

Approximation stochastique en dimension infinie

Application à la valorisation d'options américaines

Kengy Barty¹ Pierre Girardeau^{1,2} Jean-Sébastien Roy¹
Cyrille Strugarek^{1,2,3}

¹EDF Recherche & Développement

²Ecole Nationale Supérieure de Techniques Avancées

³Ecole Nationale des Ponts et Chaussées

15 décembre 2005

Introduction

Parmi les méthodes de valorisation d'options Américaines, il est relativement classique de discrétiser le temps et d'utiliser :

- La programmation dynamique (approximée) [Van Roy and Tsitsiklis, 2001] ;
- La quantification [Bally et al., 2002] ;
- La méthode de régression [Longstaff and Schwartz, 2001].

Ces méthodes requièrent toutes une forme de discrétisation de l'espace d'état, usuellement à travers le choix a priori d'une base de fonctions utilisées pour représenter la valeur de l'option.

Par ce choix, ces méthodes abandonnent dès le début l'optimalité. Mon objectif va être de vous présenter un algorithme alternatif, **nonparamétrique** pour résoudre un certain type de problèmes de programmation dynamique **sans discrétisation a priori**.

Plan de la présentation

- 1 Approximation stochastique
 - Problèmes de point fixe stochastique
 - Approximation stochastique avec des noyaux
 - Hypothèses de convergence
 - Travaux précédents
- 2 Mise en œuvre numérique
 - Application à la valorisation d'option
 - Réduction de la complexité par transformée gaussienne rapide
 - Vitesse de convergence
- 3 Conclusion et extensions
 - Conclusion
 - Extension au cas continu

Problème de point fixe

Typiquement le problème de valorisation d'une option peut être réduit à la solution d'un **problème de point fixe** :

$$\begin{aligned}u(x) &= \mathbb{E}(h(u(\mathbf{Y}), \mathbf{X}) | \mathbf{X} = x) \\ &= H(u)(x)\end{aligned}$$

où H est un opérateur contractant et \mathbf{X} et \mathbf{Y} sont deux variables aléatoires à valeur dans un ensemble d'états S .

Valorisation d'une option Bermudéenne

Description du problème

Similairement à [Van Roy and Tsitsiklis, 2001], nous cherchons à valoriser un put Bermudéen où les dates d'exercices sont restreintes à des dates équispaciées t dans $0, \dots, T$. Le prix du sous-jacent \mathbf{X}_t suit une dynamique de Samuelson discrétisée [Samuelson, 1965, Black and Scholes, 1973] :

$$\ln \frac{\mathbf{X}_{t+1}}{\mathbf{X}_t} = r - \frac{1}{2}\sigma^2 + \sigma\eta_t$$

où (η_t) est un bruit blanc Gaussien de variance unitaire et r est le taux d'intérêt sans risque. Le strike est s , et la valeur intrinsèque de l'option est $g(x) = \max(0, s - x)$ quand le prix est x . Soit $\alpha = e^{-r}$ le taux d'actualisation.

Valorisation d'une option Bermudéenne

Objectif

Soit x_0 le prix à $t = 0$. Notre objectif est d'évaluer la valeur de l'option :

$$\max_{\tau} \mathbb{E} (\alpha^{\tau} g(\mathbf{X}_{\tau}))$$

où τ est un temps d'arrêt adapté à la filtration induite par le processus de prix (\mathbf{X}_t) .

Soit $J_t(x)$ la valeur de l'option au temps t si le prix \mathbf{X}_t est égal à x . Comme l'option doit être exercée avant $T + 1$, nous avons : $J_{T+1}(x) = 0$. Ainsi, pour tout $t \leq T$:

$$J_t(x) = \max(g(x), \alpha \mathbb{E}(J_{t+1}(\mathbf{X}_{t+1}) | \mathbf{X}_t = x))$$

Valorisation d'une option Bermudéenne

fonction Q

Soit $Q_t(x)$ l'espérance du gain à t si on n'exerce pas l'option :

$$Q_t(x) = \alpha \mathbb{E} (J_{t+1}(\mathbf{X}_{t+1}) | \mathbf{X}_t = x)$$

On en déduit l'équation de point fixe :

$$Q_t(x) = \alpha \mathbb{E} (\max (g(\mathbf{X}_{t+1}), Q_{t+1}(\mathbf{X}_{t+1})) | \mathbf{X}_t = x)$$

qui en notant $Q = (Q_t)_t$, peut être exprimée comme $Q = H(Q)$ avec H un opérateur adéquat.

