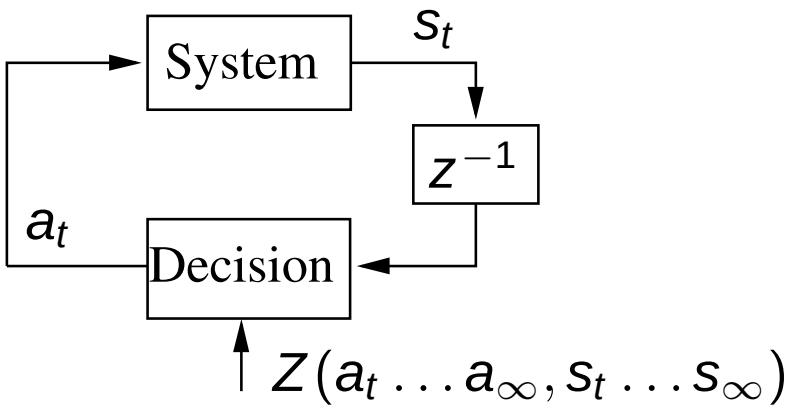
Approximate Dynamic Programming

Václav Šmídl

Seminar CSKI, 18.4.2004

Outline

Decision Making and Control



- a_t action, decision, (input),
- s_t observed state, (output),
- $Z(\cdot)$ loss function,
- Θ_t internal (unknown) quantities,

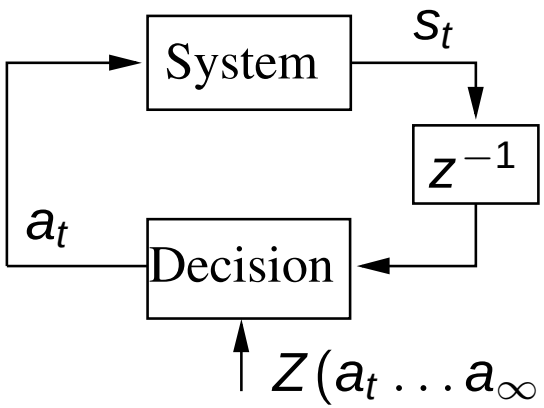
Optimal decision making (control)?:

$$a_t = \arg\min_{a_t} \{Z(a_t \dots a_\infty, s_t \dots s_\infty)\}.$$

Optimal uncertain/robust decision making:

$$a_t = \arg\min_{a_t} \mathrm{E}_{\Theta_t} Z(a_t \dots a_\infty, s_t, s_{t+1}(\Theta_t) \dots s_\infty(\Theta_t)).$$

Decision Making and Control



- a_t action, decision, (input),
- s_t observed state, (output),
- $Z(\cdot)$ loss function,
- Θ_t internal (unknown) quantities,

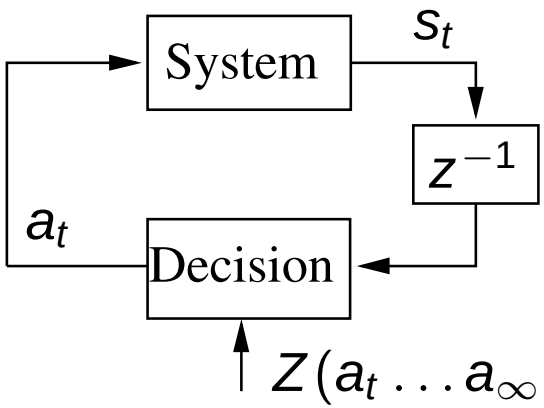
Optimal decision making (control):

$$a_t = \arg\min_{a_t} \{Z(a_t \dots a_\infty, s_t \dots s_\infty)\}.$$

Optimal uncertain/robust decision making:

$$a_t = \arg\min_{a_t} \mathrm{E}_{\Theta_t} Z(a_t \dots a_\infty, s_t, s_{t+1}(\Theta_t) \dots s_\infty(\Theta_t)).$$

Decision Making and Control



- a_t action, decision, (input),
- s_t observed state, (output),
- $Z(\cdot)$ loss function,
- Θ_t internal (unknown) quantities,

