

# Least-Squares Monte-Carlo for an American Put Option

Adrian Scheerer

July 2020

These notes reproduce the first example on least-squares Monte-Carlo from paper *Valuing American Options by Simulation: A Simple Least-Squares Approach* by Longstaff and Schwartz, [available here](#).

Consider an American put option on a share of a non-dividend-paying stock. The put option is exercisable at a strike price of 1.10 at times 1, 2, and 3, where time 3 is the expiration date of the option.

The riskless rate is assumed to be 6% with continuous compounding, giving the discount factor for one time-step:

```
( d <- exp(-0.06) )
```

```
## [1] 0.9417645
```

The following eight Monte-Carlo simulations for the underlying are given (for times 0 to 3):

```
paths <- matrix(c(1, 1.09, 1.08, 1.34,
                  1, 1.16, 1.26, 1.54,
                  1, 1.22, 1.07, 1.03,
                  1, 0.93, 0.97, 0.92,
                  1, 1.11, 1.56, 1.52,
                  1, 0.76, 0.77, 0.9,
                  1, 0.92, 0.84, 1.01,
                  1, 0.88, 1.22, 1.43),
                nrow = 8, ncol = 4, byrow = T)
```

The payoff function of a put option with strike 1.10:

```
payoff <- function(underlying) {
  pmax(1.10 - underlying, 0)
}
```

We will build the full cashflow matrix step-by-step going backwards starting at time 3.

For time 3 the cashflow is just the value of the option when it is immediately exercised.

```
cashflow_matrix <- matrix(NA, nrow = 8, ncol = 3)
cashflow_3 <- payoff(paths[, 4])
cashflow_matrix[, 3] <- cashflow_3
cashflow_matrix
```

```
##      [,1] [,2] [,3]
## [1,]   NA   NA 0.00
## [2,]   NA   NA 0.00
## [3,]   NA   NA 0.07
## [4,]   NA   NA 0.18
## [5,]   NA   NA 0.00
## [6,]   NA   NA 0.20
## [7,]   NA   NA 0.09
## [8,]   NA   NA 0.00
```

At time 2, when you are not in the money, you will not exercise the option and rather wait. If you are in the money, you need to decide whether immediately exercising or holding the option is better.

Paths that are in the money at time 2:

```
( in_the_money2 <- which(paths[, 3] < 1.10) )
```

```
## [1] 1 3 4 6 7
```

We need to compare the payoff at time 2 with the discounted payoff from time 3, but only for such paths that are in the money at time 2, i.e., will have positive payoff at time 2:

```
X <- paths[in_the_money2, 3]
```

The discounted expected cashflows from time 3 for the paths that are in the money at time 2 are

```
( Y <- d * cashflow_3 [in_the_money2] )
```

```
## [1] 0.00000000 0.06592352 0.16951762 0.18835291 0.08475881
```

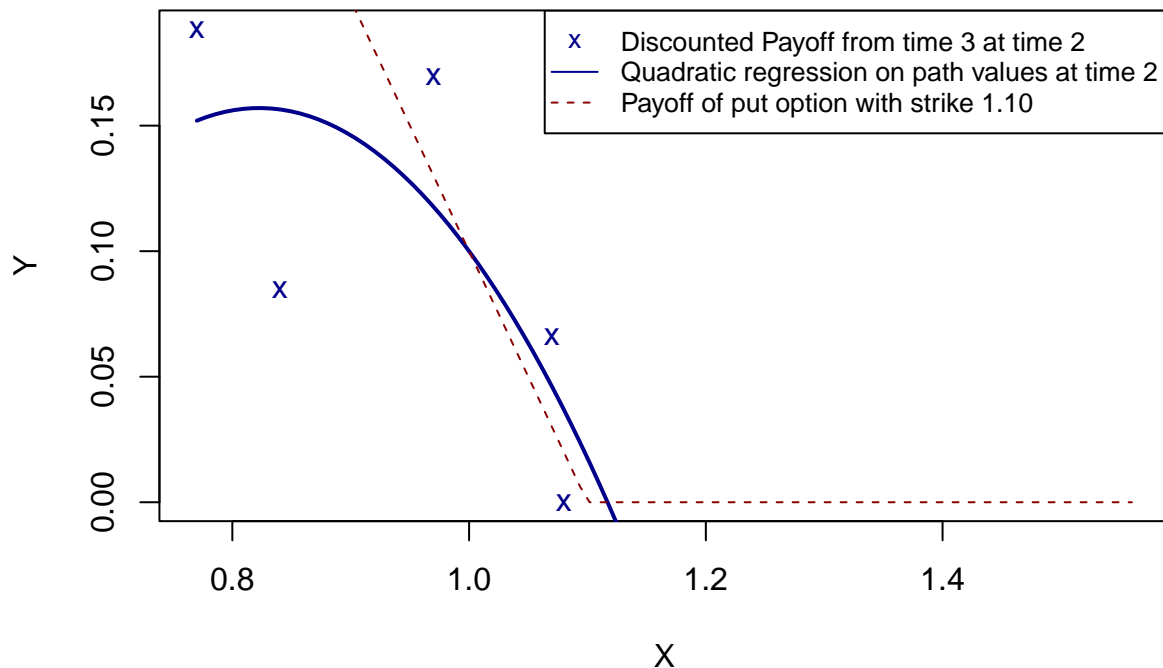
The idea is now to write the (discounted) payoff from time 3 (which we denoted as Y) as a quadratic function of the realized path values at time 2 (which we denoted as X).

```
regression_3_on_2 <- lm(Y ~ X + I(X^2))
regression_3_on_2$coefficients
```

```
## (Intercept)          X          I(X^2)
##   -1.069988    2.983411   -1.813576
```

