Weierstraß-Institut für
Angewandte Analysis und Stochastik

Leibniz-Gemeinschaft

Optimal stopping and control via reinforced regression

Vladimir Spokoiny, WIAS and HU (Berlin)

joint with Denis Belomestny (Duisburg-Essen), John Schoenmakers (WIAS Berlin) and Yury Tavyrikov (Moscow)

Table of contents

1. **Introduction**
   - Bellman principle
   - Regression methods
2. **Backward reinforced regression algorithm**
3. **Numerical examples**
4. **Some theoretical arguments**
5. **Optimal control**
6. **Summary and outlook**

Outline

1 **Introduction**

- Bellman principle
- Regression methods

2 **Backward reinforced regression algorithm**

3 **Numerical examples**

4 **Some theoretical arguments**

5 **Optimal control**

6 **Summary and outlook**

Introduction

An optimal stopping problem: for a observed Markov process $X_t \sim (\mathcal{F}_t)$, find a stopping time τ to maximize

$$\mathbb{E} g_\tau(X_\tau)$$

for a given reward function $g_t(X_t)$.

In many financial applications, $X_t \in \mathbb{R}^d$ for d relatively large, e.g. 40 or 60.

Dynamic Programming Principle

Let $\tau \in \mathcal{J} = (t_1, \ldots, t_J) \subset \{1, 2, \ldots, T\}$. Denote $Z_j = X_{t_j}$.
Let $\mathcal{T}_j$ be the set of stopping times valued in $\{j, j+1, \ldots, J\}$. Consider the optimal stopping problems

$$V_j(x) = \sup_{\tau \in \mathcal{T}_j} \mathbb{E}[g_\tau(Z_\tau) | Z_j = x], \quad x \in \mathbb{R}^d,$$

Define the continuation value

$$C_j(x) = \mathbb{E}[V_{j+1}(Z_{j+1}) | Z_j = x], \quad j < J,$$

with $V_J(x) = g_J(x), C_J(x) = 0$.
Optimal decision: terminate at j if $g_j(Z_j) \geq C_j(Z_j)$. Yields

$$V_j(x) = \max\left(g_j(x), C_j(x)\right), \quad 1 \leq j \leq J-1,$$

Numerical implementation

Even if $V_{j+1}(x)$ is given, the step

$$C_j(x) = \mathbb{E}[V_{j+1}(Z_{j+1})|Z_j = x], \quad j < J,$$

involves high dimensional integration. Forward plain (nested) Monte-Carlo methods are unfeasible.

Existing approaches:

- functional optimization approach [Andersen, 1999],
- mesh method of [Broadie and Glasserman, 1997],
- regression-based approaches of [Carriere, 1996], [Longstaff and Schwartz, 2001], [Tsitsiklis and Van Roy, 2001], [Egloff et al., 2005] and [Belomestny, 2011].
- Deep optimal stopping [Becker et al., 2018].

Regression methods. I

An estimate $C_{N,j+1}(x)$ of the continuation value $C_{j+1}(x)$ allows to use a stopping rule

$$\tau_N = \min\{1 \le j \le J \colon g_j(Z_j) \ge C_{N,j}(Z_j)\},$$

with $C_{N,J} \equiv 0$ by definition. This (suboptimal) rule provides a lower bound for the value V_0.

Suppose that for some $1 \le j < J,$ an estimate $C_{N,j+1}(x)$ for $C_{j+1}(x)$ is already constructed. Then in the j th step one needs to estimate the conditional expectation

$$\begin{aligned} C_{N,j}(x) &= \mathbb{E}\left[V_{N,j+1}(Z_{j+1}) \,\middle|\, Z_j = x\right] \\ &= \mathbb{E}\left[\max\{g_{j+1}(Z_{j+1}), C_{N,j+1}(Z_{j+1})\} \,\middle|\, Z_j = x\right], \end{aligned}$$

with $C_{N,J}(x) \equiv 0$.