Programmation dynamique approximée

Pour surmonter le problème de la dimension infinie du problème, une solution classique consiste à paramétrer la fonction u , ce qui conduit à la programmation dynamique approximée [Bellman and Dreyfus, 1959]. Soit $A = (a_i)$ le vecteur de paramètres et (f_i) une famille prédéfinie de fonctions de l'état, on cherche u comme combinaison linéaire des (f_i) :

$$u(x) = \sum_i a_i f_i(x)$$

On résout alors le problème de point fixe en dimension finie en cherchant A . C'est rarement optimal, et fréquemment on n'a pas d'indications sur l'erreur.

Quantification

La quantification [Bally et al., 2002] est un sous-cas très intéressant, présentant des caractéristiques d'optimalité dans le cadre de l'approximation d'une espérance.

L'espace d'état S est partitionné en $S = \bigcup_i P_i$ où les P_i sont des cellules de Voronoï autour de points bien choisis selon la distribution de \mathbf{X} . La famille de fonctions utilisée est constituée des fonctions indicatrices $f_i = \mathbf{1}_{P_i}$.

En contre partie d'un effort important pour l'obtention de la partition (P_i) , la méthode est très efficace et des bornes d'erreur sont connues.

Itération de la valeur

Comme pour la plupart des problèmes de point fixe, la résolution est réalisée en appliquant itérativement l'opérateur H à partir d'un point de départ u_0 quelconque, une procédure appelée itération de la valeur [Bellman, 1957] dans le contexte de la programmation dynamique :

$$u_n = H(u_{n-1})$$

Dans la plupart des cas, l'espérance présente dans H ne peut être évaluée que par tirages de Monte-Carlo, ce qui nous amène par exemple à l'algorithme d'approximation stochastique de Robbins-Monro.

Algorithme de Robbins-Monro

Pour x fixé, on réalise une estimation de l'espérance

$$H(u)(x) = \mathbb{E}(h(u(\mathbf{Y}), \mathbf{X}) | \mathbf{X} = x)$$

à travers des tirages aléatoires $(y_n(x))$ de $\mathbf{Y} | \mathbf{X} = x$, et on moyenne récursivement les valeurs obtenues. Notons :

$$\Delta_{n-1}(x, y) = h(u_{n-1}(y), x) - u_{n-1}(x)$$

On obtient l'algorithme d'approximation stochastique de Robbins-Monro [Robbins and Monro, 1951] :

$$u_n(x) = u_{n-1}(x) + \rho_n \Delta_{n-1}(x, y_n(x))$$

avec $\rho_n \downarrow 0$, $\sum_n \rho_n = \infty$ et $\sum_n \rho_n^2 < \infty$. La mise à jour est alors réalisée pour tout x .

Différences temporelles

Remarquons que l'algorithme de Robbins-Monro peut être réécrit sous la forme :

$$u_n(\cdot) = u_{n-1}(\cdot) + \rho_n \mathbb{E}(\Delta_{n-1}(\mathbf{X}, y_n) \delta_{\mathbf{X}}(\cdot))$$

Au lieu de mettre à jour la fonction u pour tous les états x , on peut randomiser l'état mis à jour à chaque itération. Soit (x_n) des tirages aléatoires de l'état \mathbf{X} . On obtient l'algorithme des différences temporelles TD(0) [Sutton, 1988] :

$$u_n(x) = \begin{cases} u_{n-1}(x_n) + \rho_n \Delta_{n-1}(x_n, y_n(x_n)) & \text{si } x = x_n, \\ u_{n-1}(x) & \text{sinon.} \end{cases}$$

Cet algorithme n'est **pas possible à mettre œuvre quand S est continu** et impraticable si S est discret mais de cardinal élevé (discrétisation fine ou d'un espace d'état de grande dimension).

Approximation d'un Dirac

Quand l'espace d'état est continu, l'algorithme TD(0) ne peut être mis en œuvre car les mises à jour sont ponctuelles en x_n . Notre suggestion est *d'approximer* le Dirac $\delta_{x_n}(\cdot)$ en utilisant un noyau de fenêtre $\epsilon_n \downarrow 0$:

$$f(\cdot) = \mathbb{E}(f(\mathbf{X}) \delta_{\mathbf{X}}(\cdot)) = \lim_{n \rightarrow \infty} \mathbb{E}\left(f(\mathbf{X}) \frac{1}{\epsilon_n} K_n(\mathbf{X}, \cdot)\right)$$

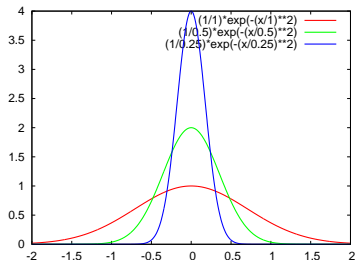


Fig.: Approximations avec des noyaux gaussiens ($\epsilon \in \{1, 0.5, 0.25\}$).