This recovers  $E[Y|X] = -1.070 + 2.983 * X - 1.813 * X^2$ . The fitted function is somewhat reminiscent of the payout function of the put option:

```
plot(X, Y, col="darkblue", pch="x", xlim = range(paths[, 3]))
curve(-1.070 + 2.983*x - 1.813*x^2, add = T, col = "darkblue", lwd = 2)
curve(payoff, add = T, col = "darkred", lty = "dashed")
legend(x = "topright",
       legend = c("Discounted Payoff from time 3 at time 2",
                  "Quadratic regression on path values at time 2",
                  "Payoff of put option with strike 1.10"),
       col = c("darkblue", "darkblue", "darkred"),
       lty = c(NA, "solid", "dashed"),
       pch = c("x", NA, NA),
       cex = 0.8)
```



To decide at time 2 whether you want to exercise or hold the option, you need to compare the current payoff with the discounted future payoff.

The immediate payoff for exercising the option at time 2 is

```
( cashflow_exercise_at_2 <- payoff(paths[in_the_money2, 3]) )
```

```
## [1] 0.02 0.03 0.13 0.33 0.26
```

Continuing to hold the option gives the following payout:

- the payout is actually Y (discounted cashflow from time 3 seen at time 2 for paths which are in the money at time 2):

```
Y
```

```
## [1] 0.00000000 0.06592352 0.16951762 0.18835291 0.08475881
```

- but the idea is instead of Y to use the fitted quadratic function of X evaluated at X:

```
( cashflow_continuation_at_2 <- predict(regression_3_on_2, data.frame(X=X)) )
```

```
##           1           2           3           4           5
## 0.03674056 0.04589834 0.11752682 0.15196921 0.15641792
```

With this approximation, for which paths would you exercise the option at time 2?

```
( paths_exercise_at_2 <- which(cashflow_exercise_at_2 > cashflow_continuation_at_2) )
```

```
## 3 4 5  
## 3 4 5
```

Now we know what the cashflow matrix looks like for times 2 and 3. For time 3 the cashflow is just the payoff of the option for immediate exercise at time 3, but only for those paths for which it is not optimal to exercise the option already at time 2. When you exercise the option at time 2, the payoff at time 3 will be 0. For time 2 the cashflow will be the payoff of the option when it is optimal to exercise it already at time 2, any 0 if you keep holding the option.

The cashflow matrix for times 2 and 3 hence looks as follows:

Zero cashflow,

```
cashflow_matrix[, 2] <- 0
```

unless you are in the money and it is optimal to exercise early,

```
cashflow_matrix[in_the_money2[paths_exercise_at_2], 2] <- cashflow_exercise_at_2[paths_exercise_at_2]
```

but then you cannot exercise at time 3 anymore.

```
cashflow_matrix[in_the_money2[paths_exercise_at_2], 3] <- 0  
cashflow_matrix
```

```
##      [,1] [,2] [,3]  
## [1,]    NA 0.00 0.00  
## [2,]    NA 0.00 0.00  
## [3,]    NA 0.00 0.07  
## [4,]    NA 0.13 0.00  
## [5,]    NA 0.00 0.00  
## [6,]    NA 0.33 0.00  
## [7,]    NA 0.26 0.00  
## [8,]    NA 0.00 0.00
```

Now for time 1.

The cashflows at time 2 are (note these are not the regression values)

```
( cashflow_2 <- cashflow_matrix[, 2] )
```

```
## [1] 0.00 0.00 0.00 0.13 0.00 0.33 0.26 0.00
```

At time 1, when you are not in the money, you will not exercise the option and rather wait. If you are in the money, you need to decide whether exercising or holding the option is better.