Regression methods. II

Suppose we are given a set of paths

$$(Z_j^{(m)}), \quad m = 1, \ldots, N.$$

Approach: Use nonparametric regression of $V_{N,j+1}(Z_{j+1})$ on Z_j to compute estimates $C_{N,j}(Z_j^{(m)})$.

Let $(\psi_1(x), \ldots, \psi_K(x))$ be a system of basis functions.

Initialize as $C_{N,J}(x) = 0$. For $j < J$, construct $C_{N,j}$ in the form

$$C_{N,j}(x) = \sum_{k=1}^{K} \gamma_k^{N,j} \psi_k(x) \quad \text{for some } \gamma^{N,j} \in I\!R^K.$$

The coefficients $\gamma_k^{N,j}$ are estimated using linear regression.

For going from $j+1>0$ down to j, define the $N\times K$ design matrix $\mathcal{M}_j$

$$\mathcal{M}_j = \Big(\psi_k(Z_j^{(m)}),\ m=1,\dots,N,\ k=1,\dots,K\Big),$$

and the (column) vector

$$\mathcal{V}_{j+1} = \Big(V_{N,j+1}(Z_{j+1}^{(1)}),\dots,V_{N,j+1}(Z_{j+1}^{(N)})\Big)^{\top},$$

$$V_{N,j+1}(Z_{j+1}^{(m)}) = \max\{g_{j+1}(Z_{j+1}^{(m)}), C_{N,j+1}(Z_{j+1}^{(m)})\}.$$

Next compute

$$\gamma^{N,j} = \underset{\gamma}{\operatorname{arginf}}\left\|\mathcal{V}_{j+1}-\mathcal{M}_j\gamma\right\|^2 = \Big(\mathcal{M}_j^{\top}\mathcal{M}_j\Big)^{-1}\mathcal{M}_j^{\top}\mathcal{V}_{j+1},$$

$$C_{N,j}(x) = \sum_{k=1}^{K}\gamma_k^{N,j}\psi_k(x).$$

Regression methods. Summary

Regression method: approximate the continuation function

$$C_j(x) = \mathbb{E}\left[V_{j+1}(Z_{j+1}) \,\middle|\, Z_j = x\right] \approx C_{N,j}(x) = \sum_{k=1}^{K} \gamma_k^{N,j} \psi_k(x) \qquad (1)$$

using the values $x = Z_j^{(m)}$ and

$$V_{N,j+1}(Z_{j+1}^{(m)}) = \max\{g_{j+1}(Z_{j+1}^{(m)}), C_{N,j+1}(Z_{j+1}^{(m)})\}, \quad m = 1, \ldots, N.$$

Pros: Linear in J complexity vs exponential one for nested Monte-Carlo.

Cons: Choice of the basis $\psi_1, \ldots, \psi_K$ is crucial, otherwise the approximation (1) is too rough.

Outline

1 Introduction

- Bellman principle
- Regression methods

2 **Backward reinforced regression algorithm**

3 Numerical examples

4 Some theoretical arguments

5 Optimal control

6 Summary and outlook

Backward reinforced regression algorithm. I

Idea: start with a computationally cheep basis set $\psi_1, \ldots, \psi_K$.

At each step $j < J$, extend this set to a larger set

$$\psi_1, \ldots, \psi_K, \psi^{N,j}_{K+1}, \ldots, \psi^{N,j}_{K+b}$$

where the auxiliary functions $\psi^{N,j}_{K+1}, \ldots, \psi^{N,j}_{K+b}$ depend on the previously computed continuation functions $C_{N,j+1}$.

