Model-Based Reinforcement Learning with Fuzzy Approximation

Lucian Buşoniu[1], Damien Ernst[2],
Bart De Schutter[1], Robert Babuška[1]
i.l.busoniu@tudelft.nl

1. Delft Center for Systems and Control, TUDelft
2. FNRS, Liége, Belgium

DCSC Colloquium
9 April 2008

Outline

1. Reinforcement learning (problem, solution, an algorithm)
2. Fuzzy Q-iteration
3. Convergence & consistency
4. Example: inverted pendulum swing-up
5. Conclusion & future work

The Problem

- Discrete-time dynamics $x_{k+1} = f(x_k, u_k)$, $x \in X$, $u \in U$
- Reward function $r_{k+1} = \rho(x_k, u_k) \in \mathbb{R}$
  evaluates each transition

Goal

- Find control policy $u = h(x)$
  to maximize discounted return:

$$R^h(x_0) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k))$$

  from any x_0; $\gamma \in [0, 1)$ discount factor

- Infinite-horizon (discounted) optimal control problem

The Problem

- Discrete-time dynamics $x_{k+1} = f(x_k, u_k)$, $x \in X$, $u \in U$
- Reward function $r_{k+1} = \rho(x_k, u_k) \in \mathbb{R}$
  evaluates each transition

Goal

- Find control policy $u = h(x)$
  to maximize discounted return:
  $$R^h(x_0) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k))$$
  from any x_0; $\gamma \in [0, 1)$ discount factor

- Infinite-horizon (discounted) optimal control problem

Solution using Q-functions

- Define Q-function of h: $Q^h(x, u) = \rho(x, u) + \gamma R^h(f(x, u))$
- Optimal Q-function: $Q^* = \max_h Q^h$

  $\Rightarrow$ optimal policy $h^*(x) = \arg\max_u Q^*(x, u)$

- Q^* satisfies Bellman equation:

$$Q^*(x, u) = \rho(x, u) + \gamma \max_{u'} Q^*(f(x, u), u')$$

$\Rightarrow$ iterative algorithms to compute Q^*

Solution using Q-functions

- Define Q-function of h: $Q^h(x,u) = \rho(x,u) + \gamma R^h(f(x,u))$
- Optimal Q-function: $Q^* = \max_h Q^h$

  $\Rightarrow$ optimal policy $h^*(x) = \arg\max_u Q^*(x,u)$

- Q^* satisfies Bellman equation:

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{u'} Q^*(f(x,u),u')$$

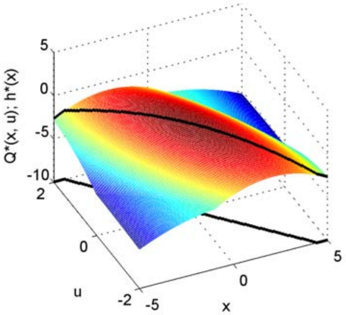
  $\Rightarrow$ iterative algorithms to compute Q^*

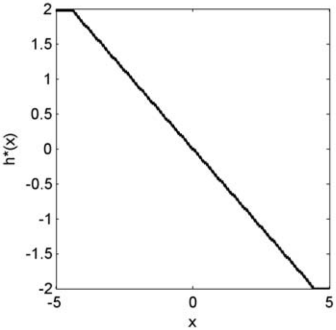
Example: discrete-time integrator

- $f(x, u) = x + K \cdot u$
  $x \in [-5, 5], \quad u \in [-2, 2], \quad K = 2$
- Goal: quadratic stabilization. Reward function:
  $\rho(x, u) = -0.1x^2 - 0.05u^2$

Example: discrete-time integrator

- $f(x, u) = x + K \cdot u$
  $x \in [-5, 5], \quad u \in [-2, 2], \quad K = 2$
- Goal: quadratic stabilization. Reward function:
  $\rho(x, u) = -0.1x^2 - 0.05u^2$





Algorithms

Model-based:

