Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement

Matthieu Geist — Olivier Pietquin

Equipe IMS
Supélec, Metz, France

RÉSUMÉ. *Le dilemme entre exploration et exploitation est un problème important en apprentissage par renforcement (AR). Les approches associées les plus efficaces utilisent généralement une information d'incertitude des valeurs estimées pendant l'apprentissage. Par ailleurs, la prise en compte de grands espaces d'état est problématique en AR et l'approximation de la fonction de valeur est devenue un domaine de recherche important pour s'attaquer à cette limitation. Ces deux problèmes co-existent dans la plupart des applications. Cependant, peu d'approches permettent à la fois d'approcher la fonction de valeur et d'en déduire une information d'incertitude sur les estimations. Dans cet article, nous montrons comment une telle information d'incertitude se déduit naturellement d'un cadre de travail basé sur les différences temporelles et le filtrage de Kalman. Nous introduisons un schéma d'apprentissage actif pour un algorithme du second ordre du type itération de la valeur (nommément KTD-Q). Ce schéma permet, entre autre, d'étendre des méthodes traitant du dilemme entre exploration et exploitation pour l'instant limitées au cas tabulaire comme il en sera discuté. Ces différentes approches sont expérimentées sur un problème simple de bandits, et la plus efficace empiriquement est également expérimentée sur un problème de gestion de dialogue dans le cadre d'une tâche concrète de dialogue homme-machine.*

ABSTRACT. *The dilemma between exploration and exploitation is an important topic in reinforcement learning (RL). Most successful approaches in addressing this problem tend to use some uncertainty information about values estimated during learning. On another hand, scalability is known as being a lack of RL algorithms and value function approximation has become a major topic of research. Both problems arise in real-world applications, however few approaches allow approximating the value function while maintaining uncertainty information about estimates. Even fewer use this information in the purpose of addressing the exploration/exploitation dilemma. In this paper, we show how such an uncertainty information can be derived from a Kalman-based Temporal Differences (KTD) framework. An active learning scheme for a second-order value-iteration-like algorithm (named KTD-Q) is proposed. We also*

RSTI - RIA – xx/2011. Numéro spécial Cap 2010, pages 1 à 25

suggest adaptations of several existing exploration/exploitation dilemma schemes. These approaches are experimented on a simple bandit problem, and the most efficient of them is also experimented on a real-world dialogue management task.

MOTS-CLÉS : *Apprentissage par renforcement , approximation de la fonction de valeur, gestion de l'incertitude, dialogue homme-machine.*

KEYWORDS: *Reinforcement learning , value function approximation, uncertainty management, dialogue management.*

1. Introduction

L'apprentissage par renforcement (AR) (Sutton *et al.*, 1996) est généralement considéré comme la réponse du domaine de l'apprentissage automatique à la problématique du contrôle optimal d'un système dynamique. Dans ce paradigme général, un agent informatique apprend à contrôler son environnement (c'est-à-dire le système dynamique) à partir d'interactions avec ce dernier. A chacune de ces interactions est associée une récompense immédiate, qui est une indication locale de la qualité du contrôle effectué par l'agent. Dans ce cadre, l'optimalité est définie par la maximisation d'un cumul de ces récompenses sur le long terme. Plus formellement, à chaque pas de temps (discret) i, le système dynamique est dans un état donné s_i. L'agent choisi une action a_i parmi un panel de contrôles disponibles et l'applique au système qui transite alors vers l'état s_{i+1} en suivant sa propre dynamique (qui peut être stochastique). L'agent reçoit alors une récompenses r_i associée à la transition (s_i, a_i, s_{i+1}) et son objectif et de maximiser le cumul moyen pondéré de ces récompenses, qu'il modélise sous la forme d'une fonction dite de valeur.

Dans le cadre le plus général, cet apprentissage se fait en ligne et l'agent doit contrôler le système tout en essayant d'apprendre une politique optimale. Cela pose un problème majeur, connu sous le nom de dilemme entre exploration et exploitation : à chaque pas de temps, l'agent doit choisir entre une action qu'il considère comme optimale, en accord avec sa connaissance imparfaite de l'environnement (exploitation) et une action qu'il considère comme sous-optimale, mais qui peut améliorer sa connaissance de l'environnement et donc la qualité du contrôle effectué (exploration). Un choix populaire de contrôle est la politique ϵ-gloutonne, qui consiste à choisir une action gloutonne (soit d'exploitation) avec une probabilité $1 - \epsilon$ (exploitation) et une action aléatoire (et donc exploratoire) avec une probabilité ϵ. Une autre stratégie classique est la politique *softmax* (Sutton *et al.*, 1996) qui génère une action aléatoirement selon un distribution de Gibbs basée sur la valeur associée aux différentes actions. Cependant, les schémas les plus efficaces pour traiter de ce dilemme tendent à utiliser une information d'incertitude, pour choisir entre exploration et exploitation mais également pour diriger l'exploration. Dearden *et al.* (1998) maintiennent une distribution pour chaque Q-valeur (valeur d'un couple état-action, qui quantifie localement la qualité du contrôle) et proposent deux schémas. Le premier consiste à échantillonner les actions en accord avec la distribution des Q-valeurs associées, le second utilise une valeur myope de l'information imparfaite, qui approche le gain qu'il peut y avoir à choisir une action exploratrice en terme d'amélioration de la qualité de la politique. Strehl *et al.* (2006) maintiennent un intervalle de confiance pour chaque Q-valeur, la politique considérée étant gloutonne respectivement à la borne supérieure de cet intervalle. Cette approche permet d'obtenir des bornes PAC (*probably approximately correct*). Sakaguchi *et al.* (2004) proposent d'utiliser une politique de Gibbs (ou *softmax*) pour laquelle le paramètre de température est remplacé par un index de fiabilité. La plupart de ces approches ont été conçues dans la cas tabulaire, c'est-à-dire lorsqu'une représentation exacte de la fonction de valeur est possible. Cependant, l'approximation de la fonction de valeur dans le cas de grands espaces d'état est un autre sujet

d'importance en apprentissage par renforcement, et ces deux problématiques sont rarement considérées conjointement : peu d'approximateurs de la fonction de valeur fournissent une information d'incertitude sur la valeur estimée. Engel (2005) propose un tel algorithme, mais l'utilisation effective de l'information d'incertitude est laissée en perspective.

Dans cet article, nous montrons comment une telle information d'incertitude peut être obtenue à partir du cadre de travail des différences temporelles de Kalman (KTD pour *Kalman Temporal Differences*), récemment introduit (Geist *et al.*, 2009a; Geist *et al.*, 2009b; Geist *et al.*, 2010c). Nous introduisons également un schéma d'apprentissage actif qui utilise cette information pour accélérer l'apprentissage dans un contexte *off-policy*, ainsi qu'un certain nombre d'adaptations d'approches traitant du dilemme entre exploration et exploitation existantes, mais conçues dans le cas tabulaire. Toutes ces contributions sont illustrées sur des benchmarks classiques. Par rapport à Geist *et al.* (2010a), que cet article étend, nous proposons également une application à un problème de gestion de dialogue dans le cadre d'une tâche concrète de dialogue homme-machine.

2. Préliminaires

2.1. *Apprentissage par renforcement*

Cet article se place dans le cadre de travail des processus décisionnels de Markov (PDM). Un PDM est un tuple $\{S, A, P, R, \gamma\}$ où S est l'espace d'état, A est l'espace d'action, $P : s, a \in S \times A \rightarrow p(.|s, a) \in \mathcal{P}(S)$ est une famille de probabilités de transitions, $R : S \times A \times S \rightarrow \mathbb{R}$ est la fonction (bornée) de récompense et γ le facteur d'actualisation. Une politique π associe à chaque état une distribution sur les actions : $\pi : s \in S \rightarrow \pi(.|s) \in \mathcal{P}(A)$. La fonction de valeur d'une politique donnée est définie par :

$$V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi] \qquad [1]$$

où r_i est la récompense immédiate observée au pas de temps i, la moyenne étant faite sur toutes les trajectoires possibles, sachant que l'agent démarre dans l'état s et suit la politique π par la suite. La Q-fonction (ou fonction de valeur sur les couples état-action) ajoute un degré de liberté supplémentaire sur le choix de la première action et est définie par :

$$Q^\pi(s, a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi] \qquad [2]$$

L'apprentissage par renforcement a pour objectif de déterminer (à partir d'interactions) la politique optimale π^* qui maximise la fonction de valeur pour chaque état : $\pi^* = \text{argmax}_\pi(V^\pi)$. Deux schémas parmi d'autres mènent à la politique optimale.