Running example:

VALUE (RELU): $\psi^{N,j}_{K+1}(x) = \max\{g_{j+1}(x), C_{N,j+1}(x)\} = V_{N,j+1}(x),$

POLICY (STAMP): $\psi^{N,j}_{K+2}(x) = 1\!\!\mathrm{I}\big(g_{j+1}(x) \geq C_{N,j+1}(x)\big).$

Backward reinforced regression algorithm. II

Suppose $C_{N,j+1}$ already constructed for $j < J$ and $\psi^{N,j}_{K+1}, \ldots, \psi^{N,j}_{K+b}$ fixed. Define $V_{N,j+1}(x) = \max(g_{j+1}(x), C_{N,j+1}(x))$,

$$\Psi^{N,j}(x) = \left(\psi_1(x), \ldots, \psi_K(x), \psi^{N,j}_{K+1}(x), \ldots, \psi^{N,j}_{K+b}(x)\right),$$

$$\mathcal{M}_j = \begin{pmatrix} \Psi^{N,j}(Z_j^{(1)}) \\ \vdots \\ \Psi^{N,j}(Z_j^{(N)}) \end{pmatrix}, \quad \mathcal{V}_{j+1} = \left(V_{N,j+1}(Z_{j+1}^{(1)}), \ldots, V_{N,j+1}(Z_{j+1}^{(N)})\right)^\top$$

and define

$$\gamma^{N,j} = \left(\mathcal{M}_j^\top \mathcal{M}_j\right)^{-1} \mathcal{M}_j^\top \mathcal{V}_{j+1},$$

$$C_{N,j}(x) = \gamma_1^{N,j}\psi_1(x) + \ldots + \gamma_K^{N,j}\psi_K(x) + \gamma_{K+1}^{N,j}\psi_{K+1}^{N,j}(x) + \ldots + \gamma_{K+b}^{N,j}\psi_{K+b}^{N,j}(x).$$

Backward reinforced regression algorithm. III

Details of implementation.

- RELU and STAMP type artificial features

$$\psi^{N,j}_{K+1}(x) = \max\{g_{j+1}(x), C_{N,j+1}(x)\} = V_{N,j+1}(x),$$
$$\psi^{N,j}_{K+a}(x) = 1\!\!1\big(g_{j+a}(x) \geq C_{N,j+a}(x)\big), \; j+1 < j+a < J.$$

- Once constructed at step j, compute and store the vector of coefficients $\gamma^{N,j}$ and the values

$$\psi^{N,j}_{K+a}(Z^{(m)}_l), \quad \ell \leq j, \, m = 1, \ldots, N.$$

Will be used at further steps $j-1, \ldots, 0$ for computing the values $C_{N,\ell}(Z^{(m)}_\ell)$, $m = 1, \ldots, N$.

Computational costs

- Pre-computation costs: $\frac{1}{2}N\mathcal{J}^2c_f + N\mathcal{J}Kc_f$, where c_f denotes the maximal cost of evaluating each function g_j, $j = 0, \ldots, J$, and ψ_k, $k = 1, \ldots, K$, at a given point.
- The cost of one backward step from $j+1$ to j can be then estimated from above by

  $$\begin{aligned} &NK^2c_* \quad \text{due to computation of } \gamma^{N,j} \\ &NKjc_* \quad \text{due to the construction of } C_{N,j} \end{aligned}$$

  where c_* denotes costs of addition and multiplication of two reals.
- Total cost of the above algorithm can be upper bounded by

  $$\frac{1}{2}NJ^2c_f + NJKc_f + NJK^2c_* + \frac{1}{2}NJ^2Kc_*$$

The procedure can be viewed as a new backward construction of a DNN.

- Start with the very last layer J and use ψ_k as nodes.
- At each step j build the layer j from static nodes ψ_k and dynamic nodes $\psi_k^{N,j}$.
- Use the coefficients $\gamma_k^{N,j}$ as DNN weights.
- Use dynamic programming (Bellman equation) as activating non-linear device.

Backward reinforced regression algorithm. Summary

The proposed approach yields a novel deep network architecture constructed by backward reinforced regression using the Bellman principle (continuation rule) as activating device.

Backward regression is an essential issue.

The DNN is sparse by construction. Very few features are constructed, each has a natural interpretation.