- f, ρ given
- E.g., Q-iteration

Model-free:

- f, ρ unknown
- Estimate Q^* from samples or trajectories $(x_k, u_k, x_{k+1}, r_{k+1})$

Algorithms

Model-based:

- f, ρ given
- E.g., Q-iteration

Model-free:

- f, ρ unknown
- Estimate Q^* from samples or trajectories $(x_k, u_k, x_{k+1}, r_{k+1})$

Q-iteration

repeat at each iteration ℓ

  for all x, u **do** $\triangleright$

    $Q_{\ell+1}(x,u) = \rho(x,u) + \gamma \max_{u'} Q_\ell(f(x,u),u')$ $\triangleright$ T

  end for $\triangleright$

until convergence

- Compare Bellman equation: $Q^*(x,u) = \rho(x,u) + \gamma \max_{u'} Q^*(f(x,u),u')$
- Write each iteration as $Q_{\ell+1} = TQ_\ell$
- T contraction: $\|T(Q) - T(Q')\|_\infty \leq \gamma \|Q - Q'\|_\infty$

Q-iteration

repeat at each iteration ℓ
  for all x, u **do** $\triangleright$
    $Q_{\ell+1}(x, u) = \rho(x, u) + \gamma \max_{u'} Q_{\ell}(f(x, u), u')$ $\triangleright$ T
  end for $\triangleright$
until convergence

- Compare Bellman equation:
  $Q^*(x, u) = \rho(x, u) + \gamma \max_{u'} Q^*(f(x, u), u')$
- Write each iteration as $Q_{\ell+1} = TQ_{\ell}$
- T contraction: $\|T(Q) - T(Q')\|_{\infty} \leq \gamma \|Q - Q'\|_{\infty}$

Why Q-iteration?

- Just one parameter: γ
- Monotonous convergence to Q^*
- Deterministic $\Rightarrow$ predictable; easy to get insight

Why approximation?

- Q-iteration requires tabular storage of Q-functions
- If X and / or U continuous – tabular storage impossible
(if X, U finite but large – tabular storage impractical)
- $\Rightarrow$ need to approximate the Q-function

Fuzzy approximation

Given:

- Membership functions $\varphi_1, \ldots, \varphi_N : X \to [0, 1]$
- Discrete actions $u_1, \ldots, u_M \in U$
- Parameter matrix θ of size $N \times M$

Approximate Q-function:

$$\widehat{Q}^{\theta}(x, u) = \sum_{i=1}^{N} \varphi_i(x)\theta_{i,j} = [\varphi_1(x) \ldots \varphi_N(x)] \begin{bmatrix} \cdots & \begin{bmatrix} \theta_{1,j} \\ \vdots \\ \theta_{N,j} \end{bmatrix} & \cdots \end{bmatrix}$$

$$j = \arg\min_{j'} \left\| u - u_{j'} \right\| \qquad \text{(nearest neighbor)}$$

Fuzzy approximation

Given:

- Membership functions $\varphi_1, \ldots, \varphi_N : X \to [0, 1]$
- Discrete actions $u_1, \ldots, u_M \in U$
- Parameter matrix θ of size $N \times M$

Approximate Q-function:

$$\widehat{Q}^{\theta}(x, u) = \sum_{i=1}^{N} \varphi_i(x)\theta_{i,j} = [\varphi_1(x) \ldots \varphi_N(x)] \begin{bmatrix} \cdots & \begin{bmatrix} \theta_{1,j} \\ \vdots \\ \theta_{N,j} \end{bmatrix} & \cdots \end{bmatrix}$$

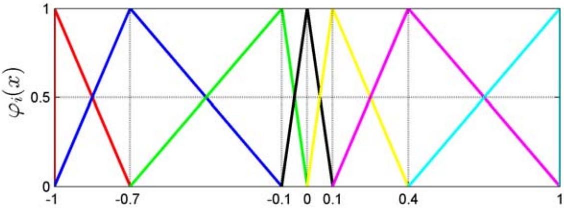
$$j = \arg\min_{j'} \left\| u - u_{j'} \right\| \qquad \text{(nearest neighbor)}$$

