Performance Bounds and Suboptimal Policies for Multi-Period Investment

Stephen Boyd, Mark Mueller, Brendan O'Donoghue, Yang Wang

26/3/2012
ETH Zürich

Outline

1. Multi-period investment problem
2. Solution via dynamic programming
3. Suboptimal policies
4. Performance bounds
5. Numerical examples
6. Summary

Outline

1. Multi-period investment problem
2. Solution via dynamic programming
3. Suboptimal policies
4. Performance bounds
5. Numerical examples
6. Summary

Multi-period investment problem

- manage portfolio of n assets over discrete time periods $t = 0, 1, \ldots, T$
- $x_t \in \mathbf{R}^n$ vector of portfolio positions at time t (in dollars)
- $u_t \in \mathbf{R}^n$ vector of trades at time t (in dollars)
- post-trade portfolio: $x_t^+ = x_t + u_t$
- starting portfolio x_0 is given

Asset returns

- portfolio propagates as

$$x_{t+1} = R_{t+1} x_t^+, \qquad t = 0, \ldots, T-1$$

- $R_{t+1} = \mathbf{diag}(r_{t+1}) \in \mathbf{R}^{n \times n}$
- $r_{t+1} \in \mathbf{R}^n$ is vector of asset returns over time period $[t, t+1]$
- $r_1, \ldots, r_T$ are independent random variables, with known first and second moments

$$\mathbf{E}(r_t) = \bar{r}_t, \qquad \mathbf{E}(r_t - \bar{r}_t)(r_t - \bar{r}_t)^T = \Sigma_t$$

- r_{t+1} is (of course) not known when u_t is chosen

Stochastic control problem

- total expected cost is

$$J = \mathbf{E} \sum_{t=0}^{T} \ell_t(x_t, u_t)$$

- $\ell_t : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R} \cup \{\infty\}$ is convex stage cost function
- $-J$ is expected revenue from the portfolio
- goal: find trading policies $\phi_0, \ldots, \phi_T : \mathbf{R}^n \to \mathbf{R}^n$, with

$$u_t = \phi_t(x_t)$$

  that minimize J
- a **convex stochastic control problem**

Stage cost

- stage cost has form

$$\ell_t(x,u) = \begin{cases} \mathbf{1}^T u + \psi_t(x,u) & x+u \in \mathcal{C}_t \\ \infty & \text{otherwise} \end{cases}$$

- $\mathbf{1}^T u_t$ is gross cash put into portfolio
  ($\mathbf{1}^T u_t < 0$ means revenue extracted from portfolio)
- ψ_t includes transaction cost, risk cost, position costs, ...
- $\mathcal{C}_t$ is the post-trade portfolio constraint set

Example post-trade constraints

position limits	$x_t^{\min} \leq x^+ \leq x_t^{\max}$
total value minimum	$\mathbf{1}^T x^+ \geq v_t^{\min}$
terminal portfolio constraint	$x_T^+ = x^{\text{term}}$
leverage limits	$\mathbf{1}^T (x^+)_- \leq \eta_t \mathbf{1}^T x^+$
sector exposure limits	$s_t^{\min} \leq F_t x^+ \leq s_t^{\max}$
sector neutrality	$F_t x^+ = 0$
concentration limits	$\sum_{i=1}^{p} (x^+)_{[i]} \leq \beta_t \mathbf{1}^T x^+$
variance risk limits	$(x^+)^T \Sigma_{t+1} x^+ \leq \gamma_t$
homogeneous risk limits	$\|\Sigma_{t+1}^{1/2} x^+\|_2 \leq \delta_t \mathbf{1}^T x^+$

Example transaction and position costs

broker commission	$(\kappa_t^{\mathrm{buy}})^T u_+ + (\kappa_t^{\mathrm{sell}})^T u_-$
bid-ask spread	$\kappa_t^T \lvert u \rvert$
quadratic price impact	$s_t^T u^2$
$3/2$ power price impact	$s_t^T \lvert u \rvert^{3/2}$
borrowing/shorting fee	$c_t^T (x^+)_-$
quadratic risk penalty	$\lambda_t (x^+)^T \Sigma_t x^+$
std. dev. risk penalty	$\lambda_t \lVert \Sigma_t^{1/2} x^+ \rVert_2$

Outline

1. Multi-period investment problem
2. Solution via dynamic programming
3. Suboptimal policies
4. Performance bounds
5. Numerical examples
6. Summary