The procedure does not require parameter tuning by non-convex optimization

Computational costs are polynomial in the number of steps.

Outline

1 Introduction

- Bellman principle
- Regression methods

2 Backward reinforced regression algorithm

3 **Numerical examples**

4 Some theoretical arguments

5 Optimal control

6 Summary and outlook

Bermudan cancelable swap

We test our algorithm in the case of the so-called complex structured asset based cancelable swap.

Consider a multi-dimensional Black-Scholes model with d assets X_l, $l = 1, \ldots, d$, under the risk-neutral measure via a system of SDEs

$$dX_l(t) = (\rho - \delta)X_l(t)dt + \sigma_l X_l(t)dW_l(t), \quad 0 \leq t \leq T, \quad l = 1, \ldots, d.$$

Here $W_1(t), \ldots, W_d(t)$ are correlated d-dimensional Brownian motions with time independent correlations $\rho_{lm} = t^{-1}\mathbb{E}[W_l(t)W_m(t)]$, $1 \leq l, m \leq d.$

The continuously compounded interest rate r and a dividend rate δ are assumed to be constant.

Bermudan cancelable swap. II

Define the asset based cancelable coupon swap. Let $t_1, \ldots, t_{\mathcal{J}}$ be a sequence of exercise dates. Fix a quantile $\alpha,\ 0 < \alpha < 1$, numbers $1 \leq n_1 < n_2 \leq d$ (we assume $d \geq 2$), and three rates s_1, s_2, s_3. Let

$$N(i) = \#\{l : 1 \leq l \leq d,\ X_l(t_i) \leq (1-\alpha)X_l(0)\},$$

that is, $N(i)$ is the number of assets which at time t_i are below $1-\alpha$ percents of the initial value. We then introduce the random rate

$$a(i) = s_1 1_{\{N(i) \leq n_1\}} + s_2 1_{\{n_1 < N(i) \leq n_2\}} + s_3 1_{\{n_2 < N(i)\}}$$

and specify the t_i -coupon to be

$$C(i) = a(i)(t_i - t_{i-1}).$$

For pricing this structured product, we need to compare the coupons $C(i)$ with risk free coupons over the period $[t_{i-1}, t_i]$ and thus to consider the discounted net coupon process

$$\mathcal{C}(i) = e^{-rt_i}(e^{r(t_i - t_{i-1})} - 1 - C(i)), \quad i = 1, \ldots, \mathcal{J}.$$

The product value at time zero may then be represented as the solution of an optimal stopping problem with respect to the adapted discounted cash-flow, obtained as the aggregated net coupon process,

$$V_0 = \sup_{\tau \in \{1, \ldots, \mathcal{J}\}} \mathbb{E}[\mathcal{Z}_\tau], \quad \mathcal{Z}_j := \sum_{i=1}^{j} \mathcal{C}(i).$$

Bermudan cancelable swap. IV

For our experiments, we choose a five-year option with semiannual exercise possibility, that is, we have

$$\mathcal{J} = 10, \quad t_i - t_{i-1} = 0.5, \quad 1 \leq i \leq 10,$$

on a basket of $d = 20$ assets. In detail, we take the following values for the parameters,

$$d = 20, \quad r = 0.05, \quad \delta = 0, \quad \sigma_l = 0.2, \quad X_l(0) = 100, \quad 1 \leq l, m \leq 20,$$
$$d_1 = 5, \quad d_2 = 10, \quad \alpha = 0.05, \quad s_1 = 0.09, \quad s_2 = 0.03, \quad s_3 = 0,$$

and

$$\rho_{lm} = \begin{cases} \rho, & l \neq m, \\ 1, & l = m. \end{cases}$$

As to the basis functions, we used a constant, the discounted net coupon process $\mathcal{C}(i)$ and the order statistics $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$.