State approximation

- Membership functions (MFs) $\varphi_1, \ldots, \varphi_N$
- Normalized MFs: $\sum_i \varphi_i(x) = 1 \quad \forall x$
- Requirements:
  - $\varphi_i(x_i) = 1$ for a unique core x_i
  - all other MFs are 0 at x_i
- Example: triangular MFs, scalar $x \in [-1, 1]$

State approximation

- Membership functions (MFs) $\varphi_1, \ldots, \varphi_N$
- Normalized MFs: $\sum_i \varphi_i(x) = 1 \quad \forall x$
- Requirements:
  - $\varphi_i(x_i) = 1$ for a unique core x_i
  - all other MFs are 0 at x_i
- Example: triangular MFs, scalar $x \in [-1, 1]$



Fuzzy Q-iteration

repeat at each iteration ℓ

  for all cores x_i, discrete actions u_j **do**

    $\theta_{\ell+1,i,j} = \rho(x_i, u_j) + \gamma \max_{j'} \widehat{Q}^{\theta_\ell}(f(x_i, u_j), u_{j'})$

  end for

until convergence

Compare: Exact Q-iteration

repeat at each iteration ℓ

  for all x, u **do**

    $Q_{\ell+1}(x, u) = \rho(x, u) + \gamma \max_{u'} Q_\ell(f(x, u), u')$

  end for

until convergence

Fuzzy Q-iteration

repeat at each iteration ℓ
for all cores x_i, discrete actions u_j **do**
$\theta_{\ell+1,i,j} = \rho(x_i, u_j) + \gamma \max_{j'} \widehat{Q}^{\theta_\ell}(f(x_i, u_j), u_{j'})$ ▷ $T\widehat{Q}$
end for
until convergence

Compare: Exact Q-iteration

repeat at each iteration ℓ
for all x, u **do**
$Q_{\ell+1}(x, u) = \rho(x, u) + \gamma \max_{u'} Q_\ell(f(x, u), u')$ ▷ T
end for
until convergence

Fuzzy Q-iteration: policy

repeat at each iteration ℓ
  for all cores x_i, discrete actions u_j **do**
    $\theta_{\ell+1,i,j} = \rho(x_i, u_j) + \gamma \max_{j'} \widehat{Q}^{\theta_\ell}(f(x_i, u_j), u_{j'})$
  end for
until convergence

$\Rightarrow \widehat{\theta}^*$, and policy:

$$\widehat{h}^*(x) = u_j, j = \arg\max_{j'} \widehat{Q}^{\widehat{\theta}^*}(x, u_{j'})$$

(Compare optimal policy: $h^*(x) = \arg\max_u Q^*(x, u)$)

1. Reinforcement learning (problem, solution, an algorithm)
2. Fuzzy Q-iteration
3. Convergence & consistency
4. Example: inverted pendulum swing-up
5. Conclusion & future work

Convergence

- $\widehat{Q}$ non-expansion: $\|\widehat{Q}^{\theta} - \widehat{Q}^{\theta'}\|_\infty \leq \|\theta - \theta'\|_\infty$
- T contraction: $\|T(Q) - T(Q')\|_\infty \leq \gamma \|Q - Q'\|_\infty$

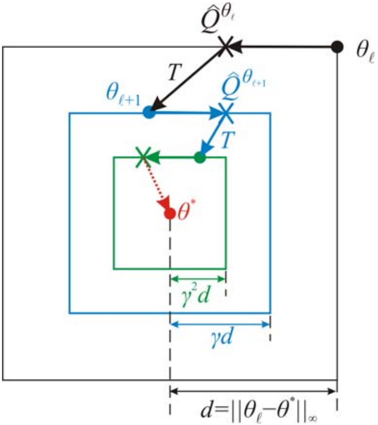
$\Rightarrow$ fuzzy Q-iteration converges monotonically to θ^*

