Un sélecteur de Dantzig pour l'apprentissage par différences temporelles

Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, Mohammad Ghavamzadeh

Un sélecteur de Dantzig pour l'apprentissage par différences temporelles

Matthieu Geist[1], Bruno Scherrer[2], Alessandro Lazaric[3] et Mohammad Ghavamzadeh[3]

[1] Supélec, IMS Research Group, Metz (France)

[2] INRIA, MAIA Project Team, Nancy (France)

[3] INRIA, SEQUEL Project Team, Lille (France)

Résumé : En apprentissage par renforcement, LSTD est l'un des algorithmes d'approximation de la fonction de valeur les plus populaires. Lorsqu'il y a plus de fonctions de base que d'exemples, un problème se pose, qui peut être traité en combinant LSTD avec une forme de régularisation. En particulier, les méthodes de régularisation ℓ_1 tendent à sélectionner les fonctions de base (en favorisant la parcimonie des solutions) et sont donc particulièrement adaptées pour les problèmes de grande dimension. Toutefois, LSTD n'est pas un simple algorithme de régression ; il résout un problème de point fixe, l'intégration d'une régularisation ℓ_1 n'est pas évidente et peut entraîner certains inconvénients (comme l'hypothèse de P-matrice pour LASSO-TD). Cette contribution introduit un nouvel algorithme qui intègre LSTD au sélecteur de Dantzig, généralisant ce dernier à l'apprentissage par différences temporelles. En particulier, nous étudions les performances de l'algorithme proposé ainsi que son lien avec les approches de l'état de l'art, notamment la façon dont il surmonte certains inconvénients des solutions existantes.

Mots-clés : Apprentissage par renforcement, estimation de la fonction de valeur, sélecteur de Dantzig

1 Introduction

Un problème important de l'apprentissage par renforcement (RL pour *Reinforcement Learning*) Sutton & Barto (1998) est l'estimation de la qualité d'une politique donnée, via le calcul de la fonction de valeur associée (par exemple, dans le cadre d'un schéma d'itération de la politique). Souvent, l'espace d'état est trop grand et la fonction de valeur doit être approchée. De plus, quand le modèle (fonction de récompense et probabilités de transition) est inconnu, cette approximation doit être calculée en utilisant un ensemble de transitions échantillonnées. Un grand nombre d'algorithmes a été proposé pour résoudre ce problème d'approximation (voir par exemple Geist & Pietquin (2010) pour un état de l'art). Parmi eux, LSTD (*Least-Squares Temporal Differences*) (Bradtke & Barto, 1996) est probablement le plus populaire. En adoptant une paramétrisation linéaire, LSTD calcule le point fixe de l'opérateur de Bellman composé avec la projection orthogonale induite par la paramétrisation.

Dans beaucoup de cas, le nombre de fonctions de base de l'approximation linéaire peut être bien plus grand que le nombre de transitions disponibles. Par exemple, on peut vouloir considérer un espace d'hypothèses très riche, de façon à ce qu'il contienne la vraie fonction de valeur. Malheureusement, dans ce cas se pose un problème de sur-apprentissage. Une approche classique est alors d'introduire une forme de régularisation, qui va pénaliser la complexité des solutions. Alors que LSTD a souvent été combiné avec une régularisation ℓ_2, ce n'est que récemment que la régularisation ℓ_1 a été envisagée (voir la section 2.2 pour un été de l'art approfondi de ces algorithmes). Ce type d'approche est particulièrement intéressant dans la mesure où la régularisation ℓ_1 tend implicitement à promouvoir les solutions parcimonieuses et par conséquent à effectuer une sélection des fonctions de base, ce qui permet de traiter les problèmes de grande dimension. En particulier, l'algorithme LASSO-TD (Kolter & Ng, 2009) peut être vu comme une extension de l'algorithme LASSO Tibshirani (1996) à l'apprentissage par différences temporelles (les deux approaches étant identiques lorsque le facteur d'actualisation est nul). Cependant, LASSO-TD ne correspond pas à un problème d'optimisation convexe correct. En conséquence, d'une part il se base sur certaines hypothèses dont la validité peut ne pas être vérifiée dans un cadre *off-policy*, d'autre part il nécessite des résolveurs ad-hoc. D'autres algorithmes ont été proposés pour surmonter ces problèmes (comme par exemple ℓ_1-PBR (Geist & Scherrer, 2011)/$\ell_{2,1}$-LSTD (Hoffman *et al.*, 2011)), mais ils présentent leurs propres inconvénients.

Cet article introduit un nouvel algorithme, Dantzig-LSTD (ou D-LSTD pour faire court, voir la section 3), qui étend le sélecteur de Dantzig (DS pour *Dantzig Selector*) de Candes & Tao (2007) à l'apprentissage par différences temporelles. Plutôt que de résoudre un problème de point fixe, D-LSTD s'exprime simplement comme un programme linéaire, ce qui permet d'utiliser n'importe quel résolveur générique. De plus, le problème d'optimisation sous-jacent étant convexe, l'apprentissage *off-policy* ne pose pas problème. Pourtant, lorsque LASSO-TD est bien défini, les deux algorithmes fournissent des solutions similaires (voir la proposition 2), de façon similaire à LASSO-TD et DS. Nous montrons que pour un choix d'oracle du facteur de régularisation, la solution de D-LSTD converge rapidement vers la solution asymptotique de LSTD (avec un taux ne dépendant que du logarithme du nombre de fonctions de base), voir le théorème 1. L'algorithme proposé soulève également quelques questions, comme la qualité de l'approximation de la fonction de valeur ou le choix pratique (à partir des données) du facteur de régularisation. Ces points sont discutés section 4. Finalement, quelques résultats expérimentaux illustratifs sont présentés section 5.

2 LSTD et travaux connexes

Un processus de Markov valué[1] (MRP pour *Markov Reward Process*) est un uplet $\{S, P, R, \gamma\}$ où S est un espace d'état fini, $P = (p(s'|s))_{1\leq s,s'\leq |S|}$ est la matrice de transitions, $R = (r(s))_{1\leq s\leq |S|}$ satisfaisant $\|R\|_\infty \leq r_{\max}$ est le vecteur de récompenses et γ est le facteur d'actualisation. La fonction de valeur V est définie, pour un état s donné, comme le cumul pondéré moyen des récompenses obtenu en partant de l'état s et en suivant la dynamique de la chaîne de Markov : $V(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s]$. C'est l'unique point fixe de l'opérateur de Bellman défini par $T : V \to R + \gamma PV$.

En pratique, le modèle du MRP (c'est-à-dire les récompense R et les transitions P) est rarement connu. La seule information disponible est donnée par un ensemble de N transitions $\{(s_i, r_i, s'_i)_{1\leq i\leq n}\}$. Nous supposerons que les états $s_1 \dots s_n$ sont tirés selon un distribution d'échantillonnage μ (qui peut être différente de la distribution stationnaire du MRP) et que les états $s'_1 \dots s'_n$ sont échantillonnés selon les probabilités de transition $p(\cdot|s_i)$. Lorsque l'espace d'état est trop grand, la fonction de valeur ne peut pas être estimée pour chaque état, un schéma d'approximation est nécessaire. Nous considérons des fonctions de valeur $\hat{V}_\theta$ définies comme étant la combinaison linéaire de p fonctions de base $\phi_i(s)$:

$$\hat{V}_\theta(s) = \sum_{i=1}^{p} \theta_i \phi_i(s) = \theta^\top \phi(s).$$

Nous notons $\Phi \in \mathbb{R}^{|S|\times p}$ l'attribut matriciel dont les lignes sont les attributs vectoriels $\phi(s)^\top$, pour chaque $s \in S$. Cela définit un espace d'hypothèses $\mathcal{H} = \{\Phi\theta | \theta \in \mathbb{R}^p\}$ qui contient toutes les fonctions de valeur représentables grâce aux attributs vectoriels $\phi(\cdot)$. L'objectif est de trouver une fonction $\hat{V}_{\theta^*}$ qui approche au mieux la vraie fonction de valeur V.

