American option pricing with machine learning: An extension of the Longstaff-Schwartz method

Jingying Lin†
Caio Almeida‡

Abstract  Pricing American options accurately is of great theoretical and practical importance. We propose using machine learning methods, including support vector regression and classification and regression trees. These more advanced techniques extend the traditional Longstaff-Schwartz approach, replacing the OLS regression step in the Monte Carlo simulation. We apply our approach to both simulated data and market data from the S&P 500 Index option market in 2019. Our results suggest that support vector regression can be an alternative to the existing OLS-based pricing method, requiring fewer simulations and reducing the vulnerability to misspecification of basis functions.

Keywords: Option pricing; Machine learning; Monte Carlo simulation; Support vector regression; Classification and regression trees.

JEL Code: C52, G13.

1. Introduction

Pricing American options has been a challenging problem in the financial industry from the perspective of both academic research and practical applications. Investors in American options can exercise the option at any time before maturity, so pricing is a typical optimal stopping problem. This has attracted a lot of research attention within academia. American options are also in high demand, because of this flexibility of exercising the option. Hence, the majority of the options market consists of American options. For example, if investors trade options on individual stocks, like Apple, General Electric, and Google, they are most likely trading American-style options. In contrast, European-style options are less common, typically traded on stock indexes, such as the S&P 500 Index, the NASDAQ Index and the Russell 2000 Index; or on currency pairs, like the U.S. dollar vs. the euro and the U.S. dollar vs. the Japanese yen. Therefore, research on pricing American options also has great practical value.

Various methods proposed to tackle the American option pricing problem include the binomial tree by Cox et al. (1979), finite difference by Schwartz...
(1977), Monte Carlo simulation by Boyle (1977), and reinforcement learning by Li et al. (2009). However, when the option value depends on several underlying stochastic processes, either because of multiple underlying assets, or because of multiple stochastic factors like volatility and risk-free rate, many pricing methodologies become inadequate. For example, for binomial tree and finite difference methods, computational time increases exponentially with the number of underlying stochastic factors, as discussed by Boyle et al. (1989). In contrast, in the Monte Carlo simulation-based method, the computational burden remains manageable as the number of underlying stochastic factors increases. Besides, as investors need to determine whether to exercise their American options early, estimating the continuation value, i.e., the value of holding the option rather than exercising it right now, needs to be factored in. The idea of combining Monte Carlo methods with approximations of the continuation value in backward induction schemes goes back to Carriere (1996). He estimates the continuation value function at each possible exercise date by minimizing squared errors using smooth splines and local regressions, and then compares the immediate payoff of exercising options with the continuation value, to determine the exercise policy. This idea was later popularized by Longstaff and Schwartz (2001), who uses the basis of polynomials to numerically estimate this conditional expectation. The rationale behind Monte Carlo-based approaches is that the convergence speed of the proposed schemes does not depend a priori on the dimension of the problem, despite the importance of the dimension, at finite distance, through the variance of the estimation error or the complexity of the algorithm, as outlined by Bouchard and Warin (2012).

In this paper, we extend the method of Longstaff and Schwartz (Longstaff and Schwartz, 2001). We are motivated by the fact that while ordinary least squares (OLS) regression is easy and efficient to implement in the regression-based Monte Carlo method for option pricing, it is subject to overfitting and misspecification of the functions used as regressors. Moreover, we may need large samples, i.e., many simulations, to obtain satisfactory fitting results if we use the OLS regression. Considering that some machine learning methods may outperform in small datasets and capture the complicated relationship between explanatory variables, we propose two alternatives: support vector regression (SVR) and classification and regression trees (CART). SVR is able to learn more effectively when the number of simulated paths is relatively small, and is robust to outliers. CART is designed specifically to deal with small datasets and assumes no prior statistical attributes of the problem.

Empirical results show that SVR outperforms other procedures in the simulated data environment in terms of root mean squared error (RMSE).
This finding is robust across different moneyness, which showcases that our method can be a good alternative to traditional OLS-based option pricing methods. However, when it comes to real market data, machine learning techniques do not continue to outperform others. This probably occurs because slightly dissimilar assets used to calibrate asset dynamics generate independent paths in the simulation, and for writing American options, and the limited hyperparameter tuning.

The paper is organized as follows. In section 2 we review the most relevant background and references on option pricing and machine learning techniques, and formulate our model accordingly. Section 3 discusses the results of our empirical analysis using both simulated data and market data from the S&P 500 Index. Finally, we draw conclusions and propose ideas for potential improvement and future development in section 4.

2. Background and literature review

2.1 American option pricing

In a complete and arbitrage-free market, we can represent the price of a derivative security as the expected value with respect to the martingale measure, as discussed by Karatzas and Shreve (1998). We assume the financial market has an underlying complete probability space \((\Omega, \mathcal{F}, \mathbb{P})\), a filtration \(\{\mathcal{F}_t\}_{0 \leq t \leq T}\) with finite time horizon \([0, T]\) and \(\mathcal{F}_T = \mathcal{F}\). Under the no-arbitrage condition, an equivalent martingale measure \(\mathbb{Q} \sim \mathbb{P}\) exists, under which the value of each security is the expected future payoff discounted with the risk-free rate.

The payoff of an American option is defined as a deterministic measurable function \(h : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}\) restricted to the square-integrable space \(\mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})\), which means the seller pays \(h(t, x)\) at time \(t\) if the option is exercised at time \(t\) and the value of the underlying asset is \(x\). For call options, \(h(t, S_t) = (S_t - K)^+\), while for put options, \(h(t, S_t) = (K - S_t)^+\), where \(K\) is the strike price and \(S_t\) is the stock price at time \(t\).

Thus, the price of an American option with maturity \(T\) is given by the value of the optimal stopping problem

\[
V_t = \sup_{\tau \in \mathcal{T}_{[t,T]}} \mathbb{E}[d(t, \tau)h(\tau, S_\tau)|\mathcal{F}_t],
\]

where \(\{S_t\}_{0 \leq t \leq T}\) is a stochastic process, \(\mathcal{T}_{[t,T]}\) is the set of stopping times with values in \([t, T]\), and \(d(s, t)\) is a non-negative \(\mathcal{F}\)-measurable discount factor satisfying \(d(s, t) = d(s, \tau)d(\tau, t)\) for any \(s \leq \tau \leq t\). For simplicity, we write

\[
f_{t,\tau}(S_\tau) = d(t, \tau)h(\tau, S_\tau),
\]
which represents the discounted payoff.