$\|\widehat{Q}^{\theta^*} - Q^*\|_\infty$
bounded

Convergence

- $\widehat{Q}$ non-expansion: $\|\widehat{Q}^{\theta} - \widehat{Q}^{\theta'}\|_{\infty} \leq \|\theta - \theta'\|_{\infty}$
- T contraction: $\|T(Q) - T(Q')\|_{\infty} \leq \gamma \|Q - Q'\|_{\infty}$

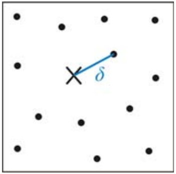
$\Rightarrow$ fuzzy Q-iteration converges monotonically to θ^*



$\|\widehat{Q}^{\theta^*} - Q^*\|_{\infty}$ bounded

Consistency

- Consistency: $\widehat{Q}^{\theta^*} \to Q^*$ as accuracy increases
- Accuracy: $\begin{cases} \delta_x = \max\limits_{x \in X} \min\limits_{i} \|x - x_i\|_2 \\ \delta_u = \max\limits_{u \in U} \min\limits_{j} \|u - u_j\|_2 \end{cases}$



- Assuming f, ρ Lipschitz:

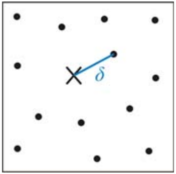
$$\|f(x,u) - f(\bar{x},\bar{u})\|_2 \le L_f(\|x - \bar{x}\|_2 + \|u - \bar{u}\|_2)$$
$$|\rho(x,u) - \rho(\bar{x},\bar{u})| \le L_\rho(\|x - \bar{x}\|_2 + \|u - \bar{u}\|_2)$$

(& certain requirements on the MFs)

$\Rightarrow \lim_{\delta_x \to 0, \delta_u \to 0} \widehat{Q}^{\theta^*} = Q^*$ — consistency

Consistency

- Consistency: $\widehat{Q}^{\theta^*} \to Q^*$ as accuracy increases
- Accuracy: $\begin{cases} \delta_x = \max_{x \in X} \min_i \|x - x_i\|_2 \\ \delta_u = \max_{u \in U} \min_j \|u - u_j\|_2 \end{cases}$



- Assuming f, ρ Lipschitz:

$$\|f(x,u) - f(\bar{x},\bar{u})\|_2 \leq L_f(\|x - \bar{x}\|_2 + \|u - \bar{u}\|_2)$$
$$|\rho(x,u) - \rho(\bar{x},\bar{u})| \leq L_\rho(\|x - \bar{x}\|_2 + \|u - \bar{u}\|_2)$$

(& certain requirements on the MFs)

$\Rightarrow \lim_{\delta_x \to 0, \delta_u \to 0} \widehat{Q}^{\theta^*} = Q^*$ — consistency

The swing-up problem

$$J\ddot{\alpha} = mgl\sin(\alpha) - b\dot{\alpha} - \frac{K^2}{R}\dot{\alpha} + K_m u$$

- $x = [\alpha, \dot{\alpha}]^{\mathrm{T}}$

  $\alpha \in [-\pi, \pi]$ angle

  $\dot{\alpha} \in [-15\pi, 15\pi]$ velocity
- $u \in [-3, 3]$ control voltage
- $T_s = 0.005$

- Goal: stabilize in unstable equilibrium (pointing up)
- Difficulty: insufficient actuation, need to swing back & forth

The swing-up problem

$$J\ddot{\alpha} = mgl\sin(\alpha) - b\dot{\alpha} - \frac{K^2}{R}\dot{\alpha} + K_m u$$

- $x = [\alpha, \dot{\alpha}]^{\mathrm{T}}$

  $\alpha \in [-\pi, \pi]$ angle

  $\dot{\alpha} \in [-15\pi, 15\pi]$ velocity
- $u \in [-3, 3]$ control voltage
- $T_{\mathrm{s}} = 0.005$