Premièrement, l'algorithme d'*itération sur les politiques* consiste à apprendre la fonction de valeur d'une politique donnée, puis à améliorer cette politique, la nouvelle étant gloutonne par rapport à la fonction de valeur apprise :

$$\pi_{i+1}(s) = \underset{a \in A}{\operatorname{argmax}}\, Q^{\pi_i}(s,a) \quad [3]$$

Cela suppose de résoudre l'équation d'évaluation de Bellman (qui se déduit de la définition de la fonction de valeur en tenant compte du caractère markovien des probabilités de transitions et de la fonction de récompense), donnée ici pour la fonction de valeur ainsi que pour la Q-fonction :

$$V^{\pi}(s) = E_{s',a|\pi,s}[R(s,a,s') + \gamma V^{\pi}(s')] \quad [4]$$

$$Q^{\pi}(s,a) = E_{s',a'|\pi,s,a}[R(s,a,s') + \gamma Q^{\pi}(s',a')] \quad [5]$$

En itérant les phases d'évaluation et d'amélioration de la politique, cet algorithme converge vers la politique optimale. Le second schéma, appelé *iteration sur les valeurs*, estime directement la politique optimale. Il nécessite de résoudre l'équation d'optimalité de Bellman :

$$Q^{*}(s,a) = E_{s'|s,a}[R(s,a,s') + \gamma \max_{b \in A} Q^{*}(s',b)] \quad [6]$$

Lorsque l'espace d'état est trop grand, il peut être difficile, voire impossible, d'obtenir des solutions exactes. Dans ce cas, il peut être nécessaire d'approcher la Q-fonction.

2.2. *Différences temporelles de Kalman - KTD*

Initialement, le filtre de Kalman (1960) est une méthode statistique ayant pour objectif de traquer en ligne l'état caché d'un système dynamique, pas nécessairement stationnaire, cela à partir d'observations indirectes de cet état. Par exemple, le filtrage de Kalman peut être utilisé pour estimer la position d'un avion en vol : l'état caché à inférer est la position de l'avion, les observations indirectes de cet état étant un ensemble de signaux radar. L'idée sous-jacente à KTD est d'exprimer le problème de l'estimation de la fonction de valeur sous la forme d'un problème de filtrage : en supposant un approximateur de fonction basé sur une famille de fonctions paramétrées, l'ensemble des paramètres (par exemples les connexions synaptiques d'un perceptron multi-couche) est l'état caché à traquer, les observations étant les récompenses, liées aux paramètres au travers d'une des équations de Bellman [4-6]. Ainsi, l'approximation de la fonction de valeur peut bénéficier des avantages inhérents du filtrage de Kalman, notamment la gestion de l'incertitude grâce à la modélisation statistique sous-jacente.

Les notations suivantes sont adoptées, suivant que l'objectif est l'estimation de la fonction de valeur, l'estimation de la Q-fonction ou l'optimisation directe de la Q-fonction (c'est-à-dire l'estimation de la Q-fonction liée à la politique optimale) :

$$t_i = \begin{cases} (s_i, s_{i+1}) \\ (s_i, a_i, s_{i+1}, a_{i+1}) \\ (s_i, a_i, s_{i+1}) \end{cases} \quad [7]$$

$$g_{t_i}(\theta_i) = \begin{cases} \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_b \hat{Q}_{\theta_i}(s_{i+1}, b) \end{cases} \quad [8]$$

où $\hat{V}_\theta$ (respectivement $\hat{Q}_\theta$) est la représentation paramétrique de la fonction de valeur (respectivement de la Q-fonction), θ étant le vecteur de paramètres. Un point de vue statistique est adopté et le vecteur de paramètres est considéré comme étant une variable aléatoire. Le problème est exprimé sous la forme d'une représentation dite *espace-d'état*[1] :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i \\ r_i = g_{t_i}(\theta_i) + n_i \end{cases} \quad [9]$$

En utilisant le vocabulaire propre au filtrage de Kalman, la première équation est l'équation dite d'évolution. Elle spécifie que le vecteur de paramètres que l'on cherche à estimer suit une marche aléatoire, dont la moyenne correspond à l'estimation optimale de la fonction de valeur au pas de temps i. Le bruit d'évolution v_i est blanc (et donc centré), indépendant et de matrice de variance P_{v_i}. Cette équation d'évolution permet de prendre en compte un problème non-stationnaire, ce qui se produit si la dynamique du système n'est pas stationnaire ou dans un cadre d'itération généralisée de la politique. Cependant, il est difficile de quantifier de façon exacte ces non-stationnarités et nous adoptons donc l'heuristique d'une simple marche aléatoire, qui se révèle efficace en pratique. La seconde équation est l'équation dite d'observation, elle lie la transition observée ainsi que la récompense associée à la fonction de valeur (ou à la Q-fonction), via les paramètres, grâce à l'une des équations de Bellman. Le bruit d'observation est également supposé blanc, indépendant et de variance P_{n_i}.

KTD est un algorithme du deuxième ordre : à chaque interaction, il met à jour le vecteur de paramètres moyens, mais également la matrice de variance associée. L'approche se décompose en trois étapes. Premièrement, les moments d'ordre un et deux des paramètres sont prédits, en accord avec l'équation d'évolution et en utilisant les précédentes estimations. Ensuite, des statistiques d'intérêt sont caculées. La troisième étape applique une correction au moments prédits, cette correction utilisant le gain de Kalman (calculé grâce aux statistiques d'intérêts obtenues lors de la deuxième étape),

1. Ce terme provient de la littérature sur le filtrage de Kalman et ne doit pas être confondu avec l'espace d'état du MDP.

la récompense prédite $\hat{r}_{i|i-1}$ ainsi que la récompense observée r_i (leur différence étant une forme d'erreur de différence temporelle, soit la différence entre la récompense et la différence entre les estimées des valeurs des deux états successifs de la transition associée).

En dehors du cas linéaire (c'est-à-dire dépendance aux paramètres linéaire de la fonction approchée), les statistiques d'intérêt ne sont généralement pas calculables analytiquement. Particulièrement, elles ne le sont pas si la paramétrisation est non-linéaire (réseaux de neurones par exemples, connus pour poser problème en apprentissage par renforcement) ou si l'équation d'optimalité de Bellman est considérée (à cause de l'opérateur $\max$). Cependant, un schéma d'approximation ne requérant pas le calcul de gradients, la transformation non-parfumée de Julier *et al.* (2004), permet d'estimer les moments d'ordres un et deux de la transformation non-linéaire d'une variable aléatoire. Soit X un vecteur aléatoire et $Y = f(X)$ sa transformation non linéaire (dans le cadre de cet article, X est l'ensemble des paramètres et f est la fonction g_{t_i}). Soit n la dimension du vecteur aléatoire X. Un ensemble de $2n+1$ *sigma-points*, ainsi que les poids associés, est calculé comme suit :

$$\begin{cases} x^{(0)} = \bar{X} & j = 0 \\ x^{(j)} = \bar{X} + (\sqrt{(n+\kappa)P_X})_j & 1 \leq j \leq n \\ x^{(j)} = \bar{X} - (\sqrt{(n+\kappa)P_X})_{n-j} & n+1 \leq j \leq 2n \end{cases} \quad [10]$$