2.1 LSTD

Soit Π_μ la projection orthogonale sur $\mathcal{H}$ respectivement à la distribution d'échantillonnage μ. Si D_μ est la matrice diagonale d'éléments $\mu(s)$ et $M_\mu = \Phi^\top D_\mu \Phi$ la matrice de Gram, alors l'opérateur de projection s'écrit $\Pi_\mu = \Phi M_\mu^{-1} \Phi^\top D_\mu$. Partant du fait que la fonction de valeur est le point fixe de l'opérateur de Bellman T, l'algorithme LSTD calcule le point fixe de l'opérateur composé $\Pi_\mu T$: $\hat{V}_{\theta^*} = \Pi_\mu T \hat{V}_{\theta^*}$. Soient $A \in \mathbb{R}^{p\times p}$ et $b \in \mathbb{R}^p$ définis par :

$$A = \Phi^\top D_\mu (I - \gamma P)\Phi \quad \text{et} \quad b = \Phi^\top D_\mu R.$$

Dans la suite de cet article, nous supposons que A et M_μ sont inversibles. Il est possible de montrer (grâce à de l'algèbre élémentaire) que $\hat{V}_{\theta^*}$ est un point fixe de $\Pi_\mu T$ si et seulement si θ^* est solution (unique) du système linéaire

$$A\theta^* = b.$$

Cette équivalence est particulièrement intéressantes dans la mesure où résoudre un problème de point fixe dans $\mathbb{R}^S$ revient à résoudre un système linéaire dans $\mathbb{R}^p$.

[1] Ces travaux s'étendent de façon évidente aux processus décisionnels de Markov, qui se réduisent à des MRP pour une politique fixée.

$\text{pen}_1 \setminus \text{pen}_2$	$\emptyset$	$\|.\|_2$	$\|.\|_1$
$\emptyset$	LSTD	✓	ℓ_1-PBR
$\|.\|_2$	✓	$\ell_{2,2}$-LSTD	$\ell_{2,1}$-LSTD
$\|.\|_1$	LASSO-TD	?	?

TAB. 1 – Résumé des approches régularisant LSTD (excepté ℓ_1-LSTD). Les croix sont des cas particuliers de $\ell_{2,2}$-LSTD et les points d'interrogation représentent des combinaisons qui n'ont pas encore été étudiés dans la littérature, à notre connaissance.

Comme P et R ne sont habituellement pas connus, il est nécessaires d'introduire des estimés basés sur les échantillons. Nous définissons :

$$\tilde{\Phi} = \begin{pmatrix} \phi(s_1)^\top \\ \vdots \\ \phi(s_n)^\top \end{pmatrix}, \tilde{\Phi}' = \begin{pmatrix} \phi(s'_1)^\top \\ \vdots \\ \phi(s'_n)^\top \end{pmatrix}, \tilde{R} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix},$$

avec $\tilde{\Phi} \in \mathbb{R}^{n \times p}$ et $\tilde{R} \in \mathbb{R}^n$. Les matrices aléatoires $\tilde{A}$ et $\tilde{b}$ sont alors définies par

$$\tilde{A} = \frac{1}{n} \tilde{\Phi}^\top \Delta\tilde{\Phi} \quad \text{et} \quad \tilde{b} = \frac{1}{n} \tilde{\Phi}^\top \tilde{R},$$

avec $\Delta\tilde{\Phi} = \tilde{\Phi} - \gamma\tilde{\Phi}'$. LSTD calcule la solution θ_0 du système linéaire (basé sur les échantillons)

$$\tilde{A}\theta_0 = \tilde{b}.$$

Nous notons que $\tilde{A}$ et $\tilde{b}$ sont tous deux des estimateurs non biaisés de respectivement A et b (c'est-à-dire que $\mathbb{E}[\tilde{A}] = A$ et $\mathbb{E}[\tilde{b}] = b$). Cela suggère que lorsque le nombre de transitions échantillonnées augmente, la solution de LSTD θ_0 converge vers la solution basées sur le modèle θ^*. Alternativement, comme LSTD calcule asymptotiquement le point fixe de l'opérateur composé $\Pi_\mu T$, sa solution θ_0 (basée sur les échantillons) peut se formuler comme étant la solution de deux problèmes d'optimisation imbriqués :

$$\begin{cases} \omega_\theta = \text{argmin}_\omega \, \|\tilde{R} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\omega\|_2^2 \\ \theta_0 = \text{argmin}_\theta \, \|\tilde{\Phi}\theta - \tilde{\Phi}\omega_\theta\|_2^2 \end{cases}. \tag{1}$$

La première équation projette l'image par l'opérateur de Bellman de la fonction de valeur estimée $\hat{V}_\theta$ sur l'espace d'hypothèses $\mathcal{H}$ et la seconde résout le problème de point fixe associé.

2.2 Travaux connexes

Lorsque le nombre de transitions n est proche du (voire plus petit que le) nombre p de fonctions de base, la matrice $\tilde{A}$ est mal conditionnée et une forme de régularisation peut être utilisée pour résoudre le problème. Dans cette section, nous proposons un état de l'art des approches régularisant LSTD.

La formulation de LSTD telle que donnée équation 1 est utile pour comprendre les différents schémas de régularisation qui peuvent lui être appliqué. En effet, chacun des problèmes de minimisation imbriqués peut être régularisé :

$$\begin{cases} \omega_\theta = \text{argmin}_\omega \, \|\tilde{R} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\omega\|_2^2 + \lambda_1 \text{pen}_1(\omega) \\ \theta_{\lambda_1,\lambda_2} = \text{argmin}_\theta \, \|\tilde{\Phi}\theta - \tilde{\Phi}\omega_\theta\|_2^2 + \lambda_2 \text{pen}_2(\theta) \end{cases}.$$

Avec cette formulation, tous les schémas de régularisation pour LSTD (excepté ℓ_1-LSTD, discuté à la fin de cette section) peuvent être résumés par le tableau 1.

La régression ridge (régularisation ℓ_2) est la forme la plus commune de régularisation, elle consiste simplement à ajouter un terme λI à $\tilde{A}$, ce qui assure son inversibilité. Cela correspond au cas $\lambda_1 \text{pen}_1(\omega) = \lambda\|\omega\|_2^2$ et $\lambda_2 = 0$, qui a été généralisé par Farahmand *et al.* (2008) avec l'algorithme $\ell_{2,2}$-LSTD pour lequel les deux termes sont des régularisations ℓ_2. Bien que ces approches permettent de traiter le mauvais conditionnement de la matrice $\tilde{A}$, elle ne sont pas particulièrement adaptées au cas $n \ll p$, pour lequel la

solution optimale est vraisemblablement parcimonieuse. En effet, il est bien connu que, contrairement à la régularisation ℓ_1, la régularisation ℓ_2 ne promeut pas la parcimonie et peut donc s'avérer inefficace lorsqu'il y a bien plus de fonctions de base que de transitions disponibles pour l'apprentissage.