Bermudan cancelable swap. V

ρ	Basis functions	Linear regression		Linear regression & $\nu_1^{N,l}$	
		Low	High	Low	High
0	$1, \mathcal{C}, X_{(i)}$	171.6(.037)	177.2(.061)	173.3(.031)	177.3(.091)
	$1, \mathcal{C}, X_{(i)}, X_{(i)}X_{(j)}$	173.6(.044)	177.3(.062)	174.3(.036)	176.6(.057)
0.2	$1, \mathcal{C}, X_{(i)}$	180.0(.060)	199.6(.125)	187.6(.057)	195.1(.121)
	$1, \mathcal{C}, X_{(i)}, X_{(i)}X_{(j)}$	188.0(.055)	197.0(.143)	188.1(.046)	196.0(.108)
0.5	$1, \mathcal{C}, X_{(i)}$	176.4(.073)	201.2(.189)	182.0(.047)	194.0(.088)
	$1, \mathcal{C}, X_{(i)}, X_{(i)}X_{(j)}$	183.4(.033)	196.6(.147)	182.9(.057)	195.0(.127)
0.8	$1, \mathcal{C}, X_{(i)}$	133.3(.065)	158.1(.197)	138.4(.087)	153.1(.106)
	$1, \mathcal{C}, X_{(i)}, X_{(i)}X_{(j)}$	140.2(.061)	153.5(.106)	139.6(.035)	152.6(.096)

Tabelle: Comparison of the standard linear regression method and the reinforced regression algorithm for the problem of pricing cancelable swaps

Outline

1. Introduction
   - Bellman principle
   - Regression methods
2. Backward reinforced regression algorithm
3. Numerical examples
4. **Some theoretical arguments**
5. Optimal control
6. Summary and outlook

Regression vs reinforced regression

Data $\left(X^{(m)}, Y^{(m)}\right)$ i.i.d., $X \sim \mu$.

Target $u(x) = \mathbb{E}\left(Y \,\middle|\, X = x\right)$.

Standard basis $\psi_1(x), \ldots, \psi_K(x)$, $\Psi = \left(\psi_k(X^{(m)})\right)$,

$$\widetilde{\beta} = \operatorname*{arginf}_{\beta} \left\| \boldsymbol{Y} - \Psi\beta \right\|^2 = \left(\Psi^\top \Psi\right)^{-1} \Psi^\top \boldsymbol{Y},$$

$$\widetilde{u}(x) = \Psi(x)\widetilde{\beta}.$$

Extended design

$$\widehat{\Psi} = \psi_1(X^{(m)}), \ldots, \psi_K(X^{(m)}), \psi^N_{K+1}(X^{(m)}), \ldots, \psi^N_{K+b}(X^{(m)})$$

$$\widehat{\beta} = \operatorname*{arginf}_{\beta} \left\| \boldsymbol{Y} - \widehat{\Psi}\beta \right\|^2 = \left(\widehat{\Psi}^\top \widehat{\Psi}\right)^{-1} \widehat{\Psi}^\top \boldsymbol{Y},$$

$$\widehat{u}(x) = \widehat{\Psi}(x)\widehat{\beta}.$$

Theorem

Suppose that $X \sim \mu$,

$$\sup_{x\in\mathbb{R}^d} |u(x)| \leq L, \qquad \sup_{x\in\mathbb{R}^d} \mathrm{Var}\left[Y \,|\, X = x\right] \leq \sigma^2,$$

then it holds with probability at least $1 - \varepsilon$

$$\int |\widetilde{u}(x) - u(x)|^2 \, \mu(dx) \lesssim \max\left(\sigma^2, L^2\right) \frac{(1 + \ln N)K + \log(\varepsilon^{-1})}{N} + \inf_{w\in\Psi_K} \int_{\mathbb{R}^d} |w(x) - u(x)|^2 \, \mu(dx)$$

where $\Psi_K :=$ span $\{\psi_1, \ldots, \psi_K\}$ *.*

Theorem

Suppose that $X \sim \mu$ *,* $\operatorname{Var}[Y \mid X = x] \leq \sigma^2$ *. Then it holds with probability at least* $1 - \varepsilon$