$$\text{et } \begin{cases} w_0 = \frac{\kappa}{n+\kappa} & j = 0 \\ w_j = \frac{1}{2(n+\kappa)} & 1 \leq j \leq 2n \end{cases} \quad [11]$$

où $\bar{X}$ est la moyenne de X, P_X sa matrice de variance, κ un facteur d'échelle permettant de contrôler la précision de l'estimation et $(\sqrt{P_X})_j$ la $j^{\text{ème}}$ colonne de la décomposition de Cholesky de P_X. Ensuite, l'image par f de chaque sigma-point est calculée : $y^{(j)} = f(x^{(j)})$, pour $0 \leq j \leq 2n$. L'ensemble des sigma-points et de leurs images peut enfin être utilisé pour calculer les approximations suivantes :

$$\begin{cases} \bar{Y} \approx \bar{y} = \sum_{j=0}^{2n} w_j y^{(j)} \\ P_Y \approx \sum_{j=0}^{2n} w_j (y^{(j)} - \bar{y})(y^{(j)} - \bar{y})^T \\ P_{XY} \approx \sum_{j=0}^{2n} w_j (x^{(j)} - \bar{X})(y^{(j)} - \bar{y})^T \end{cases} \quad [12]$$

Grâce à la transformation non-parfumée, des algorithmes pratiques peuvent être obtenus. A chaque pas de temps i, un ensemble de sigma-points est calculé à partir des prédictions des paramètres moyens $\hat{\theta}_{i|i-1}$ et de la matrice de variance associée $P_{i|i-1}$. Les prédictions des récompenses sont alors calculées comme étant l'image par g_{t_i} de ces sigma-points en utilisant l'une des équations d'observation [8]. Enfin, les sigma-points et leurs images sont utilisés pour calculer les statistiques d'intérêt de la deuxième étape. Cela fournit un algorithme générique, valide pour chacune des trois équations de Bellman et n'importe quelle représentation paramétrique de V ou Q. Cette approche est résumée dans l'algorithme 1, où p désigne le nombre de paramètres.

Algorithme 1 : KTD

Initialisation: *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i et la récompense r_i;

Phase de prédiction;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des sigma-points;
$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, 0 \leq j \leq 2p\}$;
`/* en utilisant` $\hat{\theta}_{i|i-1}$ `et` $P_{i|i-1}$ `*/`
$\mathcal{W} = \{w_j, 0 \leq j \leq 2p\}$;
$\mathcal{R}_{i|i-1} = \{\hat{r}^{(j)}_{i|i-1} = g_{t_i}(\hat{\theta}^{(j)}_{i|i-1}), 0 \leq j \leq 2p\}$;
`/* voir Eq. [8] */`

Calcul des statistiques d'intérêt;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;
$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p} w_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;

Phase de correction;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

Geist *et al.* (2010b) présentent dans le détail le cadre de travail des différences temporelles de Kalman (qui n'a été ici que brièvement introduit) et proposent également un certain nombre de résultats théoriques (telles que des preuves de convergence) et expérimentaux.

3. Calculer l'incertitude des valeurs estimées

3.1. *Principe*

Les paramètres étant modélisés comme des variables aléatoires, pour n'importe quel état donné la fonction de valeur paramétrée est également une variable aléatoire. Ce modèle statistique permet de calculer les moyenne et variance associées. Soit $\hat{V}_\theta$ la fonction de valeur approchée, paramétrée par le vecteur aléatoire θ de moyenne $\bar{\theta}$ et de matrice de variance P_θ. Soient $\hat{V}_\theta(s)$ et $\hat{\sigma}^2_{\hat{V}_\theta}(s)$ les moyenne et variance associées, pour un état s donné. Une première étape pour propager l'information d'incertitude des paramètres vers la valeur de l'état considéré est de calculer les sigma-points as-

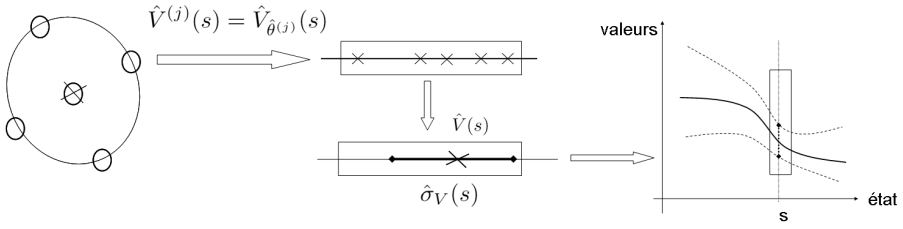


Figure 1. *Calcul de l'incertitude.*

sociés au vecteur de paramètres, c'est-à-dire $\Theta = \{\theta^{(j)},\, 0 \leq j \leq 2p\}$, ainsi que les poids correspondants, à partir de $\bar{\theta}$ et P_θ, tel qu'expliqué dans la section précédente. Ensuite, les images de ces sigma-points sont calculés en utilisant la fonction de valeur paramétrée : $\mathcal{V}_\theta(s) = \{\hat{V}_\theta^{(j)}(s) = \hat{V}_{\theta^{(j)}}(s), \quad 0 \leq j \leq 2p\}$. Connaissant ces images et les poids associés, les statistiques d'intérêt peuvent être approchées :

$$\begin{cases} \bar{V}_\theta(s) = \sum_{j=0}^{2p} w_j \hat{V}_\theta^{(j)}(s) \\ \hat{\sigma}^2_{V_\theta}(s) = \sum_{j=0}^{2p} w_j (\hat{V}_\theta^{(j)}(s) - \bar{V}_\theta(s))^2 \end{cases} \quad [13]$$

Cela est illustré sur la figure 1 et l'extension à la Q-fonction est évidente. Ainsi, à chaque pas de temps, une information d'incertitude peut être estimée dans le cadre de travail des différences temporelles de Kalman.

3.2. *Illustration*

La première expérience vise à illustrer l'information d'incertitude disponible sur une simple problème de navigation. L'espace d'état bi-dimensionnel et continu est le carré unité $(x, y) \in [0, 1]^2$. Les actions consistent à se déplacer dans les quatre directions (haut, bas, gauche, droite), l'amplitude du déplacement étant de $0,05$ dans chaque cas. La récompense est de $+1$ si l'agent quitte la pièce dans la zone $\{x \in [\frac{3}{8}, \frac{5}{8}], y = 1\}$, -1 si l'agent le quitte dans la zone $\{x \in [0, \frac{3}{8}[\cup]\frac{5}{8}, 1], y = 1\}$, 0 sinon. L'algorithme considéré est KTD-V (c'est-à-dire que l'équation d'évaluation de Bellman pour la fonction de valeur [4] est considérée). La fonction de valeur est un réseau RBF (*radial basis function*), plus précisément neuf noyaux gaussiens équi-répartis (centrés en $\{0; 0,5; 1\} \times \{0; 0,5; 1\}$ et d'écart-type $0,5$). Les paramètres sont donc les poids de chaque gaussienne. Le facteur d'actualisation est fixé à $\gamma = 0,9$. L'agent commence chaque épisode dans une position aléatoire (x_0, y_0) où x_0 est échantillonné selon une distribution gaussienne, $x_0 \sim \mathcal{N}(\frac{1}{2}, \frac{1}{8})$, et y_0 est échantillonné selon une distribution uniforme, $y_0 \sim \mathcal{U}_{[0;0,05]}$. La politique suivie par l'agent (dont la fonction de valeur est apprise par KTD-V) consiste à aller vers le haut avec une probabilité de $0,9$ et à aller dans une des trois autres directions avec probabilité $\frac{0,1}{3}$. Les *a priori* sont fixés à $\theta_{0|0} = 0$ et $P_{0|0} = 10I$ et les variances des bruits à $P_{n_i} = 1$ et $P_{v_i} = 0I$.