La régularisation ℓ_1 a été introduite plus récemment par LASSO-TD, algorithme pour lequel la projection orthogonale est remplacée par une projection pénalisée par la norme ℓ_1 du vecteur de paramètres. Dans ce cas, les problèmes d'optimisation imbriqués de l'équation 1 peuvent s'exprimer comme le problème de point fixe suivant (si bien défini) :

$$\theta_{l,\lambda} = \operatorname*{argmin}_{\theta} \|\tilde{R} + \gamma\tilde{\Phi}'\theta_{l,\lambda} - \tilde{\Phi}\theta\|_2^2 + \lambda\|\theta\|_1. \tag{2}$$

Cet algorithme a été originellement introduit par Kolter & Ng (2009) sous le nom de LARS-TD, sa résolution utilisant une variation ad-hoc de l'algorithme LARS (Efron *et al.*, 2004). Une condition nécessaire pour que LARS-TD trouve une solution est que $\tilde{A}$ soit une P-matrice[2]. Malheureusement, cette condition peut ne pas se vérifier lorsque la distribution d'échantillonnage et la distribution stationnaire sont différentes (apprentissage *off-policy*). Bien que cela ne semble pas affecter les performances de l'algorithme en pratique (voir les expériences reportées par Kolter & Ng (2009)), il est souhaitable de supprimer cette condition. L'idée de LARS-TD est développée plus avant par Johns *et al.* (2010), où LASSO-TD est réexprimé comme un problème linéaire complémentaire (LCP pour *Linear Complementary Problem*). Cela permet d'utiliser des résolveurs standards de LCP[3], mais l'hypothèse de P-matrice est toujours nécessaire, dans la mesure où elle est liée au problème d'optimisation sous-jacent et non à la façon dont il est résolu. Enfin, les propriétés théoriques de LASSO-TD ont été analysées par Ghavamzadeh *et al.* (2011), qui donnent des bornes à échantillon fini sur l'erreur de prédiction, dans le cas *on-policy* et motif fixe (*fixed design*, ce qui signifie que la performance est évaluée en les états utilisés lors de l'apprentissage ; il ne s'agit donc pas de bornes de généralisation). En particulier, ils montrent que, de façon similaire à LASSO pour la régression, l'erreur de prédiction dépend de la parcimonie de la projection de la fonction de valeur (c'est-à-dire la norme ℓ_0 du vecteur de paramètre θ correspondant à la projection $\Pi_\mu V$ de la vraie fonction de valeur V) et ne dépend que du logarithme du nombre de fonctions de base. Cela implique que même si la dimension de l'espace d'hypothèse $\mathcal{H}$ est bien plus grande que le nombre d'exemples, LASSO-TD estime précisément la vraie fonction de valeur, dans un cadre *on-policy* et pour les états rencontrés lors de l'apprentissage.

De façon à éviter l'hypothèse de P-matrice, les algorithmes ℓ_1-PBR (*Projected Bellman residual*) (Geist & Scherrer, 2011) et $\ell_{2,1}$-LSTD (Hoffman *et al.*, 2011) ont été proposés (publiés simultanément et indépendamment, ces deux algorithmes sont identiques). L'idée est de placer le terme de régularisation ℓ_1 dans l'équation de point fixe plutôt que dans l'équation de projection. De façon équivalente, cela correspond à ajouter un terme de régularisation ℓ_1 à la minimisation du résidu de Bellman projeté (nous écrivons $\tilde{\Pi}$ la projection empirique et $\tilde{T}$ l'opérateur de Bellman échantillonné) :

$$\theta_{\text{pbr},\lambda} = \operatorname*{argmin}_{\theta} \|\tilde{\Pi}(\tilde{\Phi}\theta - \tilde{T}(\tilde{\Phi}\theta))\|_2^2 + \lambda\|\theta\|_1.$$

Comme c'est un problème d'optimisation convexe, il n'est pas nécessaire que $\tilde{A}$ soit une P-matrice et des résolveurs standards de LASSO peuvent être utilisés. Toutefois, cela n'est possible qu'au prix d'un plus grand coût algorithmique lorsque $n \ll p$ (en raison du calcul de la projection empirique). De plus, aucune analyse théorique n'est fournie.

Finalement, une dernière approche a été introduite par Pires (2011). L'idée est de voir LSTD comme résolvant un système linéaire ($\tilde{A}\theta_0 = \tilde{b}$) et d'y ajouter directement un terme de régularisation ℓ_1 :

$$\theta_{1,\lambda} = \operatorname*{argmin}_{\theta} \|\tilde{A}\theta - \tilde{b}\|_2^2 + \lambda\|\theta\|_1.$$

Nous nommons cette approche ℓ_1-LSTD. Etant définie via un problème d'optimisation convexe, elle ne présente aucune contrainte dans le cadre *off-policy* et n'importe quel résolveur standard peut être utilisé. Notons toutefois que pour $\gamma = 0$, ℓ_1-LSTD ne se spécialise pas en un algorithme connu d'apprentissage supervisé.

[2]Une P-matrice est une matrice ayant tous ses mineurs principaux positifs. Cela généralise la notion de matrice définie positive (non nécessairement symétrique).

[3]Notamment, certains de ces résolveurs permettent une "initialisation à chaud" (*warm start*), ce qui est intéressant dans un contexte d'itération de la politique.

3 Dantzig-LSTD

L'algorithme Dantzig-LSTD (ou D-LSTD pour faire court), contribution de cet article, produit un estimé $\theta_{d,\lambda}$ (donc une fonction de valeur $V_{\theta_{d,\lambda}}$) de faible norme ℓ_1 sous la contrainte que la norme ℓ_∞ du résidu de Bellman corrélé ($\tilde{\Phi}^\top(\tilde{R} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\theta) = \tilde{b} - \tilde{A}\theta$, avec $\tilde{R} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\theta$ le résidu de Bellman) soit plus petite qu'un paramètre λ. Formellement, D-LSTD résout :

$$\theta_{d,\lambda} = \operatorname*{argmin}_{\theta\in\mathbb{R}^p} \|\theta\|_1 \quad \text{sujet à } \|\tilde{A}\theta - \tilde{b}\|_\infty \leq \lambda. \tag{3}$$

Ce problème d'optimisation est convexe et peut facilement s'exprimer comme un problème de programmation linéaire (LP pour *Linear Programming*) :

$$\min_{u,\theta\in\mathbb{R}^p} \mathbf{1}^\top u \quad \text{sujet à } \begin{cases} -u \leq \theta \leq u \\ -\lambda\mathbf{1} \leq \tilde{A}\theta - \tilde{b} \leq \lambda\mathbf{1} \end{cases}.$$

D-LSTD est proche de l'algorithme DS (Candes & Tao, 2007), auquel il se réduit lorsque $\gamma = 0$. Dans la mesure où le problème d'optimisation sous-jacent est convexe, il n'est pas nécessaire que $\tilde{A}$ soit une P-matrice pour qu'il y ait une solution et n'importe quel résolveur de programme linéaire peut être utilisé (notamment le solveur de Candes & Tao (2007), qui utilise l'identité matricielle de Woodbury lorsque $n \ll p$).

3.1 Une analyse non-asymptotique

Dans cette section nous étudions la qualité de l'approximation de θ^* (solution de LSTD connaissant le modèle, c'est-à-dire satisfaisant $A\theta^* = b$) par $\theta_{d,\lambda}$. L'analyse s'inspire de celle de ℓ_1-LSTD (Pires, 2011). Dans la suite, nous supposons que les états $s_1, \ldots, s_n$ sont échantillonnés de façon i.i.d. selon une distribution d'échantillonnage arbitraire μ. Nous réservons l'étude du cas markovien à de futurs travaux.

Théoreme 1. *Soit $B_{\infty,\phi} = \max_{s\in S} \|\phi(s)\|_\infty$, la solution $\theta_{d,\lambda}$ de D-LSTD (voir l'équation 3) vérifie*

$$\inf_\lambda \|A\theta_{d,\lambda} - b\|_\infty \leq 2\left(\|\theta^*\|_1(1+\gamma)B_{\infty,\phi} + r_{max}\right) B_{\infty,\phi}\sqrt{\frac{4}{n}\ln\frac{8p}{\delta}},$$

avec probabilité d'au moins $1-\delta$.

