Optimization under uncertainty: modeling and solution methods

Paolo Brandimarte
Dipartimento di Scienze Matematiche
Politecnico di Torino

e-mail: paolo.brandimarte@polito.it
URL: http://staff.polito.it/paolo.brandimarte

Lecture 8: Stochastic dynamic programming

REFERENCES

- D.P. Bertsekas. *Dynamic Programming and Optimal Control (Vols. I & II)*. Athena Scientific, 2005/2012.
- P. Brandimarte. *Numerical Methods for Finance and Economics: A Matlab-Based Introduction*, (2nd ed.). Wiley, 2006.
- P. Brandimarte. *Handbook in Monte Carlo Simulation: Applications in Financial Engineering, Risk Management, and Economics*. Wiley, late 2013.
- L. Buşoniu, R. Babuška, B. De Schutter, D. Ernst. *Reinforcement learning and Dynamic Programming Using Function Approximations.* CRC Press, 2010
- K.L. Judd. *Numerical Methods in Economics*. MIT Press, 1998.
- F.A. Longstaff, E.S. Schwartz. Valuing American Options by Simulation: a Simple Least-Squares Approach. *The Review of Financial Studies*, **14** (2001) 113-147.
- M.J. Miranda, P.L. Fackler. *Applied Computational Economics and Finance*. MIT Press, 2002.
- W.B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, (2nd ed.). Wiley, 2011.
- R.S. Sutton, A.G. Barto. *Reinforcement learning*. MIT Press, 1998.

I also recommend the Web page on computational stochastic optimization, maintained by Warren Powell of Princeton University: `http://www.castlelab.princeton.edu/cso.htm`

OUTLINE

DYNAMIC PROGRAMMING

Dynamic programming is arguably the most powerful optimization strategy available.

In principle, it can be used to deal with:

- discrete and continuous time models;
- finite and infinite time horizons;
- continuous and discrete state variables;
- continuous and discrete decision variables;
- deterministic and stochastic problems.

There is a huge literature on Markov Decision Processes, i.e., stochastic DP with discrete state and discrete decisions, featuring significant links with Artificial Intelligence and Machine Learning. Here we illustrate only stochastic DP in discrete time with continuous state variables.

SDP IN DISCRETE TIME, CONTINUOUS STATE

The starting point of a DP approach is a dynamic model based on a state transition function:

$$\mathbf{s}_{t+1} = \mathbf{g}_t(\mathbf{s}_t, \mathbf{x}_t, \epsilon_{t+1}), \tag{1}$$

where $\mathbf{s}_t$ is the state at time t, $\mathbf{x}_t$ is the decision made after observing the state, and ϵ_{t+1} is a random disturbance occurring *after* we have made our decision.

The time horizon can be finite, in which case we have to make decisions at $t = 0, 1, \ldots, T$, or infinite. In the latter case, it is typical to consider stationary state transition functions, i.e., $\mathbf{g}$ does not depend on time.

When we make decision $\mathbf{x}_t$ in state $\mathbf{s}_t$, we incur an immediate cost or reward $f_t(\mathbf{s}_t, \mathbf{x}_t)$. In finite-horizon problems, it is also possible to assign a value/cost $F_{T+1}(\mathbf{s}_{T+1})$ to the terminal state.

Optimal policies

In a stochastic dynamic decision problem we cannot anticipate optimal decisions $\mathbf{x}_t^*$. The decisions are a stochastic process depending on the realization of the disturbances and the resulting state trajectory.

In SDP we look for an *optimal policy in feedback form*, i.e., a mapping from state to decisions

$$\mathbf{x}_t = A^\pi(\mathbf{s}_t). \tag{2}$$

The mapping must be admissible, in the sense that we may have to comply with constraints on decisions and state variables.
Let Π be the set of admissible policies. We want to solve the problems

$$\max_{\pi \in \Pi} \ \mathrm{E}\left[\sum_{t=0}^{T} \beta^t f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta^{T+1} F_{T+1}(\mathbf{s}_{T+1})\right] \tag{3}$$

or

$$\max_{\pi \in \Pi} \ \mathrm{E}\left[\sum_{t=0}^{\infty} \beta^t f(\mathbf{s}_t, \mathbf{x}_t)\right] \tag{4}$$

with discount factor $\beta \in (0, 1]$ ($\beta < 1$ is needed in the latter case).