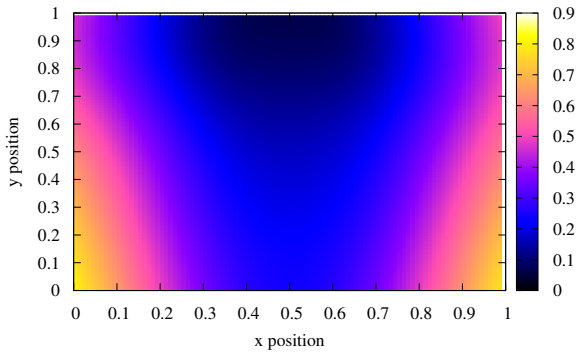


Figure 2. *Illustration de l'incertitude.*

L'apprentissage est fait sur trente épisodes, les résultats étant présentés sur la figure 2 qui montre l'écart-type approché de la fonction de valeur comme une fonction des états. Si l'on considère l'abscisse, l'incertitude est plus faible au centre qu'aux bords, ce qui s'explique par le fait que les trajectoires ont lieu plus souvent dans le centre du domaine pendant l'apprentissage (politique considérée et état initial sélectionné suivant une gaussienne). En considérant l'ordonnée, l'incertitude est plus faible près de la borne supérieure ($y = 1$) que près de la borne inférieure ($y = 0$), ce qui s'explique par le fait que les valeurs rétro-propagées sont d'autant moins certaines qu'elles sont éloignées de la source de la récompense.

4. Une forme d'apprentissage actif

4.1. *Principe*

Nous expliquons dans cette section comment l'information d'incertitude disponible peut être utilisée dans une forme d'apprentissage actif. La spécialisation de KTD à l'apprentissage de la Q-fonction optimale (c'est-à-dire l'algorithme 1 en considérant la troisième équation de [8]) est appelée KTD-Q. C'est un algorithme *off-policy*: il estime la Q-fonction optimale à partir de trajectoires générées en suivant une politique comportementale sous-optimale b. Une question naturelle se pose alors : quelle politique comportementale choisir afin d'accélérer l'apprentissage ? Une réponse triviale, mais impraticable, est de suivre la politique optimale. La solution que nous proposons consiste à échantillonner les actions relativement à l'incertitude de leurs valeurs estimées. Soit i l'index temporel courant. Le système est dans un état s_i et l'agent doit choisir une action a_i. Les prédictions $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$ sont disponibles et peuvent être utilisées pour estimer l'incertitude de la Q-function paramétrée par $\theta_{i|i-1}$ en l'état

s_i, et ce pour toute action a. Soit $\hat{\sigma}^2_{Q_{i|i-1}}(s_i, a)$ la variance associée. Nous proposons d'échantillonner l'action a_i selon la politique (heuristique) suivante :

$$b(.|s_i) = \frac{\hat{\sigma}_{Q_{i|i-1}}(s_i, .)}{\sum_{a \in A} \hat{\sigma}_{Q_{i|i-1}}(s_i, a)} \qquad [14]$$

Cette politique totalement exploratrice favorise les actions dont le résultat est le moins certain. L'approche correspondante, appelée "KTD-Q actif", est résumée par l'algorithme 2.

Algorithme 2 : KTD-Q actif

Initialisation: *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$, state s_1;

for $i \leftarrow 1, 2, \ldots$ **do**

Phase de prédiction;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des sigma-points et échantillonnage de l'action;
$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, \quad 0 \leq j \leq 2p\}$;
`/* en utilisant` $\hat{\theta}_{i|i-1}$ `et` $P_{i|i-1}$ `*/`
$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \quad \}$;
pour $a \in A$ **faire**
$\mathcal{Q}_{i|i-1}(s_i, a) = \{\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a), 0 \leq j \leq 2p\}$;
$\bar{Q}_{i|i-1}(s_i, a) = \sum_{j=0}^{2p} w_j \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a)$;
$\hat{\sigma}^2_{Q_{i|i-1}}(s_i, a) = \sum_{j=0}^{2p} w_j (\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a) - \bar{Q}_{i|i-1}(s_i, a))^2$;
Echantillonner a_i selon $b(.|s_i)$, voir Eq. [14];
Observer r_i et s_{i+1} ;
$\mathcal{R}_{i|i-1} = \{\hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i)$
$\quad - \gamma \max_{a \in A} \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a), 0 \leq j \leq 2p\}$;

Calculer les statistiques d'intérêt;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;
$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p} w_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;

Phase de correction;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

4.2. *Expérimentation*

La deuxième expérimentation que nous proposons est celle du pendule inversé. Cette tâche requiert de maintenir à la verticale un pendule de longueur et masse inconnues en appliquant des forces au chariot auquel est attachée sa base. Ce benchmark est totalement décrit par Lagoudakis *et al.* (2003) et nous utilisons la même paramétrisation (un réseau RBF). L'objectif ici est de comparer deux algorithmes de type itération de la valeur, nommément KTD-Q et Q-learning, tous deux ayant pour objectif l'apprentissage direct et *off-policy* de la Q-fonction optimale. Autant que nous le sachions, KTD-Q est le premier algorithme d'ordre deux de type itération de la valeur, la difficulté principale étant de prendre en compte correctement l'opérateur $\max$ (à noter que Yu *et al.* (2007) proposent également un tel algorithme, cependant pour une classe plus restreinte de processus décisionnels de Markov). C'est pourquoi nous le comparons à un algorithme du premier ordre (Q-learning). Le schéma d'apprentissage actif proposé est également expérimenté : il utilise l'information d'incertitude calculée par KTD pour accélérer la convergence.

Pour Q-learning, le taux d'apprentissage est fixé à $\alpha_i = \alpha_0 \frac{n_0+1}{n_0+i}$ avec $\alpha_0 = 0.5$ et $n_0 = 200$, en accord avec Lagoudakis *et al.* (2003). Pour KTD-Q, les paramètres sont fixés à $P_{0|0} = 10I$, $P_{n_i} = 1$ et $P_{v_i} = 0I$. Pour tous les algorithmes le vecteur de paramètres initial est nul. Les trajectoires utilisées pour l'entraînement sont générées en utilisant une politique totalement aléatoire (échantillonnage uniforme des actions). L'agent commence chaque épisode en un état aléatoire correspondant à une légère perturbation de la position d'équilibre. La performance est mesurée comme étant le nombre moyen d'interactions dans un épisode de test (un tel épisode applique la politique gloutonne respectivement à la Q-fonction estimée, l'apprentissage étant figé, et un maximum de 3 000 interactions étant autorisé, ce qui correspond à maintenir le pendule à la verticale pendant cinq minutes). Les résultats présentés sur la figure 3 en échelle semi-logarithmique, moyennés sur 100 apprentissages indépendants, comparent KTD-Q et Q-learning.

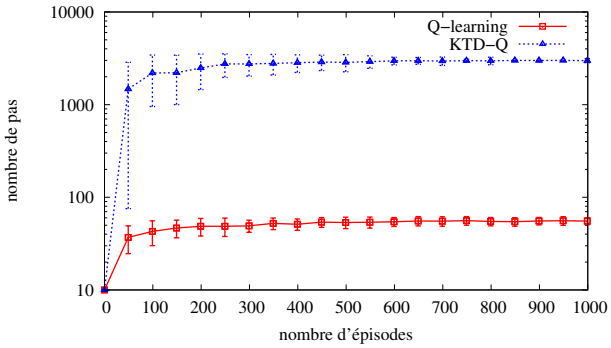


Figure 3. *Apprentissage de la politique optimale.*

Il apparaît clairement, d'après cette figure, que KTD-Q permet d'apprendre un politique optimale (c'est-à-dire maintenir le pendule à la verticale le plus longtemps possible) asymptotiquement et que des politiques quasi-optimales sont apprises très rapidement, après seulement quelques dizaines d'épisodes. Avec le même nombre de trajectoires, l'algorithme Q-learning utilisant la même paramétrisation échoue à apprendre une politique qui permette de maintenir le pendule à la verticale pendant plus de quelques secondes. Des résultats similaires sont obtenus par Lagoudakis *et al.* (2003). Il est intéressant de noter que KTD-Q est tout à fait comparable à l'algorithme LSPI (Least-Squares Policy Iteration) en termes de performances et de vitesse d'apprentissage, voir Lagoudakis *et al.* (2003, figure 16), et ce en utilisant moins d'échantillons (LSPI utilise tous les échantillons à chacune de ses itérations, alors que KTD-Q n'utilise chaque échantillon qu'une seule fois).

