Temporal Difference Learning with Kernels

Theory and Application to Bermudan option pricing

Kengy Barty[2] Jean-Sébastien Roy[1] Cyrille Strugarek[1]

[1]EDF R&D

[2]Ecole Nationale des Ponts et Chaussées

7th july 2005

Introduction

Among the various methods used to price American-style options, a classical one is to discretize time and to use either:

- Approximate dynamic programming [Van Roy and Tsitsiklis, 2001];
- Quantization [Bally et al., 2002];
- The regression method of [Longstaff and Schwartz, 2001].

Beside the time discretization, these methods require some kind of state space discretization, usually through an a priori choice of functional basis used to represent the value of the option.

By choosing an a priori functional basis, these methods usually give up optimality. My objective will be to present an alternative, nonparametric algorithm to solve dynamic programming problems without a priori discretisation.

Presentation outline

Fixed point problem

Typically, the pricing of a Bermudan option can be reduced to the solution of a fixed point problem in $\mathcal{L}^2$ such as:

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

where H is a contraction mapping and $\mathbf{X}$ and $\mathbf{Y}$ are two random variables with values in S.

Such fixed point problems arise for example from dynamic programming equations such as:

$$J(x) = \mathbb{E}\left(g(x, \mathbf{W}) + \alpha J(f(x, \mathbf{W}))\right)$$

where x is the state of the system, $\mathbf{W}$ a random noise, g the immediate cost, f the dynamic, α a discount factor, and J the expected cost we try to evaluate. Here, $\mathbf{Y} = f(\mathbf{X}, \mathbf{W})$.

Approximate Dynamic Programing

To alleviate the infinite dimension problem, a classical solution consists in parametrizing function u, which leads to approximate dynamic programming [Bellman and Dreyfus, 1959]. Let $A = (a_i)$ a parameter vector and (f_i) a predefined family of functions of the state, we search u among the linear combinations of (f_i):

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

The resolution is then performed by solving a finite dimensional fixed point problem on A.

It is usually not optimal, and we usually have no idea of the error.

Quantization [Bally et al., 2002] is a subcase where the state space S is discretized into a partition $S = \bigcup_i P_i$ and $f_i = \mathbf{1}_{P_i}$.

Value iteration

As with most fixed point problems, resolution is performed by iteratively applying the operator H from any starting point u_0, a procedure called value iteration [Bellman, 1957] in the dynamic programming context:

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

In most cases, the expectation in H can only be estimated through Monte-Carlo simulation, which leads, for example, to the Robbins-Monro stochastic approximation algorithm.

Robbins-Monro algorithm

For a fixed x, we perform an estimation of the expectation

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

through random samples $(y_n(x))$ of $\mathbf{Y}$, and recursively average the values obtained. Let:

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

We obtain the Robbins-Monro stochastic approximation algorithm [Robbins and Monro, 1951]:

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

with $\rho_n \downarrow 0$, $\sum_n \rho_n = \infty$ and $\sum_n \rho_n^2 < \infty$. The update is then performed on all x.

Temporal differences

Remark that the Robbins-Monro algorithm can be rewritten as:

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

Instead of updating the u function for all states x, we could randomize the updated state at each iteration. Let (x_n) be random draws of the state $\mathbf{X}$. We obtain the TD(0) temporal difference algorithm [Sutton, 1988]:

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

This algorithm is not implementable when S is continuous and not practical when S is discrete with a large cardinal number (as with fine discretization of a high dimensional state space).

Approximation of a Dirac

When the state space is continuous, the TD(0) algorithm cannot be implemented since the updates are pointwise in x_n. We suggest to *approximate* the Dirac $\delta_{x_n}(\cdot)$ using a kernel of bandwidth $\epsilon_n \downarrow 0$:

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

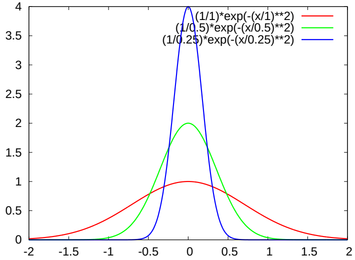


Figure: Approximations with Gaussian kernels ($\epsilon \in \{1, 0.5, 0.25\}$).

TD(0) with kernels

We therefore propose the following temporal difference learning with kernels algorithm:

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

Usually $K_n(x_n, \cdot) = K\left(\frac{x_n - \cdot}{\eta_n}\right)$ with $\epsilon_n = \eta_n^d$ and K a d-dim. kernel.

This algorithm avoid the a priori parametrization of the function u, and we proved this algorithm converge in [Barty et al., 2005c].