Dynamic programming 'solution'

- Bellman recursion: $V_{T+1} = 0$,

$$V_t(x) = \inf_u \left(\ell_t(x,u) + \mathbf{E}\, V_{t+1}(R_{t+1}(x+u))\right), \quad t = T, T-1, \ldots, 0$$

- abstractly $V_t = \mathcal{T}_t V_{t+1}$ (Bellman operator)
- optimal policy

$$\phi_t^\star(x) \in \operatorname*{argmin}_u \left(\ell_t(x,u) + \mathbf{E}\, V_{t+1}(R_{t+1}(x+u))\right)$$

- optimal cost $J^\star = V_0(x_0)$
- in general, **intractable to compute** (indeed, even represent) V_t

Exception: The quadratic problem

- suppose ℓ_t are convex quadratic (can include linear equality constraints)
- multi-period trading problem is quadratic stochastic control problem

- V_t are convex quadratic, via recursion:
  - $V_{T+1} = 0$ is convex quadratic
  - convex quadratic functions preserved under expectation and partial minimization, so Bellman operator $\mathcal{T}_t$ preserves convex quadratic
- optimal policy affine is affine: $\phi_t^\star(x) = J_t x + k_t$
- can compute J_t, k_t from problem data

Outline

1. Multi-period investment problem

2. Solution via dynamic programming

3. Suboptimal policies

4. Performance bounds

5. Numerical examples

6. Summary

Approximate dynamic programming

- replace V_t with (convex) approximation $\hat{V}_t$ to get ADP policy

$$\phi_t^{\mathrm{adp}}(x) \in \operatorname*{argmin}_u \left(\ell_t(x,u) + \mathbf{E}\,\hat{V}_{t+1}(R_{t+1}(x+u)\right)$$

- choose $\hat{V}$ so
  - policy evaluation is easy (minimization above can be done fast)
  - performance is good (hopefully, $J \approx J^\star$)
- one reasonable choice: exact value function of related quadratic problem
- we'll see another choice later

Model predictive control

- a.k.a. receding (or shrinking) horizon control
- at time t, solve (open loop) control problem using mean returns

$$\begin{array}{ll} \text{minimize} & \sum_{\tau=t}^{T} \ell_\tau(z_\tau, v_\tau) \\ \text{subject to} & z_{\tau+1} = \mathbf{diag}(\bar{r}_{\tau+1})(z_\tau + v_\tau), \quad \tau = t, \ldots, T-1 \\ & z_t = x_t \end{array}$$

  over z_τ, v_τ, $\tau = t, \ldots, T$
- we interpret $v_t^\star, \ldots, v_T^\star$ as **trading plan**, assuming future returns take on their mean values
- policy is $\phi_t^{\mathrm{mpc}}(x_t) = v_t^\star$ (first trade in trading plan)

Evaluating ADP and MPC policies

- evaluating ADP and MPC policies reduce to solving
  - convex optimization problems in general
  - QPs when ℓ_t are QP-representable and $\hat{V}_t$ are quadratic
- $O(n)$ variables for ADP, $O(n(T-t))$ variables for MPC
- new methods (code generation) allow us to solve both very quickly
  - $O(n^3)$ flops for ADP, $O(n^3(T-t))$ flops for MPC
  - for $n = 30$ assets, $T = 99$: 50μs for ADP, 10ms for MPC
  - 1000–10000$\times$ faster than generic methods

Performance of ADP and MPC policies

- can evaluate $J^{\rm adp}$ and $J^{\rm mpc}$ by Monte Carlo
- fast evaluation of policies critical for Monte Carlo simulations
- suboptimal policies appear to do well (with good choice of $\hat{V}_t$ for ADP)

- leads to obvious question:

  how suboptimal are ADP and MPC policies?