Dans un second temps, nous comparons l'algorithme KTD-Q, pour lequel les trajectoires sont générées selon une politique totalement aléatoire (c'est-à-dire échantillonnage uniforme des actions, indépendamment de l'état courant) à l'algorithme KTD-Q actif, pour lequel les actions sont échantillonnées en accord avec l'heuristique [14] qui utilise l'information d'incertitude disponible, les conditions expérimentales étant les mêmes. Les résultats sont présentés figure 4.

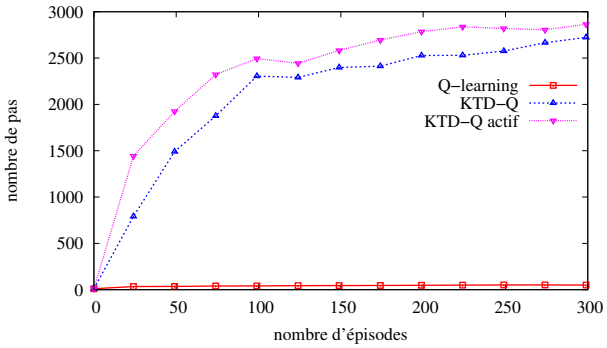


Figure 4. *Apprentissage actif.*

Il est à noter que la longueur moyennne d'un épisode lorsque la politique est totalement aléatoire est de dix interactions, tandis que pour la politique comportementale de l'apprentissage actif elle est de onze. En conséquence, le nombre d'interactions par épisode n'explique que partiellement l'accélération de l'apprentissage (au plus 10%, bien moins que l'amélioration réelle qui peut atteindre 100% au départ). D'après la figure 4, il est clair qu'échantillonner les actions en accord avec la politique [14] accélère l'apprentissage. Par exemple, KTD-Q classique nécessite presque 50 épisodes pour atteindre les mêmes performances que KTD-Q actif au bout de 25 épisodes. Ce schéma d'apprentissage actif est donc efficace, du moins sur cet exemple. Il est à noter que le schéma d'apprentissage actif n'a pas été considéré combiné avec l'algorithme Q-learning. La raison en est simple : ce dernier ne produit pas d'information d'incertitude.

5. Dilemme entre exploration et exploitation

Dans cette section sont proposées un certain nombre d'approches visant à traiter du dilemme entre exploration et exploitation. La première est le schéma classique ϵ-glouton, qui servira de référence. Les autres sont inspirées de la littérature (cas tabulaire) et utilisent l'information d'incertitude disponible (voir la section 3 pour son calcul). La combinaison de KTD-SARSA (c'est-à-dire l'algorithme 1 en considérant la deuxième équation de [8]) avec un schéma de contrôle est présentée algorithme 3.

Algorithme 3 : KTD-SARSA et contrôle

Initialisation: *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$, état s_1, action a_1;

for $i \leftarrow 1, 2, \ldots$ **do**

Appliquer a_i en l'état s_i ;

Observer r_i et s_{i+1} ;

Phase de prédiction;

$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;

$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des sigma-points et échantillonnage;

$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, \quad 0 \leq j \leq 2p\}$;

`/* en utilisant` $\hat{\theta}_{i|i-1}$ `et` $P_{i|i-1}$ `*/`

$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \quad \}$;

for $a \in A$ **do**

$\mathcal{Q}_{i|i-1}(s_{i+1}, a) = \{\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a), 0 \leq j \leq 2p\}$;

$\bar{Q}_{i|i-1}(s_{i+1}, a) = \sum_{j=0}^{2p} w_j \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a)$;

$\hat{\sigma}^2_{Q_{i|i-1}}(s_{i+1}, a) = \sum_{j=0}^{2p} w_j (\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a) - \bar{Q}_{i|i-1}(s_{i+1}, a))^2$;

Echantillonner a_{i+1} selon $\pi(.|s_{i+1})$, voir Eq. [15-18];

$\mathcal{R}_{i|i-1} = \{\hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i)$

$\quad - \gamma \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a_{i+1}), 0 \leq j \leq 2p\}$;

Calcul des statistiques d'intérêt;

$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;

$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;

$P_{r_i} = \sum_{j=0}^{2p} w_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;

Phase de correction;

$K_i = P_{\theta r_i} P_{r_i}^{-1}$;

$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;

$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

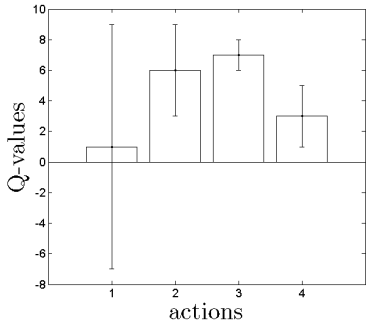


Figure 5. *Q-valeurs et incertitude associée.*

5.1. *Politique ϵ-gloutonne*

Avec une politique ϵ-gloutonne, l'agent choisit une action optimale (respectivement à la Q-fonction estimée courante) avec une probabilité $1-\epsilon$, une action aléatoire (tirage uniforme) avec une probabilité ϵ (δ étant le symbole de Kronecker) :

$$\begin{aligned}\pi(a_{i+1}|s_{i+1}) = (1-\epsilon)\delta(a_{i+1} = \underset{b\in A}{\operatorname{argmax}}\, \bar{Q}_{i|i-1}(s_{i+1}, b)) \\ + \epsilon\delta(a_{i+1} \neq \underset{b\in A}{\operatorname{argmax}}\, \bar{Q}_{i|i-1}(s_{i+1}, b))\end{aligned} \quad [15]$$

Cette politique est peut-être la plus basique, bien qu'elle soit très utilisée, et elle n'utilise aucune information d'incertitude. Une Q-fonction arbitraire pour un état donné et quatre actions différentes est illustrée figure 5. Pour chaque action, la figure donne la Q-valeur estimée ainsi que l'incertitude associée (c'est-à-dire ici $\pm$ l'écart-type associé). Par exemple, l'action 3 présente la plus grande valeur et la plus faible incertitude, tandis que l'action 1 a la plus faible valeur mais la plus grande incertitude. La distribution sur les actions associée à la politique ϵ-gloutonne est illustrée sur la figure 6.a. La plus grande probabilité est associée à l'action 3, et les autres actions ont la même (faible) probabilité, malgré le fait qu'elles présentes des Q-valeurs estimées et incertitudes associées sensiblement différentes.

5.2. *Politique confiante-gloutonne*

La deuxième approche considérée consiste à agir de façon gloutonne par rapport à la borne supérieure d'un intervalle de confiance estimé. L'approche n'est pas nouvelle (Kaelbling, 1993), mais des garanties PAC (probablement approximativement correct) ont récemment été données par Strehl *et al.* (2006) pour le cas tabulaire (pour lequel l'écart-type est proportionnel à l'inverse de la racine carrée du nombre de visites de la paire état-action considérée). Dans le cas traité ici, nous postulons que la largeur de l'intervalle de confiance est proportionnel à l'écart-type estimé (ce qui est

vrai si la distribution des paramètres est gaussienne). Soit α un paramètre libre positif, la politique confiante-gloutonne est définie comme suit :

$$\pi(a_{i+1}|s_{i+1}) = \delta\Big(a_{i+1} = \underset{b\in A}{\operatorname{argmax}} \big(\bar{Q}_{i|i-1}(s_{i+1}, b) + \alpha\hat{\sigma}_{Q_{i|i-1}}(s_{i+1}, b)\big)\Big) \quad [16]$$

La même Q-fonction arbitraire est considérée (voir figure 5) et la politique confiante-gloutonne est illustrée figure 6.b, qui représente la borne supérieure de l'intervalle de confiance par rapport à laquelle l'agent agit de façon gloutonne. L'action 1 est choisie car elle a le plus grand score, malgré le fait qu'elle présente la plus faible valeur estimée. Il est à noter que l'action 3, qui présente la plus grande Q-valeur estimée, n'est qu'en troisième position pour cette politique.