- Goal: stabilize in unstable equilibrium (pointing up)
- Difficulty: insufficient actuation, need to swing back & forth

The swing-up problem

$$J\ddot{\alpha} = mgl\sin(\alpha) - b\dot{\alpha} - \frac{K^2}{R}\dot{\alpha} + K_m u$$

- $x = [\alpha, \dot{\alpha}]^{\mathrm{T}}$

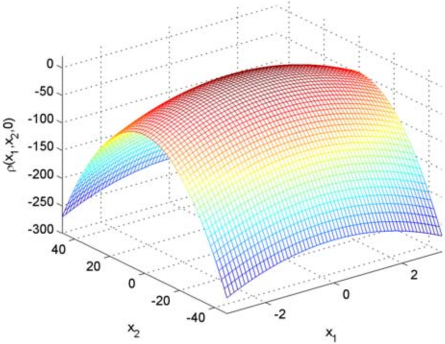
  $\alpha \in [-\pi, \pi]$ angle

  $\dot{\alpha} \in [-15\pi, 15\pi]$ velocity
- $u \in [-3, 3]$ control voltage
- $T_{\mathrm{s}} = 0.005$

- Goal: stabilize in unstable equilibrium (pointing up)
- Difficulty: insufficient actuation, need to swing back & forth

Reward function

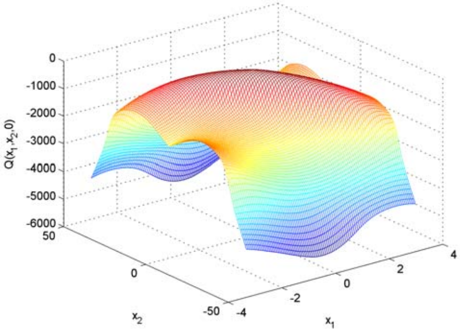
- Reward function: $\rho(x, u) = -x^{\mathrm{T}} \begin{bmatrix} 5 & 0 \\ 0 & 0.1 \end{bmatrix} x - u^{\mathrm{T}} 1 u$

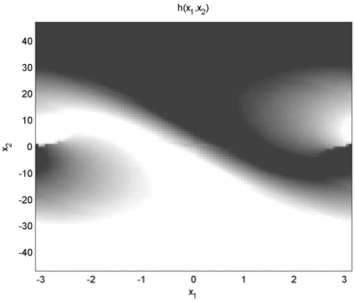


- Discount factor: $\gamma = 0.98$

Near-optimal solution

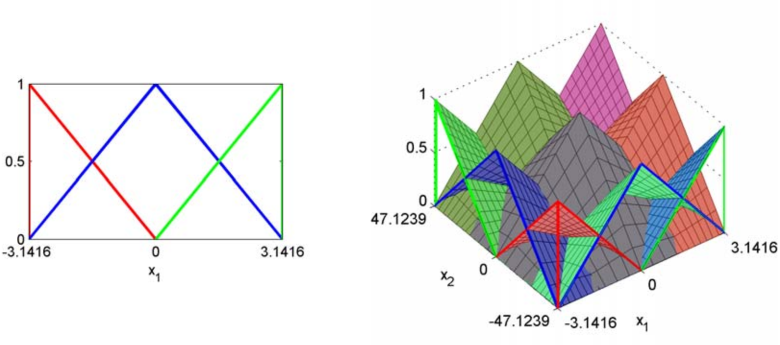
- Left: Q-function for $u = 0$; right: policy





Approximator setup

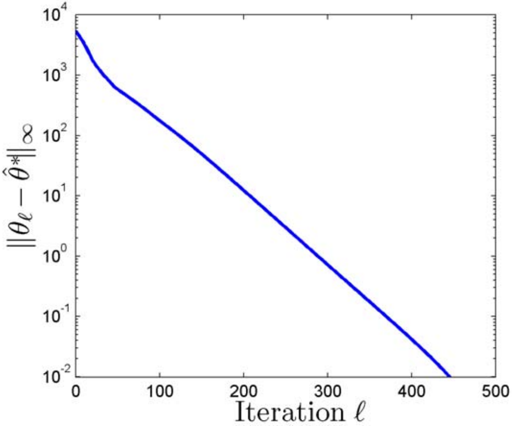
- N' equidistant triangular MFs on each axis ($\Rightarrow N = N'^2$)
- 2D MFs: products of 1D MFs. Example: $N' = 3$