Many choices for the stochastic process \( \{S_t\}_{0\leq t\leq T} \) exist. One of the widely used dynamics is the one-dimensional geometric Brownian motion \( dS_t = \mu S_t \, dt + \sigma S_t \, dW_t \), as in the well-known Black-Scholes setting (Black and M., 1973). Here, we consider more general and realistic models which incorporate stochastic volatility. Specifically, we use the two-dimensional Heston model, first introduced by Heston (1993). Let \( S_t \) be the value of the underlying asset under the martingale measure with variance \( v_t \) that follows a CIR process:

\[
\begin{align*}
    dS_t &= \mu S_t \, dt + \sqrt{v_t} S_t \, dW^{(1)}_t, \\
    dv_t &= \kappa (\theta - v_t) \, dt + \xi \sqrt{v_t} \, dW^{(2)}_t, \\
    \langle dW^{(1)}_t, dW^{(2)}_t \rangle &= \rho dt,
\end{align*}
\]

where \( W^{(1)}_t \) and \( W^{(2)}_t \) are correlated Brownian motions.

In terms of option pricing, the most notable result is the Black-Scholes formula (Black and M., 1973). This gives the closed-form solution to price a European option when the underlying asset follows geometric Brownian motion. However, no analytical solution to American option pricing exists, even in the Black-Scholes framework, except for calls on non-dividend-paying stocks, whose prices are equal to their European counterparts. Likewise, under the Heston model, analytical solutions to the European call and European put prices exist. We express a European call option price as follows:

\[
C_0 = S_0 \cdot \Pi_1 - e^{rT} K \cdot \Pi_2,
\]

and give probabilities \( \Pi_1 \) and \( \Pi_2 \) by

\[
\begin{align*}
    \Pi_1 &= \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re \left[ \frac{e^{-iw \ln K} \psi_{\ln S_T}(w - i)}{iw \psi_{\ln S_T}(-i)} \right] \, dw, \\
    \Pi_2 &= \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re \left[ \frac{e^{-iw \ln K} \psi_{\ln S_T}(w)}{iw} \right] \, dw,
\end{align*}
\]
where the characteristic function $\Psi$ is

\[
\Psi_{\ln S_T}(w) = e^{C(t,w)\theta + D(t,w)v_0 + iw\ln(S_0e^{rT})},
\]

\[
C(t,w) = \kappa \left[ r_- t - \frac{2}{\xi^2} \ln \left( \frac{1 - g \cdot e^{-h\Delta t}}{1 - g} \right) \right],
\]

\[
D(t,w) = r_- \frac{1 - e^{-h\Delta t}}{1 - g \cdot e^{-h\Delta t}};
\]

\[
r_{\pm} = \frac{\beta \pm h}{\xi^2},
\]

\[
h = \sqrt{\beta^2 - 4\alpha\gamma},
\]

\[
g = \frac{r_-}{r_+},
\]

\[
\alpha = -\frac{w^2}{2} - \frac{iw}{2},
\]

\[
\beta = \alpha - \rho \xi iw;
\]

\[
\gamma = \frac{\xi^2}{2}.
\]

To solve for $C_0$, we must evaluate the integrands in the $\Pi_1$ and $\Pi_2$ terms using a selected numeric technique suited to integration from 0 to $\infty$, and obtain the internal characteristic function terms using complex-number computation. We can obtain the analytical solution to the European put price from the put-call parity, as follows:

\[
C_0 - P_0 = S_0 - e^{-rT} K.
\]

However, no analytical solution to American option pricing exists under the Heston model, except for calls on non-dividend-paying stocks. Consequently, we consider numerical solutions, of which the most well-known is the binomial model suggested by Cox et al. (1979). The binomial tree models the stock price’s evolution, with parameters chosen to match volatility $\sigma$ and drift term $r$ in geometric Brownian motion. The maturity $T$ is divided into $N$ equidistant time steps $\Delta t = \frac{T}{N}$. After the $n$th time step $t_n = n\Delta t$, the stock price $S_n$ can move up to $uS_n$ or down to $dS_n$, with probabilities $p$ and $1 - p$, respectively. To enable the tree to recombine, we choose $d = \frac{1}{u}$, and to match the volatility in geometric Brownian motion, we choose $u = \exp(\sigma\sqrt{\Delta t})$. Then, at the $n$th time step, we label the $n + 1$ nodes as $S_{n,1}, \ldots, S_{n,n+1}$. Given the risk-neutral pricing, the expected stock price should grow at the risk-free rate. Thus, we choose the probability to be $p = \frac{e^{\sigma\Delta t - d}}{u - d}$, to guarantee the martingale
restriction. Under such specifications, when the number of time steps is sufficiently large, the binomial tree for the stock price will converge to geometric Brownian motion.

To price a European option $V_0(S_0)$ with the payoff function $h(S_T)$, we use the terminal condition

$$V_N(S_{N,i}) = h(S_{N,i}) \quad \text{for } i \in \{1, \ldots, N + 1\}.$$  

Also, at time step $n - 1$, we use the martingale condition and let

$$V_{n-1}(S_{n-1,i}) = e^{-r\Delta t} \mathbb{E}[V_n(S_n)|S_{n-1,i}]$$

$$= e^{-r\Delta t} (pV_n(uS_{n-1,i}) + (1-p)V_n(dS_{n-1,i})).$$

Then, working backwards in the binomial tree from $N$ to $0$ yields the option price $V_0(S_0)$.

To apply the binomial tree method to American option pricing, we restrict ourselves to a discrete time scale and consider Bermudan options, i.e., options that can only be exercised on specific dates before maturity. Then, if time $n - 1$ is an exercise date, let

$$\hat{V}_{n-1}(S_{n,i}) = e^{-r\Delta t} (pV_n(uS_{n-1,i}) + (1-p)V_n(dS_{n-1,i}))$$

and set the value of the option at time $n - 1$ as follows:

$$V_{n-1}(S_{n,i}) = \max \{\hat{V}_{n-1}(S_{n,i}), h(S_{n,i})\}.$$  

Then, we can work out the price at time 0 by backward induction.

However, in the binomial model, the only stochastic factor is the price of the underlying asset. Other factors, like volatility, risk-free rates, etc. are considered constant. If other stochastic factors are taken into account, the binomial model will be computationally expensive, as the number of nodes in the tree grows exponentially with the number of factors, as outlined by Boyle et al. (1989) and noted by Stentoft (2004).