Moreover it is easily implementable, requiring only at each iteration the storage of the vector $\alpha_n := \frac{\rho_n}{\epsilon_n} \Delta_{n-1}(x_n, y_n(x_n))$, the vector x_n and the shape of K_n (usually defined by its bandwidth ϵ_n). so that:

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

Hypotheses for on kernels

We assume H is a contraction mapping, i.e. $\exists \beta \in [0,1[$ s.t.

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

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

Let $r_n(x) = \mathbb{E}\left(\Delta_n(\mathbf{X},\mathbf{Y}) \middle| \mathbf{X} = x\right)$,

- $\exists b_1 \geq 0$ s.t. $\left\| r_{n-1}(\cdot) - \mathbb{E}\left(r_{n-1}(\mathbf{X}) \frac{1}{\epsilon_n} K_n(\mathbf{X},\cdot)\right)\right\| \leq b_1 \eta_n \left(1 + \|r_{n-1}(\cdot)\|\right)$, i.e. the bias is controlled and asymptotically zero,
- $\exists b_2 \geq 0$ s.t. $\mathbb{E}\left(\left\| r_{n-1}(\mathbf{X}) \frac{1}{\epsilon_n} K_n(\mathbf{X},\cdot)\right\|^2 \right) \leq b_2 \left(1 + \frac{1}{\epsilon_n} \|r_{n-1}(\cdot)\|^2\right)$, i.e. the variance of the error is controlled.

Hypotheses on the steps and the bandwidth

The sequences (ρ_n), (ϵ_n) and (η_n) must be positive and satisfy:

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

These hypotheses are quite similar to those found in other stochastic approximation algorithms with biased estimates such as in [Kiefer and Wolfowitz, 1952].

For example, if $S = \mathbb{R}^d$, suitable sequences are $\rho_n = \frac{1}{n}$, $\epsilon_n = \frac{1}{\sqrt{n}}$ and $\eta_n = \epsilon_n^{\frac{1}{d}}$.

Previous works

Many authors [Kushner and Clark, 1978, Kulkarni and Horn, 1996, Delyon, 1996, Chen and White, 1998] and especially [Hiriart-Urruty, 1975] have proved the convergence of this kind of algorithms, but these approaches have limitations that make them difficult to use in our case:

- Either they are restricted to the finite dimensional case;
- Or they cannot cope with constraints on u.

But the main limitation in our case is that in an infinite dimensional space, it is difficult to obtain an implementable unbiased estimate of a descent direction.

A more general, perturbed gradient framework for the previous theorem can be found in [Barty et al., 2005a].

Bermudan option pricing

Problem description

Similarly to [Van Roy and Tsitsiklis, 2001], we try to price a Bermudan put option where exercise dates are restricted to equispaced dates t in $0, \ldots, T$. The underlying price $\mathbf{X}_t$ follow a discretized Black-Scholes [Black and Scholes, 1973] dynamic:

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

where $(\boldsymbol{\eta}_t)$ is a Gaussian white noise of variance unity and r is the risk free interest rate. The strike is s, and the intrinsic option price is $g(x) = \max(0, s - x)$ when the price is x. Let the discount factor $\alpha = e^{-r}$.

Bermudan option pricing

Objective

Let x_0 the price at $t = 0$. Our objective is to evaluate the value of the option:

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

where τ is taken among the stopping times adapted to the filtration induced by the price process $(\mathbf{X}_t)$.

Let $J_t(x)$ the option value at time t if the price $\mathbf{X}_t$ is equal to x. Since the option must be exercised before $T+1$, we have: $J_{T+1}(x) = 0$. Therefore, for all $t \leq T$:

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

Bermudan option pricing
Q function

Let $Q_t(x)$ the expected gain at t if we do not exercise the option:

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

We derive the fixed point equation:

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

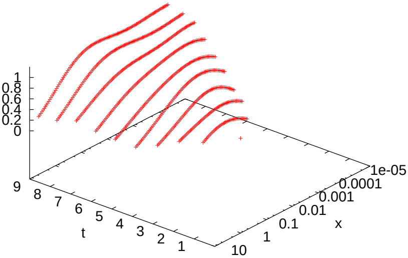
which by letting $Q = (Q_t)_t$, can be expressed as $Q = H(Q)$ with H a suitable contraction mapping.