The Bellman equation: finite horizon

Problems (3) and (4) are intractable in general, but we may take advantage of the Markovian structure of the problem in order to decompose it with respect to time, obtaining a sequence of single period subproblems.

By "Markovian structure" we mean the structure of the state transition equation (1) and the assumption of intertemporal independence of the disturbance terms.

We refrain from giving a rigorous treatment, but the intuition behind DP is fairly easy to grasp. Let us define the *value function* $V_t(\mathbf{s}_t)$, for time $t = 0, 1, \ldots, T$, which is the optimal value obtained by applying an optimal policy starting from state $\mathbf{s}_t$ at time t.

Then, the following Bellman equation allows us to find the set of value functions:

$$V_t(\mathbf{s}_t) = \max_{\mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t)} \{ f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta \mathrm{E}[V_{t+1}(\mathbf{g}_t(\mathbf{s}_t, \mathbf{x}_t, \epsilon_{t+1})) \mid \mathbf{s}_t, \mathbf{x}_t] \} \tag{5}$$

THE BELLMAN EQUATION: FINITE HORIZON

Eq. (5) is a recursive functional equation: if we knew the value function $V_{t+1}(\mathbf{s}_{t+1})$, we could solve a set of one-step optimization problems in order to find the value function $V_t(\mathbf{s}_t)$ for each $\mathbf{s}_t$.

This requires unfolding the recursion *backward* from the last time period. If a terminal state value $F_{T+1}(\mathbf{s}_{T+1})$ is given, then clearly we start from the terminal condition $V_{T+1}(\mathbf{s}_{T+1}) = F_{T+1}(\mathbf{s}_{T+1})$.

The first problem we have to solve is:

$$V_T(\mathbf{s}_T) = \max_{\mathbf{x}_T \in \mathcal{X}(\mathbf{s}_T)} \{f_T(\mathbf{s}_T, \mathbf{x}_T) + \beta \mathrm{E}[V_{T+1}(\mathbf{g}_T(\mathbf{s}_T, \mathbf{x}_T, \epsilon_{T+1}))]\} \qquad (6)$$

for all values of the state variable $\mathbf{s}_T$ in order to find the value function $V_T(\mathbf{s}_T)$.

Given V_T we step backwards and find V_{T-1}:

$$V_{T-1}(\mathbf{s}_{T-1}) = \max_{\mathbf{x}_{T-1} \in \mathcal{X}(\mathbf{s}_{T-1})} \{f_{T-1}(\mathbf{s}_{T-1}, \mathbf{x}_{T-1}) + \beta \mathrm{E}[V_T(\mathbf{g}_{T-1}(\mathbf{s}_{T-1}, \mathbf{x}_{T-1}, \epsilon_T))]\}$$

THE BELLMAN EQUATION: FINITE HORIZON

We proceed recursively and finally find the value function V_1:

$$V_1(\mathbf{s}_1) = \max_{\mathbf{x}_1 \in \mathcal{X}(\mathbf{s}_1)} \{f_1(\mathbf{s}_1, \mathbf{x}_1) + \beta \mathrm{E}[V_2(\mathbf{g}_1(\mathbf{s}_1, \mathbf{x}_1, \epsilon_2))]\} \tag{7}$$

Given V_1 we find the optimal decision *now*, $\mathbf{x}_0^*$, for the given initial state $\mathbf{s}_0$:

$$\max_{\mathbf{x}_0 \in \mathcal{X}(\mathbf{s}_0)} \{f_0(\mathbf{s}_0, \mathbf{x}_0) + \beta \mathrm{E}[V_1(\mathbf{g}_0(\mathbf{s}_0, \mathbf{x}_0, \epsilon_1))]\} \tag{8}$$

It is also important to notice that the knowledge of the value functions allows us to derive and apply the optimal policy in feedback form:

- starting from the initial state $\mathbf{s}_0$ and based on $V_1(\mathbf{s})$, we solve an optimization problem to obtain the optimal decision $\mathbf{x}_0^*$;
- we apply $\mathbf{x}_0^*$ and, given the realization ϵ_1, we end up in the next state $\mathbf{s}_1$;
- there, given $V_2(\mathbf{s})$, we find $\mathbf{x}_1^*$, and so on.

