995] Archibald, T., McKinnon, K., and Thomas, L. (1995). On the Generation of Markov Decision Processes. *Journal of the Operational Research Society*, 46 :354–361.

[Azuma, 1967] Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19 :357–367.

[Barto et al., 1995] Barto, A. G., Bradtke, S. J., and Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1–2) :81 – 138.

[Bellman, 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition.

[Bernstein, 1927] Bernstein, S. (1927). Theory of probability. *Collected works*, 4 :570–574.

[Bertsekas and Tsitsiklis, 1996] Bertsekas, D. P. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.

[Bertsekas and Yu, 2009] Bertsekas, D. P. and Yu, H. (2009). Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227(1) :27 – 50.

[Bosq, 1975] Bosq, D. (1975). Inégalité de Bernstein pour les processus stationnaires et mélangeants. *CRAS Comptes Rendus de L'académie des sciences*, 281 :1095–1098.

[Boutilier et al., 1996] Boutilier, C., Dean, T., and Hanks, S. (1996). Planning under uncertainty : Structural assumptions and computational leverage. In *Proceedings of the Second European Workshop on Planning*, pages 157–171. IOS Press.

[Boyan, 2002] Boyan, J. A. (2002). Technical update : Least-squares temporal difference learning. *Machine Learning*, 49(2–3) :233–246.

[Bradley, 2005] Bradley, R. (2005). Basic properties of strong mixing conditions. a survey and some open questions. *Probability Survey*.

[Bradtke and Barto, 1996] Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22 :33–57.

[Burch et al., 1994] Burch, J. R., Clarke, E. M., Long, D. E., McMillan, K. L., and Dill, D. L. (1994). Symbolic model checking for sequential circuit verification. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 13(4) :401–424.

[Chafai, 2012] Chafai, D. (2012). Quelques mots sur l'inégalité d'Azuma-Hoeffding. Notes d'exposé du mardi 20 mars 2012 au groupe de travail Analyse, Statistique, et Probabilités (ASPro).

[Chanda, 1974] Chanda, K. (1974). Strong mixing properties of linear stochastic processes. *journal of Applied probabilities*, 11(2) :401–408.

[Dietterich, 1999] Dietterich, T. G. (1999). Hierarchical reinforcement learning with the MAXQ value function decomposition. *CoRR*, cs.LG/9905014.

[Dräger et al., 2009] Dräger, K., Finkbeiner, B., and Podelski, A. (2009). Directed model checking with distance-preserving abstractions. *STTT*, 11(1) :27–37.

[Ferns et al., 2012] Ferns, N., Panangaden, P., and Precup, D. (2012). Metrics for finite Markov Decision Processes. *CoRR*, abs/1207.4114.

[Givan et al., 2003] Givan, R., Dean, T., and Greig, M. (2003). Equivalence notions and model minimization in Markov Decision Processes. *Artificial Intelligence*, 147(1–2) :163 – 223.

[Givan et al., 1997] Givan, R., Leach, S. M., and Dean, T. (1997). Model reduction techniques for computing approximately optimal solutions for Markov Decision Processes. *Artificial Intelligence*, 122(1-2) :1–8.

[Givan et al., 2000] Givan, R., Leach, S. M., and Dean, T. (2000). Bounded-parameter Markov Decision Processes. *Artificial Intelligence*, 122(1-2) :71–109.

[Hartmanis and Stearns, 1966] Hartmanis, J. and Stearns, R. (1966). *Algebraic structure theory of sequential machines*. Prentice Hall, Engelwoodcliffs, N.J.

[Hoeffding, 1963] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58 :13–30.

[Howard, 1960] Howard, R. A. (1960). Dynamic programming and Markov processes.

[Hunter, 2002] Hunter, D. (2002). http ://sites.stat.psu.edu/dhunter/asymp/fall2002/lectures /ln05.pdf.

[Kattenbelt et al., 2010] Kattenbelt, M., Kwiatkowska, M., Norman, G., and Parker, D. (2010). A game-based abstraction-refinement framework for Markov Decision Processes. *Formal Methods in System Design*, 36(3) :246–280.

[Kevin et al., 2009] Kevin, W., David, S., Michael, B., and Duane, S. (2009). Abstraction pathologies in extensive games. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, AAMAS '09, pages 781–788.