An alternative pricing method is simulation, which we describe in detail in the next section. Also well-known is the finite difference method proposed by Schwartz (1977). Consider the Black-Scholes ordinary differential equation

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0,$$

where $S \in (0, \infty)$, $t \in (0, T)$ and $V(S,T) = \max(K - S, 0)$. To discretize the equation, set an upper bound $S_{\text{max}}$ on $S$, and then divide the range of possible
American option pricing with machine learning

S into $N$ intervals of length $\Delta S = \frac{S_{\max}}{N}$, and the time into $M$ equal time steps of length $\Delta t = \frac{T}{M}$. Then, we can use a finite central difference to approximate the derivatives in $S$ as follows:

$$\frac{\partial V}{\partial S}(S_n, t_m) = \frac{V(S_{n+1}, t_m) - V(S_{n-1}, t_m)}{2\Delta S} + O(\Delta S^2),$$

$$\frac{\partial^2 V}{\partial S^2}(S_n, t_m) = \frac{V(S_{n+1}, t_m) - 2V(S_n, t_m) + V(S_{n-1}, t_m)}{2\Delta S^2} + O(\Delta S^2),$$

in which $S_n$ represents the stock price in the $n$-th step in the discretized stock price evolution path, and $t_m$ represents the time in the $m$-step after we discretize the time interval.

For the derivative with respect to time, we can choose between a backward difference

$$\frac{\partial V}{\partial t}(S_n, t_m) = \frac{V(S_n, t_m) - V(S_n, t_m - 1)}{2\Delta t} + O(\Delta t)$$

and a forward difference

$$\frac{\partial V}{\partial t}(S_n, t_m) = \frac{V(S_n, t_{m+1}) - V(S_n, t_m)}{2\Delta t} + O(\Delta t).$$

Using the backward difference in $t$ and the central difference in $S$ gives the discretization as follows:

$$\frac{V^m - V^{m-1}}{\Delta t} + \frac{1}{2} \sigma^2 \Delta S^2 n^2 \frac{V^m_{n+1} - 2V^m_n + V^m_{n-1}}{\Delta S^2} + r\Delta S n \frac{V^m_{n+1} - V^m_{n-1}}{2\Delta S} - rV^m_n = 0.$$

Thus, we can solve for $V^{m-1}$ explicitly, and this gives the explicit Euler scheme below:

$$V^{m-1}_n = A^m_n V^{m-1}_{n-1} + B^m_n V^m_n + C^m_n V^m_{n+1},$$

where

$$A^m_n = \frac{1}{2} n^2 \sigma^2 \Delta t - \frac{1}{2} n r \Delta t,$$

$$B^m_n = 1 - n^2 \sigma^2 \Delta t - r \Delta t,$$

and

$$C^m_n = \frac{1}{2} n^2 \sigma^2 \Delta t + \frac{1}{2} n r \Delta t.$$
Thus, we cannot solve for $V^m_n$ explicitly, but can express it implicitly, and this gives the implicit Euler scheme:

$$V^{m+1}_n = a^m_n V^m_{n-1} + b^m_n V^m_n + c^m_n V^m_{n+1},$$

where

$$a^m_n = -\frac{1}{2} n^2 \sigma^2 \Delta t + \frac{1}{2} nr \Delta t,$$

$$b^m_n = 1 + n^2 \sigma^2 \Delta t + r \Delta t,$$

$$c^m_n = -\frac{1}{2} n^2 \sigma^2 \Delta t - \frac{1}{2} nr \Delta t.$$

To price European put options, use the boundary condition

$$V^M_n = \max(K_n - n \Delta S, 0)$$

and work backwards in the time dimension to obtain $V^0_n$. To apply this method to price Bermudan options with $L$ exercise dates, for the Bermudan put options, set $V^M_n = \max(K - S_n, 0)$, and then solve backwards to each date $m$ to obtain an approximation $\hat{V}^m_n$. If $m$ is an early-exercise date, set $V^m_n = \max(\hat{V}^m_n, \max(K - S_n, 0))$, and otherwise $V^m_n = \hat{V}^m_n$.

The finite difference method cannot be extended to more than two or at most three stochastic factors, as outlined by Stentoft (2004). In contrast, a simulation-based method is expected to be a much better choice in such situations.

In recent years, machine learning algorithms and deep learning techniques have also been widely applied to the sector of American option pricing. The pricing can be solved by finding an optimal exercise policy, i.e., a rule that determines when to exercise the American option, given a strike price, in order to maximize the expected total discounted return. This is an optimal stopping problem that belongs to the general class of Markov decision processes (MDP). Reinforcement learning solves the finite MDP problem via the value iteration algorithm. Therefore, reinforcement learning techniques such as least-squares policy iteration suggested by Antos et al. (2007) and further developed by Li et al. (2009) have been widely researched. However, when the size of an MDP is large, the curse of dimensionality cannot be ignored, and requires effective sampling-based techniques to approximate. For example, Becker et al. (2019) develop a deep learning method for optimal stopping problems which directly learns the optimal stopping rule from Monte Carlo samples by decomposing an optimal stopping time into a sequence of 0-1 stopping decisions and approximating them recursively with a sequence
of multilayer feedforward neural networks, and test the approach on complicated option pricing in high-dimensional situations. They show that such neural network policies can approximate optimal stopping times to any desired degree of accuracy. Furthermore, researchers have applied deep learning methods to solve the high-dimension partial differential equation (PDE) problem, and then derive the option prices accordingly. For example, Han et al. (2018) reformulate the parabolic PDEs using backward stochastic differential equations, and approximate the gradient of the unknown solution by neural networks. Their numerical test on the nonlinear Black-Scholes equation suggests that the proposed algorithm is quite effective in high dimensions, in terms of both accuracy and cost. Nevertheless, most of these tests are conducted only on simulated data. In our study, we not only test our methods on simulated data, but also on real market data to see whether the proposed methods can be practically applicable and robust.

2.2 Monte Carlo formulation

As mentioned previously, one well-established pricing method is simulation. In a simulation, the state variables change based on pre-specified distributions in each step, rather than changing by some factor of proportionality, as in the binomial model. A random draw from the corresponding distribution determines values in the next period. One can simulate a large number of paths, and use the average of the prices obtained from each path as an estimator. Therefore, the number of nodes remains constant across time steps, and grows only linearly in the number of stochastic factors.

