Efficient Reinforcement Learning
with
Multiple Reward Functions
for
Randomized Controlled Trial Analysis

Dan Lizotte, Michael Bowling, Susan A. Murphy

University of Michigan, University of Alberta

Overview

- **Why are we interested in multiple reward functions?**
  - Medical research applications for RL
- Algorithm for discrete (tabular) case
  - More efficient than previous approaches
- Algorithm for linear function approximation
  - Previous approaches not applicable

Application: Medical Decision Support

- Our goal is to use RL as a tool for data analysis for decision support:

  1. Take comparative effectiveness clinical trial data
  2. Produce a policy (or Q-function) based on patient features (state)
  3. Give the policy to a clinician

- But really, a policy is too prescriptive.
  - Our data are noisy and incomplete, causing uncertainty in the learned policy - other projects at Michigan and elsewhere.
  - Our output is intended for an "agent" whose reward function we do not know.

Example: Schizophrenia

- In treatment of schizophrenia, one wants few symptoms but good functionality. This is often unachievable.

  1. This is a chronic disease. Patient state changes over time.
  2. The effect of different treatments varies from patient to patient.
  3. Different people may have very different preferences about which to give up. Each has a different reward function/objective.

- Properties 1. and 2. make the problem amenable to RL. The goal of this work is to deal with 3. by not committing to a single reward function.

- Let's look at an idealized version of batch RL, and compare with the type of data we actually have.

Fixed-Horizon Batch RL, Idealized Version

- Get trajectories $s_1^i, a_1^i, r_1^i, s_2^i, a_2^i, r_2^i, \ldots, s_T^i, a_T^i, r_T^i$
- Find a policy that chooses actions to maximize expected sum of future rewards. (Note fixed horizon and knowledge of t.)
- One possibility: Fitted Q-iteration
  - Learn $Q_T(s_T, a_T) \approx E[R_T | s_T, a_T]$
  - Move backward through time:
  - $Q_t(s_t, a_t) \approx E[R_t + \max_{a'_{t+1}} Q_{t+1}(S_{t+1}, a'_{t+1}) | s_t, a_t]$
- Expectations are approximated using data

Fixed-Horizon Batch RL, Realistic Version

- Get trajectories $o_0^i, a_1^i, o_1^i, a_2^i, o_2^i, \ldots, a_T^i, o_T^i$
- The o_t^i include many measurements, including symptoms, side-effects, genetic information, ...
- We must define (much like in state-feature construction)
    - $s_t^i = s_t^i(o_{0:t-1}^i, a_{1:t-1}^i)$
    - $r_t^i = r_t^i(o_{0:t}^i, a_{1:t}^i)$
- Now we have $s_1^i, a_1^i, r_1^i, s_2^i, a_2^i, r_2^i, \ldots, s_T^i, a_T^i, r_T^i$, can do fitted-Q
- How can we defer the choice of $r_t^i(o_{0:t}^i, a_{1:t}^i)$?

Multiple Reward Functions

- Consider a pair of important objectives. Suppose $r_t^{(0)}$ reflects level of symptoms and $r_t^{(1)}$ reflects level of functionality.

- Consider the set of convex combinations of reward functions, e.g.

$$r_t(s,a,\delta) \equiv (1 - \delta) \cdot r_t^{(0)}(s,a) + \delta \cdot r_t^{(1)}(s,a)$$

- Each δ identifies a specific reward function, and induces a corresponding $Q_t(\cdot,\cdot,\delta)$. Depending on δ, the optimal policy "cares more" about $r_t^{(0)}$ or $r_t^{(1)}$.

- Standard approach: "Preference Elicitation"
  - Try to determine the decision-maker's *true* value of δ via time tradeoff, standard gamble, visual analog scales,...