TD(0) avec noyaux

Algorithme

Nous proposons donc l'algorithme de différences temporelles à noyaux suivant :

$$u_n(\cdot) = u_{n-1}(\cdot) + \rho_n \Delta_{n-1}(x_n, y_n(x_n)) \frac{1}{\epsilon_n} K_n(x_n, \cdot)$$

Usuellement $K_n(x_n, \cdot) = K\left(\frac{x_n - \cdot}{\eta_n}\right)$ avec $\epsilon_n = \eta_n^d$ et K un noyau d -dimensionnel. x_n et $y_n(x_n)$ sont tous deux tirés aléatoirement, respectivement selon la loi de \mathbf{X} et de $\mathbf{Y} | \mathbf{X} = x_n$.

Cet algorithme évite une paramétrisation a priori de la fonction u , et nous avons démontré qu'il convergeait [Barty et al., 2005c].

TD(0) avec noyaux

Mise en œuvre

De plus cet algorithme est **facile à mettre en œuvre**, et ne requiert à chaque itération que le stockage de $\alpha_n := \frac{\rho_n}{\epsilon_n} \Delta_{n-1}(x_n, y_n(x_n))$, x_n et de la forme du noyau K_n (usuellement défini à partir de sa fenêtre η_n). Ainsi :

$$u_n(x) = \sum_{i \leq n} \alpha_i K_i(x_i, x)$$

Hypothèses sur les noyaux

On suppose que H est un opérateur contractant, i.e. $\exists \beta \in [0, 1[$
 t.q.

$$\|H(u) - H(u')\| \leq \beta \|u - u'\|$$

avec $\|u\| = \sqrt{\mathbb{E}(\|u(\mathbf{X})\|^2)}$.

Soit $r_n(x) = \mathbb{E}(\Delta_n(\mathbf{X}, \mathbf{Y}) | \mathbf{X} = x)$,

- $\exists b_1 \geq 0$ t.q.
 $\left\| r_{n-1}(\cdot) - \mathbb{E}\left(r_{n-1}(\mathbf{X}) \frac{1}{\epsilon_n} K_n(\mathbf{X}, \cdot)\right) \right\| \leq b_1 \eta_n (1 + \|r_{n-1}(\cdot)\|)$,
 i.e. le biais est contrôlé et asymptotiquement nul,
- $\exists b_2 \geq 0$ t.q.
 $\mathbb{E}\left(\left\| r_{n-1}(\mathbf{X}) \frac{1}{\epsilon_n} K_n(\mathbf{X}, \cdot) \right\|^2\right) \leq b_2 \left(1 + \frac{1}{\epsilon_n} \|r_{n-1}(\cdot)\|^2\right)$, i.e. la
 variance de l'erreur est contrôlée.

Hypothèses sur les pas et la fenêtre

Les suites (ρ_n) , (ϵ_n) et (η_n) doivent être positives et satisfaire :

- $\sum \rho_n = \infty$,
- $\sum \frac{(\rho_n)^2}{\epsilon_n} < \infty$,
- et $\sum b_1 \rho_n \eta_n < \infty$.

Ces hypothèses sont très similaires à celles faites dans d'autres algorithmes d'approximation stochastique avec des estimations biaisées tels que [Kiefer and Wolfowitz, 1952].

Par exemple, si $S = \mathbb{R}^d$, des suites admissibles sont $\rho_n = \frac{1}{n}$, $\epsilon_n = \frac{1}{\sqrt{n}}$ et $\eta_n = \epsilon_n^{\frac{1}{d}}$.

Travaux précédents

De nombreux auteurs [Kushner and Clark, 1978, Kulkarni and Horn, 1996, Delyon, 1996, Chen and White, 1998] et particulièrement [Hiriart-Urruty, 1975] ont prouvé la convergence de ce type d'algorithmes, mais leurs approches sont en général limitées à la dimension finie ce qui les rends inadaptées à notre cas.

En effet, dans le cas de la dimension infinie, il est difficile d'obtenir une estimation sans biais d'une direction de descente. Notre preuve étend la convergence au cas asymptotiquement sans biais requis.