Démonstration. (esquisse) Nous avons d'abord besoin d'une inégalité de concentration pour la norme ℓ_∞. Soient $x_1, \ldots, x_n$ des vecteurs aléatoires i.i.d. de moyenne $\bar{x} \in \mathbb{R}^d$ et bornés presque sûrement, $\|x_i\|_\infty \leq B$. En utilisant l'inégalité de Hoeffding et une borne d'unions, il est aisé de montrer qu'avec probabilité d'au moins $1-\delta$ on a :

$$\left\|\frac{1}{n}\sum_{i=1}^n x_i - \bar{x}\right\|_\infty \leq B\sqrt{\frac{2}{n}\ln\frac{2d}{\delta}}.$$

Soient $\Delta_{A,\max} = \|A - \tilde{A}\|_{\max}$ (norme $\max$ "par composante"[4]) et $\Delta_{b,\max} = \|b - \tilde{b}\|_\infty$. Nous avons l'inégalité de consistence suivante : $\|A\theta\|_\infty \leq \|A\|_{\max}\|\theta\|_1$. Combinée à l'inégalité triangulaire, elle donne :

$$\left|\|A\theta - b\|_\infty - \|\tilde{A}\theta - \tilde{b}\|_\infty\right| \leq \Delta_{A,\max}\|\theta\|_1 + \Delta_{b,\max}.$$

Choisissons $\lambda = \Delta_{A,\max}\|\theta^*\|_1 + \Delta_{b,\max}$. L'inégalité précédente implique que $\|\tilde{A}\theta^* - \tilde{b}\|_\infty \leq \lambda$ (nous rappelons que $A\theta^* = b$). En utilisant de plus le fait que $\theta_{d,\lambda}$ minimise l'équation 3, nous avons $\|\theta_{d,\lambda}\|_1 \leq \|\theta^*\|_1$. Combinés, ces résultats montrent que :

$$\|A\theta_{d,\lambda} - b\| \leq 2\Delta_{A,\max}\|\theta^*\|_1 + 2\Delta_{b,\max}.$$

Le résultat de concentration pour la norme ℓ_∞ peut être enfin utilisé pour borner avec forte probabilité les termes $\Delta_{A,\max}$ et $\Delta_{b,\max}$, ce qui donne le résultat final. Pour cela, nous avons utiliser les majorations suivantes : $\|\phi(s_i)(\phi(s_i) - \gamma\phi(s_i'))\|_\infty \leq B_{\infty,\phi}^2(1+\gamma)$ et $\|\phi(s_i)r_i\|_\infty \leq B_{\infty,\phi}r_{\max}$. □

[4] Soit $M \in \mathbb{R}^{d\times d}$, cette norme vérifie $\|M\|_{\max} = \max_{1\leq i,j\leq d}|M_{i,j}|$.

Comme l'algorithme est spécifiquement proposé pour traiter le cas de grande dimension ($n \ll p$), il est critique d'étudier l'influence des termes n et p sur la performance de D-LSTD. A des constantes près, la borne précédente peut s'écrire

$$\inf_{\lambda} \|A\theta_{d,\lambda} - b\|_\infty \leq O\Big(\|\theta^*\|_1 \sqrt{\frac{1}{n} \ln \frac{p}{\delta}}\Big).$$

Tout d'abord, notons que lorsque le nombre de transitions augmente, l'erreur de $\theta_{d,\lambda}$ tend vers zéro, ce qui implique que l'on atteint asymptotiquement les performances de LSTD basé sur le modèle, soit θ^*. De plus, la dépendance en le nombre p de fonctions de base est seulement logarithmique et la norme ℓ_1 de θ^* est supposée petite dès que la solution est parcimonieuse. Cela suggère que D-LSTD est adapté au cas $n \ll p$, dès lors que la solution asymptotique est parcimonieuse. Finalement, nous notons qu'aucune hypothèse n'a été faite sur le cadre de l'apprentissage (*on-policy* ou *off-policy*), mis à part l'inversibilité de A. Cela est particulièrement important, dans la mesure où cela signifie que, contrairement aux autres analyses théoriques concernant LSTD (voir par exemple Ghavamzadeh *et al.* (2011)), ce résultat est valable également dans un cadre *off-policy*. L'inconvénient majeur de notre analyse est qu'elle se base sur un choix d'oracle pour λ (le paramètre λ permettant de vérifier la borne nécessite de connaître la solution θ^*). Nous discutons le choix pratique (à partir des données) de λ dans la section 4.

3.2 Comparaison aux autres algorithmes

Comme ℓ_1-PBR et $\ell_{2,1}$-LSTD, D-LSTD est basé sur un problème d'optimisation convexe standard et bien défini. Ainsi, contrairement à LASSO-TD, $\tilde{A}$ ne doit pas être une P-matrice et des résolveurs standards peuvent être utilisés. Toutefois, D-LSTD a seulement un méta-paramètre (plutôt que deux) et, en général, a un coût computationnel plus faible (comparativement à la résolution des problèmes d'optimisation imbriqués de ℓ_1-PBR et $\ell_{2,1}$-LSTD).

D-LSTD est également lié à LASSO-TD au travers de la proposition suivante.

Proposition 2. *La solution $\theta_{l,\lambda}$ de LASSO-TD (voir équation 2), lorsqu'elle existe, satisfait les contraintes de D-LSTD :*

$$\|\tilde{A}\theta_{l,\lambda} - \tilde{b}\|_\infty \leq \lambda.$$

Démonstration. Les conditions d'optimalité de LASSO-TD sont obtenues en s'assurant que zéro appartient bien au sous-gradient de $\frac{1}{2}\|\tilde{\Phi}\theta - (\tilde{R} + \gamma\tilde{\Phi}'\theta_{l,\lambda})\|_2^2 + \lambda\|\theta\|_1$, puis en substituant $\theta_{l,\lambda}$ à θ (Kolter & Ng, 2009) :

$$\begin{cases} -\lambda \leq (\tilde{b} - \tilde{A}\theta_{l,\lambda})_i \leq \lambda, \forall 1 \leq i \leq p \\ (\tilde{b} - \tilde{A}\theta_{l,\lambda})_i = \lambda \Rightarrow (\theta_{l,\lambda})_i \geq 0 \\ (\tilde{b} - \tilde{A}\theta_{l,\lambda})_i = -\lambda \Rightarrow (\theta_{l,\lambda})_i \leq 0 \\ -\lambda < (\tilde{b} - \tilde{A}\theta_{l,\lambda})_i < \lambda \Rightarrow (\theta_{l,\lambda})_i = 0. \end{cases}$$

Ceci implique le résultat. □

Ainsi, D-LSTD et LASSO-TD satisfont les mêmes contraintes, mais $\|\theta_{l,\lambda}\|_1 \geq \|\theta_{d,\lambda}\|_1$, ce qui suggère une solution plus parcimonieuse. Cela n'est pas surprenant, car D-LSTD est lié à LASSO-TD comme DS est lié à LASSO (Bickel *et al.*, 2009). Cependant, grâce à sa formulation sous forme d'un problème d'optimisation convexe, D-LSTD évite les principaux inconvénients de LASSO-TD (notamment l'hypothèse de P-matrice).