The Bellman equation: infinite horizon

In the case of an infinite horizon, we usually suppress time dependence from the immediate contribution function f and the state transition function $\mathbf{g}$.

In this case the Bellman equation essentially calls for the determination of a time-independent value function as a fixed point:

$$V(\mathbf{s}) = \max_{\mathbf{x} \in \mathcal{X}(\mathbf{s})} \{f(\mathbf{s}, \mathbf{x}) + \beta \mathrm{E}[V(g(\mathbf{s}, \mathbf{x}, \epsilon))]\} \tag{9}$$

In this case, too, we skip issues related to existence and uniqueness of the value function, and take for granted that by finding the value function we do solve the original dynamic optimization problem (for a more rigorous treatment see, e.g., Bertsekas).

THE THREE CURSES OF DIMENSIONALITY

The DP principle, although extremely appealing, suffers from severe computational difficulties.

Solving the Bellman equation with finite time horizon, discrete and finite state and action spaces, and a known matrix of transition probabilities is fairly easy.

However, when we deal with more complicated problems we have to face three curses of dimensionality:

1. when the state space is multidimensional, the value function is a difficult object to manage (more so, when the state space is continuous and we need some form of discretization);
2. when the disturbance is a multidimensional random variable, the expectation in the recursive equations is tough to compute;
3. the single-step optimization problem itself may have to cope with many decision variables and possibly difficult constraints.

THE FINITE HORIZON CASE

In the continuous state case, we have to apply some form of discretization. One possibility is to discretize the state space using a grid. Given an approximation $\widehat{V}_{t+1}(\mathbf{s})$ for states on the grid, we may use, say, interpolation with cubic splines to obtain values outside the grid.

Another form of discretization we need is sampling, in order to approximate the expectation with respect to the disturbance term ϵ_{t+1}. We may generate scenarios with values ϵ_{t+1}^s and probability π^s (see Lecture 5).

Then, given the approximation $\widehat{V}_{t+1}(\mathbf{s})$, we use numerical optimization methods to solve

$$\widehat{V}_t(\mathbf{s}_t) = \max_{\mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t)} \left\{ f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta \sum_{s=1}^{S} \pi^s \widehat{V}_{t+1} \left(\mathbf{g}_t(\mathbf{s}_t, \mathbf{x}_t, \epsilon_{t+1}^s) \right) \right\}$$

yielding the value function at time t for states $\mathbf{s}_t$ on the grid.

THE INFINITE HORIZON CASE: COLLOCATION METHOD

The infinite horizon case can be dealt with by projecting the value function on a linear space spanned by a set of basis functions and using the collocation method.

The collocation method requires choosing a set of basis functions to approximate the value function:

$$V(\mathbf{s}) \approx \sum_{j=1}^{n} c_j \phi_j(\mathbf{s}).$$

Cubic splines could be a choice, but there are alternatives.

Given collocation nodes $\mathbf{s}_1, \ldots, \mathbf{s}_n$, the Bellman equation for each state $\mathbf{s}_i$ reads:

$$\sum_{j=1}^{n} c_j \phi_j(\mathbf{s}_i) = \max_{\mathbf{x} \in \mathcal{X}(\mathbf{s}_i)} \left\{ f(\mathbf{s}_i, \mathbf{x}) + \beta \, \mathrm{E} \left[\sum_{j=1}^{n} c_j \phi_j \left(\mathbf{g}(\mathbf{s}_i, \mathbf{x}, \epsilon) \right) \right] \right\} \qquad (10)$$

THE INFINITE HORIZON CASE: COLLOCATION METHOD

Eqs. (10) are a set of nonlinear equations in the unknown coefficients c_j:

$$\Phi \mathbf{c} = \boldsymbol{\nu}(\mathbf{c}),$$

where Φ is the collocation matrix with elements $\Phi_{ij} = \phi_j(\mathbf{s}_i)$, and the collocation function $\boldsymbol{\nu}(\mathbf{c})$ is a vector function with components:

$$\nu_i(\mathbf{c}) = \max_{\mathbf{x} \in \mathcal{X}(\mathbf{s}_i)} \left\{ f(\mathbf{s}_i, \mathbf{x}) + \beta \, \mathrm{E} \left[\sum_{j=1}^{n} c_j \phi_j \left(\mathbf{g}(\mathbf{s}_i, \mathbf{x}, \epsilon) \right) \right] \right\}.$$