Cette méthode s'inscrit dans le cadre plus général des algorithmes de gradient perturbés [Barty et al., 2005a] et nous l'avons appliqué dans le contexte de l'optimisation stochastique.

Valorisation d'une option Bermudéenne

fonction Q

Soit $Q_t(x)$ l'espérance du gain à t si on n'exerce pas l'option :

$$Q_t(x) = \alpha \mathbb{E}(\max(g(\mathbf{X}_{t+1}), Q_{t+1}(\mathbf{X}_{t+1})) | \mathbf{X}_t = x)$$

qui en notant $Q = (Q_t)_t$, peut être exprimée comme $Q = H(Q)$ avec H un opérateur contractant adéquat.

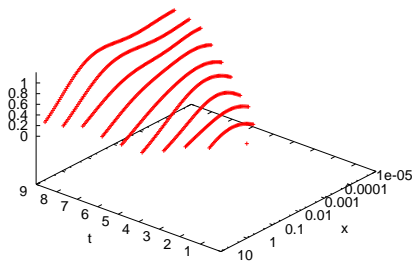
La mise à jour est donnée pour tout t par :

$$Q_t^n(\cdot) = Q_t^{n-1}(\cdot) + \rho_n \Delta_t^{n-1}(x_t^n, x_{t+1}^n) \frac{1}{\epsilon_n} K_n(x_t^n, \cdot)$$

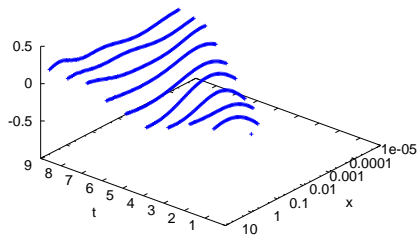
$$\Delta_t^{n-1}(x, x') = \alpha \max(g(x'), Q_{t+1}^n(x')) - Q_t^{n-1}(x)$$

Valorisation d'une option Bermudéenne

100 itérations



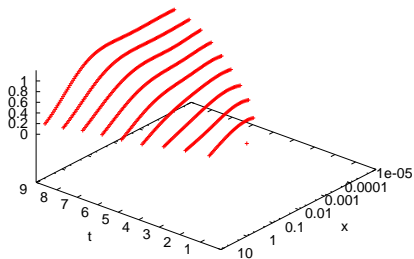
Q^{100}



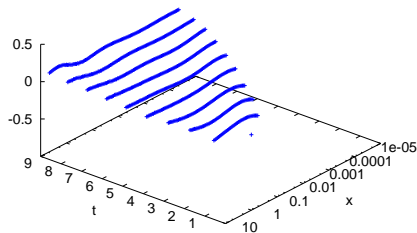
$Q^{100} - Q^*$

Valorisation d'une option Bermudéenne

1000 itérations



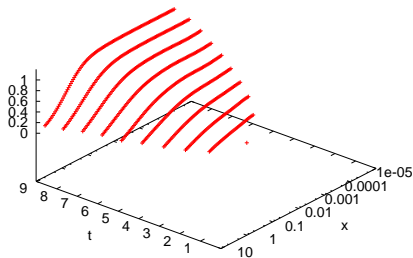
Q^{1000}



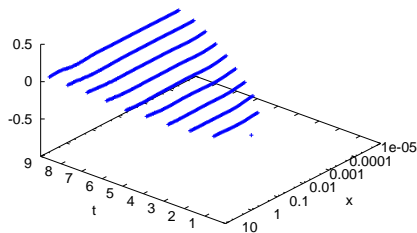
$Q^{1000} - Q^*$

Valorisation d'une option Bermudéenne

10000 itérations



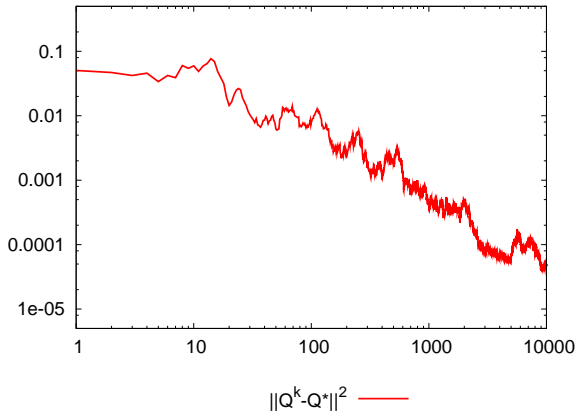
Q^{10000}



$Q^{10000} - Q^*$

Valorisation d'une option Bermudéenne

Vitesse de convergence



Benchmarks

Nous comparons nos résultats pour un problème en dimension 3 avec les résultats de [Barraquand and Martineau, 1995].