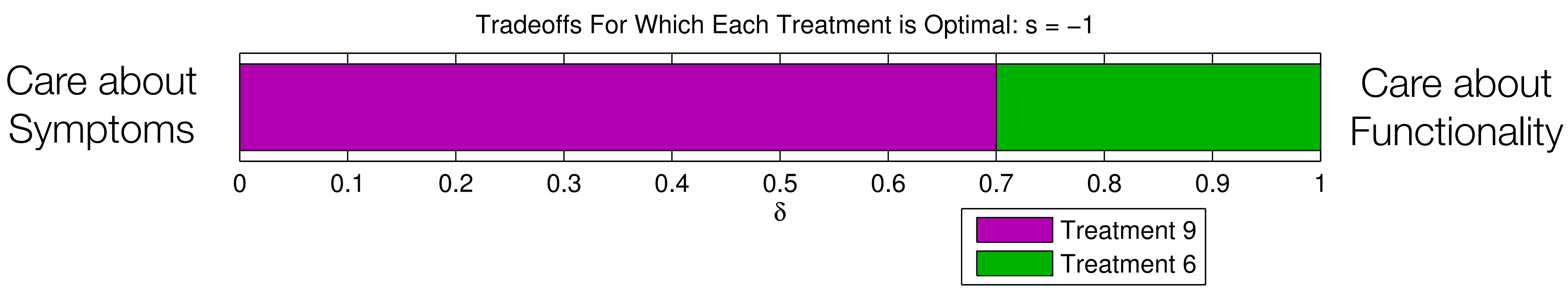
“Inverse”
Preference Elicitation

- We propose a different approach

- Take $r(s,a,\delta) \equiv (1 - \delta) \cdot r^{(0)}(s,a) + \delta \cdot r^{(1)}(s,a)$

- Run analysis to find optimal actions *given all* δ,
  i.e. learn $Q_t(s,a,\delta)$ and $V_t(s,\delta)$ for all $t \in \{1,2,\ldots,T\}$
  and for all $\delta \in [0, 1]$

- Given a new patient’s state, report, for each action, the range of δ for which it is optimal.



Overview

- Why are we interested in multiple reward functions?
  - Medical research applications for RL
- **Algorithm for discrete (tabular) case**
  - More efficient than previous approaches
- Algorithm for linear function approximation
  - Previous approaches not applicable

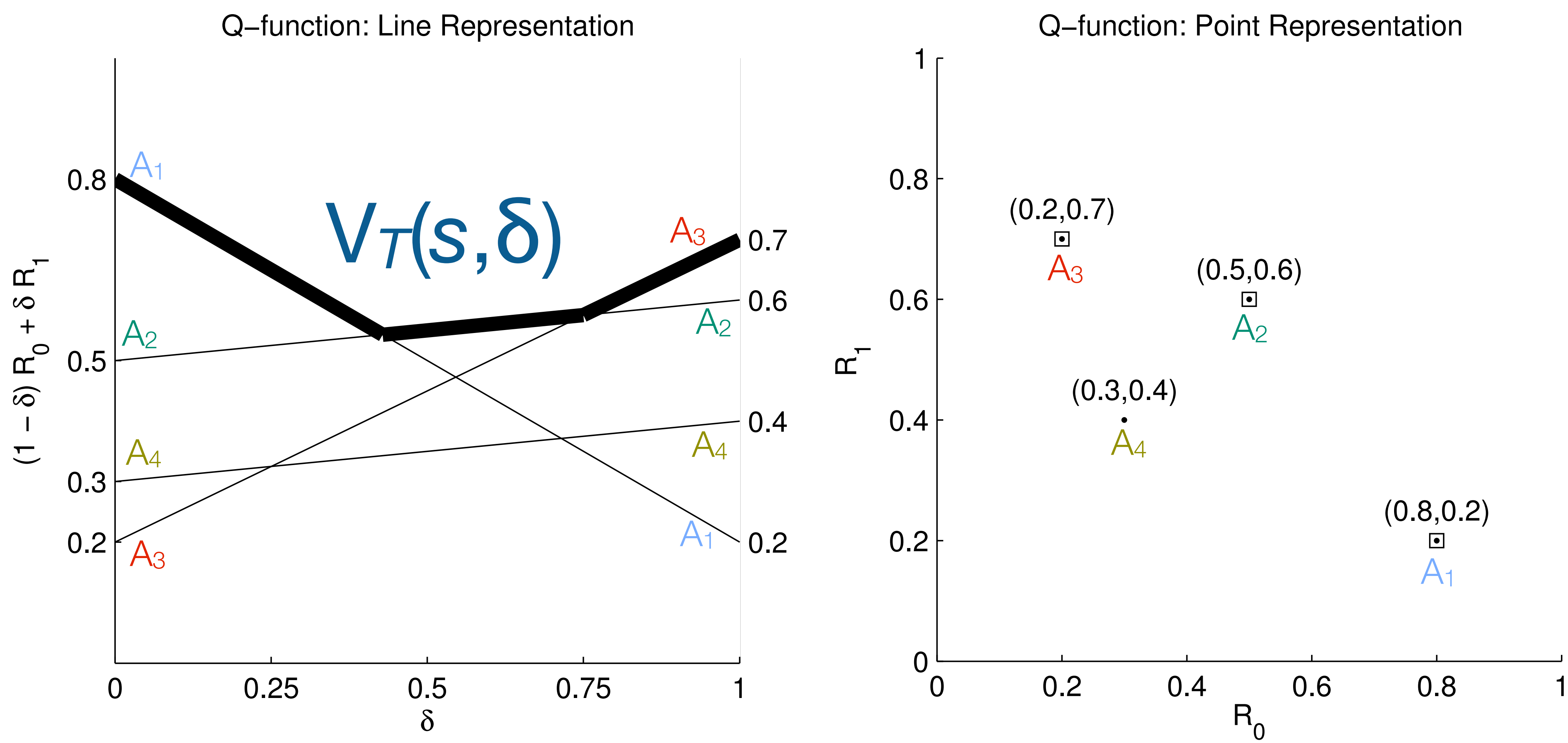
Algorithm for Discrete State Space: Preview

