KERNEL-BASED STOCHASTIC GRADIENT ALGORITHMS FOR OPTION PRICING

KENGY BARTY, PIERRE GIRARDEAU, JEAN-SÉBASTIEN ROY, AND CYRILLE STRUGAREK

Abstract. To overcome the curse of dimensionality usually encountered in dynamic programming problems with high-dimensional state spaces such as in option pricing problems, most approaches so far have proposed some sort of discretization, either on the state space or on the underlying functional space, for example by approximating the value function as a linear combination of a predefined finite functional basis.

Because of this discretization, these methods give up optimality from the beginning. We introduce an alternative approach based on functional gradient descent and using an infinite kernel basis, that preserves optimality under very light conditions while being implementable in practice.

We present a temporal difference scheme adapted to this approach, applied to Bermudean option pricing, and propose a comprehensive methodology on how to efficiently implement this algorithm. We also compare our results to numerous existing methods, and provide empirical statistical results.

1. Extended abstract

Since the early 70s, the field of mathematical finance and, more specifically, the option valuation theory, has been studied very thoroughly. Numerous techniques exist to price portfolios relying on multiple assets, and most of them rely on a parametrization : either they discretize the state space (see e.g. the binomial or multinomial approaches in [Ross, 1976], [Cox et al., 1979], the stochastic mesh method in [Broadie and Glasserman, 1997], or the quantization algorithm in [Bally et al., 2005]), or they consider a truncated basis of L^2 for the computation of the conditional expectation (see e.g. [Longstaff and Schwartz, 2001] and [Tsitsiklis and Roy, 1999]) and try to optimize the coefficients of the linear combination of this basis, in a least squares approach.

We propose an alternative approach for these problems that is non-parametric and avoids any discretization of space. Our method relies on the convolution by kernels, typically Gaussians, and has been introduced and its convergence proved in [Barty et al., 2005]. The functional subspace of L^2 that we consider grows along with the number of iterations, and the coefficients of kernel functions are not optimized by regression, but obtained once for all by a single temporal difference computation.

Our method can be introduced as an extension of the Robbins-Monro stochastic approximation algorithm [Robbins and Monro, 1951] and, more recently, the temporal difference-based algorithm TD(0) [Sutton, 1988]. Indeed, it extends those methods to the case of infinite-dimensional stochastic approximation.

In order to speed up the algorithm, we make use of the Improved Fast Gauss Transform to approximate sums of Gaussian kernels. This technique has already been used in mathematical finance, see for instance its applying to the multinomial and the stochastic mesh methods in [Broadie and Yamamoto, 2003] for the case of discrete-time American-style options. This technique enables us to ensure our algorithm perform a constant amount of computation at each iteration which make it suitable for a very large number of iterations.

Several other techniques are presented and used to improve the rate of convergence of the algorithm, such as averaging of the iterates[Polyak and Juditsky, 1992] and making use of low discrepancy random number generators combined with a brownian bridge. The performance of

Date: January 30, 2006.

our implementation is assessed through numerous numerical experiments and comparisons against most well-known pricing algorithms.

2. Overview of our algorithm

2.1. **Problem.** We here consider Bermudean options with exercise dates belonging to a finite subset of $[0,T]$. The price of a Bermudean option with evenly spaced exercise dates $\{t_0, t_1, \ldots, t_N = T\}$, maturity T, discount factor $\alpha = B(t_1, t_2)$ and initial stock price x is given by

$$J_0(x) = \max_{\tau \in \{t_0, t_1, \ldots, t_N\}} \mathbb{E}\left[\alpha^\tau g(X_\tau) \mid X_{t_0} = x\right], \tag{2.1}$$

where the price process X is a Markov chain $\left\{X_{t_j} \in S, 0 \leq j \leq N\right\}$ with $S = \mathbb{R}^d$ the multi-dimensional state space. The intrinsic value of the option is $g : S \to \mathbb{R}^+$.

Let us now introduce the dynamic programming counterpart of (2.1).

$$J_{N+1}(x) = 0, \tag{2.2a}$$

$$J_j(x) = \max\left(g(x), \alpha\mathbb{E}\left[J_{j+1}(X_{t_{j+1}}) \mid X_{t_j} = x\right]\right), \ \forall\, 0 \leq j \leq N. \tag{2.2b}$$

We can equivalently write the equations (2.2) with the so-called Q-functions :

$$Q_j(x) = \alpha\mathbb{E}\left[J_{j+1}(X_{t_{j+1}}) \mid X_{t_j} = x\right],$$

i.e. the expected payoff if we do not exercise the option. Hence it comes:

$$\forall 0 \leq j \leq N, \quad J_j(x) = \max\left(g(x), Q_j(x)\right). \tag{2.3}$$

Equation (2.2) now reads :

$$Q_N(x) = 0 \tag{2.4a}$$

$$Q_j(x) = \alpha\mathbb{E}\left[\max\left(g\left(X_{t_{j+1}}\right), Q_{j+1}\left(X_{t_{j+1}}\right)\right) \mid X_{t_j} = x\right], \ \forall\, 0 \leq j \leq N-1. \tag{2.4b}$$

Equations (2.4) can of course be rewritten as an infinite horizon stochastic dynamic program, by letting the state be defined by $(j,x) \in \{0 \leq j \leq N\} \times S$, and defining the associated nonhomogeneous Markov chain. Finally, the function $\hat{Q} : \{0 \leq j \leq N\} \times S \to \mathbb{R}^N$ defined by $\hat{Q}(j,x) = Q_j(x)$ verifies the following fixed point equation:

$$\hat{Q}(j,x) = \alpha\mathbb{E}\left[\max\left(\hat{g}\left(\hat{V}\right), \hat{Q}\left(\hat{V}\right)\right) \mid V = (j,x)\right], \tag{2.5}$$

with

$$\mathcal{L}\left(\hat{V}\middle| V = (j,x)\right) = \begin{cases} \left(j+1, \mathcal{L}\left(X_{t_{j+1}} | X_{t_j} = x\right)\right), & \text{if } j \leq N, \\ (j+1, V^\infty), & \text{else, with } V^\infty = +\infty, \end{cases}$$

and for all $x \in S$, $\hat{g}(j,x) = g(x)$, and $\hat{g}(j, V^\infty) = 0$.