- Les trois processus de prix (\mathbf{X}_t^i) , $i = 1, 2, 3$, suivent une dynamique de Samuelson discrétisée, avec un écart-type σ_i ($\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\sigma_3 = 0.5$), une corrélations ρ et une valeur initiale $x_0^i = 40$. Le taux d'intérêt sans risque est $r = \ln(1.05)$ (5% annuellement).
- La valeur intrinsèque de l'option est $g(x) = (s - \max_j (x^j))_+$ dans le cas d'un put et $g(x) = (\max_j (x^j) - s)_+$ dans le cas d'un call, avec s le strike.
- L'horizon de temps est fixé à un mois de 30 jours discrétisé en 10 dates d'exercice ($\Delta t = \frac{3}{365}$).
- On effectue 100 000 itérations soit env. 1 min. de calcul.

Benchmarks

put option				
	$\rho = 0$		$\rho = 0.5$	
s	référence	noyaux	référence	noyaux
35	0.00	0.00	0.00	0.00
40	0.23	0.23	0.49	0.49
45	5.00	5.00	5.00	5.00

call option				
	$\rho = 0$		$\rho = 0.5$	
s	référence	noyaux	référence	noyaux
35	8.59	8.64	7.78	7.84
40	3.84	3.78	3.18	3.21
45	0.90	0.91	0.83	0.83

Complexité

L'itération n nécessitant l'évaluation d'une somme de n noyaux, la complexité en temps de l'algorithme est quadratique en le nombre d'itérations ce qui peut s'avérer une limitation importante. En espace, la complexité est linéaire.

Pour réduire la complexité, il faut soit réduire le nombre de noyaux à sommer, soit accélérer la somme de n noyaux.

La première solution, via des Kd-Tree par exemple, n'est pas directement applicable dans notre cas compte tenu des hypothèses sur la fenêtre. La deuxième solution est rendue possible pour les noyaux gaussiens, via la transformée gaussienne rapide [Yang et al., 2003, Greengard and Strain, 1991], une méthode déjà utilisée par ailleurs pour des problèmes de valorisation [Broadie and Yamamoto, 2003].

Evaluation dans une boule

Nous choisissons de considérer des noyaux gaussiens :

$$K_n(x, y) = e^{-\left(\frac{\|x-y\|}{\epsilon_n}\right)^2}$$

avec x et y dans \mathbb{R}^d . Dans notre algorithme, on fait varier la fenêtre par paliers et l'on s'intéresse donc au cas de l'évaluation d'une somme de noyaux K de fenêtre constante ϵ . L'espace est partitionné en B boules de rayon ϵ centrées en $\{c_b, b = 1, \dots, B\}$.

Soit c_b tel que $\|x - c_b\| \leq \epsilon$ et $\Delta y = \frac{y - c_b}{\epsilon}$ et $\Delta x = \frac{x - c_b}{\epsilon}$, on a :

$$K(x, y) = e^{-\frac{\|y-x\|^2}{\epsilon^2}} = e^{-\|\Delta y\|^2} e^{-\|\Delta x\|^2} e^{2\Delta y \cdot \Delta x}$$

Développement en série entière

$$e^{2\Delta y \cdot \Delta x} = \sum_{j \geq 0} \frac{2^j (\Delta x \cdot \Delta y)^j}{j!} = \sum_{\beta \geq 0} \frac{2^{|\beta|}}{\beta!} \Delta x^\beta \Delta y^\beta$$

car $(\Delta x \cdot \Delta y)^j = \sum_{|\beta|=j} \frac{j!}{\beta!} \Delta x^\beta \Delta y^\beta$,

avec $\beta = (\beta_1, \dots, \beta_d)$, $\beta! = \beta_1! \cdots \beta_d!$ et $\Delta x^\beta = \Delta x_1^{\beta_1} \cdots \Delta x_d^{\beta_d}$.