$$\int |\widetilde{u}(x) - u(x)|^2 \mu(dx) \lesssim$$

$$\inf_{w \in \Psi_K \cup \mathcal{V}_b} \Bigg[\max\left(\sigma^2, L_w^2\right) \frac{(1 + \ln N)K + \log(\varepsilon^{-1})}{N}$$

$$+ \int_{\mathbb{R}^d} |w(x) - u(x)|^2 \mu(dx) \Bigg]$$

where $\Psi_K := \operatorname{span}\{\psi_1, \ldots, \psi_K\}$ *,* $\mathcal{V}_b := \operatorname{span}\big\{\widehat{\psi}_{K+1}, \ldots, \widehat{\psi}_{K+b}\big\}$ *and*

$$L_w = \sup\big|u(x) - w(x)\big|.$$

Main issues for theoretical study:

- the auxiliary basis functions $\psi_k^{N,j}(x)$ are random and data dependent.

  So far we assume that for each j we use a separate set $\left(Z_j^{(m)}\right)$. Yields a linear in J increase of numerical complexity.

- The starting basis $\psi_1, \ldots, \psi_K$ is essential. Otherwise the value functions $V_{N,j+1}(x)$ are not informative enough and no (or minor) improvement by reinforcing.

Outline

1. Introduction
   - Bellman principle
   - Regression methods
2. Backward reinforced regression algorithm
3. Numerical examples
4. Some theoretical arguments
5. **Optimal control**
6. Summary and outlook

Optimal control with reinforced regression. I

Let Z_j be a controlled Markov process, where $u_j \in \mathcal{U}$ is a control, $u_j = u_j(Z_j)$.

Objective functional (reward)

$$g = g(Z_1, \ldots, Z_J),$$

$$V^* = \sup_{u(\cdot)} \mathbb{E} g(Z_1, \ldots, Z_J).$$

Decomposition of the reward: for each j

$$g(Z_1, \ldots, Z_J) = g_{1,j-1}(Z_{1,j-1}) + g_{j,J}(Z_{j,J})$$

with $Z_{j,J} = (Z_j, \ldots, Z_J)$. Typical examples:

- $g(Z_1, \ldots, Z_J) = g_1(Z_1) + \ldots + g_J(Z_J)$;
- $g(Z_1, \ldots, Z_J) = g(Z_J)$.

For $j \le J$, define $U_{j,J} = \{u_{j,J} = (u_j, \dots, u_J)\}$ as the set of all policies from step j, $u_{j,J} = u_{j,J}(Z_j)$.

Dynamic programming: define the continuation value $V_j(x)$:

$$V_j(x) \overset{\text{def}}{=} \max_{u \in \mathcal{U}_{j,J}} I\!E\big[g_{j,J}(Z_{j,J}(u)) \,\big|\, Z_j = x\big]$$

Initialize with $V_{J+1} \equiv 0$. Fix $j \le J$ and suppose $V_{j+1}(\cdot)$ computed.

Define

$$\begin{aligned} V_j(x) &= \max_{u \in \mathcal{U}} I\!E\big[V_{j+1}(Z_{j+1}) \,\big|\, Z_j = x, u_j = u\big] = \max_{u \in \mathcal{U}} V_j(x, u), \\ u_j(x) &= \operatorname*{argmax}_{u \in \mathcal{U}} V_j(Z_j, u_j). \end{aligned}$$

Optimal control with reinforced regression. II

Let $\mathcal{U}$ be finite and do not vary with j.

Objects of interest:

- Partial value functions $V_j(x,u)$;
- Support function of u_t:

$$\mathbb{1}\big(u_t(x) = u\big) = \mathbb{1}\left(V_j(x,u) = \max_{u' \in \mathcal{U}} V_j(x,u')\right)$$

Finite $\mathfrak{U}$

Let $\mathfrak{U}$ be small. For each $u \in \mathfrak{U}$, generate and store $Z_j^{(m,u)}$, $m = 1, \ldots, M$, $j = 1, \ldots, J$.