The algorithm builds sequences Q^k that converge to Q^*. Then we come back to J^* using equation (2.3).

2.2. **Algorithm.** We use the following notation convention: we note $x_{t_j}^k$ the k-th draw (realization) of the random process X_{t_j}. We propose the following algorithm:

Algorithm 2.1. Step -1 : initialize $Q_j^0(\cdot) = 0$ for all $0 \leq j \leq N-1$,
Step $k \geq 0$:

- Draw $x_{t_j}^k, \forall 1 \leq j \leq N$ independently from the past drawings, starting from $x_{t_0}^k = x$ and with respect to the law of the Markov chain X ;

- Update:

$$\left\{\begin{array}{rcl} Q_N^{k+1}(\cdot) & = & 0, \\ Q_{N-1}^{k+1}(\cdot) & = & Q_{N-1}^k(\cdot) + \rho_{N-1}^k\ \Delta_{N-1}^k\ K_{N-1}^k(x_{t_{N-1}}^k, \cdot), \\ & \vdots & \\ Q_j^{k+1}(\cdot) & = & Q_j^k(\cdot) + \rho_j^k\ \Delta_j^k\ K_j^k(x_{t_j}^k, \cdot), \\ & \vdots & \\ Q_0^{k+1}(\cdot) & = & Q_0^k(\cdot) + \rho_0^k\ \Delta_0^k\ K_0^k(x_{t_0}^k, \cdot). \end{array}\right.$$

where

$$\Delta_j^k = \alpha \max\left(g\left(x_{t_{j+1}}^k\right), Q_{t_{j+1}}^{k+1}\left(x_{t_{j+1}}^k\right)\right) - Q_j^k(x_{t_j}^k).$$

where K-functions are chosen kernels, i.e. bounded mappings $S \times S \to \mathbb{R}$, with $K(x, \cdot)$ non-zero on a subset of S centered on x. A typical choice of these kernels is Gaussian functions:

$$K^k(x, y) = exp\left\{\left(\frac{x-y}{\varepsilon_k}\right)^2\right\}.$$

where ε_k decreases to zero when k goes to infinity.

Let us note $(Q^k) = (Q_j^k)_{0 \le j \le N}$. We proved that the sequence $(Q^k)_{j \in \mathbb{N}}$ strongly converges to Q^*. Steps ρ_j^k and radius of the kernels must be decreasing real sequences, that satisfy the following relations:

$$\sum_{k \in \mathbb{N}} \rho_j^k \varepsilon_k = \infty, \quad \sum_{k \in \mathbb{N}} \left(\rho_j^k\right)^2 \varepsilon_k < \infty, \quad \sum_{k \in \mathbb{N}} \rho_j^k \varepsilon_k \eta_k < \infty,$$

where $\eta_k = (\varepsilon_k)^{\frac{1}{d}}$ and d is the dimension of the space.

As one can see, we are working directly in the infinite dimension state space to which the solution belongs. In spite of the infinite dimension, this method remains numerically tractable since in order to compute Q^{k+1} one only needs to keep in memory $\left\{Q^k, \Delta^k, X^{k+1}\right\}$:

$$Q_j^k(\cdot) = \sum_{i=0}^{k} \rho_j^i\ \Delta_j^i\ K_j^i(x_{t_j}^i, \cdot) + Q_j^0(\cdot), \quad \forall j \le N.$$

References

[Bally et al., 2005] Bally, V., Pagès, G., and Printems, J. (2005). A quantization method for pricing and hedging multi-dimensional american style options. *Mathematical Finance*, pages 15(1), 119–168.

[Barty et al., 2005] Barty, K., Roy, J.-S., and Strugarek, C. (2005). A perturbed gradient algorithm in Hilbert spaces. *Optimization Online*. http://www.optimization-online.org/DB_HTML/2005/03/1095.html.

[Broadie and Glasserman, 1997] Broadie, M. and Glasserman, P. (1997). A stochastic mesh method for pricing high-dimensional american options. *Working Paper, Columbia University*.

[Broadie and Yamamoto, 2003] Broadie, M. and Yamamoto, Y. (2003). Application of the fast gauss transform to option pricing. *Management Science*, 49(8):1071–1088.

[Cox et al., 1979] Cox, J., Ross, S., and Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics*, pages vol 7:229–263.

[Longstaff and Schwartz, 2001] 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 and Juditsky, 1992] Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, pages 838 – 855.

[Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

[Ross, 1976] Ross, S. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, pages 13:341–360.

[Sutton, 1988] Sutton, R. (1988). Learning to predict by the method of temporal difference. *IEEE Trans. Autom. Control*, 37:332–341.

[Tsitsiklis and Roy, 1999] Tsitsiklis, J. and Roy, B. V. (1999). Optimal stopping for markov processes: Hilbert space theory, approximation algorithm and an application to pricing high-dimensional financial derivatives. *IEEE Trans. Autom. Controll*, pages 44:1840–1851.