Comme ℓ_1-LSTD, D-LSTD se base sur la vision de LSTD résolvant un système linéaire d'équations. Les deux approches relâchent la condition $\tilde{A}\theta = \tilde{b}$ (en utilisant une norme ℓ_2 sur le résidu pour ℓ_1-LSTD et une norme ℓ_∞ pour D-LSTD) tout en pénalisant la complexité des solutions au travers de la norme ℓ_1 du vecteur de paramètres. Les deux algorithmes ont les mêmes avantages comparés à LASSO-TD et à ℓ_1-PBR/$\ell_{2,1}$-LSTD. Leur différence principale réside dans leur vitesse de convergence. Un résultat similaire au théorème 1 existe pour ℓ_1-LSTD Pires (2011) :

$$\inf_{\lambda} \|A\theta_{1,\lambda} - b\|_2 \leq O\Big(\|\theta^*\|_1 \sqrt{\frac{p^2}{n} \ln \frac{1}{\delta}}\Big).$$

Bien que contrôler la norme ℓ_2 (pour ℓ_1-LSTD) puisse être plus dur que contrôler la norme ℓ_∞ (pour D-LSTD), ℓ_1-LSTD a une très mauvaise dépendance en p qui rend la borne non informative dans le cas où

$n \ll p$. De son côté, D-LSTD a une bien meilleur dépendance, logarithmique en le nombre de fonctions de base.

4 Discussion

Dans cette section, nous discutons comment l'erreur $\|A\theta - b\|$ (que nous contrôlons) est liée à l'erreur de prédiction sur la fonction de valeur, ainsi que le choix pratique du paramètre de régularisation.

4.1 Des paramètres à la valeur

De façon similaire à ce que font Yu & Bertsekas (2010), nous pouvons lier le terme $(V - \hat{V}_\theta)$ au terme $(A\theta - b)$.

Théoreme 3. *Pour chaque fonction de valeur $\hat{V}_\theta = \Phi\theta$, nous avons l'égalité par composantes suivante :*

$$V - \hat{V}_\theta = (I - \gamma\Pi_\mu P)^{-1}\left((V - \Pi_\mu V) + \Phi M_\mu^{-1}(A\hat{\theta} - b)\right). \tag{4}$$

Démonstration. Rappelons que pour la vraie fonction de valeur V nous vérifions $V = TV$ et que pour une fonction de valeur $\hat{V}_\theta = \Phi\theta$ de l'espace d'hypothèses nous avons $\hat{V}_\theta = \Pi_\mu \hat{V}_\theta$. Alors :

$$\begin{aligned} V - \Pi_\mu V &= V - \Pi_\mu TV - (\hat{V}_\theta - \Pi_\mu T\hat{V}_\theta) + (\hat{V}_\theta - \Pi_\mu T\hat{V}_\theta) \\ &= (I - \gamma\Pi_\mu P)(V - \hat{V}_\theta) + \Pi_\mu(\hat{V}_\theta - T\hat{V}_\theta), \\ V - \hat{V}_\theta &= (I - \gamma\Pi_\mu P)^{-1}\big((V - \Pi_\mu V) + \Pi_\mu(T\hat{V}_\theta - \hat{V}_\theta)\big). \end{aligned}$$

Avec l'égalité $\Pi_\mu(T\hat{V}_\theta - \hat{V}_\theta) = \Phi M_\mu^{-1}(b - A\theta)$, nous obtenons le résultat. □

Pour obtenir un résultat sur l'erreur de prédiction, nous appliquons la norme ℓ_∞ à l'équation. 4 et dérivons les inégalités associées (en utilisant la norme matricielle induite). Soit $L_\mu^\phi = \max_s \|M_\mu^{-1}\phi(s)\|_1$, un corollaire du théorème 1 est donc :

$$\inf_\lambda \|V - \hat{V}_{\theta_{d,\lambda}}\|_\infty \leq \|(I - \gamma\Pi_\mu P)^{-1}\|_\infty \left(\|V - \Pi_\mu V\|_\infty + O\left(\|\theta^*\|_1 L_\mu^\phi \sqrt{\frac{1}{n}\ln\frac{p}{\delta}}\right)\right).$$

En général, l'expression précédente ne peut pas être simplifiée plus avant. Toutefois, lorsqu'on considère un problème de grande dimension il est raisonnable d'également supposer que $\Pi_\mu P = P$ et $\Pi_\mu R = R$ (espace d'hypothèses suffisamment riche). Ainsi, l'espace d'hypothèses $\mathcal{H}$ est stable par l'opérateur de Bellman et $V \in \mathcal{H}$. Dans ce cas, nous avons $\|V - \Pi_\mu V\|_\infty = 0$ et il est possible de montrer que $\|(I - \gamma\Pi_\mu P)^{-1}\|_\infty = \frac{1}{1-\gamma}$. Ainsi, nous obtenons la borne suivante (également valide dans un cadre *off-policy*) :

$$\inf_\lambda \|V - \hat{V}_{\theta_{d,\lambda}}\|_\infty \leq O\left(\frac{\|\theta^*\|_1 L_\mu^\phi}{1-\gamma}\sqrt{\frac{1}{n}\ln\frac{p}{\delta}}\right).$$

Le terme le plus critique dans cette borne est L_μ^ϕ, qui peut cacher une dépendance en le nombre p de fonctions de base. En fait, bien que la valeur spécifique de L_μ^ϕ dépende de l'espace d'hypothèse, il est possible de construire des cas où la dépendance est en $\sqrt{p}$, ce qui neutralise malheureusement la faible dépendance en p du théorème 1. Savoir si cette dépendance en p est intrinsèque à l'algorithme ou un artefact de notre technique de preuve reste une question ouverte. A vrai dire, si $\theta_{d,\lambda}$ résout le système d'équations (défini par A et b) précisément, alors nous espérons que la fonction de valeur correspondante $\hat{V}_{\theta_{d,\lambda}}$ soit presque aussi efficace que la solution basée sur le modèle, $\hat{V}_{\theta^*}$. Les expériences de la section 5 semblent confirmer cette conjecture.

4.2 Validation croisée

Les résultats du théorème 1 sont vrais pour un choix d'oracle du paramètre de régularisation. En pratique, le choix de λ ne peut se baser que sur les données disponibles. C'est un problème pratique très important, qui n'est toutefois pas souvent discuté dans la littérature d'apprentissage par renforcement (notamment celle concernant les variations ℓ_1 de LSTD).

En apprentissage supervisé, les algorithmes minimisent généralement un risque empirique qui est la moyenne empirique d'une certaine fonction de perte. La validation croisée consiste à utiliser un échantillon de données indépendant pour estimer le risque réel (moyenne théorique de cette même fonction de perte). Le ou les méta-paramètres sont alors choisis de façon à minimiser le risque réel. Cependant, pour l'estimation de fonction de valeur, il n'y a pas de telle fonction de risque, la validation croisée ne peut donc pas être utilisée. Une approche générale pour la sélection de modèle dans le cadre de l'estimation de fonction de valeur a été proposée par Farahmand & Szepesvári (2011). Toutefois, il pourrait être préférable de concevoir une méthode ad-hoc (et plus simple) pour D-LSTD. Comme cet algorithme est défini via un problème d'optimisation standard (conduisant même à un algorithme classique d'apprentissage supervisé lorsque $\gamma = 0$), on pourrait être tenté d'utiliser la validation croisée standard. Malheureusement, les choses ne sont pas directes. En effet, $\|\tilde{A}\theta - \tilde{b}\|_\infty$ est la perte ($\|.\|_\infty$) d'une moyenne empirique ($\tilde{A}$ et $\tilde{b}$) et non pas la moyenne empirique d'une perte. Toutefois, il est toujours possible d'envisager certaines heuristiques.

Supposons que l'on veuille estimer la quantité $\|A\theta - b\|_\infty$ pour un vecteur de paramètres θ donné. Soient $\tilde{A}$ et $\tilde{b}$ des estimateurs non biaisés de A et b, alors on a par l'inégalité de Jensen que $\|A\theta - b\|_\infty \leq E[\|\tilde{A}\theta - \tilde{b}\|_\infty]$. Ainsi, étant donné un jeu indépendant de transitions, nous avons à disposition un estimateur non biaisé d'une majoration de $\|A\theta - b\|_\infty$.