In order to apply simulation techniques to American option pricing, we restrict ourselves to a discrete time scale and consider Bermudan options, i.e., options that can only be exercised at specific dates before maturity. We consider a uniform discretization of the time interval $[0,T]$ into $N$ time steps with distance $\Delta t = \frac{T}{N}$, and $0 = t_1 < t_2 < \cdots < t_N = T$ equidistant exercise opportunities. We simulate the asset value $S_{t_i}$ at the same time steps $\{t_i\}_{i \in \{0,\ldots,N\}}$ as the discretization. The discretization error affects the simulation of $S_{t_1}, S_{t_2}, \ldots, S_{t_N}$, as Kloeden and Platen (2013) describe in detail.

A backward dynamic programming formulation which recursively estimates the value of the option forms the basis of Monte Carlo methods for pricing American options. Denote the underlying stochastic process by $X = (S, v)$, the payoff of exercising the option by $h(S_{t_i})$, and the discounted value of the option at time $t_i$ by $V_{t_i}$. We define the recursive estimation of the
option value as follows:

\[ V_{t_N} = h(S_{t_N}), \]
\[ V_{t_{i-1}} = \max \left\{ h(S_{t_{i-1}}), \mathbb{E}[V_{t_i}|X_{t_{i-1}}] \right\}, \quad i = 1, 2, \ldots, N. \]  

(1)

The equation above states that the value of the option at expiration time \( t_N \) is exactly the payoff of the option \( h(S_{t_N}) \). At any other time \( 0 \leq t_i < t_N \), the value of the option \( V_{t_{i-1}} \) is the maximum between immediate exercise \( h(S_{t_{i-1}}) \) and the continuation value \( \mathbb{E}[V_{t_i}|X_{t_{i-1}}] \), which is the discounted present value of holding the option rather than exercising it at time \( t_i \). We denote the continuation value as

\[ \Phi_{t_{i-1}} = \mathbb{E}[V_{t_i}|X_{t_{i-1}}], \quad i = 1, 2, \ldots, N - 1. \]

As the payoff \( h(S_{t_N}) \) is always non-negative, we can set \( \Phi_{t_N} = 0 \). That is, at expiry it is no longer optimal to hold the option. Conversely, we obtain the value function \( V_{t_i} \) for \( i = 0, 1, \ldots, N \) as

\[ V_{t_i} = \max \{ h(S_{t_i}), \Phi_{t_i} \}. \]

At the last step \( t_0 = 0 \), we estimate the option value as

\[ V_0 = \frac{1}{M} \sum_{j=1}^{M} V_{t_0}^{(j)}, \]

where \( M \) is the number of simulated paths of the Heston process \( \{X_t\} \).

### 2.3 Regression-based Monte Carlo methods

The previous section shows that it suffices to determine the continuation values \( \Phi_{t_0}, \ldots, \Phi_{t_{N-1}} \) in order to find the optimal stopping time. The basic idea of the regression-based Monte Carlo method is to use regression estimates as numerical procedures to compute the above conditional estimations approximately. Thus, we generate artificial samples of the price process, and use them to construct data for the regression estimates. The algorithm approximates the continuation value of the option at each possible exercise time by a linear combination of a set of basis functions \( \{\phi_j(X)\}_{j=1}^{N_B} \) which depend on the underlying stochastic process \( \{X_t\} \).

The classical Longstaff-Schwartz Algorithm is described below.
Longstaff-Schwartz algorithm

**Step 1** Generate $M$ paths for the values of the state variables at all possible exercise times.

**Step 2** At the terminal time $t_N$, set the option value $V$ equal to the payoff

$$V\left(S_{t_N}^{(m)}, t_N\right) = h\left(S_{t_N}^{(m)}, t_N\right), \quad m = 1, \ldots, M.$$ 

**Step 3** For the set of paths $\{i_l\}_{l=1}^L$, for which the option is in the money, i.e., $h(S_{t_{N-1}}^{i_l}, t_{N-1}) > 0$, find coefficients $a^*_j(t_{N-1})$ to minimize the norm

$$\left\| \sum_{j=1}^{N_b} a_j(t_{N-1}) \begin{pmatrix} \phi_j(X_{t_{N-1}}^{(i_1)}) \\ \phi_j(X_{t_{N-1}}^{(i_2)}) \\ \vdots \\ \phi_j(X_{t_{N-1}}^{(i_L)}) \end{pmatrix} e^{-r(t_{N-1})} \right\| \begin{pmatrix} V(S_{t_N}^{(i_1)}, t_N) \\ V(S_{t_N}^{(i_2)}, t_N) \\ \vdots \\ V(S_{t_N}^{(i_L)}, t_N) \end{pmatrix}.$$ 

**Step 4** For each path update the value function at time $t_{N-1}$ as follows:

$$V\left(S_{t_{N-1}}^{(m)}, t_{N-1}\right) = \begin{cases} h\left(S_{t_{N-1}}^{(m)}\right), & \text{if } h\left(S_{t_{N-1}}^{(m)}\right) \geq \sum_{j=1}^{N_b} a^*_j(t_{N-1}) \phi_j\left(X_{t_{N-1}}^{(m)}\right) \\ V\left(S_{t_N}^{(m)}, t_N\right) e^{r(t_{N-1})}, & \text{otherwise.} \end{cases}$$

**Step 5** Repeat Steps 3 and 4 for possible exercise times $t_{N-2}, t_{N-3}, \ldots$, until time $t_0$.

The inputs of the Longstaff-Schwartz Algorithm are the number of Monte Carlo paths $M$, the basis functions $\{\phi_j\}_{j=1}^{N_b}$, and the vector norm $\| \cdot \|$. Longstaff and Schwartz (2001) use polynomials for the basis functions and the $L_2$ vector norm, which leads to the OLS regression.

### 2.4 Machine learning methods

Even though OLS can be implemented easily, it is susceptible to overfitting and misspecification of the basis functions, such as the degree of polynomials and the interaction terms between state variables. In addition, OLS may need a large number of samples to achieve satisfactory fitting results.
Given these shortcomings, alternatives such as matching projection pursuit, by Tompaidis and Yang (2014), Gaussian process regression, by Mu et al. (2018), and the upgraded GPR-MC approach, proposed by Goudenège et al. (2020), which replaces the Monte Carlo-based computation of the continuation value by a binomial tree or an exact integration at each time step, have been suggested. Likewise, we consider several machine-learning alternatives to the classic linear regression employed in the Longstaff-Schwartz algorithm, including support vector regression (SVR) and classification and regression tree (CART). We expect each of these to address some drawbacks of the linear regression method. For example, SVR can learn more effectively in relatively small datasets and remain robust to outliers. CART is also capable of dealing with small datasets, and assumes no prior statistical attributes of the problem.