Newton's method for nonlinear equations yields the iteration scheme:

$$\mathbf{c} \leftarrow \mathbf{c} - \left[\Phi - \boldsymbol{\Theta}(\mathbf{c}) \right]^{-1} \left[\Phi \mathbf{c} - \boldsymbol{\nu}(\mathbf{c}) \right],$$

where the Jacobian matrix $\boldsymbol{\Theta}(\mathbf{c})$ may be computed by the envelope theorem,

$$\theta_{ij}(\mathbf{c}) = \frac{\partial \nu_i}{\partial c_j}(\mathbf{c}) = \beta \, \mathrm{E}[\phi_j \left(\mathbf{g}(\mathbf{s}_i, \mathbf{x}_i, \epsilon) \right)],$$

and $\mathbf{x}_i$ is the optimal decision for the optimization problem we solve to get the collocation function value $\nu_i(\mathbf{c})$.

THE INFINITE HORIZON CASE: COLLOCATION METHOD

In practice, conditional expectation is approximated by quadrature formulae. Assume weights (probabilities) and nodes (discretized shock values) are given by π^s and ϵ^s, respectively, $s = 1, \ldots, S$.

Then we solve the following optimization problem to evaluate the collocation function:

$$\nu_i(\mathbf{c}) = \max_{\mathbf{x} \in \mathcal{X}(\mathbf{s}_i)} \left\{ f(\mathbf{s}_i, \mathbf{x}) + \beta \sum_{s=1}^{S} \sum_{j=1}^{n} \pi^s c_j \phi_j \left(\mathbf{g}(\mathbf{s}_i, \mathbf{x}, \epsilon^s) \right) \right\}$$

with Jacobian

$$\theta_{ij}(c) = \beta \sum_{s=1}^{S} \pi^s \phi_j \left(\mathbf{g}(\mathbf{s}_i, \mathbf{x}_i, \epsilon^s) \right).$$

See (Judd, 1998) or (Miranda and Fackler, 2002) for additional information on computational DP in economics, or (Buşoniu et al., 2010) for engineering applications.

ADP: Learning by Monte Carlo methods

Numerical DP allows to use a powerful principle to a set of stochastic problems with limited dimensionality.

With respect to stochastic programming, we have the advantage of getting the solution in (implicit) feedback form; the disadvantage is that we have to impose some constraints on the structure of the disturbance terms.

However, large-scale problems cannot be tackled by standard discretization, and we must somehow resort to Monte Carlo methods.

A fundamental insight is that we really need only a decent approximation of the value function to find good, possibly near optimal policies.

We may use Monte Carlo methods and reinforcement learning to find such an approximation.

ADP: Learning by Monte Carlo methods

A comprehensive reference for ADP is (Powell, 2011), where several variants of learning algorithms are described.

There are a few features distinguishing ADP from the numerical approaches previously described:

1. we try to learn the expected value function *directly*, using optimality equations based on post-decision state variables;
2. typical learning algorithms use forward passes, rather than backward recursion;
3. we may use batch or recursive learning algorithms, based on statistical estimation.

Optimality equations around the post-decision state variables

The typical history of a stochastic decision process is:

$$(\mathbf{s}_0, \mathbf{x}_0, \epsilon_1, \mathbf{s}_1, \mathbf{x}_1, \epsilon_2, \ldots, \mathbf{s}_{t-1}, \mathbf{x}_{t-1}, \epsilon_t, \ldots)$$

Let us introduce a post-decision state variable $\mathbf{s}_t^x$, which represents the state of the system after our decision $\mathbf{x}$, but *before* the realization of the disturbance.

The state evolution is now:

$$\left(\mathbf{s}_0, \mathbf{x}_0, \mathbf{s}_0^x, \epsilon_1, \mathbf{s}_1, \mathbf{x}_1, \mathbf{s}_1^x, \epsilon_2, \ldots, \mathbf{s}_{t-1}, \mathbf{x}_{t-1}, \mathbf{s}_{t-1}^x, \epsilon_t, \ldots\right)$$

The exact definition of $\mathbf{s}_t^x$ depends on the context. As an example, it could be the inventory state after we replenish, but before we satisfy demand.