[Korf, 1997] Korf, R. E. (1997). Finding optimal solutions to Rubik's Cube using pattern databases. In Kuipers, B. J. and Webber, B., editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 700–705, Portland, OR. MIT Press.

[Lagoudakis and Parr, 2003] Lagoudakis, M. and Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, 4 :1107–1149.

[Larsen and Skou, 1991] Larsen, K. G. and Skou, A. (1991). Bisimulation through probabilistic testing. *Information and Computation*, 94(1) :1 – 28.

[Lazaric et al., 2012] Lazaric, A., Ghavamzadeh, M., and Munos, R. (2012). Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13 :3041–3074.

[Lee and Yannakakis, 1992] Lee, D. and Yannakakis, M. (1992). Online minimization of transition systems (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 264–274, New York, NY, USA. ACM.

[Li et al., 2006] Li, L., Walsh, T. J., and Littman, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, pages 531–539. ISAIM.

[Littman et al., 1995] Littman, M. L., Dean, T. L., and Kaelbling, L. P. (1995). On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh International Conference on Uncertainty in Artificial Intelligence*.

[Maillard et al., 2010] Maillard, O. A., Munos, R., Lazaric, A., and Ghavamzadeh, M. (2010). Finite sample analysis of Bellman residual minimization. In Sugiyama, M. and Yang, Q., editors, *Asian Conference on Machine Learpning. JMLR : Workshop and Conference Proceedings*, volume 13, pages 309–324.

[Mausam and Kolobov, 2012] Mausam and Kolobov, A. (2012). *Planning with Markov Decision Processes*. Morgan and Claypool Publishers.

[Meyn and Tweedie, 1993] Meyn, S. P. and Tweedie, R. L. (1993). *Markov chains and stochastic stability*. Springer-Verlag, London.

[Milner, 1990] Milner, R. (1990). Operational and algebraic semantics of concurrent processes. In *Handbook of Theoretical Computer Science, Volume B : Formal Models and Sematics (B)*, pages 1201–1242. Elsevier Science Publishers B.V. (North-Holland).

[Modha and Masry, 1996] Modha, D. S. and Masry, E. (1996). Minimum complexity regression estimation with weakly dependent observations. *IEEE Transactions on Information Theory*, 42(6) :2133–2145.

[Munos, 2003] Munos, R. (2003). Error bounds for approximate policy iteration.

[Munos, 2007] Munos, R. (2007). Analyse en norme lp de l'algorithme d'itérations sur les valeurs avec approximations. *Revue d'Intelligence Artificielle*, 21(1) :53–74.

[Nedic and Bertsekas, 2002] Nedic, A. and Bertsekas, D. P. (2002). Least squares policy evaluation algorithms with linear function approximation. *Theory and Applications*, 13 :79–110.

[Ortner, 2011] Ortner, R. (2011). Adaptive aggregation for reinforcement learning in average reward Markov Decision Processes. *Annals of Operationnal Research*.

[Parr and Russell, 1997] Parr, R. and Russell, S. (1997). Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems 10*, pages 1043–1049. MIT Press.

[PDMIA, 2008] PDMIA (2008). *Processus décisionnels de Markov en intelligence artificielle*, volume 1 - principes généraux et applications of *IC2 - informatique et systèmes d'information*. Lavoisier - Hermes Science Publications.

[Puterman, 1994] Puterman, M. L. (1994). *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

[Rosenblatt, 1956] Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. In Smith, P., editor, *National Academy of Sciences of the United States of America*, volume 42, pages 43–47. National Academy of Sciences.

[Sabnani et al., 1989] Sabnani, K., Aleta, M. L., and Ümit Uyar, M. (1989). An algorithmic procedure for checking safety properties of protocols. *IEEE Transactions on Communications*, 37(9) :940–948.

[Sandholm and Singh, 2012] Sandholm, T. and Singh, S. (2012). Lossy stochastic game abstraction with bounds. In *ACM Conference on Electronic Commerce*, EC '12, pages 880–897. ACM.

[Scherrer, 2010] Scherrer, B. (2010). Should one compute the temporal difference fix point or minimize the bellman residual ? the unified oblique projection view. In *ICML*.

[Scherrer, 2014] Scherrer, B. (2014). What is the approximate policy iteration scheme with the best performance guarantees. In *ICML*.

[Scherrer and Lesner, 2012] Scherrer, B. and Lesner, B. (2012). On the use of non-stationary policies for stationary infinite-horizon Markov decision processes. In *NIPS 2012 - Neural Information Processing Systems*, South Lake Tahoe, United States.