Optimal decision making (control):

$$a_t = \arg\min_{a_t} \{Z(a_t \dots a_\infty, s_t \dots s_\infty)\}.$$

Optimal uncertain/robust decision making:

$$a_t = \arg\min_{a_t} \mathrm{E}_{\Theta_t} Z(a_t \dots a_\infty, s_t, s_{t+1}(\Theta_t) \dots s_\infty(\Theta_t)).$$

Dynamic Programing

In case of additive loss function:

$$Z(a_t \dots a_\infty, s_t \dots s_\infty) = \sum_{k=0}^{\infty} \gamma^k z\left(a_{t+k}, s_{t+k}, a_{t+k-1}, \dots\right).$$

e.g. $Z = \sum_{k=1}^{\infty}(s_{t+k} - s_{t+k}^{\text{ref}})^2 + a_t^2$, the solution can be simplified.

Theorem

A strategy of actions a_t which minimizes

$$V(s_t) = \min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V\left(s_{t+1}\right)\right] \tag{1}$$

in time t also minimizes the overall loss $Z(a_t \dots a_\infty, s_t \dots s_\infty)$.

Equation (1) is known as (i) Hamilton-Jacobi from physics, (ii) Bellman, (iii) Hamilton-Jacobi-Bellman.

- Optimal robust control (solved by PDE)
- Operational research: Markov Decision Processes (discrete)
- Artificial inteligence: Neurocontrol (continuous)

Dynamic Programing

In case of additive loss function:

$$Z(a_t \dots a_\infty, s_t \dots s_\infty) = \sum_{k=0}^{\infty} \gamma^k z\left(a_{t+k}, s_{t+k}, a_{t+k-1}, \dots\right).$$

e.g. $Z = \sum_{k=1}^{\infty}(s_{t+k} - s_{t+k}^{\text{ref}})^2 + a_t^2$, the solution can be simplified.

Theorem

A strategy of actions a_t which minimizes

$$V(s_t) = \min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V\left(s_{t+1}\right)\right] \tag{1}$$

in time t also minimizes the overall loss $Z(a_t \dots a_\infty, s_t \dots s_\infty)$.

Equation (1) is known as (i) Hamilton-Jacobi from physics, (ii) Bellman, (iii) Hamilton-Jacobi-Bellman.

- Optimal robust control (solved by PDE)
- Operational research: Markov Decision Processes (discrete)
- Artificial inteligence: Neurocontrol (continuous)

Dynamic Programing

In case of additive loss function:

$$Z(a_t \dots a_\infty, s_t \dots s_\infty) = \sum_{k=0}^{\infty} \gamma^k z\left(a_{t+k}, s_{t+k}, a_{t+k-1}, \dots\right).$$

e.g. $Z = \sum_{k=1}^{\infty}(s_{t+k} - s_{t+k}^{\text{ref}})^2 + a_t^2$, the solution can be simplified.

Theorem

A strategy of actions a_t which minimizes

$$V(s_t) = \min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V\left(s_{t+1}\right)\right] \tag{1}$$

in time t also minimizes the overall loss $Z(a_t \dots a_\infty, s_t \dots s_\infty)$.

Equation (1) is known as (i) Hamilton-Jacobi from physics, (ii) Bellman, (iii) Hamilton-Jacobi-Bellman.

- Optimal robust control (solved by PDE)
- Operational research: Markov Decision Processes (discrete)
- Artificial inteligence: Neurocontrol (continuous)

Aim of the research

Long-term aims:

Apply formal method of decision making in ‘difficult’ areas of multi-agent systems, multiple participant decision making.