En nous basant sur cette remarque, nous proposons une validation croisée à K plis (*K-fold cross-validation*) comme heuristique pour D-LSTD. Supposons que la base d'apprentissage est divisée en K ensembles $\mathcal{F}_k$. Soit $\theta^{-k}_{d,\lambda}$ l'estimé calculé en utilisant la base d'apprentissage privée de $\mathcal{F}_k$. Soient également $\tilde{A}_{\mathcal{F}_k}$ et $\tilde{b}_{\mathcal{F}_k}$ les quantités calculées en utilisant uniquement les données de $\mathcal{F}_k$. Une heuristique possible est de choisir le facteur de régularisation λ qui minimise

$$J_1(\lambda) = \frac{1}{K}\sum_{i=1}^{K} \|\tilde{A}_{\mathcal{F}_k}\theta^{-k}_{d,\lambda} - \tilde{b}_{\mathcal{F}_k}\|_\infty. \tag{5}$$

Toutefois, nous sommes intéressés par le cas $n \ll p$ et l'estimé $\tilde{A}_{\mathcal{F}_k}$ n'est calculé qu'avec $\frac{n}{K}$ transitions, il peut y avoir une forte variance. Une alternative (empiriquement plus efficace) consiste, au prix de l'ajout d'un biais, à choisir λ selon

$$J_2(\lambda) = \frac{1}{K}\sum_{i=1}^{K} \|\tilde{A}\theta^{-k}_{d,\lambda} - \tilde{b}\|_\infty. \tag{6}$$

Des heuristiques similaires peuvent être utilisées pour ℓ_1-LSTD.

Bien que l'heuristique précédente fonctionne bien dans nos expériences, elle ne présente aucune garantie théorique. Une approche de selection de modèle différente a été proposée pour ℓ_1-LSTD par Pires (2011). Elle consiste à choisir

$$\hat{\lambda} = \operatorname*{argmin}_{[a,b]} \|\tilde{A}\theta_{1,\lambda} - \tilde{b}\|_2^2 + \lambda' \|\theta_{1,\lambda}\|_1 \tag{7}$$

où $[a, b]$ est une grille exponentielle de valeurs du paramètre de régularisation et λ' peut être calculé en utilisant uniquement les données (pas de choix d'oracle). Cela ne nécessite pas de diviser la base d'entraînement (et donc de réduire le nombre de transitions disponibles) tout en assurant une borne pour $\|A\theta_{1,\hat{\lambda}} - b\|_2$ (qui n'est bien sûr pas meilleure que la borne d'oracle). Nous laissons à de futurs travaux l'adaptation de cette procédure de sélection du modèle à D-LSTD.

5 Illustration and expériences

La section 5.1 présente un exemple qui montre que D-LSTD évite le problème potentiel posé par l'apprentissage *off-policy*. La section 5.2 étudie un problème plus complexe qui illustre le cas $n \ll p$, dans un cadre *on-* et *off-policy* et qui étudie brièvement la validation croisée (heuristique).

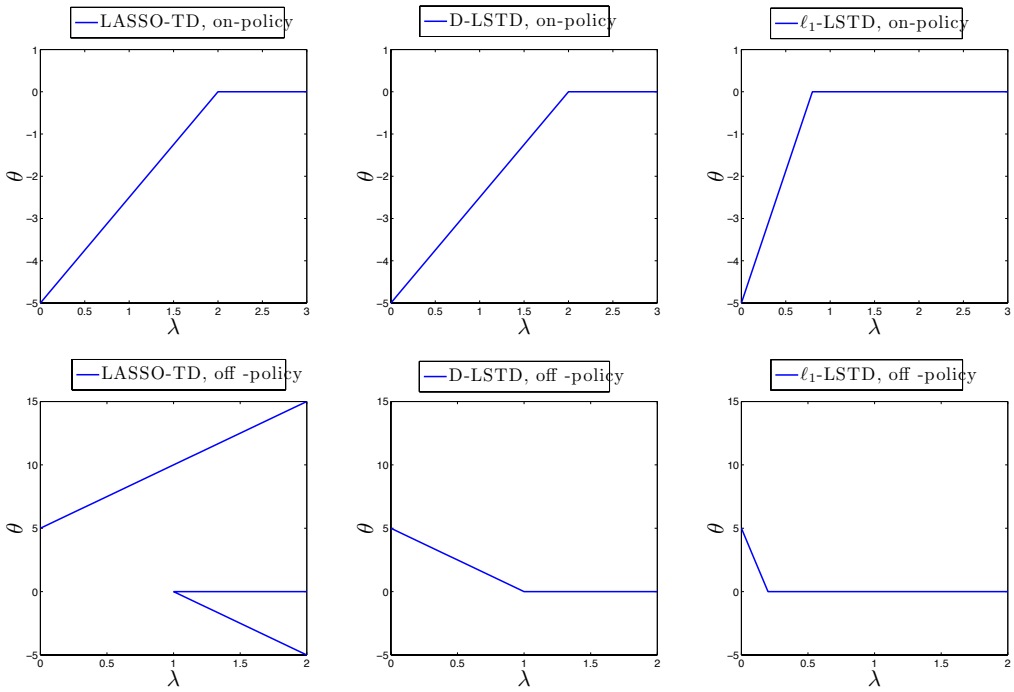

FIG. 1 – MDP pathologique, chemins de régularisation.

5.1 Un MDP pathologique

Nous considérons un simple MDP à deux états (voir par exemple Kolter & Ng (2009), mais l'exemple est plus ancien). La matrice de transition et le vecteur de récompense sont donnés par

$$P = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \text{ et } R = \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \tag{8}$$

La fonction de valeur est donc $V = \frac{-1}{1-\gamma} \begin{pmatrix} \gamma & 1 \end{pmatrix}^\top$ où γ est le facteur d'actualisation usuel. Nous considérons une unique fonction de base $\Phi = \begin{pmatrix} 1 & 2 \end{pmatrix}^\top$. L'objectif de cet exemple est de comparer les chemins de régularisation (asymptotiques) de LASSO-TD, ℓ_1-LSTD et D-LSTD, dans les cas *on-policy* et *off-policy* (pour lequel LASSO-TD échoue).

Cas *on-policy*. Dans le cas *on-policy*, la distribution d'échantillonnage est la distribution stationnaire, soit $\mu^\top = \begin{pmatrix} 0 & 1 \end{pmatrix}$. Les chemins de régularisation de chaque algorithme peuvent aisément être déterminés analytiquement en résolvant les conditions d'optimalité (il n'y a qu'un paramètre). Ils sont reportés sur la figure 1, dans les panneaux du haut. LASSO-TD et D-LSTD ont le même chemin. Cela était prévisible, dans la mesure où il n'y a ici qu'un seul paramètre (et à la lumière de la proposition 2). Ce n'est toutefois pas vrai dans le cas général (rappelons que LASSO-TD et D-LSTD héritent des mêmes différences qu'entre LASSO et DS).

Cas *off-policy*. Considérons maintenant la distribution uniforme $\mu^\top = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}$. Pour $\gamma > \frac{5}{6}$, A n'est plus une P-matrice et LASSO-TD n'a plus une unique solution, ni n'admet un chemin de régularisation continu et linéaire par morceaux. Les chemins sont montrés sur la figure 1, panneaux du bas. Le chemin de ℓ_1-LSTD est toujours bien défini et LASSO-TD a plus d'une solution. Dans ce cas, le chemin de D-LSTD est bien défini, il est même ce que l'on aurait attendu de LASSO-TD s'il n'y avait pas de problème lié à la condition de P-matrice non satisfaite. Notons que dans tous les cas (*on-* et *off-policy*), tous les algorithmes fournissent la solution de LSTD pour $\lambda = 0$.