[Schoknecht, 2002] Schoknecht, R. (2002). Optimality of reinforcement learning algorithms with linear function approximation. In Becker, S., Thrun, S., and Obermayer, K., editors, *NIPS*, pages 1555–1562. MIT Press.

[Schwarz, 1965] Schwarz, S. (1965). New theroems on non negative matrices. *Czechoslovak Mathematical Journal*.

[Schweitzer and Seidman, 1985] Schweitzer, P. and Seidman, A. (1985). Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110 :568–582.

[Shapley, 1953] Shapley, L. (1953). Stochastic games. In *Proceedings of The National Academy of Sciences of the United States of America*, pages 1095–1100.

[Sutton et al., 1999] Sutton, R., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs : A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112 :181–211.

[Sutton, 1988] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3(1) :9–44.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning : An Introduction*. The MIT Press.

[Tsitsiklis and Roy, 1997] Tsitsiklis, J. N. and Roy, B. V. (1997). An analysis of temporal-difference learning with function approximation. Technical report, IEEE Transactions on Automatic Control.

[Volonskii and Rozanov, 1959] Volonskii, V. and Rozanov, Y. (1959). Some limit theorems for random functions. *Theory of Probability and Its Applications*, 4(2) :178–197.

[Watkins, 1989] Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK.

[Watkins and Dayan, 1992] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4) :279–292.

[Williams and Baird, 1993] Williams, R. and Baird, L. (1993). Tight performance bounds on greedy policies based on imperfect value functions.

[Withers, 1981] Withers, C. (1981). Conditions for linear processes to be strong mixing. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(2) :477–480.

[Yu, 1994] Yu, B. (1994). Rates of convergence for empirical processes stationnary mixing consequences. *The Annals of Probability*, 19 :3041–3074.

[Yu, 2010] Yu, H. (2010). Convergence of least-squares temporal difference methods under general conditions. In *ICML*.

[Yu and Bertsekas, 2008] Yu, H. and Bertsekas, D. P. (2008). New error bounds for approximations from projected linear equations. Technical report, Dept. Computer Science, Univ. of Helsinki.

Résumé

Les processus décisionnels de Markov (MDP) sont un formalisme mathématique des domaines de l'intelligence artificielle telle que la planification, l'apprentissage automatique, l'apprentissage par renforcement...Résoudre un MDP permet d'identifier la stratégie (politique) optimale d'un agent en interaction avec un environnement stochastique. Lorsque la taille de ce système est très grande il devient difficile de résoudre ces processus par les moyens classiques.

Cette thèse porte sur la résolution des MDP de grande taille. Elle étudie certaines méthodes de résolutions : comme les abstractions et les méthodes dites de projection. Elle montre les limites de certaines abstractions et identifie certaines structures "les bisimulations" qui peuvent s'avérer intéressantes pour une résolution approchée du problème. Cette thèse s'est également intéressée à une méthode de projection l'algorithme *Least square temporal difference* LSTD(λ). Une estimation de la borne sur la vitesse de convergence de cet algorithme a été établie avec une mise en valeur du rôle joué par le paramètre λ. Cette analyse a été étendue pour déduire une borne de performance pour l'algorithme *Least square non stationary policy iteration* LS(λ)NSPI en estimant la borne d'erreur entre la valeur calculée à une itération fixée et la valeur sous la politique optimale qu'on cherche à identifier.

Abstract

Markov Decision Processes (MDP) are a mathematical formalism of many domains of artifical intelligence such as planning, machine learning, reinforcement learning...Solving an MDP means finding the optimal strategy or policy of an agent interacting in a stochastic environment.

When the size of this system becomes very large it becomes hard to solve this problem with classical methods.

This thesis deals with the resolution of MDPs with large state space. It studies some resolution methods such as : abstractions and the projection methods. It shows the limits of some approachs and identifies some structures that may be interesting for the MDP resolution. This thesis focuses also on projection methods, the *Least square temporal difference* algorithm LSTD(λ). An estimate of the rate of the convergence of this algorithm has been derived with an emphasis on the role played by the parameter λ. This analysis has then been generalized to the case of *Least square non stationary policy iteration* LS(λ)NSPI . We compute a performance bound for LS(λ)NSPI by bounding the error between the value computed given a fixed iteration and the value computed under the optimal policy, that we aim to determine.