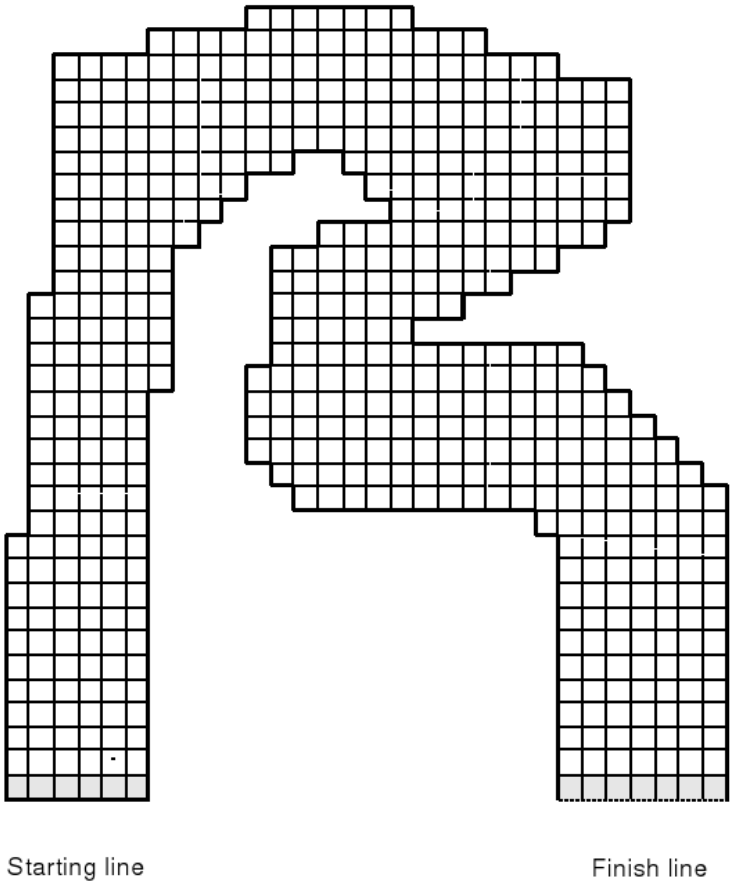
We seek decision making algorithms that:

1. take uncertainty into account,
2. are computationally tractable,
3. are as close to the optimality as possible,
4. operates reliably in changing enviroment
    1. on-line learning (system identification)
    2. on-line design (improvement) of decision making strategy.

This talk offers summary of state-of-the art.

Example: Markov process

RACE TRACK; Stochastic shortest path problem, [Bradtke, 94]



- Model:

$$\begin{aligned} x_{t+1} &= x_t + \dot{x}_t + a_{x;t}, \\ y_{t+1} &= y_t + \dot{y}_t + a_{y;t}, \\ \dot{x}_{t+1} &= \dot{x}_t + a_{x;t}, \\ \dot{y}_{t+1} &= \dot{y}_t + a_{y;t}. \end{aligned}$$

If $[x_{t+1}, y_{y+1}]$ outside bounds:

$$\begin{aligned} x_{t+1} &= x_t, y_{t+1} = y_t, \\ \dot{x}_{t+1} &= \dot{y}_{t+1} = 0. \end{aligned}$$

- Control actions $a_{x;t}, a_{y;t} \in \{-1, 0, 1\}$ are executed with **probability** p.
- Loss function: $z(s_t, s_{t+1}, a_t) = 1$.

Exact Solutions of Dynamic Programming

We seek such a function $V(s_t)$ (lookup table) for which it holds:

$$V(s_t) = \min_{a_t} \sum_{s_{t+1}} f\left(s_{t+1}|s_t, a_t\right)\left(z(s_t, s_{t+1}, a_t) + \gamma V(s_{t+1})\right),$$

Solutions:

Value iteration: the solution is iterated for all possible values of a_t,

Policy iteration: we start with a guess of optimal policy $a_t = a^{(0)}(s_t)$

$$V(s_t) = \sum_{s_{t+1}} f\left(s_{t+1}|s_t, a(s_t)\right)\left(z(s_t, s_{t+1}, a(s_t)) + \gamma V(s_{t+1})\right),$$

and recompute the new estimate of the policy.

Linear programming: existence of upper bounds, constraint optimization.