The paths that are in the money at time 1:

```
( in_the_money1 <- which(paths[, 2] < 1.10) )
```

```
## [1] 1 4 6 7 8
```

Need to compare payoff at time 1 with discounted payoff from time 2 only for such paths that are in the money at time 2, i.e., will have positive payoff at time 2:

```
( X <- paths[in_the_money1, 2] )
```

```
## [1] 1.09 0.93 0.76 0.92 0.88
```

The discounted expected cashflows for these in the money paths at time 2

```
( Y <- d * cashflow_2 [in_the_money1] )
```

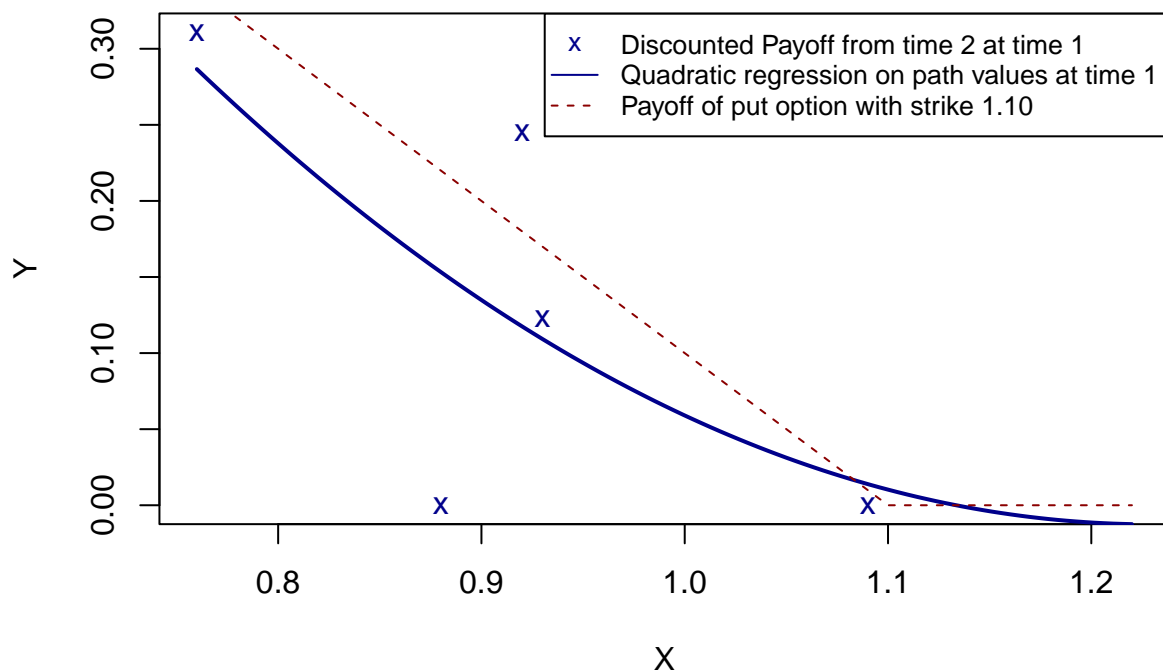
```
## [1] 0.0000000 0.1224294 0.3107823 0.2448588 0.0000000
```

We now again write the (discounted) payoff from time 2 as a function of the path values at time 1.

```
regression_2_on_1 <- lm(Y ~ X + I(X^2))
regression_2_on_1$coefficients
```

```
## (Intercept)          X          I(X^2)
##    2.037512   -3.335443    1.356457
```

For time 1, the quadratic approximation looks like this:



We again need to decide whether it is optimal to exercise the option at time 1 or hold it.

The immediate payoff for exercising the option at time 1 is

```
( cashflow_exercise_at_1 <- payoff(paths[in_the_money1, 2]) )
```

```
## [1] 0.01 0.17 0.34 0.18 0.22
```

If we continue to hold the option:

- the actual payoff is Y (discounted cashflow from time 2 seen at time 1 for paths at time 1 which are in the money):

```
Y
```

```
## [1] 0.0000000 0.1224294 0.3107823 0.2448588 0.0000000
```

- but instead of Y we use the quadratic function of the path values at time 1 evaluated at these path values

```
( cashflow_continuation_at_1 <- predict(regression_2_on_1, data.frame(X=X)) )
```

```
##           1           2           3           4           5
## 0.01348511 0.10874928 0.28606468 0.11700927 0.15276213
```

For which paths would you exercise the option at time 1?

```
( paths_exercise_at_1 <- which(cashflow_exercise_at_1 > cashflow_continuation_at_1) )
```

```
## 2 3 4 5
## 2 3 4 5
```

Finally, we can update the cashflow matrix with the cashflow at time 1. This cashflow is zero,

```
cashflow_matrix[, 1] <- 0
```

unless you are in the money at time 1 and it is optimal to exercise early,

```
cashflow_matrix[in_the_money1[paths_exercise_at_1], 1] <- cashflow_exercise_at_1[paths_exercise_at_1]
```

but then you cannot exercise at time 2 (or 3) anymore.

```
cashflow_matrix[in_the_money1[paths_exercise_at_1], 2:3] <- 0
cashflow_matrix
```

```
##      [,1] [,2] [,3]
## [1,] 0.00  0 0.00
## [2,] 0.00  0 0.00
## [3,] 0.00  0 0.07
## [4,] 0.17  0 0.00
## [5,] 0.00  0 0.00
## [6,] 0.34  0 0.00
## [7,] 0.18  0 0.00
## [8,] 0.22  0 0.00
```

Now, the price of the option is the average of the discounted cashflow over all paths:

```
mean(cashflow_matrix %*% c(d, d^2, d^3))
```

```
## [1] 0.1144343
```