Initialize $V_{J+1} \equiv 0$. Assuming $V_{N,j+1}(x)$ already computed, estimate

$$V_{N,j}(x,u) = \mathbb{E}\left[V_{N,j+1}(Z_{j+1}) \,\middle|\, Z_t = x, u_t = u\right]$$

using $\left(Z_j^{(m,u)}, Z_{j+1}^{(m,u)}\right)$, $m = 1, \ldots, N$:

$$V_{N,j}(x,u) \approx \sum_k \gamma_k^{N,j}(u)\psi_k^{N,j}(x)$$

with

$$\gamma^{N,j}(u) = \operatorname*{arginf}_{\gamma} \sum_m \left| V_{N,j+1}(Z_{j+1}^{(m,u)}) - \sum_k \gamma_k \psi_k^{N,j}(Z_j^{(m,u)}) \right|^2.$$

Enlarging the basis set

Idea: start with a computationally cheep basis set $\psi_1, \dots, \psi_K$.

At each step $j < J$, extend this set to a larger set

$$\psi_1, \dots, \psi_K, \psi^{N,j}_{K+1}, \dots, \psi^{N,j}_{K+b}$$

where the auxiliary functions $\psi^{N,j}_{K+1}, \dots, \psi^{N,j}_{K+b}$ depend on the previously computed value functions $V_{N,j+1}(x)$.

Running example: for each $u \in \mathfrak{U}$,

$$\text{VALUE}: \quad \psi^{u,j}(x) = V_{j+1}(x,u),$$

$$\text{POLICY}: \quad \psi^{u,j}(x) = \mathbb{1}\left(u = \operatorname*{argmax}_{u' \in \mathfrak{U}} V_{N,j+1}(x,u')\right).$$

Computation at step $j < J$ given $V_{N,j+1}(x)$

Let $\Psi^{N,j}(x) = \big(\psi_1(x), \ldots, \psi_K(x),\ \psi^{N,j}_{K+1}(x), \ldots, \psi^{N,j}_{K+b}(x)\big)$ fixed. Define

$$\mathcal{M}_j(u) = \begin{pmatrix} \Psi^{N,j}(Z_j^{(1,u)}) \\ \vdots \\ \Psi^{N,j}(Z_j^{(N,u)}) \end{pmatrix}, \quad \mathcal{V}_{j+1}(u) = \begin{pmatrix} V_{N,j+1}(Z_{j+1}^{(1,u)}) \\ \vdots \\ V_{N,j+1}(Z_{j+1}^{(N,u)}) \end{pmatrix}$$

for each $u \in \mathcal{U}$ and

$$\gamma^{N,j}(u) = \operatorname*{arginf}_{\gamma} \left\| \mathcal{V}_{j+1}(u) - \mathcal{M}_j(u)\gamma \right\|^2 = \left\{ \mathcal{M}_j^\top(u)\, \mathcal{M}_j(u) \right\}^{-1} \mathcal{M}_j^\top(u)\, \mathcal{V}_{j+1}(u)$$

$$\begin{aligned} V_{N,j}(x,u) &= \gamma_1^{N,j}(u)\psi_1(x) + \ldots + \gamma_K^{N,j}(u)\psi_K(x) \\ &\qquad + \gamma^{N,j}_{K+1}(u)\psi^{N,j}_{K+1}(x) + \ldots + \gamma^{N,j}_{K+b}(u)\psi^{N,j}_{K+b}(x), \\ V_{N,j}(x,u) &= \max_{u \in \mathcal{U}} V_{N,j}(x,u). \end{aligned}$$

Compute and store $\psi_k^{N,\ell}(x)$ for all $x = Z_\ell^{(m,u)}$, $\ell < j$, $u \in \mathcal{U}$.