2.4.1 Support vector regression

Unlike linear regression, which aims to minimize the sum of squared errors, and its extensions, which add an additional penalty parameter to minimize complexity and/or reduce the number of features used, support vector regression (SVR) tries to fit the best line within a threshold value, which is the distance between the hyperplane and boundary line. Therefore, SVR gives us more flexibility to define how much error is acceptable in the model. It finds a decision boundary at a distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

The objective function of SVR is to minimize the coefficients, more specifically, the $L_2$-norm of the coefficient vector, instead of the sum of squared error. The constraints handle the error term; we set the absolute error to be less than or equal to a specified margin $\varepsilon$. $\varepsilon$ can be tuned to gain the desired accuracy of the model. In addition, for any value that falls outside of $\varepsilon$, we can have a slack variable denoted by $\xi$ to tolerate deviation from the margin, to a certain extent. The optimization problem is as follows:

$$\min \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{n} |\xi_i|, \quad \text{s.t.} \quad |y_i - \omega_i x_i| \leq \varepsilon + |\xi_i|.$$ 

Moreover, SVR can adopt the kernel trick. For all $x$ and $x'$ in the input space $\mathcal{X}$, we can express certain functions $k(x, x')$ as an inner product in another space $\mathcal{V}$, and the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the kernel function. Kernel functions make possible the operation in a high-dimensional, implicit feature space without computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs.
of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. Combined with support vector machine, the kernel trick avoids the explicit mapping needed for linear learning algorithms to learn a nonlinear hyperplane in the higher-dimensional space. The most widely used kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

In the context of pricing American options, the ultimate goal of modeling the continuation value is to determine the exercise policy. Consequently, we are only concerned about reducing error to a certain degree. In this situation, SVR seems to be a sensible alternative. It allows us to choose how tolerant we are of errors, both through an acceptable error margin and through tuning the tolerance to falling outside the acceptable error rate. Also, the statistical learning theory reviewed by Vapnik (1999) demonstrates that using structural risk minimization in SVM improves its generalization ability, making it more suitable for small-sample classification and regression problems by finding the decision boundary based on the representation samples, i.e., support vectors, selected from all samples. Therefore, we expect that SVR can potentially achieve a similarly satisfactory fitting result with fewer samples, i.e., fewer simulated paths in the context of Monte Carlo pricing of American options.

2.4.2 Classification and regression tree

Breiman et al. (1984) develop the classification and regression tree (CART) method. This nonparametric method predicts values of continuous dependent variables, in regression problems, or classifies categorical variables, in classification problems. We represent CART as a binary tree, in which each root node is a single input variable \( x \), and a split point on the variable if that variable is numeric. The leaf nodes of the tree contain an output variable \( y \) which generates a prediction. Given a new input, we traverse the tree by evaluating the specific input started at the root node of the tree.

CART makes no assumptions on the relationships between the explanatory variables and the dependent variables. The main idea of CART is to partition the space spanned by the predictors into a set of rectangles, and then fit a simple model in each rectangle. We perform the process of partitioning recursively, to balance between the size of the tree and the quality of data fitting. New data filters through the tree and lands in one of the rectangles. The output value for that rectangle is the prediction made by the model. The intuition is that a very large tree might overfit the data, while a small tree might overlook important details in the data.

Because of its nonparametric nature, CART is well suited for problems where there is little a priori knowledge regarding the relationship between
predictors and dependent variables. This is the situation we face when pricing American-style options by Monte Carlo simulation, where the continuation value is unknown, problem-dependent, nonparametric, and nonlinear.

3. Empirical result

We conduct a numerical test to compare the proposed alternatives with the Longstaff-Schwartz method. In the first part, we use simulated data to compare different methods and study the effect of the number of simulated paths and the number of discretized time steps on pricing accuracy. Then, in the second part, we compare the prices obtained from these methods with the market prices by first calibrating the Heston model to the market data of S&P 500 European options. Also, we define at-the-money (ATM) options as those with moneyness between 0.95 and 1.05. In-the-money (ITM) options have moneyness between 1.05 and 1.3 for call options, and between 0.7 and 0.95 for put options. Conversely, out-of-the-money (OTM) is defined as moneyness between 0.7 and 0.95 for calls, and 1.05 and 1.3 for puts.

3.1 Simulated data study

Let $t_0 = 0, t_1, \ldots, t_N = T$ be the equidistant discretization of the interval $[0, T]$ with the difference $\Delta t = \frac{T}{N}$. Then for $i = 1, 2, \ldots, N$, we can rewrite the discrete risk-neutral Heston dynamics for sufficiently small $\Delta t$:

$$S_{t_i} = S_{t_{i-1}} + \mu S_{t_{i-1}} \Delta t + \sqrt{\nu_{t_{i-1}} S_{t_{i-1}}} \sqrt{\Delta t} Z_{t_i}^{(1)},$$

$$\nu_{t_i} = \nu_{t_{i-1}} + \kappa (\theta - \nu_{t_{i-1}}) \Delta t + \xi \sqrt{\nu_{t_{i-1}}} \sqrt{\Delta t} \left( \rho Z_{t_i}^{(1)} + \sqrt{1 - \rho^2} Z_{t_i}^{(2)} \right),$$

where $\{Z_{t_i}^{(1)}, Z_{t_i}^{(2)}\}_{i=1,\ldots,N}$ are i.i.d. random variables with standard normal distribution.

In the Longstaff-Schwartz algorithm, we estimate the continuation value $\Phi(X)$ with the following basis functions in terms of stock prices $S$ and variance $\nu$:

$$\phi(S, \nu) = \{1, S, \nu, S\nu, S^2, \nu^2, S^2\nu, S^3, \nu^3, S^2\nu^2\}.$$

The interaction terms between $S$ and $\nu$ are necessary to capture the correlation of the Brownian motion of the stock price and its volatility.

In Figure 1, we visualize the continuation values approximated with different methods based on the simulated data at time $t - 1$. From the plot, the Longstaff-Schwartz method seems to yield the lowest estimate. The estimated continuation value by SVR appears to be the highest. It exhibits the highest
nonlinearity among the three. Moreover, the $v$ term seems to affect the CART estimate much more than the $S$ term does.