5.2 Chaîne corrompue

Nous considérons le même problème de chaîne corrompue que Kolter & Ng (2009) et Hoffman *et al.* (2011). Chaque état $\mathbf{s}$ a $\bar{s} + 1$ composantes. La première est un entier naturel ($\mathbf{s}^1 \in \{1 \dots 20\}$) qui évolue

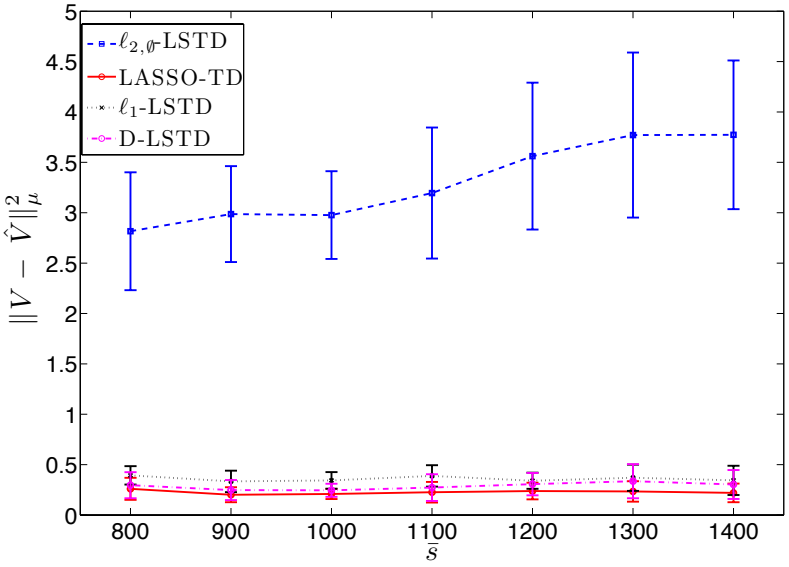

FIG. 2 – Chaîne corrompue, cas *on-policy*.

selon une chaîne à 20 états et 2 actions (les états sont connectés via une chaîne unidimensionnelle, les actions permettent de choisir la direction de déplacement et la probabilité de succès est de 0,9). Toutes les autres composantes d'état sont des bruits aléatoires gaussiens, $\mathbf{s}_t^{i+1} \sim \mathcal{N}(0,1)$. La récompense est de $+1$ si $\mathbf{s}_t^1 = 1$ ou 20. L'attribut vectoriel $\phi(\mathbf{s}) \in \mathbb{R}^{\bar{s}+6}$ comporte une fonction constante, cinq fonctions à base radiale pour la première composante de l'état et $\bar{s}$ fonctions identité pour les composantes de bruit :

$$\phi(\mathbf{s}) = \begin{pmatrix} 1 & \text{RBF}_1(\mathbf{s}^1) & \dots & \text{RBF}_5(\mathbf{s}^1) & \mathbf{s}^2 & \dots & \mathbf{s}^{\bar{s}+1} \end{pmatrix}^\top .$$

Nous comparons LASSO-TD (considérant son implémentation dérivée de LARS), ℓ_1-LSTD et D-LSTD (pour lesquels nous avons utilisé la librairie ℓ_1-magic Romberg (2005)). Nous standardisons les données en supprimant la fonction constante, en centrant les observations, en centrant et normalisant l'attribut matriciel empirique $\tilde{\Phi}$ et en appliquant la même transformation (calculée à partir de $\tilde{\Phi}$) à $\tilde{\Phi}'$. La fonction constante peut être calculée analytiquement, c'est l'erreur de Bellman moyenne (sans régularisation, cela permet de retrouver la solution de LSTD). Nous considérons également $\ell_{2,\emptyset}$-LSTD, l'approche la plus standard de régularisation ℓ_2 pour LSTD.

Evaluation *on-policy*. Nous étudions d'abord le cas *on-policy*. La politique évaluée est la politique optimale (aller à gauche si $\mathbf{s}^1 \leq 10$, à droite sinon). L'apprentissage se fait à partir de 400 transitions (20 trajectoires de longueur 20, initialisées aléatoirement dans $\{1 \dots 20\}$), le nombre $\bar{s}$ de composantes inutiles varie entre 800 et 1400. Les résultats présentés sur la figure 2 sont moyennés sur 20 essais indépendants. Pour LARS-TD, nous calculons tout le chemin de régularisation (du moins jusqu'à ce que trop de fonctions de base soient ajoutées) et nous entraînons les autres algorithmes pour un jeu de paramètres de régularisation (exponentiellement espacés entre 10^{-3} et 10). Pour chaque algorithme et chaque point de la courbe, nous reportons la meilleure erreur de prédiction (estimée sur 500 états test, dont la première composante est uniformément échantillonnée sur $\{1 \dots 20\}$), calculée respectivement à la vraie fonction de valeur. C'est donc un choix d'oracle, la vraie fonction de valeur étant utilisée pour choisir le facteur de régularisation. Toutes les approches de type régularisation ℓ_1 sont bien plus efficaces que la régularisation ℓ_2, ce qui montre que leur performance ne dépend que peu du nombre p de fonctions de bases (comme prévu par le théorème 1, pour D-LSTD). Parmi elles, LASSO-TD semble être meilleur de façon consistante, suivie de près par D-LSTD, puis par ℓ_1-LSTD. Pour LASSO-TD et $\ell_{2,\emptyset}$-LSTD, ces résultats sont consistants avec ceux donnés par Hoffman *et al.* (2011). Notons qu'il y a un plus grand choix de paramètres de régularisation pour LASSO-TD, dans la mesure où tout le chemin de régularisation est calculé. Cela pourrait expliquer les meilleurs résultats de LASSO-TD, comparés à D-LSTD.

Validation croisée heuristique. Tous les résultats de la figure 2 reposent sur un choix d'oracle pour le paramètre de régularisation considéré. Ce n'est envisageable en pratique. Comme expliqué section 4, ℓ_1-

Algorithme	Erreur (moyenne $\pm$ écart-type)
$\ell_{2,\emptyset}$-LSTD (oracle)	$2,82 \pm 0,58$
LASSO-TD (oracle)	$0,26 \pm 0,10$
ℓ_1-LSTD (validation croisée, J_1/J_2)	$4,14 \pm 0,84$ / $0,34 \pm 0,12$
D-LSTD (validation croisée, J_1/J_2)	$0,65 \pm 0,18$ / $0,23 \pm 0,11$

TAB. 2 – Validation croisée.

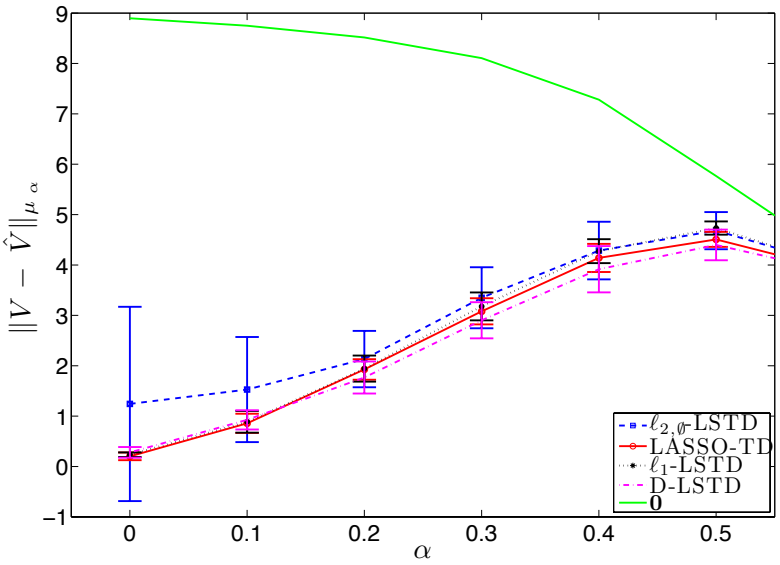

FIG. 3 – Chaîne corrompue, cas *off-policy*.