- we'll address this by computing a **numerical** lower bound on optimal objective value, $J^{\rm lb} \leq J^\star$
- if $J^{\rm adp}$, $J^{\rm mpc}$ are not far above $J^{\rm lb}$, we **know** they are nearly optimal

Outline

Performance bound from Bellman inequalities

- suppose $V_t^{\mathrm{lb}} \leq V_t$ (elementwise)
- yields **performance bound**

$$J^{\mathrm{lb}} = V_0^{\mathrm{lb}}(x_0) \leq V_0(x_0) = J^\star$$

- sufficient condition for $V_t^{\mathrm{lb}} \leq V_t$: **Bellman inequalities**

$$V_{T+1}^{\mathrm{lb}} = 0, \quad V_t^{\mathrm{lb}} \leq \mathcal{T}_t V_{t+1}^{\mathrm{lb}}$$

- *cf.* Bellman equalities

$$V_{T+1} = 0, \quad V_t = \mathcal{T}_t V_{t+1}$$

- follows from monotonicity of Bellman operators

Optimizing performance bound via convex optimization

general approach:

- linearly parametrize candidate $V_t^{\mathrm{lb}} = \sum_{i=1}^N \alpha_{ti} V^{(i)}$
  ($V^{(i)}$ are basis elements)
- derive convex condition on α_{ti} that implies Bellman inequalities
- maximize (linear function of α_{ti}) J^{lb} via convex optimization
- yields best performance bound (for basis, Bellman inequality condition)

- maximizer of performance bound is **excellent candidate** for $\hat{V}_t$ in ADP

Bellman inequality condition

(simplified case)

- assume stage cost function of form

$$\ell_t(x,u) = \begin{cases} \psi_t(x,u) & (x,u) \in \mathcal{C} \\ \infty & \text{otherwise} \end{cases}$$

  ψ_t convex quadratic

- constraint set described by quadratic inequalities

$$\mathcal{C} = \{(x,u) \mid f_1(x,u) \geq 0, \ldots, f_M(x,u) \geq 0\}$$

  $f_1, \ldots, f_M$ (not necessarily convex) quadratic

Bellman inequality condition II

- parameterize V_t^{lb} as general convex quadratic function

$$V_t^{\mathrm{lb}}(x) = (1/2) \left[\begin{array}{c} x \\ 1 \end{array}\right]^T \left[\begin{array}{cc} P_t & p_t \\ p_t^T & r_t \end{array}\right] \left[\begin{array}{c} x \\ 1 \end{array}\right], \qquad P_t \geq 0$$

- Bellman inequality has form

$$V_t^{\mathrm{lb}}(x) \leq \psi_t(x,u) + \mathbf{E}\, V_{t+1}^{\mathrm{lb}}(R_{t+1}(x+u)), \quad \forall (x,u) \in \mathcal{C}$$

- write inequality as $f_0(x,u) \geq 0$, with f_0 quadratic
- coefficients of f_0 are affine in parameters $P_t, P_{t+1}, p_t, p_{t+1}, r_t, r_{t+1}$
- Bellman inequality has form

$$f_0(x,u) \geq 0 \quad \text{whenever} \quad f_1(x,u) \geq 0, \ldots, f_M(x,u) \geq 0$$

i.e., a quadratic function is nonnegative whenever a set of M others are

$\mathcal{S}$-procedure

- a sufficient condition (called $\mathcal{S}$-procedure): $\exists \lambda_1, \ldots, \lambda_M \geq 0$

$$f_0(x,u) \geq \lambda_1 f_1(x,u) + \cdots + \lambda_M f_M(x,u) \quad \forall (x,u)$$

- equivalent to a matrix inequality in the coefficients of $f_0, \ldots, f_M$
- this matrix inequality is an **affine function** of λ and the parameters $P_t, P_{t+1}, p_t, p_{t+1}, r_t, r_{t+1}$, *i.e.*, it is a linear matrix inequality (LMI)
- maximizing $J^{\rm lb}$ subject to our ($\mathcal{S}$-procedure based) sufficient condition for Bellman inequalities is a semidefinite program (SDP)

- hence, **we can effectively solve it**

Outline

Numerical examples

- $n = 30$ assets
- $T = 99$ periods
- $x_0 = x_T = 0$
- returns IID log-normal: $\log(r_t) \sim \mathcal{N}(\mu, \tilde{\Sigma})$
- we consider quadratic case and four others

Transaction cost and constraints

- for quadratic example:

$$\psi(x,u) = s^T u_t^2 + \lambda (x_t^+)^T \Sigma x_t^+$$

(price impact, risk penalty)

- for other cases:

$$\psi(x,u) = c^T (x_t^+)_- + \kappa^T |u_t| + s^T u_t^2 + \lambda (x_t^+)^T \Sigma x_t^+$$

(includes additional shorting cost, bid-ask spread)