- M equidistant discrete actions

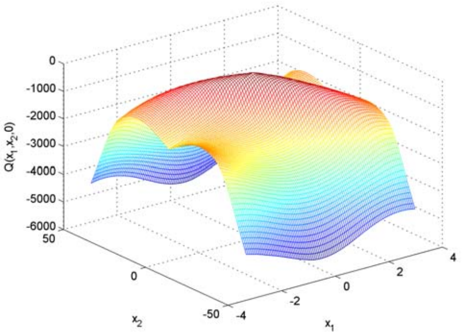
Example: Convergence

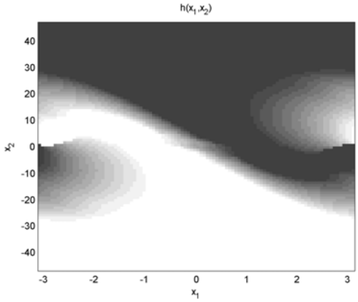
- $N' = 41$, $M = 15$
- Criterion: $\|\theta_{\ell+1} - \theta_\ell\|_\infty \leq 10^{-2}$
- Monotonic convergence



Example: Solution

- $N' = 41$, $M = 15$
- Left: Q-function for $u = 0$; right: policy
- Close to optimal





Consistency & discontinuous rewards

- Consistency requires Lipschitz rewards
- Study effect of discontinuous rewards
- Introduce discontinuity without altering the problem

$$\rho'(x, u) = \rho(x, u) + \gamma\psi(f(x, u)) - \psi(x)$$

- ρ' preserves quality of policies, $Q^h_{\rho'} - Q^*_{\rho'} = Q^h_\rho - Q^*_\rho$
- ψ discontinuous, positive around origin:

$$\psi(x) = \begin{cases} 30 & \text{if } |x_1| \leq \pi/4 \text{ and } |x_2| \leq 2\pi \\ 0 & \text{otherwise} \end{cases}$$

Consistency & discontinuous rewards

- Consistency requires Lipschitz rewards
- Study effect of discontinuous rewards
- Introduce discontinuity without altering the problem

$$\rho'(x, u) = \rho(x, u) + \gamma\psi(f(x, u)) - \psi(x)$$

- ρ' preserves quality of policies, $Q^h_{\rho'} - Q^*_{\rho'} = Q^h_\rho - Q^*_\rho$
- ψ discontinuous, positive around origin:

$$\psi(x) = \begin{cases} 30 & \text{if } |x_1| \leq \pi/4 \text{ and } |x_2| \leq 2\pi \\ 0 & \text{otherwise} \end{cases}$$

Consistency & discontinuous rewards

- Consistency requires Lipschitz rewards
- Study effect of discontinuous rewards
- Introduce discontinuity without altering the problem

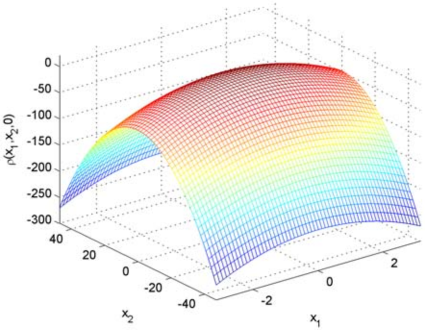
$$\rho'(x, u) = \rho(x, u) + \gamma\psi(f(x, u)) - \psi(x)$$