Continuous or large $\mathcal{U}$: generating $Z_j^{(m,u)}$ for all u is too expensive.

Idea: Generate and store the set of controlled paths $(Z_j^{(m)}, u_j^{(m)})$. Use them to estimate

$$V_{N,j}(x,u) = \mathbb{E}\left[V_{N,j+1}(Z_{j+1}) \,\middle|\, Z_j = x, u_j = u\right] \approx \sum_k \gamma_k^{N,j} \psi_k^{N,j}(x,u)$$

with

$$\gamma^{N,j} = \underset{\gamma}{\operatorname{arginf}} \sum_m \left| V_{N,j+1}(Z_{j+1}^{(m)}) - \sum_k \gamma_k \, \psi_k^{N,j}(Z_j^{(m)}, u_j^{(m)}) \right|^2$$

and

$$V_{N,j}(x) = \max_{u \in \mathcal{U}} V_{N,j}(x,u).$$

Enlarging the basis set

Idea: start with a computationally cheep basis set $\psi_1(x,u), \ldots, \psi_K(x,u)$.

At each step $j < J$, extend this set to a larger set

$$\psi_1, \ldots, \psi_K, \psi_{K+1}^{N,j}, \ldots, \psi_{K+b}^{N,j}$$

where the auxiliary functions $\psi_{K+1}^{N,j}, \ldots, \psi_{K+b}^{N,j}$ depend on the previously computed value functions $V_{N,j+1}(x)$.

Running example:

$$\text{VALUE}: \quad \psi^{N,j}(x,u) = V_{N,j+1}(x,u),$$

$$\text{POLICY}: \quad \psi^{N,j}(x,u) = K\Big(u - u_{N,j+1}(x)\Big).$$

for a kernel $K(\cdot)$ and $u_{N,j}(x) = \mathrm{argmax}_{u' \in \mathfrak{U}} V_{N,j}(x,u')$.

Outline

1 Introduction

- Bellman principle
- Regression methods

2 Backward reinforced regression algorithm

3 Numerical examples

4 Some theoretical arguments

5 Optimal control

6 **Summary and outlook**

- The proposed approach yields a novel deep network architecture constructed by backward reinforced regression using the Bellman principle (continuation rule) as activating device.
- The procedure does not require parameter tuning by non-convex optimization
- Computational costs are polynomial in the number of steps.
- The method demonstrates expected gain in numerical power.
- There is some theoretical evidence that the extended basis yields a better quality of approximation.

Outlook

- Non-Markovian processes X_t .
- Rough paths models
- Extension to optimal control
- More theoretical results
- Relation between deep networks and optimal control via Bellman principle.

References

Andersen, L. B. (1999).

A simple approach to the pricing of bermudan swaptions in the multi-factor libor market model.

Journal of Computational Finance, 3:5–32.

Becker, S., Cheridito, P., and Jentzen, A. (2018).

Deep optimal stopping.

arXiv preprint arXiv:1804.05394.

Belomestny, D. (2011).

Pricing bermudan options by nonparametric regression: optimal rates of convergence for lower estimates.

Finance and Stochastics, 15(4):655–683.

Broadie, M. and Glasserman, P. (1997).

Pricing american-style securities using simulation.

Journal of Economic Dynamics and Control, 21(8):1323–1352.

Carriere, J. F. (1996).

Valuation of the early-exercise price for options using simulations and nonparametric regression.

Insurance: mathematics and Economics, 19(1):19–30.

Egloff, D. et al. (2005).

Monte carlo algorithms for optimal stopping and statistical learning.

The Annals of Applied Probability, 15(2):1396–1432.

References

Longstaff, F. and Schwartz, E. (2001).

Valuing american options by simulation: a simple least-squares approach.

Review of Financial Studies, 14(1):113–147.

Tsitsiklis, J. and Van Roy, B. (2001).

Regression methods for pricing complex american style options.

IEEE Trans. Neural. Net., 12(14):694–703.