$\forall i$, soit I_b l'ensemble i tels que c_b est le centre le plus proche de x_i .

$$\sum_i \alpha_i K(x_i, x) = \sum_s \sum_{i \in I_b} \alpha_i K(x_i, x)$$

avec, pour $i \in I_b$, en notant $\Delta x_i = \frac{x_i - c_b}{\epsilon}$,

$$\alpha_i K(x_i, x) = \sum_{\beta > 0} \frac{2^{|\beta|}}{\beta!} \left(\alpha_i e^{-\Delta x_i^2} \Delta x_i^\beta \right) e^{-\left(\frac{\|x - c_b\|}{\epsilon}\right)^2} \left(\frac{x - c_b}{\epsilon}\right)^\beta$$

Evaluation et mise à jour

Soit $C_b(\beta) = \sum_{i \in I_b} \alpha_i e^{-\Delta x_i^2} \Delta x_i^\beta$, alors :

$$\sum_{i \leq n} \alpha_i K(x_i, x) = \sum_b \sum_{\beta \geq 0} \frac{2^{|\beta|}}{\beta!} C_b(\beta) e^{-\left(\frac{\|x - c_b\|}{\epsilon}\right)^2} \left(\frac{x - c_b}{\epsilon}\right)^\beta$$

En pratique on tronque la somme en β et on ne considère que les centres c_b proches de x . Le temps de calcul de la somme des noyaux est alors **constant**.

Pour l'ajout d'un noyau de centre x et de hauteur α , on met à jour C_b , avec b l'indice du centre le plus proche de x :

$$\forall \beta, C_b(\beta) := C_b(\beta) + \alpha \cdot e^{-\Delta x^2} \Delta x^\beta$$

Temps de calcul

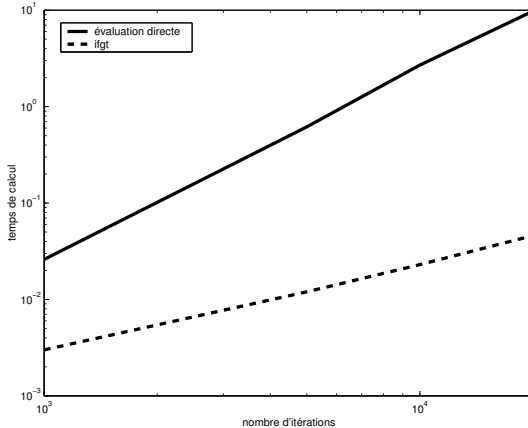


Fig.: Comparaison de temps de calcul avec ou sans FGT, en fonction du nombre d'itérations.

Moyennisation à la Polyak-Juditsky

La vitesse de convergence peut être améliorée par l'utilisation de la moyennisation telle que proposée dans [Polyak and Juditsky, 1992], qui consiste à moyenner, à posteriori les itérés u_k :

$$\hat{u}_n(\cdot) = \frac{1}{n} \sum_{k=1}^n u_k(\cdot)$$

Cette méthode permet d'obtenir des itérés \hat{u}_k de variance plus faible : [Polyak and Juditsky, 1992] démontre un théorème de limite centrale pour cet algorithme, où la variance décroît comme la racine carrée du nombre d'itérations.

Pour des résultats intéressants, il est conseillé de commencer à moyenner à partir d'un certain rang, et de choisir des vitesses de descente des pas plus petites.

Moyennisation à la Polyak-Juditsky

Comparatif

On peut observer son efficacité :

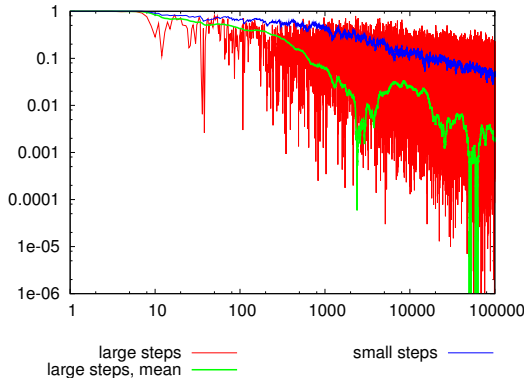


Fig.: Evolution de l'erreur le long des itérations

Heuristique de calcul des pas et des fenêtres

Il est usuel de prendre les pas et la fenêtre sous la forme $\epsilon_n = (\alpha_\epsilon + \beta_\epsilon n)^{\gamma_\epsilon}$, $\rho_n = (\alpha_\rho + \beta_\rho n)^{\gamma_\rho}$ et $\eta_n = \epsilon_n^{\frac{1}{d}}$. Néanmoins, ce choix ne tient pas compte de la distribution des états. On peut par exemple souhaiter que la décroissance des pas et des fenêtres soit fonction du nombre de tirages réalisés dans le voisinage de l'état considéré.