5.3. *Politique bonus-gloutonne*

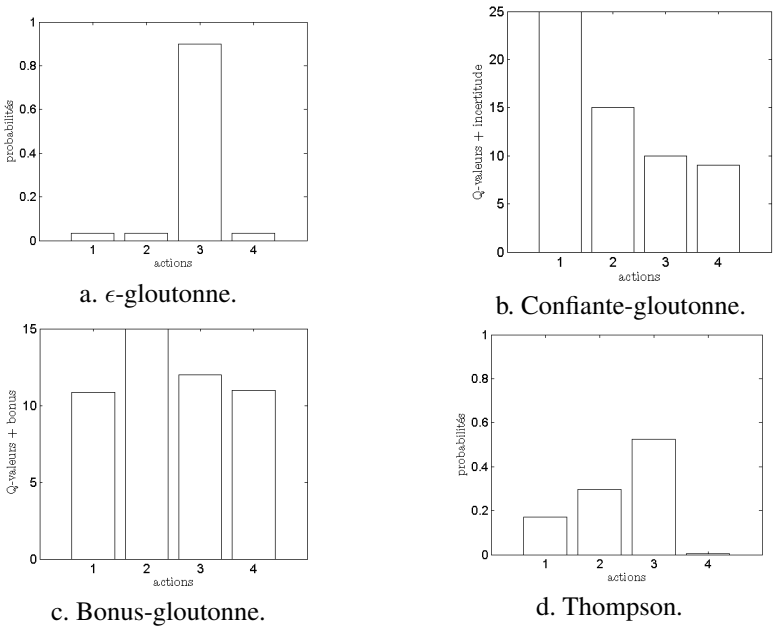
La troisième approche proposée s'inspire de la méthode proposée par Kolter *et al.* (2009). La politique qu'ils utilisent est gloutonne par rapport à la Q-valeur estimée plus un bonus, ce bonus étant proportionnel à l'inverse du nombre de visites de la paire état-action d'intérêt. Cela peut s'interpréter comme une variance, de la même façon que l'écart-type pour la politique confiante-gloutonne est assimilé à la racine carrée de cette même quantité. La politique bonus-gloutonne que nous proposons utilise donc la variance estimée, et est définie par (β_0 et β étant deux paramètres libres) :

$$\pi(a_{i+1}|s_{i+1}) = \delta\Big(a_{i+1} = \underset{b\in A}{\operatorname{argmax}} \big(\bar{Q}_{i|i-1}(s_{i+1}, b) + \beta\frac{\hat{\sigma}^2_{Q_{i|i-1}}(s_{i+1}, b)}{\beta_0 + \hat{\sigma}^2_{Q_{i|i-1}}(s_{i+1}, b)}\big)\Big) \quad [17]$$

Cette politique bonus-gloutonne est illustrée figure 6.c, toujours respectivement la Q-fonction arbitraire de la figure 5. L'action 2 a le plus grand score, elle est donc choisie. A noter que les autres actions ont approximativement le même score, malgré le fait qu'elle présentent des Q-valeurs estimées sensiblement différentes.

5.4. *Politique de Thompson*

Rappelons que KTD maintient les moments d'ordre un et deux du vecteur de paramètres. En supposant que ce vecteur aléatoire suit une distribution gaussienne, nous proposons d'échantillonner un jeu de paramètres en accord avec la distribution estimée, puis d'agir de façon gloutonne respectivement à la Q-fonction résultante. Ce type d'approche à d'abord été proposé par Thompson (1933) dans le cadre d'un problème de bandits, puis a été plus récemment introduit en apprentissage par renforcement



a. ϵ-gloutonne. b. Confiante-gloutonne.

c. Bonus-gloutonne. d. Thompson.

Figure 6. *Politiques.*

dans le cas tabulaire (Dearden *et al.*, 1998; Strens, 2000). La politique de Thompson est définie ici comme suit :

$$\pi(a_{i+1}|s_{i+1}) = \underset{b \in A}{\operatorname{argmax}}\, \hat{Q}_{\xi}(s_{i+1}, b) \text{ avec } \xi \sim \mathcal{N}(\hat{\theta}_{i|i-1}, P_{i|i-1}) \quad [18]$$

Cette politique est illustrée figure 6.d, qui montre la distribution de l'action gloutonne (les paramètres étant des variables aléatoires, l'action gloutonne l'est également). La plus grand probabilité est associée à l'action 3. Cependant, il est à noter qu'une plus grande probabilité est associée à l'action 1 qu'à la 4 : la première action a une Q-valeur estimée plus faible, mais moins certaine.

5.5. *Expérimentation*

Le problème du bandit à N bras est un PDM à un état et N actions. Chaque action a implique une récompense de 1 avec probabilité p_a et une récompense de 0 avec probabilité $1 - p_a$. Pour une action a^* (choisie aléatoirement au début de chaque expérimentation), cette probabilité est choisie égale à $p_{a^*} = 0,6$. Pour toutes les autres actions, la probabilité associée est choisie aléatoirement et uniformément entre 0 et $0,5$: $p_a \sim \mathcal{U}_{[0;0,5]}, \forall a \neq a^*$. Les résultat présentés sont moyennés sur 1000 expérimentations. La performance d'une approche est mesurée comme étant le pourcentage de fois où l'action optimale a été choisie, en fonction du nombre d'interactions entre

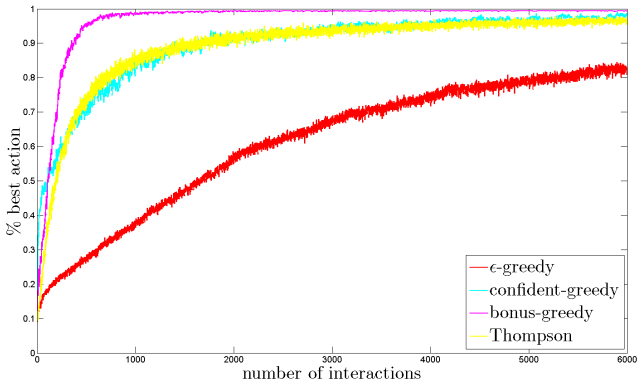


Figure 7. *Résultats du bandit.*

l'agent et le bandit. Une représentation tabulaire est adoptée pour KTD-SARSA, et les paramètres suivants sont utilisés : $N = 10$, $P_{0|0} = 0,1I$, $\theta_{0|0} = I$, $P_{n_i} = 1$, $\epsilon = 0,1$, $\alpha = 0,3$, $\beta_0 = 1$ et $\beta = 10$. Comme le bandit considéré a $N = 10$ bras, une politique aléatoire a une performance de $0,1$. Notons qu'une politique purement gloutonne choisirait systématiquement la première action pour laquelle l'agent aurait observé une récompense positive.

Les résultats de la figure 7 comparent les quatre schémas introduits. La politique ϵ-gloutonne sert de référence et tous les schémas proposés, qui utilisent l'information d'incertitude disponible dans le cadre de KTD, montrent de meilleures performances. La politique de Thompson et la politique confiante-gloutonne présentent des résultats similaires, et les meilleurs résultats sont obtenus par la politique bonus-gloutonne. Bien sûr, ces résultats préliminaires ne permettent pas de conclure sur l'efficacité générale des approches proposées. Cependant, ils tendent à montrer que l'information d'incertitude fournie par KTD a du sens et peut s'avérer utile pour traiter du dilemme entre exploration et exploitation. De plus, la section suivante traite d'un problème du monde réel avec succès, ce qui tend à corroborer ces résultats.