In Table 1, we present the results for the American put options with $S_0 = 100$, $T$ ranging from one month to two years, and the strike $K$ ranging from 70 to 130 to account for ITM, ATM, and OTM options with different maturities. For the sake of comparison, we use the price obtained from the finite difference method with the number of grids set sufficiently large as the benchmark, and report root mean squared error (RMSE) of the prices obtained from different methods.

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.6796</td>
<td>0.4724</td>
<td>0.8187</td>
</tr>
<tr>
<td>ITM</td>
<td>0.8385</td>
<td>0.5975</td>
<td>0.9836</td>
</tr>
<tr>
<td>ATM</td>
<td>0.7557</td>
<td>0.5687</td>
<td>0.8591</td>
</tr>
<tr>
<td>OTM</td>
<td>0.5260</td>
<td>0.3087</td>
<td>0.6885</td>
</tr>
</tbody>
</table>

We see that SVR outperforms Longstaff-Schwartz in the whole samples. Its better performance remains robust across each moneyness bucket. CART
performs worse than Longstaff-Schwartz in all cases.

Next, we would like to study the effects of increasing the number of simulated paths and the number of time steps on pricing accuracy. With the same parameters for the Heston Model and the same set of option specifications, we report the results in Tables 2 and 3, respectively.

From Table 2, we see that as the number of paths increases, the errors exhibit a downward trend. Error reduction is most significant in the Longstaff-Schwartz algorithm, while this is reversed for SVR in some situations. This observation also corroborates our previous discussion that SVR might be able to achieve results similarly satisfactory to the classical Longstaff-Schwartz method, with fewer simulated paths. Moreover, when the number of paths is large enough, the Longstaff-Schwartz method outperforms the other two in the ITM options. SVR still remains the top performer in ATM and OTM options.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Effect of the number of simulated paths, with 100 time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of paths</td>
<td>moneyness</td>
</tr>
<tr>
<td>total</td>
<td>500</td>
</tr>
<tr>
<td>ITM</td>
<td>1.2760</td>
</tr>
<tr>
<td>ATM</td>
<td>1.0316</td>
</tr>
<tr>
<td>OTM</td>
<td>0.8376</td>
</tr>
<tr>
<td>total</td>
<td>1000</td>
</tr>
<tr>
<td>ITM</td>
<td>0.8385</td>
</tr>
<tr>
<td>ATM</td>
<td>0.7557</td>
</tr>
<tr>
<td>OTM</td>
<td>0.5260</td>
</tr>
<tr>
<td>total</td>
<td>1500</td>
</tr>
<tr>
<td>ITM</td>
<td>0.6572</td>
</tr>
<tr>
<td>ATM</td>
<td>0.5452</td>
</tr>
<tr>
<td>OTM</td>
<td>0.3976</td>
</tr>
<tr>
<td>total</td>
<td>2000</td>
</tr>
<tr>
<td>ITM</td>
<td>0.4419</td>
</tr>
<tr>
<td>ATM</td>
<td>0.5125</td>
</tr>
<tr>
<td>OTM</td>
<td>0.3127</td>
</tr>
</tbody>
</table>

From Table 3, we see that as the number of time steps increases, SVR always performs the best in OTM options. CART outperforms the other two in ITM options when the number of time steps is larger, i.e., when the time discretization is finer. Also, the relatively large drop in the RMSE of CART
in ATM option pricing when the number of time steps is 200 reflects the potential limitation in CART that trees can be very non-robust, i.e., a small change in the training data can result in a large change in the tree, and thus the final prediction.

Table 3  
Effect of the number of time steps, with 1000 simulated paths

<table>
<thead>
<tr>
<th>number of time steps</th>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>0.5949</td>
<td>0.5274</td>
<td>0.8997</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>0.7152</td>
<td>0.6735</td>
<td>1.1107</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>0.6942</td>
<td>0.6392</td>
<td>0.9321</td>
</tr>
<tr>
<td></td>
<td>OTM</td>
<td>0.4633</td>
<td>0.3381</td>
<td>0.7268</td>
</tr>
<tr>
<td>50</td>
<td>Total</td>
<td>0.6796</td>
<td>0.4724</td>
<td>0.8187</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>0.8385</td>
<td>0.5975</td>
<td>0.9836</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>0.7557</td>
<td>0.5687</td>
<td>0.8591</td>
</tr>
<tr>
<td></td>
<td>OTM</td>
<td>0.5260</td>
<td>0.3087</td>
<td>0.6885</td>
</tr>
<tr>
<td>100</td>
<td>Total</td>
<td>0.6725</td>
<td>0.6598</td>
<td>0.6551</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>0.8626</td>
<td>0.8955</td>
<td>0.7691</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>0.6322</td>
<td>0.6867</td>
<td>0.6728</td>
</tr>
<tr>
<td></td>
<td>OTM</td>
<td>0.5362</td>
<td>0.3972</td>
<td>0.5660</td>
</tr>
<tr>
<td>150</td>
<td>Total</td>
<td>0.7087</td>
<td>0.6177</td>
<td>0.6202</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>0.8941</td>
<td>0.7873</td>
<td>0.7734</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>0.7370</td>
<td>0.7779</td>
<td>0.4268</td>
</tr>
<tr>
<td></td>
<td>OTM</td>
<td>0.5407</td>
<td>0.3726</td>
<td>0.5505</td>
</tr>
<tr>
<td>200</td>
<td>Total</td>
<td>0.7087</td>
<td>0.6177</td>
<td>0.6202</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>0.8941</td>
<td>0.7873</td>
<td>0.7734</td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td>0.7370</td>
<td>0.7779</td>
<td>0.4268</td>
</tr>
<tr>
<td></td>
<td>OTM</td>
<td>0.5407</td>
<td>0.3726</td>
<td>0.5505</td>
</tr>
</tbody>
</table>

3.2 Market data study

We would like to apply the proposed methods to the real market to test their applicability. Thus, we consider the S&P 500 Index (SPX), which has enough market depth and trading volume, as well as a relatively large number of options written on it. Therefore, we expect the corresponding option prices to be less subject to artificial manipulation.