Rather than writing optimality equations around $V_t(\mathbf{s}_t)$, we may use $V_t^x(\mathbf{s}_t^x)$, the value of being in state $\mathbf{s}_t^x$ just after the decision.

Optimality equations around the post-decision state variables

We have

$$V^x_{t-1}(\mathbf{s}^x_{t-1}) = \mathrm{E}\left[V_t(\mathbf{s}_t) \mid \mathbf{s}^x_{t-1}\right] \tag{11}$$

$$V_t(\mathbf{s}_t) = \max_{\mathbf{x}_t} \left(f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta V^x_t(\mathbf{s}^x_t)\right] \tag{12}$$

$$V^x_t(\mathbf{s}^x_t) = \mathrm{E}\left[V_{t+1}(\mathbf{s}_{t+1}) \mid \mathbf{s}^x_t\right] \tag{13}$$

Clearly, we obtain the standard Bellman equation by plugging (13) into (12):

$$V_t(\mathbf{s}_t) = \max_{\mathbf{x}_t} \left[f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta \mathrm{E}[V_{t+1}(\mathbf{s}_{t+1})]\right)$$

But if we substitute (12) into (11) we find

$$V^x_{t-1}(\mathbf{s}^x_{t-1}) = \mathrm{E}\left\{\max_{\mathbf{x}_t} \left[f_t(\mathbf{s}_t, \mathbf{x}_t) + \beta V^x_t(\mathbf{s}^x_t)\right]\right\} \tag{14}$$

Big advantage: we have to solve a deterministic optimization problem.

American option pricing by Monte Carlo

We illustrate Monte Carlo-based DP with a simple example: pricing an American-style put option on a non-dividend paying stock share.

Let $S(t)$ be the price of a stock share. A possible continuous-time model of prices is Geometric Brownian Motion (GBM), described by the stochastic differential equation:

$$dS(t) = \mu S(t)\, dt + \sigma S(t)\, dW(t),$$

where μ is a drift, σ is a volatility, and $W(t)$ is a standard Wiener process.

Using tools from stochastic calculus (Ito's lemma), we may discretize the differential equation and simulate the process:

$$S_j = S_{j-1} \exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)\delta t + \sigma\sqrt{\delta t}\epsilon_j\right\}$$

where $S_j \equiv S(j \cdot \delta t)$ for some discretization time step δt, and variables ϵ_j, $j = 1, \ldots, T$, are a sequence of independent standard normals.

American option pricing by Monte Carlo

A put option on a stock share gives the right to sell a the underlying asset at a given strike price K fixed at $t = 0$.

Clearly, the option is exercised only if it is "in the money", i.e., when the intrinsic value of the option $K - S(t)$ is non-negative. The option payoff is $\max\{0, K - S(t)\}$.

European-style options can only be exercised at maturity $t = T$; American-style options can be exercised at any time $t \in [0, T]$ before expiration.

Using no arbitrage arguments, it can be shown that options should be priced by taking the expected value of the payoff *under a risk neutral* measure. In the GBM case, this boils down to replacing the drift μ by the (continuously compounded) risk-free rate r.

American option pricing by Monte Carlo

Pricing an American-style option entails solving an optimal stopping problem: when the option is in the money, we have to decide if we exercise immediately, earning the intrinsic value, or wait for better opportunities.

This is a dynamic stochastic optimization problem that can be tackled by dynamic programming, since GBM is a Markov process.

The state is (in principle) continuous, but the set of decisions is quite simple: either we exercise or we continue.

Nevertheless, pricing high-dimensional American options is a tough problem. We illustrate an ADP approach proposed in (Longstaff and Schwartz, 2001), based on the approximation of the value function by linear regression.

American option pricing by Monte Carlo

Using simple Monte Carlo, we generate sample paths $(S_0, S_1, \ldots, S_j, \ldots, S_N)$, where $T \cdot \delta t = N$ is the option expiration.

Let $I_j(S_j) = \max\{K - S_j, 0\}$ be the payoff from the immediate exercise of the option at time j, and $V_j(S_j)$ be the value of the option at time j when state is S_j.