- constraint sets:
  - unconstained: $\mathcal{C}_t = \mathbf{R}^n$
  - long-only: $\mathcal{C}_t = \mathbf{R}_+^n$
  - leverage limit: $\mathcal{C}_t = \{x^+ \mid \mathbf{1}^T (x^+)_- \leq 0.3(\mathbf{1}^T x^+)\}$
  - sector neutral: $\mathcal{C}_t = \{x^+ \mid F x^+ = 0\}$, $F \in \mathbf{R}^{2\times n}$

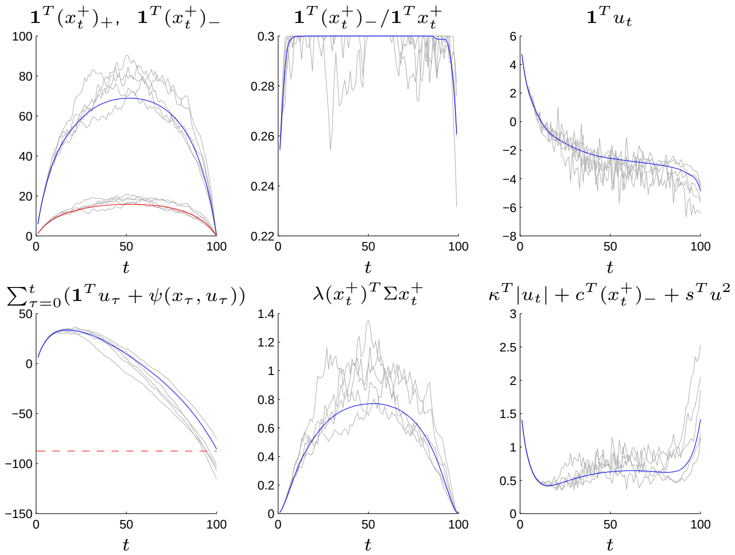
Results

- evaluate J^{adp} and J^{mpc} via Monte Carlo
  - for ADP, 50000 samples (5 million QPs, 3 minutes on 8 cores)
  - for MPC, 5000 samples (0.5 million QPs, a few hours)

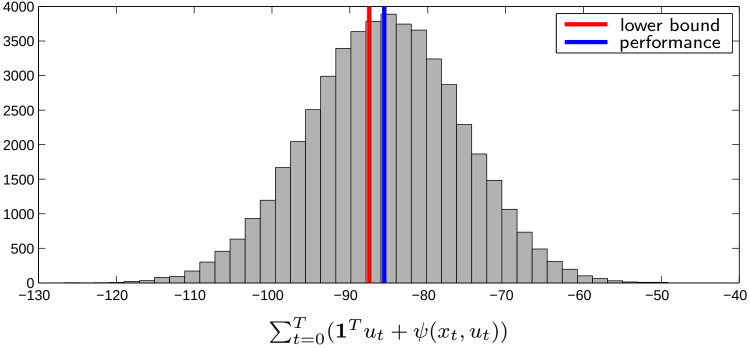
Example	J^{lb}	J^{adp}	J^{mpc}
quadratic	-450.1	-450.3	-444.3
unconstrained	-132.6	-131.9	-130.6
long-only	-41.3	-41.0	-40.6
leverage limit	-87.5	-85.6	-84.7
sector neutral	-121.3	-118.9	-117.5

- conclusion: **ADP and MPC are nearly optimal**

Time traces (leverage limit example, ADP policy)



Cost histogram (leverage limit example, ADP policy)



Outline

1. Multi-period investment problem
2. Solution via dynamic programming
3. Suboptimal policies
4. Performance bounds
5. Numerical examples
6. Summary

Summary

- by using **value** of assets as variables, dynamics is linear (but random)
- hence get **convex stochastic control problem**
  - even with complicated practical constraints and transaction costs
- can solve exactly in quadratic case
- using SDP we compute a **numerical bound on performance**
- ADP and MPC suboptimal policies
  - rely on solving convex optimization problem in each step
  - often achieve **provably near-optimal performance**

Final comments on numerical performance bounds

- **no, we cannot guarantee that** $J^{\mathrm{adp}} - J^{\mathrm{lb}}$ **is small**
- and we **do not apologize**

- we can only compute it for any given problem
- it is **exceedingly useful** in practice (and we think, in theory)
- we doubt a generic theoretical type bound would have any practical value