The update is given for all t by:

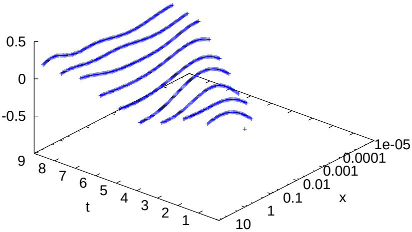
$$Q_t^n(\cdot) = Q_t^{n-1}(\cdot) + \rho_n \Delta_t^{n-1}\left(x_t^n, x_{t+1}^n\right) \frac{1}{\epsilon_n} K_n\left(x_t^n, \cdot\right)$$

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

Bermudan option pricing
100 iterates



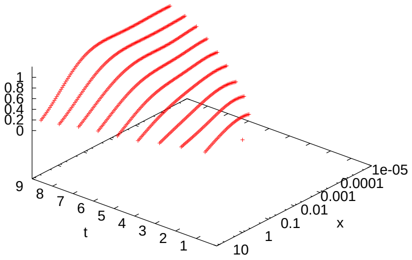
Q^{100}



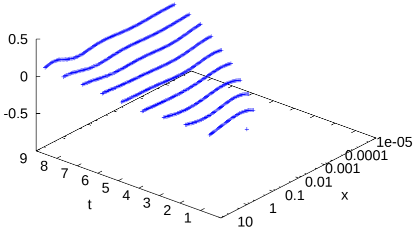
$Q^{100} - Q^*$

Bermudan option pricing

1000 iterates



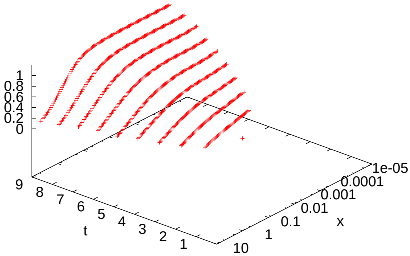
Q^{1000}



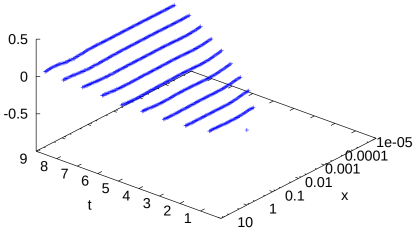
$Q^{1000} - Q^{*}$

Bermudan option pricing

10000 iterates

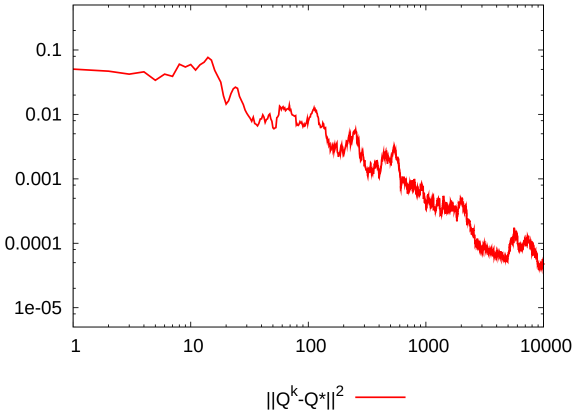


Q^{10000}



$Q^{10000} - Q^*$

Bermudan option pricing
Convergence speed



Conclusion

I have presented a convergent nonparametric method for dynamic programming that does not require an a priori discretization. The method is easy to implement, and the ideas can be used to solve closed loop stochastic programming problems [Barty et al., 2005b]. Many extensions are possible, notably:

- Accelerate the convergence using larger step sizes and averaging [Polyak and Juditsky, 1992];
- Define good heuristics for the window and the steps;
- Extend our results to Q-Learning. Our first experiments shows it should be possible.

More importantly, the numerical behavior of the algorithm in high dimensional state space is still unknown: we plan to experiment this soon.

Bibliography 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.

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.

Bibliography II

Barty, K., Roy, J.-S., and Strugarek, C. (2005b).
A stochastic gradient type algorithm for closed loop problems.
SPEPS.
`http://www.speps.info/`.

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.

Bibliography III

Bellman, R. and Dreyfus, S. (1959).
Functional approximations and dynamic programming.
Math tables and other aides to computation, 13:247–251.

Black, F. and Scholes, M. (1973).
The pricing of options and corporate liabilities.
Journal of Political Economy, 81(3):637–654.

Chen, X. and White, H. (1998).
Nonparametric learning with feedback.
Journal of Economic Theory, 82:190–222.

Bibliography IV

Delyon, B. (1996).
General results on the convergence of stochastic algorithms.
IEEE Transactions on Automatic and Control, 41(9):1245–1255.

Hiriart-Urruty, J.-B. (1975).
Algorithmes de résolution d'équations et d'inéquations variationnelles.
Z. Wahrscheinlichkeitstheorie verw. Gebiete, 33:167–186.

Kiefer, J. and Wolfowitz, J. (1952).
Stochastic estimation of the maximum of a regression function.
Annals of Mathematical Statistics, 23:462–466.

Bibliography V

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.

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.

Bibliography VI

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.

Sutton, R. S. (1988).
Learning to predict by the method of temporal differences.
Machine Learning, 3:9–44.

Bibliography VII

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.