Curses of dimensionality

1. Size of the state-space -> size of the value function
   1. aggregation of states,
   2. using continuous functions
2. Taking expectations over the whole space -> time consuming
   1. sampling,
   2. tree search,
   3. roll-out heuristics,
   4. post-decision formulation of DP.
3. Size of the space of decisions -> expensive minimization
   1. model-less (Q) learning

Curses of dimensionality

1. Size of the state-space -> size of the value function
   1. aggregation of states,
   2. using continuous functions
1. Taking expectations over the whole space -> time consuming
   1. sampling,
   2. tree search,
   3. roll-out heuristics,
   4. post-decision formulation of DP.
1. Size of the space of decisions -> expensive minimization
   1. model-less (Q) learning

Curses of dimensionality

1. Size of the state-space -> size of the value function
   1. aggregation of states,
   2. using continuous functions
2. Taking expectations over the whole space -> time consuming
   1. sampling,
   2. tree search,
   3. roll-out heuristics,
   4. post-decision formulation of DP.
3. Size of the space of decisions -> expensive minimization
   1. model-less (Q) learning

Iterative methods

Estimates of $V(s_t)$ in the kth iteration is $V^{(k)}(s_t)$.

- Synchronous policy. For all states do:

$$V^{(k+1)}(s_t) = \min_{a_t} \sum_{s_{t+1}} f\left(s_{t+1}|s_t, a_t\right) \left(z(s_t, s_{t+1}, a_1) + \gamma V^{(k)}(s_{t+1})\right),$$

  Each state is visited once in each sweep.

- Asynchronous policy. For states in a chosen set do:

$$V^{(k+1)}(s_t) = \min_{a_t} \sum_{s_{t+1}} f\left(s_{t+1}|s_t, a_t\right) \left(z(s_t, s_{t+1}, a_1) + \gamma V^{(k)}(s_{t+1})\right),$$

  and generate a new set.

Real-time Dynamic Programming

Assynchronous plicy: new set is generated by applying the decision based on current estimate.
(Requires many trial simulation runs to converge.)

Theorem

Assymptotic convergence is guaranteed if:

1. *there is at least one optimal strategy,*
2. *all step loss functions* $z(s_t, a_t) > 0$,
3. *initial values for all states are non-overestimating,* $V_0(s_t) \leq V(s_t)$.

Variants:

RTDP: basic variant with known model,

ARTDP: a variant with model with unknown parameters which are estimated using ML,

RTQ: variant with unknown model.

Model-less learning

In case of unknown model, we may extend the Bellman equation as follows:

$$\begin{aligned} V(s_t) &= \min_{a_t} Q(s_t, a_t), \\ Q(s_t, a_t) &= \sum_{s_{t+1}} f(s_{t+1}|s_t, a_t)\left(z(s_t, s_{t+1}, a_1) + \gamma V(s_{t+1})\right), \end{aligned}$$

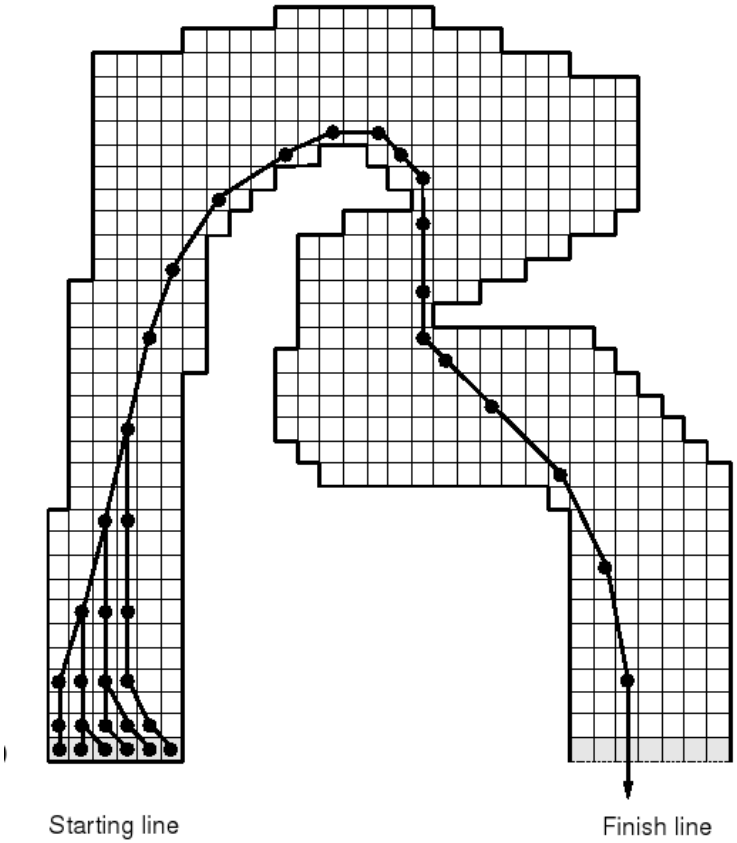
yielding recursion:

$$Q(s_t, a_t) = \sum_{s_{t+1}} f(s_{t+1}|s_t, a_t)\left(z(s_t, s_{t+1}, a_1) + \gamma \min_{a_{t+1}} Q(s_{t+1}, a_{t+1})\right).$$

Advantages: (i) no need of explicit model, (ii) easier computation of optimal actions.

Disadvantages: (i) larger function to store, (ii) learning is more time- and data-consuming.

Experimental comparison: Race track [Bradtke: 94]



States: 22 546 total,
9 836 effective.

Number of operations to reach convergence:

	# of trials (passes)	# of backups
GS	38 p	835 468
RTDP	1000 t	517 356
ARTDP	800 t	653 774
RTQ	60 000 t	10 milion

Number of visited states:

	<100×	<10×	0×
RTDP	97%	70%	8%
RTQ	50%	8%	2%

Linear Quadratic Control

Linear model with fully observable state, with Gaussian noise:

$$s_{t+1} = As_t + Ba_t + e_t, \quad e_t \sim \mathcal{N}(0, \Sigma),$$

and quadratic loss function

$$z(s_t, a_t) = s_t' C s_t + a_t' D a_t.$$

The Bellman equation is a quadratic in x_t:

$$V(s_t) = [s_t', 1] \Phi \begin{bmatrix} s_t \\ 1 \end{bmatrix}.$$

Obtained as solutions of Riccatti equation.
In other cases the problem is *intractable*. LQG still serves as an etalon for tests.

Adaptive Critic Algorithms

Continuous case of policy iteration approach used in discrete cases. However, it is a *forward-dynamic-programming* structure.

$$V(s_t) = \min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V(s_{t+1})\right] \tag{2}$$

Parameterized approximants:

$$\begin{aligned} V(s_t) &\approx V^{(k)}(s_t, w_k), \\ a_t &\approx a^{(k)}(s_t, \alpha_k). \end{aligned}$$

Iterations:

1. Policy improvement:

$$a^{(k+1)}(s_t, \alpha_{k+1}) = \arg \min_{a_t(s_t,\alpha)} \mathrm{E}\left[z(s_t, a_t(s_t, \alpha)) + \gamma V^{(k)}(s_{t+1}, w_k)\right],$$

2. Value determination:

$$V^{(k+1)}(s_t, w_{k+1}) = \mathrm{E}\left[z(s_t, a^{(k)}(s_t, \alpha_k)) + \gamma V^{(k)}(s_{t+1}, w_k)\right].$$

Properties of Adaptive Critics

In deterministic case:

1. In each step, the new control law $a^{(k+1)}(\cdot)$ achieves better overall performance than the previous one.
2. In each step, the new value function $V^{(k+1)}(\cdot)$ is closer approximation of the Bellman function.
3. The algorithm terminates on optimal control is such exist.
4. No backward iteration is needed.

In the stochastic case, no proof found, but probably exist (via stochastic approximations).

Design of Adaptive Critics

Choose functional approximations that are:

1. differentiable,
2. less complex than lookup table,
3. algorithms for computing parameter values exist,
4. capable of approximation of the optimal shape withing the desired accuracy.