6. Une application au dialogue homme-machine

Le dialogue homme-machine est un pan important du domaine des interactions homme-machine. Dans ce type d'application, un utilisateur interagit avec une machine en utilisant le language naturel, l'objectif étant souvent d'accéder à des systèmes d'information ou de réservation (Raux *et al.*, 2005). Un système de dialogue typique est composé de trois éléments : un module de reconnaissance de la parole, un gestionnaire de dialogue et un module de synthèse de la parole. Le module de reconnaissance de la parole est typiquement composé d'un algorithme de reconnaissance vocale combiné à un analyseur sémantique, et le module de synthèse de parole est composé d'un

générateur de language naturel combiné à un synthétiseur vocal. Le module qui nous intéresse ici est le gestionnaire de dialogue. Il consiste à choisir la prochaine action de dialogue, étant donné la reconnaissance imparfaite de ce que l'utilisateur a dit. C'est un problème de décision séquentielle qui peut être traité par l'apprentissage par renforcement (Levin *et al.*, 1997; Singh *et al.*, 1999; Pietquin *et al.*, 2006).

6.1. *Système de dialogue considéré*

Dans cet article, nous considérons un système de dialogue orienté vers l'obtention d'informations touristiques, proche de celui proposé par Lemon *et al.* (2006). L'objectif du système est de fournir à l'utilisateur des informations concernant un restaurant, plus précisément sa localisation dans la ville (imaginaire), le type de cuisine et l'échelle de prix. Du point de vue de l'agent, l'objectif est donc de remplir les trois *slots* correspondants (localisation, cuisine et prix) afin de proposer le bon restaurant à l'utilisateur. Les actions possibles de dialogue correspondent à :

– demander la valeur d'un slot (par exemple "Quelle type de cuisine recherchez-vous ?"), ce qui totalise 3 actions ;

– confirmer explicitement la valeur d'un slot (par exemple "Pouvez vous confirmer que vous cherchez un restaurant dans le centre ?"), ce qui totalise 3 actions ;

– confirmer implicitement un slot en demandant la valeur d'un autre (par exemple "Vous recherchez un restaurant indien, dans quelle zone de la ville ?"), ce qui totalise 6 actions ;

– clore le dialogue en proposant un restaurant ("Vous rechercher un restaurant français bon marché dans le centre, nous vous proposons...").

Il y a donc au total 13 actions possibles.

Il est nécessaire de modéliser le système de dialogue comme un PDM pour y appliquer un algorithme d'apprentissage par renforcement. Les actions sont celles présentées précédemment. Concernant l'état, il n'est pas réaliste de travailler directement avec la sortie du module de reconnaissance de parole. Nous utilisons le paradigme de l'état d'information (*State Information paradigm*) (Larsson *et al.*, 2001), qui à partir de l'historique des sorties du module de reconnaissance de parole fournit deux valeurs par slot : une probabilité de remplissage (*filling confidence*) et une probabilité de confirmation (*confirmation confidence*). Nous considérons un espace d'état à 3 dimensions, chaque composante étant la moyenne de ces deux probabilités pour un slot donné. Il est également nécessaire de définir une fonction de récompense. Cette dernière est nulle tout le temps, sauf lorsque l'action de clore le dialogue est choisie. Dans ce cas, l'agent reçoit une récompense de $+25$ par slot correctement rempli, de -75 par slot incorrectement rempli et de -300 par slot vide. Le facteur d'actualisation est choisi égal à $\gamma = 0,95$. Etant donné que nous expérimentons des algorithmes ne nécessitant pas de connaître les probabilités de transitions, ces dernières n'ont pas à être définies.

Idéalement, nous souhaiterions implémenter nos algorithmes sur un problème de dialogue réel, cependant les expériences présentées ici sont conduites en utilisant un simulateur d'utilisateur. Le simulateur d'utilisateur est combiné au gestionnaire de dialogue DIPPER (Lemon *et al.*, 2006) pour générer des dialogues.

6.2. *Algorithmes considérés et Q-fonction paramétrée*

Nous considérons trois algorithmes ici : LSPI (Least-Squares Policy Iteration) (Lagoudakis *et al.*, 2003), KTD-SARSA combiné à un schéma d'exploration ϵ-glouton et KTD-SARSA combiné à un schéma d'exploration bonus-glouton. L'algorithme LSPI est hors-ligne (du moins tel que nous le considérons ici), mais il est reconnu comme étant très efficace et il a déjà fait ses preuves dans le domaine du dialogue homme-machine (Li *et al.*, 2009; Chandramohan *et al.*, 2010). Il nous servira de référence en termes de performances. Deux schémas d'explorations combinés à KTD-SARSA sont considérés : la politique ϵ-gloutonne sert de référence, et nous avons choisi la politique bonus-gloutonne car elle présente les meilleurs résultats expérimentaux sur le problème de bandit. L'objectif ici est de montrer que les contributions de cet article s'appliquent également, et avec succès, à un problème bien plus complexe que les benchmarks jouets considérés jusqu'à présent.

L'espace d'état étant continu (un cube de côté 1), il est nécessaire de choisir une architecture paramétrée pour la Q-fonction. Ici, la Q-fonction est représentée par un réseau RBF (Park *et al.*, 1991) par action. Il y a trois noyaux gaussiens par dimension, d'écart-type $0,25$, et ce pour chaque action. Il y a donc au total 351 (c'est-à-dire $3^3 \times 13$) fonctions de base. Tous les algorithmes considérés utilisent la même paramétrisation pour la Q-fonction.

6.3. *Résultats*

Avant de présenter les résultats, les différents paramètres des algorithmes considérés sont donnés. Pour LSPI, il n'y a pas de paramètre à fixer excepté le critère d'arrêt, que nous choisissons ici être un nombre maximum de changement de choix d'action entre les politiques de deux itérations successives. Pour KTD-SARSA, les *a priori* sont $\hat{\theta}_{0|0} = 0$ et $P_{0,0} = I$, et le bruit d'observation est $P_{n_i} = 1$. Pour la politique ϵ-gloutonne, le facteur d'exploration est choisi constant égal à $\epsilon = 0,1$. Pour la politique bonus-gloutonne, les paramètres libres sont choisis égaux à $\beta_0 = 1$ et $\beta = 5$. Les résultats sont présentés figure 8 en échelle semi-logarithmique (performance de la politique apprise en fonction du nombre de transitions utilisées pour l'apprentissage). Pour KTD-SARSA, la performance à l'origine correspond à une politique aléatoire (politique initiale avant apprentissage). Concernant LSPI, les premiers résultats sont donnés à partir de 5 000 transitions observées. En effet, cet algorithme implique d'effectuer des inversions matricielles, avec moins d'échantillons le conditionnement pose des problèmes de stabilité numérique.