The value of the option is the maximum between the immediate payoff and the continuation value. But the value of the continuation is the discounted expected value of the option value at time $j+1$, under the risk neutral measure $\mathbb{Q}$:

$$\mathrm{E}_j^{\mathbb{Q}}\left[e^{-r\cdot\delta t} V_{j+1}(S_{j+1}) \middle| S_j \right]$$

The dynamic programming recursion for the value function $V_j(S_j)$ is

$$V_j(S_j) = \max\left\{ I_j(S_j)\,,\, \mathrm{E}_j^{\mathbb{Q}}\left[e^{-r\cdot\delta t} V_{j+1}(S_{j+1}) \middle| S_j \right] \right\}. \tag{15}$$

APPROXIMATION BY LINEAR REGRESSION

We are going to approximate the conditional expectation inside (15) directly as a function of S_j, using linear regression and a set of basis functions $\psi_k(S_j)$, $k = 1, \ldots, L$:

$$\mathrm{E}_j^{\mathbb{Q}}\left[e^{-r\cdot\delta t} V_{j+1}(S_{j+1}) \middle| S_j \right] \approx \sum_{k=1}^{L} \alpha_{kj} S_j^{k-1}.$$

We use the same set of basis function for each time instant, but the weights in the linear combination do depend on time.

The simplest choice we can think of is regressing the conditional expectation against a basis of monomials: $\psi_1(S) = 1$, $\psi_2(S) = S$, $\psi_3(S) = S^2$, etc. The approximation is nonlinear in S_j, but it is linear in terms of the weights.

Also note that:

- the approach corresponds to DP with post-decision state variables;
- linear regression on functions of S_{j-1} ensures nonanticipativity of the resulting policy.

THE BACKWARD RECURSION

In order to illustrate the method, we should start from the last time period. Assume we have generated N sample paths, and let us denote by S_{ji} the price at time j on sample path $i = 1, \ldots, N$.

When $j = M$, i.e., at expiration, the value function is trivially:

$$V_M(S_{Mi}) = \max\{K - S_{Mi}, 0\}$$

for each sample path i. These values can be used, in a sense, as the Y-values in a linear regression, where the X values are the prices at time $j = M - 1$.

More precisely, we consider the regression model:

$$e^{-r\cdot\delta t} \max\{K - S_{Mi}, 0\} = \sum_{k=1}^{L} \alpha_{k,M-1} S_{M-1,i}^{k-1} + e_i, \qquad i = 1, \ldots, N,$$

where e_i is the residual for each sample path and weights $\alpha_{k,M-1}$ are obtained by ordinary least squares.

THE MONEYNESS CRITERION

In the original paper it is suggested to include only the subset of sample paths for which we have a decision to make at time $j = M - 1$, i.e., the sample paths in which the option is in the money at time $j = M - 1$ (the authors refer to the idea as "moneyness criterion," and its advantage is debated).

Denoting this subset by $\mathcal{I}_{M-1}$ and assuming $L = 3$, we solve the following least squares problem:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{I}_{M-1}} e_i^2 \\ \text{s.t.} \quad & \alpha_{1,M-1} + \alpha_{2,M-1} S_{M-1,i} + \alpha_{3,M-1} S_{M-1,i}^2 + e_i \\ & \qquad = e^{-r \cdot \delta t} \max\{K - S_{Mi}, 0\}, \qquad i \in \mathcal{I}_{M-1}. \end{aligned} \tag{16}$$

The output of this problem is a set of weights, which allow us to approximate the continuation value. Note that the weights are linked to the time period, and not to sample paths. Using the same approximation for each sample path in $\mathcal{I}_{M-1}$, we may decide if we exercise or not.

A NUMERICAL EXAMPLE

Assume that we must price an American put with strike price $K = 1.1$ and that eight sample paths have been sampled.

Path	$j = 0$	$j = 1$	$j = 2$	$j = 3$
1	1.00	1.09	1.08	1.34
2	1.00	1.16	1.26	1.54
3	1.00	1.22	1.07	1.03
4	1.00	0.93	0.97	0.92
5	1.00	1.11	1.56	1.52
6	1.00	0.76	0.77	0.90
7	1.00	0.92	0.84	1.01
8	1.00	0.88	1.22	1.34

Path	$j = 1$	$j = 2$	$j = 3$
1	-	-	.00
2	-	-	.00
3	-	-	.07
4	-	-	.18
5	-	-	.00
6	-	-	.20
7	-	-	.09
8	-	-	.00

For each sample path, we also have a set of cash flows at expiration; cash flows are positive where the option is in the money.