Par exemple, si la densité f de \mathbf{X} est connue, soit (ϵ'_n) une suite adéquate de fenêtres, on peut prendre $\epsilon_n = \epsilon'_{\lfloor n \cdot f(x_n) \rfloor}$. De même pour le pas. Alternativement on peut aussi diviser le pas et la fenêtre par la densité.

Quand la densité est inconnue, on peut utiliser à la place de $n \cdot f(x_n)$ une estimation non paramétrique de la densité, par exemple $\sum_{i < n} \frac{1}{\epsilon_i} K_i(x_i, x_n)$.

Générateurs a discrépance faible

Enfin, une dernière stratégie d'accélération de la convergence est d'utiliser un générateur de nombres aléatoires à discrépance faible par exemple un générateur de Sobol[Sobol, 1967]. Par exemple pour un call en dimension 1 (10 pas de temps, 100 000 itérations, 15 secondes) :

Méthode	Valeur
Exacte	1
Random	1.10
Sobol	1.01

Tab.: Performance via Quasi-MC

Cette méthodologie est à combiner avec des techniques telles que le pont brownien.

Conclusion

Je vous ai présenté une méthode convergente, non paramétrique pour la programmation dynamique qui **ne requiert pas de discrétisation a priori**. La méthode est **simple à mettre en œuvre**, et ces idées peuvent être utilisées pour résoudre des problèmes de programmation stochastique en boucle fermée [Barty et al., 2005b]. De nombreuses extensions sont possibles, notamment étendre les nos résultats au Q-Learning. Nos premières expérimentations montrent que c'est possible.

Enfin, le comportement numérique en grande dimension reste inconnu : nous sommes en train de l'expérimenter.

Définition du problème

Soit (X_t) , pour $t \geq 0$, un processus de Markov en temps continu dans $S \subseteq \mathbb{R}^d$, $\beta > 0$ un facteur d'actualisation et $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$ une fonction de récompense immédiate, tels que la fonction valeur

$$V(x) := \mathbb{E} \left(\int_0^\infty e^{-\beta t} f(X_t) dt \mid X_0 = x \right)$$

existe $\forall x \in S$.

On cherche à approximer V .

Discrétisation en temps

Soit $\delta > 0$ un pas de temps et $\forall x \in S$,

$$V^\delta(x) := \mathbb{E} \left(\sum_{k \in \mathbb{N}} \delta e^{-b\delta k} f(X_{\delta k}) \middle| X_0 = x \right)$$

une approximation en temps discret de $V(x)$.

On suppose, $\forall x \in S$, $V^\delta(x) \rightarrow V(x)$, p.s., quand $\delta \rightarrow 0$.

Soit $h^\delta : \mathbb{R}^p \times S \rightarrow \mathbb{R}^p$, $h^\delta(Y, Z) := \delta f(Z) + e^{-\beta\delta} Y$

et $H^\delta : \mathcal{L}^2(S, \mathbb{R}^p) \rightarrow \mathcal{L}^2(S, \mathbb{R}^p)$,

$$H^\delta(U) := \mathbb{E} \left(h^\delta(U(X_\delta), X_0) \middle| X_0 = x \right)$$

Alors $V^\delta = H^\delta(V^\delta)$: on retrouve un problème de point fixe.

Approximation convergente

On suppose que H^δ est $e^{-\beta\delta}$ contractant pour la norme $\|U\| = \sqrt{\mathbb{E}(\|U(X)\|^2)}$ avec $U \in \mathcal{L}^2(S, \mathbb{R}^p)$. C'est le cas si la distribution de (X_t) est invariante.

Notre algorithme de différences temporelles avec noyaux nous permet d'obtenir une approximation convergente de V^δ .

Nous proposons donc un algorithme de diagonalisation pour générer une suite (V_n) d'approximation de V obtenues en faisant simultanément décroître la fenêtre et le pas de temps δ .

On suppose que l'on dispose d'une suite (δ_n) , t.q.
$$\sum_n \|V^{\delta_n} - V^{\delta_{n+1}}\| < \infty.$$

Proposition

Soit $V_0 \in \mathcal{L}^2(S, \mathbb{R}^p)$, (x_n, x'_n) des tirages de (X_0, X_{δ_n}) , et des suites (δ_n) , (ρ_n) et (ϵ_n) et des noyaux (K_n) tels que définis précédemment. Nous proposons la mise à jour suivante :

$$V_{n+1}(\cdot) = V_n(\cdot) + \rho_n \left[h^{\delta_n} (V_n(x'_n), x_n) - V_n(x_n) \right] \frac{1}{\epsilon_n} K_n(x_n, \cdot)$$

Alors $\forall x \in S$, $V_n(x) \rightarrow V(x)$, p.s., quand $n \rightarrow \infty$.