LSTD et D-LSTD peuvent bénéficier de schémas de validation croisée heuristique. Nous expérimentons la validation croisée à K plis (avec $K = 5$) sur ce problème, avec les schémas J_1 (équation 5) et J_2 (équation 6), pour $n = 400$ transitions d'entraînement et $\bar{s} = 800$ composantes inutiles. Les résultats, reportés tableau 2, sont moyennés sur 20 essais indépendants. L'erreur est estimée comme avant, mais ici elle n'est pas utilisée pour choisir le meilleur facteur de régularisation. Les résultats pour J_1 sont plutôt mauvais, probablement en raison de la grande variance de l'estimateur utilisé (toutefois, des résultats non reportés ici tendent à montrer que D-LSTD choisi souvent le bon facteur de régularisation, au prix de quelques *outliers*). Le schéma J_2 est bien meilleur, il est même comparable au schéma d'oracle (voir également la figure 2). La comparaison des résultats de cette heuristique J_2 en utilisant un test de Behrens-Fisher(test d'égalité de moyennes) montre que ℓ_1-LSTD et LASSO-TD sont différents (risque de 5 %), mais pas D-LSTD et LASSO-TD (même risque).

Evaluation *off-policy*. Nous étudions maintenant le cas *off-policy*. Soit π_{opt} la politique optimale (aller à gauche si $\mathbf{s}^1 \leq 10$, à droite sinon) et $\pi_{\text{worst}} = 1 - \pi_{\text{opt}}$ (aller à droite si $\mathbf{s}^1 \leq 10$, à gauche sinon). Nous définissons $\pi_\alpha = (1-\alpha)\pi_{\text{opt}} + \alpha\pi_{\text{worst}}$, avec $\alpha \in [0, \frac{1}{2}]$. Soit également μ_α la distribution stationnaire associée et rappelons que V désigne la vraie fonction de valeur. Nous considérons le même problème que précédemment, avec $\bar{s} = 800$. Pour des valeurs de α variant de 0 à $0,5$, nous échantillons $n = 400$ états de la chaîne corrompue selon la distribution μ_α ainsi que les transitions associées, échantillonnées selon la politique optimale. Le paramètre de régularisation est choisi de telle façon à minimiser l'erreur entre la fonction de valeur et son estimée (donc procédure d'oracle), pour chaque algorithme. Les résultats sont moyennés sur 50 essais indépendants. La figure 3 montre l'erreur $\|\hat{V}_\alpha - V\|_{\mu_\alpha}$ comme une fonction de α. Le terme **0** correspond à la prédiction nulle, c'est-à-dire d'erreur correspondante $\|V\|_{\mu_\alpha}$. Dans tous les cas, D-LSTD semble être légèrement meilleur que les autres algorithmes, les résultats empirant lorsqu'on s'éloigne de la distribution stationnaire (lorsque α augmente). En aucun cas LASSO-TD ne semble souffrir du contexte *off-policy*, ce qui suggère qu'ici la condition de P-matrice est satisfaite. On constate également

que la différence entre les schémas de régularisation ℓ_1 et ℓ_2 s'amenuise lorsque α augmente. Un schéma de régularisation ℓ_1 peut s'avérer utile lorsqu'il y a plus de fonctions de base que d'exemples, mais il y a peu à faire lorsque le décalage entre les distributions stationnaire et d'échantillonnage est trop important. Bien que ce ne soit pas reporté sur la figure, toutes les approches sont également inefficaces lorsque α tend vers 1, dans la mesure où il n'y a plus d'information utile dans les données.

6 Conclusion

Dans cet article, nous avons introduit l'algorithme Dantzig-LSTD, avec pour objectif de palier aux inconvénients des approches de régularisation ℓ_1 existantes pour l'apprentissage par différences temporelles. Comme D-LSTD est défini via un programme linéaire standard, il ne nécessite pas que $\tilde{A}$ soit une P-matrice et il peut être calculé en utilisant n'importe que résolveur. La solution de D-LSTD est une bonne approximation de la solution asymptotique de LSTD, comme exprimé par le théorème 1. Elle est également proche de celle de LASSO-TD (lorsque cette dernière est bien définie), tel qu'exprimé par la proposition 2. En fait, D-LSTD et LASSO-TD héritent vraisemblablement de différences semblables à celles qui existent entre DS et LASSO. De plus, les résultats expérimentaux préliminaires présentés section 5 montrent que D-LSTD est au moins aussi efficace que LASSO-TD.

Toutefois, il reste un certain nombre de points à éclaircir. Comme nous l'avons discuté section 4, lorsqu'on considère l'erreur de prédiction plutôt que l'erreur du système linéaire, un terme additionnel de dépendance en le nombre de fonctions de base semble apparaître. Nous ne savons pas si c'est un artefact de la preuve ou si c'est inhérent à l'algorithme (bien que les résultats expérimentaux nous fasse pencher vers la première option). Concernant le choix pratique du paramètre de régularisation, nous prévoyons d'adapter le schéma de sélection de modèle de ℓ_1-LSTD (Pires, 2011) à D-LSTD et de l'expérimenter. Finalement, nous prévoyons également d'étudier et d'expérimenter D-LSTD dans un cadre de contrôle (itération de la politique approchée).

Références

BICKEL P. J., RITOV Y. & TSYBAKOV A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, **37**(4), 1705–1732.

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

CANDES E. & TAO T. (2007). The Dantzig selector : statistical estimation when p is much larger than n. *Annals of Statistics*, **35**(6), 2313–2351.

EFRON B., HASTIE T., JOHNSTONE I. & TIBSHIRANI R. (2004). Least Angle Regression. *Annals of Statistics*, **32**(2), 407–499.

FARAHMAND A., GHAVAMZADEH M., SZEPESVÁRI C. & MANNOR S. (2008). Regularized Policy Iteration. In *Proc. of NIPS 21*.

FARAHMAND A. M. & SZEPESVÁRI C. (2011). Model selection in reinforcement learning. *Machine Learning Journal*, **85**(3), 299–332.

GEIST M. & PIETQUIN O. (2010). *A Brief Survey of Parametric Value Function Approximation*. Rapport interne, Supélec.

GEIST M. & SCHERRER B. (2011). ℓ_1-penalized projected Bellman residual. In *Proc. of EWRL 9*.

GHAVAMZADEH M., LAZARIC A., MUNOS R. & HOFFMAN M. (2011). Finite-Sample Analysis of Lasso-TD. In *Proc. of ICML*.

HOFFMAN M. W., LAZARIC A., GHAVAMZADEH M. & MUNOS R. (2011). Regularized Least Squares Temporal Difference learning with nested ℓ_2 and ℓ_1 penalization. In *Proc. of EWRL 9*.

JOHNS J., PAINTER-WAKEFIELD C. & PARR R. (2010). Linear Complementarity for Regularized Policy Evaluation and Improvement. In *Proc. of NIPS 23*, p. 1009–1017.

KOLTER J. Z. & NG A. Y. (2009). Regularization and Feature Selection in Least-Squares Temporal Difference Learning. In *Proc. of ICML*.

PIRES B. A. (2011). Statistical analysis of ℓ_1-penalized linear estimation with applications. Master's thesis, University of Alberta.

ROMBERG J. (2005). ℓ_1-magic matlab library. `http://users.ece.gatech.edu/~justin/l1magic/`.

SUTTON R. S. & BARTO A. G. (1998). *Reinforcement Learning : an Introduction*. The MIT Press.

TIBSHIRANI R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, **58**(1), 267–288.

YU H. & BERTSEKAS D. P. (2010). Error Bounds for Approximations from Projected Linear Equations. *Mathematics of Operations Research*, **35**, 306–329.