- ρ' preserves quality of policies, $Q^h_{\rho'} - Q^*_{\rho'} = Q^h_{\rho} - Q^*_{\rho}$
- ψ discontinuous, positive around origin:

$$\psi(x) = \begin{cases} 30 & \text{if } |x_1| \leq \pi/4 \text{ and } |x_2| \leq 2\pi \\ 0 & \text{otherwise} \end{cases}$$

Discontinuous reward

- Left, for comparison: original ρ; right: discontinuous ρ'



Consistency study

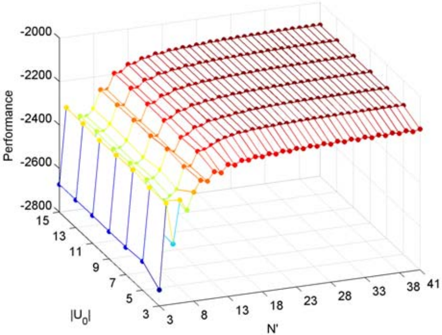
- $N' \in \{3, 4, 5, \ldots, 41\}$ equidistant MFs
- $M \in \{3, 5, \ldots, 15\}$ equidistant actions
(odd to always include $u = 0$)
- Fuzzy Q-iteration with continuous ρ and discontinuous ρ'
- Always evaluate with ρ, average return from initial states:
$X_0 = \{-\pi, -5\pi/6, -4\pi/6, \ldots, \pi\} \times \{-16\pi, -14\pi, \ldots, 16\pi\}$

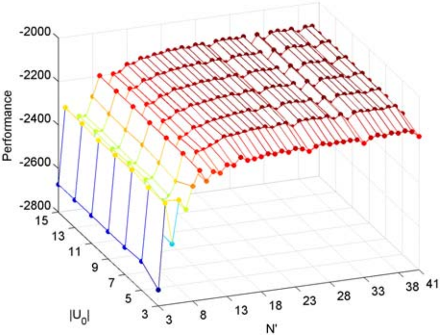
Consistency study

- $N' \in \{3, 4, 5, \ldots, 41\}$ equidistant MFs
- $M \in \{3, 5, \ldots, 15\}$ equidistant actions
  (odd to always include $u = 0$)
- Fuzzy Q-iteration with continuous ρ and discontinuous ρ'
- Always evaluate with ρ, average return from initial states:
  $X_0 = \{-\pi, -5\pi/6, -4\pi/6, \ldots, \pi\} \times \{-16\pi, -14\pi, \ldots, 16\pi\}$

Consistency results (simulation)

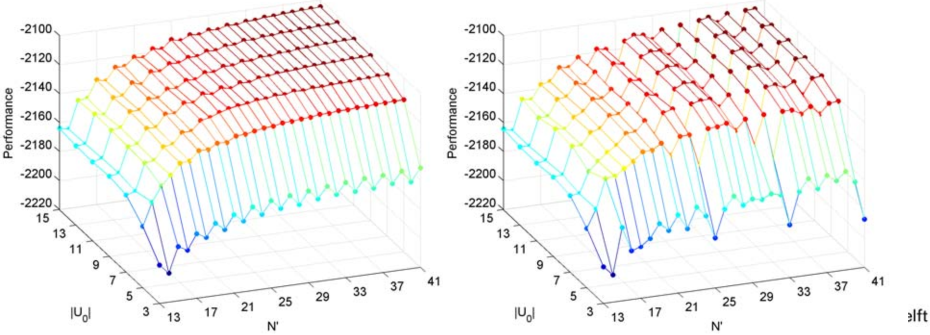
- Left: continuous ρ; right: discontinuous ρ'
- Performance variation decreases for ρ, not for ρ'
- Performance not monotonous as N, M increase
- M not very important





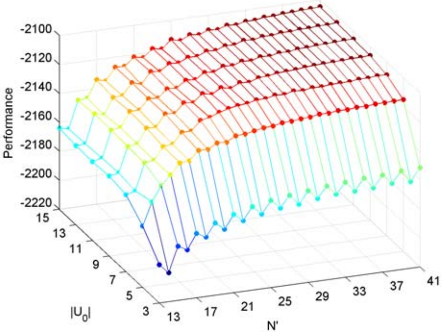
Consistency results (simulation)