Then:

$$a^{(k+1)}(s_t, \alpha_{k+1}) = \arg\min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V^{(k)}(s_{t+1}, w_k)\right],$$

holds if:

$$\frac{\partial}{\partial a_t}\left[z(s_t, a_t) + \gamma V^{(k)}(s_{t+1}, w_k)\right]_{a_t=\overline{a_t}} = 0.$$

The closest parametrized approximation:

$$\alpha_{k+1} \equiv \arg\min_{\alpha} \left(\overline{a_t} - a(s_t, \alpha)\right).$$

Design of Adaptive Critics

Choose functional approximations that are:

1. differentiable,
2. less complex than lookup table,
3. algorithms for computing parameter values exist,
4. capable of approximation of the optimal shape withing the desired accuracy.

Then:

$$a^{(k+1)}(s_t, \alpha_{k+1}) = \arg\min_{a_t} \mathrm{E}\left[z(s_t, a_t) + \gamma V^{(k)}(s_{t+1}, w_k)\right],$$

holds if:

$$\frac{\partial}{\partial a_t}\left[z(s_t, a_t) + \gamma V^{(k)}(s_{t+1}, w_k)\right]_{a_t=\overline{a_t}} = 0.$$

The closest parametrized approximation:

$$\alpha_{k+1} \equiv \arg\min_{\alpha} \left(\overline{a_t} - a(s_t, \alpha)\right).$$

Variants of Adaptive Critics

1. Dual Heuristic Programming.
   All that is required for update of the control law is partial derivatives of $V^{(k)}(s_{t+1}, w_k)$. Hence, we approximate directly the derivatives:
   $$\lambda(s_t, l_k) \approx \frac{\partial V(s_t)}{\partial s_t}.$$
2. Action Dependent Dynamic Programming:
   Similar idea to that of Q-learning. We do not approximate $V(s_t)$ but $Q(s_t, a_t)$.
   $$Q^{(k)}(s_t, a_t, q_k) \approx Q(s_t, a_t).$$

Same requirements as for traditional critic design (differentiability, accuracy, etc.).

Critics using Neural Networks

The functional approximantions are neural networks:

$$V(s_t) \approx NN(w_k), \quad a(s_t) \approx NN(\alpha_k),$$

Learning rules are replaced by gradient descent:

$$w^{(k+1)} = w^{(k)} - \beta \frac{\partial E}{\partial w}, \quad E = [e(t)]^2,$$

$e(t)$ is the error in Bellman equation (or its derivative, or both).
Variants in derivations of error, E:

partial

$$\frac{\partial E}{\partial w} = 2e(t)\left[-\frac{\partial V(s_t,w)}{\partial w}\right]$$

total (Galerkin PDE)

$$\frac{\partial E}{\partial w} = 2e(t)\left[-\frac{\partial V(s_t,w)}{\partial w} + \gamma \frac{\partial V(s_{t+1},w)}{\partial w}\right]$$

Issues of Adaptive Critics

- Initialization: techniques used for improvement (i) Robust control, (ii) Model Predictive Control.
- Convergence tested on LQG case in two criteria:
  1. correct equilibrium, i.e. zero derivation when optimum is found.
     - none of the total gradient (Galerkin) variants posses this property,
     - all partial gradient variants do,
  2. quadratic unconditional stability, (quadratic Lyapunov function, strictly monotonous decrease in stochastic case)
     - all total gradients posses this property,
     - none of the partial gradient does.

  New definitions of error that meet these criteria have appeared recently.

Issues of Adaptive Critics

- Initialization: techniques used for improvement (i) Robust control, (ii) Model Predictive Control.
- Convergence tested on LQG case in two criteria:
  1. correct equilibrium, i.e. zero derivation when optimum is found.
     - none of the total gradient (Galerkin) variants posses this property,
     - all partial gradient variants do,
  2. quadratic unconditional stability, (quadratic Lyapunov function, strictly monotonous decrease in stochastic case)
     - all total gradients posses this property,
     - none of the partial gradient does.

  New definitions of error that meet these criteria have appeared recently.

Conclusion

- Wide range of approximate methods for approximate programming exist. Also wide range of problem specific approaches (tips & tricks) can be found.
- The only rigorous general theory are stochastic approximations. Most of the proofs are based on this theory.
- For our purpose, some variants of the Adaptive Critics can be elaborated. Similar tools are being used (estimation of parameters via distance minimization)