A NUMERICAL EXAMPLE

Cash flows are discounted back to time $j = 2$ and used for the first linear regression. Assuming a risk free rate of 6% per period, the discount factor is $e^{-0.06} = 0.94176$.

The data for the regression are given in the table below: X corresponds to current underlying asset price and Y corresponds to discounted cash flows in the future.

Path	Y	X
1	$.00 \times .94176$	1.08
2	-	-
3	$.07 \times .94176$	1.07
4	$.18 \times .94176$	0.97
5	-	-
6	$.20 \times .94176$	0.77
7	$.09 \times .94176$	0.84
8	-	-

Only the sample paths in which the option is in the money at time $j = 2$ are used. The following approximation is obtained:

$$\mathrm{E}[Y \mid X] \approx -1.070 + 2.983X - 1.813X^2.$$

A NUMERICAL EXAMPLE

Now we may compare at time $j = 2$ the intrinsic value and the continuation value.

Path	Exercise	Continue
1	.02	.0369
2	-	-
3	.03	.0461
4	.13	.1176
5	-	-
6	.33	.1520
7	.26	.1565
8	-	-

Path	$j = 1$	$j = 2$	$j = 3$
1	-	.00	.00
2	-	.00	.00
3	-	.00	.07
4	-	.13	.00
5	-	.00	.00
6	-	.33	.00
7	-	.26	.00
8	-	.00	.00

Given the exercise decisions, we update the cash flow matrix.
Note that the exercise decision does not exploit knowledge of the future.
Consider sample path 4: we exercise, making \$0.13; on that sample path, we would regret our decision, because we could make \$0.18 at time $j = 3$.
We should also note that on some paths we exercise at time $j = 2$, and this is reflected by the updated cash flow matrix in the table.

A NUMERICAL EXAMPLE

To carry out the regression, we must consider the cash flows on each path, resulting from early exercise decisions. If we are at time j, for each sample path i there is an exercise time j_e^* (set conventionally to $M+1$ if the option will never be exercised in the future).

Then the regression problem (16) should be rewritten, for the generic time period j, as:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{I}_j} e_i^2 \\ \text{s.t.} \quad & \alpha_{1j} + \alpha_{2j} S_{ji} + \alpha_{3j} S_{ji}^2 + e_i \\ & \quad = \begin{cases} e^{-r(j_e^* - j)\,\delta t} \max\{K - S_{j_e^*,i}\,,\, 0\} & \text{if } j_e^* \leq M \\ 0 & \text{if } j_e^* = M+1 \end{cases} \qquad i \in \mathcal{I}_j. \end{aligned} \tag{17}$$

Since there can be at most one exercise time for each path, it may be the case that after comparing the intrinsic value with the continuation value on a path, the exercise time j_e^* is reset to a previous period.

A NUMERICAL EXAMPLE

Stepping back to time $j = 1$, we have the following regression data:

Path	Y	X
1	$.00 \times .88692$	1.09
2	-	-
3	-	-
4	$.13 \times .94176$	0.93
5	-	-
6	$.33 \times .94176$	0.76
7	$.26 \times .94176$	0.92
8	$.00 \times .88692$	0.88

The discount factor $e^{-2 \cdot 0.06} = 0.88692$ is applied on paths 1 and 8. Since the cash flow there is zero, the discount factor is irrelevant, but note that we are discounting cash flows from time period $j = 3$.
Least squares yield the approximation:

$$\mathrm{E}[Y \mid X] \approx 2.038 - 3.335X + 1.356X^2.$$

A NUMERICAL EXAMPLE

Based on this approximation of the continuation value, we obtain the following exercise decisions:

Path	Exercise	Continue
1	.01	.0139
2	-	-
3	-	-
4	.17	.1092
5	-	-
6	.34	.2866
7	.18	.1175
8	.22	.1533

Path	$j = 1$	$j = 2$	$j = 3$
1	.00	.00	.00
2	.00	.00	.00
3	.00	.00	.07
4	.17	.00	.00
5	.00	.00	.00
6	.34	.00	.00
7	.18	.00	.00
8	.22	.00	.00

Discounting all cash flows back to time $j = 0$ and averaging over the eight sample paths, we get an estimate of the continuation value of \$0.1144, which is larger than the intrinsic value \$0.1; hence, the option should not be exercised immediately.