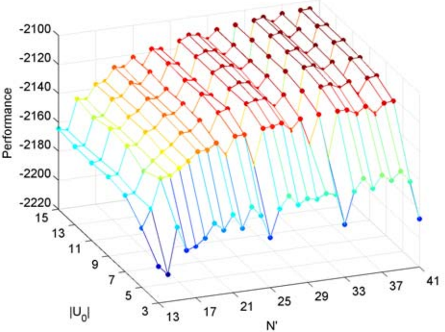
- Left: continuous ρ; right: discontinuous ρ'
- Performance variation decreases for ρ, not for ρ'
- Performance not monotonous as N, M increase
- M not very important



Consistency results (simulation)

- Left: continuous ρ; right: discontinuous ρ'
- Performance variation decreases for ρ, not for ρ'
- Performance not monotonous as N, M increase
- M not very important





Demo

Demo

$N' = 41, M = 15$

Conclusion and future work

- Fuzzy Q-iteration: fuzzy approx in X; discretization of U
- Algorithm is convergent & consistent
- Good performance in simulation & with real system
- Continuous reward functions important in practice

Ongoing & future work

- Automated discovery of MFs
- Sample-based and online techniques

Conclusion and future work

- Fuzzy Q-iteration: fuzzy approx in X; discretization of U
- Algorithm is convergent & consistent
- Good performance in simulation & with real system
- Continuous reward functions important in practice

Ongoing & future work

- Automated discovery of MFs
- Sample-based and online techniques

Thank you

Thank you!
Questions?

Thanks to BSIK-ICIS project #BSIK03024
(Interactive Collaborative Information Systems)
& NWO Van Gogh grant #VGP 79-99
& STW-VIDI project #DWV.6188

References I

D. Bertsekas (2007).
Dynamic Programming and Optimal Control,
Athena Scientific, vol. 2, 3rd ed.

D. Bertsekas and J.N. Tsitsiklis (1996).
Neuro-Dynamic Programming,
Athena Scientific.

J.N. Tsitsiklis and B. Van Roy (1996).
Feature-based methods for large scale dynamic programming.
Machine Learning, 22:59–94.

Action approximation

- Discrete actions $u_1, \ldots, u_M$

$$\widehat{Q}^\theta(x,u) = \sum_{i=1}^{N} \varphi_i(x)\theta_{i,j} \qquad j = \arg\min_{j'} \|u - u_{j'}\|$$

$\Rightarrow \widehat{Q}^\theta$ constant in Voronoi cell of each u_j

- Example: Voronoi partitions of $U = [-1,1] \times [-1,1]$ for random & equidistant discretizations

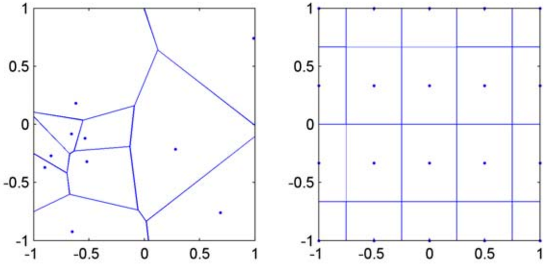
Action approximation

- Discrete actions $u_1, \dots, u_M$

$$\widehat{Q}^{\theta}(x, u) = \sum_{i=1}^{N} \varphi_i(x)\theta_{i,j} \qquad j = \arg\min_{j'} \left\| u - u_{j'} \right\|$$

$\Rightarrow \widehat{Q}^{\theta}$ constant in Voronoi cell of each u_j

- Example: Voronoi partitions of $U = [-1, 1] \times [-1, 1]$ for random & equidistant discretizations



Consistency results (cont'd)

- Average performance over M, for every N'
- Performance with ρ usually at least as good as with ρ'