L'algorithme semble aussi applicable dans le cas d'un horizon fini, les tirages sont alors des trajectoires d'états discrétisées selon δ_n et les noyaux incorporent une dimension supplémentaire, le temps.

Bibliographie I



Bally, V., Pagès, G., and Printems, J. (2002).

First order schemes in the numerical quantization method.
Prépublications du laboratoire de probabilités et modèles aléatoires, (735) :21–41.



Barraquand, J. and Martineau, D. (1995).

Numerical valuation of high dimensional multivariate american securities.
Journal of Financial and Quantitative Analysis, pages vol.30, no 3, pp.383–405.

Bibliographie II



Barty, K., Roy, J.-S., and Strugarek, C. (2005a).

A perturbed gradient algorithm in hilbert spaces.

Optimization Online.

http://www.optimization-online.org/DB_HTML/2005/03/1095.html.



Barty, K., Roy, J.-S., and Strugarek, C. (2005b).

A stochastic gradient type algorithm for closed loop problems.

Stochastic Programming E-Print Series.

<http://hera.rz.hu-berlin.de/speps/>.

Bibliographie III



Barty, K., Roy, J.-S., and Strugarek, C. (2005c).

Temporal difference learning with kernels.

Optimization Online.

http://www.optimization-online.org/DB_HTML/2005/05/1133.html.



Bellman, R. (1957).

Dynamic Programming.

Princeton University Press, New Jersey.






Bellman, R. and Dreyfus, S. (1959).

Functional approximations and dynamic programming.

Math tables and other aides to computation, 13 :247–251.

Bibliographie IV

-  Black, F. and Scholes, M. (1973).
The pricing of options and corporate liabilities.
Journal of Political Economy, 81(3) :637–654.
-  Broadie, M. and Yamamoto, Y. (2003).
Application of the fast gauss transform to option pricing.
Management Science, 49(8) :1071–1088.
-  Chen, X. and White, H. (1998).
Nonparametric learning with feedback.
Journal of Economic Theory, 82 :190–222.

Bibliographie V



Delyon, B. (1996).

General results on the convergence of stochastic algorithms.
IEEE Transactions on Automatic and Control,
41(9) :1245–1255.



Greengard, L. and Strain, J. (1991).

The fast gauss transform.
SIAM Journal on Scientific and Statistical Computing.






Hiriart-Urruty, J.-B. (1975).




Algorithmes de résolution d'équations et d'inéquations
variationnelles.

Z. Wahrscheinlichkeitstheorie verw. Gebiete, 33 :167–186.

Bibliographie VI

-  Kiefer, J. and Wolfowitz, J. (1952).
Stochastic estimation of the maximum of a regression function.
Annals of Mathematical Statistics, 23 :462–466.
-  Kulkarni, S. and Horn, C. (1996).
An alternative proof for convergence of stochastic approximation algorithms.
IEEE Transactions on Automatic Control, 41(3) :419–424.
-  Kushner, H. and Clark, D. (1978).
Stochastic Approximation for Constrained and Unconstrained Systems.
Springer-Verlag.

Bibliographie VII

-  Longstaff, F. A. and Schwartz, E. S. (2001).
Valuing american options by simulation : A simple least squares approach.
Rev. Financial Studies, 14(1) :113–147.
-  Polyak, B. T. and Juditsky, A. B. (1992).
Acceleration of stochastic approximation by averaging.
SIAM Journal on Control and Optimization, 30 :838–355.
-  Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
Annals of Mathematical Statistics, 22 :400–407.

Bibliographie VIII



Samuelson, P. A. (1965).

Rational theory of warrant pricing.

Industrial Management Review, 6 :13–31.



Sobol, I. (1967).

On the distribution of points in a cube and the approximate evaluation of integrals.

Zh. Vychisl. Mat. i Mat. Phys., 7 :784–802.



Sutton, R. S. (1988).

Learning to predict by the method of temporal differences.

Machine Learning, 3 :9–44.

Bibliographie IX



Van Roy, B. and Tsitsiklis, J. N. (2001).

Regression methods for pricing complex american-style options.

IEEE Trans. on Neural Networks, 12(4) :694–703.



Yang, C., Duraiswami, R., Gumerov, N., and Davis, L. (2003).

Improved fast gauss transform and efficient kernel density estimation.

IEEE International Conference on Computer Vision, pages 464–471.