First, we calibrate the Heston model based on the SPX European options. Our data is from WRDS OptionMetrics Database, which contains the date of the contract, the strike price, maturity date, implied volatility, best ask and best bid of each option, and the forward price of the SPX index. We also obtain the SPX spot prices from the WRDS database, and the risk-free rate from FRED. To calibrate the Heston model, we try to minimize the mean squared
difference between the Black-Scholes implied volatility of SPX European options derived from the market prices and the prices from the Heston model as follows:

$$\min_{\kappa, v_0, \theta, \xi, \rho} \sqrt{\frac{1}{|O|} \sum_{o \in O} (\sigma_{imp}^{mkt} - \sigma_{imp}^{Heston})^2},$$

where $|O|$ is the set of options on the same index, i.e. SPX European options, traded on the corresponding day, $\sigma_{imp}^{mkt}$ is the implied volatility based on the market price and $\sigma_{imp}^{Heston}$ is the implied volatility based on the prices obtained from the Heston model. To make this calibration more robust, we use the global optimizer differential evolution here. Also, since we test on the American options traded on each Wednesday of 2019, the European options used for calibration are those traded on the last day before the corresponding Wednesday.

For more reliable calibration results without the impact of outliers, we use the options with trading volume greater than 0, implied volatility no greater than 1, and moneyness between 0.7 and 1.3. To illustrate, we present the calibrated parameters of the Heston model for January 2019 in Table 4.

<table>
<thead>
<tr>
<th>test date</th>
<th>calibration date</th>
<th>$\theta$</th>
<th>$\kappa$</th>
<th>$\xi$</th>
<th>$\rho$</th>
<th>$v_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9/2019</td>
<td>1/8/2019</td>
<td>0.0449</td>
<td>11.1236</td>
<td>2.0196</td>
<td>−0.6592</td>
<td>0.0406</td>
</tr>
<tr>
<td>1/16/2019</td>
<td>1/15/2019</td>
<td>0.0440</td>
<td>7.0026</td>
<td>1.9903</td>
<td>−0.6518</td>
<td>0.0289</td>
</tr>
<tr>
<td>1/23/2019</td>
<td>1/22/2019</td>
<td>0.0468</td>
<td>6.8600</td>
<td>2.2994</td>
<td>−0.6953</td>
<td>0.0444</td>
</tr>
<tr>
<td>1/30/2019</td>
<td>1/29/2019</td>
<td>0.0376</td>
<td>9.1716</td>
<td>2.0465</td>
<td>−0.6607</td>
<td>0.0397</td>
</tr>
</tbody>
</table>

As only European options are traded on SPX, we consider testing our methods on the pricing of American options of SPDR S&P 500 Trust ETF (SPY), which is an exchange-traded fund that closely tracks SPX and has only American options, and not European options, traded on it. Even though SPY closely tracks SPX, tracking errors still exist, to some extent. Another difference between SPX and SPY is that SPY pays quarterly dividends, which impacts the prices of its options when the dividends are paid out. To deal with this issue, in simulating the stochastic process, we assume the dividend is paid out continuously. The continuous dividend yield is calculated from the TTM (trailing twelve months) dividend yield on each dividend payout date,
respectively. Thus, the stochastic process becomes as follows:

\[ dS_t = (\mu - \delta)S_t dt + \sqrt{v_t}S_t dW^{(1)}_t, \]
\[ dv_t = \kappa(\theta - v_t)dt + \xi \sqrt{v_t}dW^{(2)}_t, \]
\[ \langle dW^{(1)}_t, dW^{(2)}_t \rangle = \rho dt, \]

where \( W^{(1)}_t \) and \( W^{(2)}_t \) are correlated Brownian motions, and \( \delta \) is the continuous dividend yield.

Similarly, the discretization becomes as follows:

\[ S_{t_i} = S_{t_{i-1}} + (\mu - \delta)S_{t_{i-1}} \Delta t + \sqrt{v_{t_{i-1}}}S_{t_{i-1}} \sqrt{\Delta t} Z^{(1)}_{t_i}, \]
\[ v_{t_i} = v_{t_{i-1}} + \kappa(\theta - v_{t_{i-1}})\Delta t + \xi \sqrt{v_{t_{i-1}}} \sqrt{\Delta t} \left( \rho Z^{(1)}_{t_i} + \sqrt{1 - \rho^2}Z^{(2)}_{t_i} \right), \]

where \( \{Z^{(1)}_{t_i}, Z^{(2)}_{t_i}\}_{i=1,...,N} \) are i.i.d. random variables with standard normal distribution.

Then, combined with the calibrated parameters, we simulate the stochastic process for SPY on each Wednesday of 2019, respectively, and apply our methods to price all the American options traded on that day. Similar to the treatment in calibrating the Heston model parameters, we only use the options with trading volume greater than 0 to reflect actual market conditions. Also, we only consider the options with maturity between one month and one year, to make pricing more robust. The market price of the option is calculated as the mid point of the best bid and the best ask. The bid price is the maximum price a buyer is willing to pay for the option, and the best bid is the highest price among all the bid prices. Likewise, the ask price is the minimum price for which a seller is willing to sell the option, and the best ask is the lowest among all the ask prices. The SPY American option data used for testing is summarized in Table 5.

To test the applicability of SVR and CART in the real market, out of concern for efficiency, we first choose fixed hyperparameters for both SVR and CART, and compare the root mean squared error (RMSE) of Longstaff-Schwartz, SVR, and CART. The result is summarized in Table 6.

In most cases, Longstaff-Schwartz outperforms our proposed methods. However, SVR still performs much better in the OTM put option pricing, and runs close to Longstaff-Schwartz in the OTM call option pricing, which is consistent with the result from the simulated data. We would like to know whether tuning hyperparameters will be of some help, and thus test this idea on the American options traded during the first quarter of 2019. However, due
Table 5
Data summary

<table>
<thead>
<tr>
<th></th>
<th>total</th>
<th>ITM</th>
<th>ATM</th>
<th>OTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>62619</td>
<td>6581</td>
<td>31178</td>
<td>24860</td>
</tr>
<tr>
<td>mean price</td>
<td>8.3370</td>
<td>31.7633</td>
<td>8.1647</td>
<td>2.3517</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>total</th>
<th>ITM</th>
<th>ATM</th>
<th>OTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>27326</td>
<td>4674</td>
<td>15861</td>
<td>6791</td>
</tr>
<tr>
<td>mean price</td>
<td>10.5517</td>
<td>32.4205</td>
<td>8.0354</td>
<td>1.3773</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>total</th>
<th>ITM</th>
<th>ATM</th>
<th>OTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>35293</td>
<td>1907</td>
<td>15317</td>
<td>18069</td>
</tr>
<tr>
<td>mean price</td>
<td>6.6224</td>
<td>30.1525</td>
<td>8.2987</td>
<td>2.7180</td>
</tr>
</tbody>
</table>