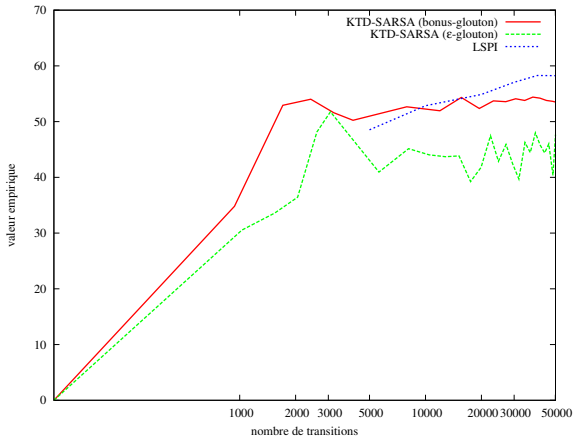


Figure 8. *Résultats sur le problème de dialogue.*

L'algorithme LSPI sert de référence, et présente une performance asymptotique moyenne d'environ 58 (cumul moyen pondéré des récompenses). En pratique, la politique correspondante permet de mener à terme le dialogue en seulement quelques interactions, avec satisfaction de l'utilisateur et avec une bonne robustesse au bruit lié au module de reconnaissance de la parole. KTD-SARSA présente également de bons résultats, quelle que soit le schéma d'exploration considéré, et l'algorithme apprend à interagir de façon satisfaisante avec l'utilisateur en seulement quelques centaines d'épisodes. Si l'on compare les deux schémas d'exploration, on observe que le schéma utilisant l'information d'incertitude (la politique bonus-gloutonne) produit plus rapidement de meilleures politiques qui s'avèrent également être plus stables que la politique ϵ-gloutonne. Il est également à noter que KTD-SARSA combiné avec la politique bonus-gloutonne produit des politiques quasiment aussi bonnes que LSPI (performance d'environ 54 au lieu de 58). Avec une politique ϵ-gloutonne la performance oscille entre 40 et 47. De plus, KTD-SARSA produit déjà des politiques de qualité après un millier de transitions (ce qui correspond à seulement quelques dizaines d'épisodes, les premiers étant très longs) alors qu'il n'y a pas encore suffisamment d'échantillons pour pouvoir utiliser l'algorithme LSPI. Cela montre, sur un problème bien plus complexe que celui du bandit, que l'information d'incertitude fournie par KTD est utile, et que ce cadre de travail général permet de combiner efficacement approximation de la fonction de valeur et dilemme entre exploration et exploitation.

7. Conclusion

Dans cet article, nous avons montré comment une information d'incertitude relative aux valeurs estimées pouvait être dérivée du cadre de travail général des différences temporelles de Kalman. Nous avons également introduit un schéma d'appren-

tissage actif, dans le cadre de l'apprentissage *off-policy*, qui accélère la convergence en tirant les actions relativement à leur incertitude, et nous avons adapté un certain nombre de schémas traitant du dilemme entre exploration et exploitation à ce cadre. Quatre expérimentations ont été proposées. La première illustre sur un simple problème de navigation que l'information d'incertitude estimée à du sens : l'incertitude est d'autant plus faible qu'un endroit a été souvent visité, elle est également plus faible lorsque elle est proche des sources de récompenses. La seconde expérience montre que KTD-Q, qui est un algorithme du second ordre de type itération de la valeur, est efficace en termes d'échantillons. Elle montre également que le schéma d'apprentissage actif proposé accélère effectivement la convergence. Les schémas proposés pour traiter du dilemme entre exploration et exploitation ont été expérimentés sur un problème de bandit, où ils se sont tous montrés plus efficaces qu'une classique politique ϵ-gloutonne. Nous avons également montré sur un problème complexe de gestion de dialogue que l'utilisation de l'information d'incertitude disponible permettait de traiter du dilemme entre exploration et exploitation, ce qui permet d'accélérer, d'améliorer et de rendre plus stable l'apprentissage. Ceci s'avérera particulièrement important lors de la mise en ligne de tels algorithmes sur des systèmes de dialogue homme-machine impliquant de vrais utilisateurs. Les contributions de cet article sont essentiellement heuristiques, et une prochaine étape sera de proposer des garanties formelles concernant les méthodes proposées.

Remerciements

Les auteurs souhaitent remercier la Région Lorraine ainsi que la Communauté Européenne (projet CLASSiC, FP7/2007-2013, subvention 216594) pour leur support financier.

8. Bibliographie

Chandramohan S., Geist M., Pietquin O., « Optimizing Spoken Dialogue Management with Fitted Value Iteration », *Proceedings of the International Conference on Speech Communication and Technologies (Interspeech 2010)*, Makuhari (Japan), September, 2010.

Dearden R., Friedman N., Russell S. J., « Bayesian Q-Learning », *AAAI/IAAI*, p. 761-768, 1998.

Engel Y., Algorithms and Representations for Reinforcement Learning, PhD thesis, Hebrew University, April, 2005.

Geist M., Pietquin O., « Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement », *actes de la conférence francophone sur l'apprentissage automatique (CAP 2010)*, PUG, Clermont-Ferrand (France), p. 101-112, May, 2010a.

Geist M., Pietquin O., « Kalman Temporal Differences », *Journal of Artificial Intelligence Research (JAIR)*, 2010b. A paraître.

Geist M., Pietquin O., Fricout G., « Kalman Temporal Differences: the deterministic case », *Proceedings of the IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, Nashville, TN, USA, April, 2009a.

Geist M., Pietquin O., Fricout G., « Tracking in Reinforcement Learning », *Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009)*, Springer, Bangkok (Thailand), December, 2009b.

Geist M., Pietquin O., Fricout G., « Différences temporelles de Kalman : cas déterministe », *Revue d'Intelligence Artificielle*, 2010c. A paraître.

Julier S. J., Uhlmann J. K., « Unscented filtering and nonlinear estimation », *Proceedings of the IEEE*, vol. 92, n° 3, p. 401-422, 2004.

Kaelbling L. P., *Learning in embedded systems*, MIT Press, 1993.

Kalman R. E., « A New Approach to Linear Filtering and Prediction Problems », *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, n° Series D, p. 35-45, 1960.

Kolter J. Z., Ng A. Y., « Near-Bayesian Exploration in Polynomial Time », *Proceedings of the 26th international conference on Machine learning (ICML 09)*, ACM, New York, NY, USA, 2009.

Lagoudakis M. G., Parr R., « Least-Squares Policy Iteration », *Journal of Machine Learning Research*, vol. 4, p. 1107-1149, 2003.

Larsson S., Traum D., « Information state and dialogue management in the TRINDI dialogue move engine toolkit », *Natural language engineering*, vol. 6, n° 3&4, p. 323-340, 2001.

Lemon O., Georgila K., Henderson J., Stuttle M., « An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system », *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 119-122, 2006.

Levin E., Pieraccini R., Eckert W., « Learning Dialogue Strategies within the Markov Decision Process Framework », *Proc. ASRU'97*, December, 1997.

Li L., Balakrishnan S., Williams J., « Reinforcement Learning for Dialog Management using Least-Squares Policy Iteration and Fast Feature Selection », *Proceedings of the International Conference on Speech Communication and Technologies (InterSpeech'09)*, Brighton (UK), 2009.

Park J., Sandberg I., « Universal approximation using radial-basis-function networks », *Neural computation*, vol. 3, n° 2, p. 246-257, 1991.

Pietquin O., Dutoit T., « A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning », *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, n° 2, p. 589-599, mar, 2006.

Raux A., Langner B., Bohus D., Black A., Eskenazi M., « Let's go public! taking a spoken dialog system to the real world », *Proc. of Interspeech 2005*, 2005.

Sakaguchi Y., Takano M., « Reliability of internal prediction/estimation and its application: I. adaptive action selection reflecting reliability of value function », *Neural Networks*, vol. 17, n° 7, p. 935-952, 2004.

Singh S., Kearns M., Litman D., Walker M., « Reinforcement Learning for Spoken Dialogue Systems », *Proc. NIPS'99*, 1999.

Strehl A. L., Littman M. L., « An Analysis of Model-Based Interval Estimation for Markov Decision Processes », *Journal of Computer and System Sciences*, 2006.

Strens M., « A Bayesian Framework for Reinforcement Learning », *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, p. 943-950, 2000.

Sutton R. S., Barto A. G., *Reinforcement Learning: An Introduction*, The MIT Press, 1996.

Thompson W. R., « On the likelihood that one unknown probability exceeds another in view of two samples », *Biometrika*, n° 25, p. 285-294, 1933.

Yu H., Bertsekas D. P., « Q-Learning Algorithms for Optimal Stopping Based on Least Squares », *Proceedings of European Control Conference*, Kos, Greece, 2007.