Table 6
RMSE for the year 2019, without hyperparameter tuning

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.8285</td>
<td>1.3002</td>
<td>1.9044</td>
</tr>
<tr>
<td>ITM</td>
<td>1.0285</td>
<td>1.1918</td>
<td>1.8488</td>
</tr>
<tr>
<td>ATM</td>
<td>0.8495</td>
<td>1.6735</td>
<td>1.9083</td>
</tr>
<tr>
<td>OTM</td>
<td>0.7374</td>
<td>0.6084</td>
<td>1.9140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.7356</td>
<td>1.2103</td>
<td>1.3144</td>
</tr>
<tr>
<td>ITM</td>
<td>0.8875</td>
<td>1.1467</td>
<td>1.7351</td>
</tr>
<tr>
<td>ATM</td>
<td>0.7718</td>
<td>1.4223</td>
<td>1.3688</td>
</tr>
<tr>
<td>OTM</td>
<td>0.4942</td>
<td>0.5141</td>
<td>0.7099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.8938</td>
<td>1.3658</td>
<td>2.2576</td>
</tr>
<tr>
<td>ITM</td>
<td>1.3115</td>
<td>1.2958</td>
<td>2.1015</td>
</tr>
<tr>
<td>ATM</td>
<td>0.9232</td>
<td>1.8989</td>
<td>2.3393</td>
</tr>
<tr>
<td>OTM</td>
<td>0.8101</td>
<td>0.6402</td>
<td>2.2024</td>
</tr>
</tbody>
</table>

to efficiency concerns, we tune the hyperparameters within a relatively narrow range. The result is reported in Table 7. For comparison, we also present the RMSE on the options traded during this period when the hyperparameters are not tuned in Table 8.

The hyperparameter tuning enhances the performance of SVR and CART in some cases, like the ATM call options for SVR and all the put options for CART, but does not always help reduce the error. Moreover, regarding relative performance, with hyperparameter tuning, SVR retains its advantage in OTM call and put options, and becomes the best performer in ATM call options, but does even worse in ITM put options. CART still shows no out-
American option pricing with machine learning

Table 7
RMSE for the first quarter of 2019, with hyperparameter tuning

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.7836</td>
<td>1.1981</td>
<td>1.7318</td>
</tr>
<tr>
<td>ITM</td>
<td>1.0297</td>
<td>1.3743</td>
<td>2.6978</td>
</tr>
<tr>
<td>ATM</td>
<td>0.7400</td>
<td>1.4973</td>
<td>1.8268</td>
</tr>
<tr>
<td>OTM</td>
<td>0.7669</td>
<td>0.6318</td>
<td>1.2748</td>
</tr>
</tbody>
</table>

Table 8
RMSE for the first quarter of 2019, without hyperparameter tuning

<table>
<thead>
<tr>
<th>moneyness</th>
<th>LS</th>
<th>SVR</th>
<th>CART</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>0.7823</td>
<td>1.1758</td>
<td>1.9084</td>
</tr>
<tr>
<td>ITM</td>
<td>1.04201</td>
<td>1.1505</td>
<td>1.9159</td>
</tr>
<tr>
<td>ATM</td>
<td>0.7414</td>
<td>1.5283</td>
<td>1.8240</td>
</tr>
<tr>
<td>OTM</td>
<td>0.7583</td>
<td>0.5424</td>
<td>1.9998</td>
</tr>
</tbody>
</table>

Given the outperformance of SVR in simulated data, it is surprising that this trend does not persist in the market data. Apart from the possible lack of performance over Longstaff-Schwartz.
of robustness of the machine learning techniques we propose, we think the poor performance with the market data can be attributed to the consideration of dividends and inherent tracking error between SPX and SPY. As a result, our simulated stock price, i.e., the one generated by the Heston process calibrated with SPX, may be consistently lower than the stock price that should be generated by the Heston process, based on what the American options are actually written on, i.e., SPY. In this case, SVR may overestimate the continuation value, and thus ITM options suffer from mispricing the most. Consequently, with the market data, even the hyperparameter tuning fails to improve the performance of SVR in ITM options. In addition, even though we tune the hyperparameters, we conduct this in a relatively narrow range, which reduces its power of tuning to some extent. With simulated data, we tune the hyperparameters in a wider range. Still, this empirical result shows that machine learning techniques can be useful alternatives. Nevertheless, to give full play to their power, details like the setup of the regression problem and the hyperparameter tuning procedures deserve enough attention from researchers and practitioners. Otherwise, the established classical method may work better.

4. Conclusion

We propose two machine learning techniques: support vector regression and classification and regression trees, to replace the OLS regression step in estimating the continuation value of American options and thus extend the classical Longstaff-Schwartz method under the regression-based Monte Carlo framework. With simulated data, SVR demonstrates robust outperformance, and its excellent performance is particularly consistent in the OTM bucket. CART can perform better in the ITM bucket when the discretization across the time interval is finer. This showcases that machine learning techniques can be powerful alternatives to the traditional Longstaff-Schwartz method in American option pricing. However, in the market data, SVR outperforms others only in the OTM bucket, but not in the whole sample. This inconsistency in the performance between simulated data and market data is alleviated to some extent but still not fully resolved, even after we tune the hyperparameters. This might be caused by the design of our empirical study and the relatively narrow range when we tune the hyperparameters.

There is still work to be done in the future. First, we do not tune the hyperparameters in SVR and CART within a relatively wide range. It would be of interest to see whether tuning in a wider range can further enhance the performance of these two machine learning methods. Second, to study whether the poor performance of our proposed methods in the market data
is caused by the inherent difference between SPX and SPY, we can test the Longstaff-Schwartz method and our proposed methods on options on individual stocks, with the calibration of Heston model conducted on the European options on exactly the same stocks. Third, CART does not yield satisfactory performance, possibly because it is subject to over-fitting. Thus, using such mechanisms as pruning or extensions based on decision trees like random forest may yield better performance. Fourth, within the Monte Carlo framework, adding European option price as a control variable should reduce the standard error of the simulation. It may help improve the performance of our proposed methods, especially for CART, given its vulnerability to the over-fitting problem and non-robustness when we change the parameters of the simulation, such as the number of discretized time steps. Finally, since the simulation technique overcomes the curse of dimensionality, we can apply our proposed methods to the pricing of American options on a basket of a large number of underlying assets, or on the assets whose price dynamics include a large number of stochastic factors, to see whether these methods are able to better match market prices.

References