- $r_T(s,a,\delta) \equiv (1 - \delta) \cdot r_T^{(0)}(s,a) + \delta \cdot r_T^{(1)}(s,a)$
- $Q_T(s,a,0)$, $Q_T(s,a,1)$ are average rewards, $Q_T(s,a,\delta)$ is linear in δ
- $V_T(s,\delta)$ is continuous and piecewise linear in δ
  - Knots introduced by pointwise max over a
- $Q_{T\text{-}1}(s,a,\delta)$ is continuous and piecewise linear in δ
  - Average of $V_T(s',\delta)$ over trajectories with s,a,s' tuples.

[WLOG considering terminal rewards only]

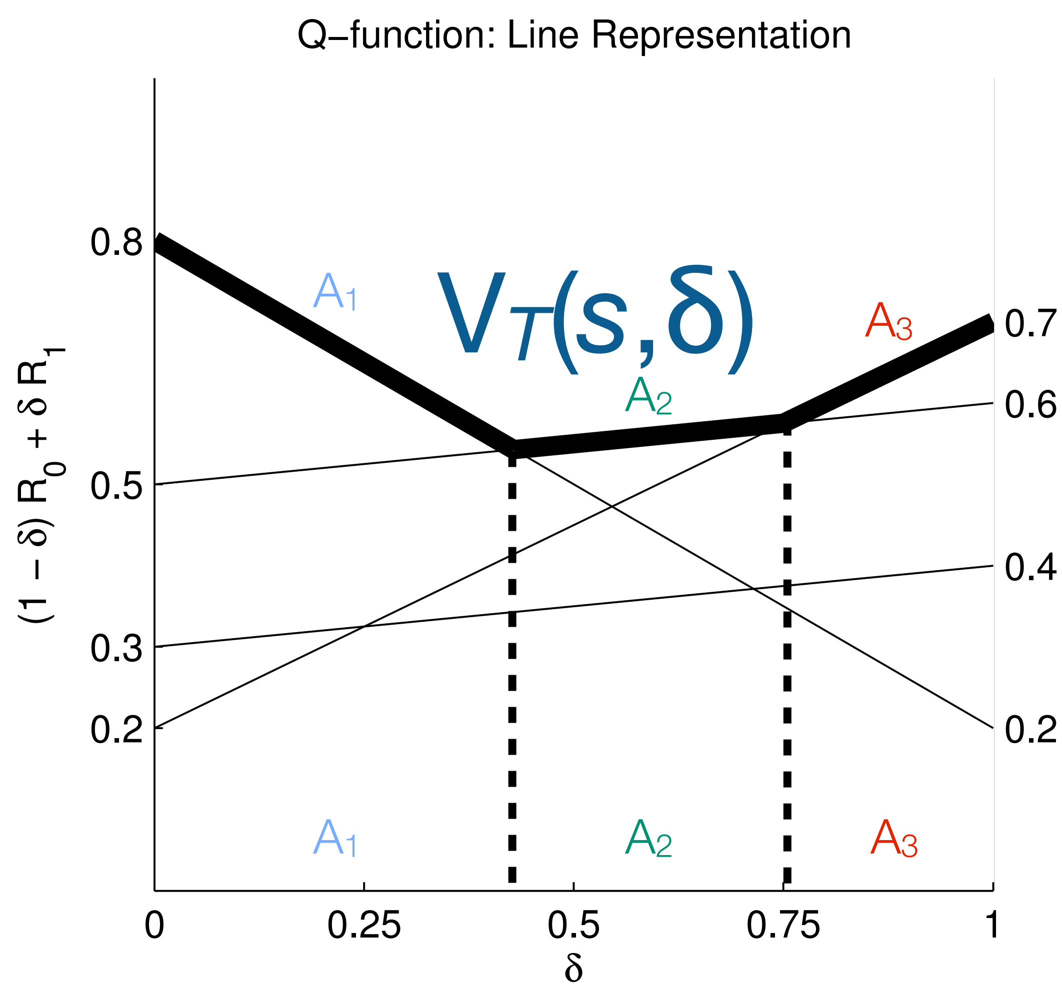
Value Backup: Max Over Actions

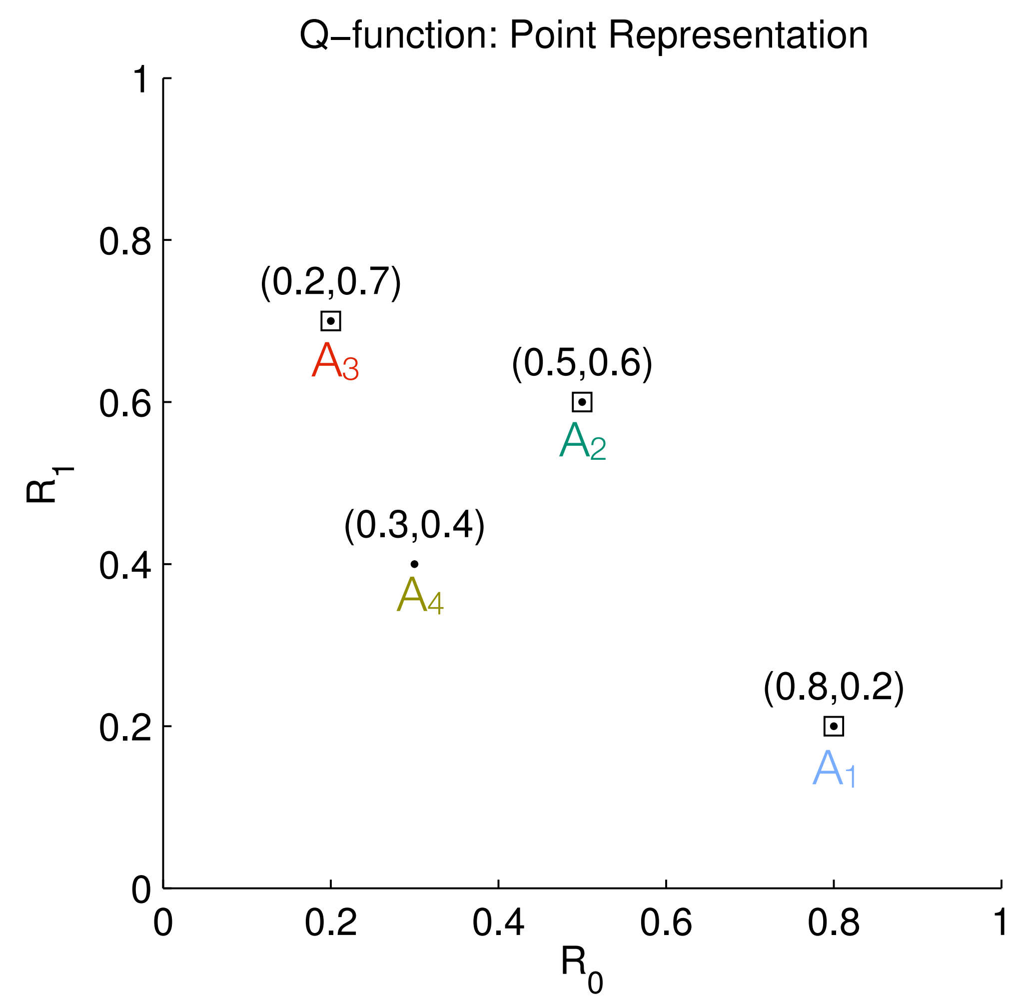
- $Q_T(s,a,0)$, $Q_T(s,a,1)$ are average rewards, $Q_T(s,a,\delta)$ is linear in δ
- $V_T(s,\delta)$ is continuous and piecewise linear in δ
  - Knots introduced by pointwise max over *a* found by convex hull



Value Backup: Max Over Actions

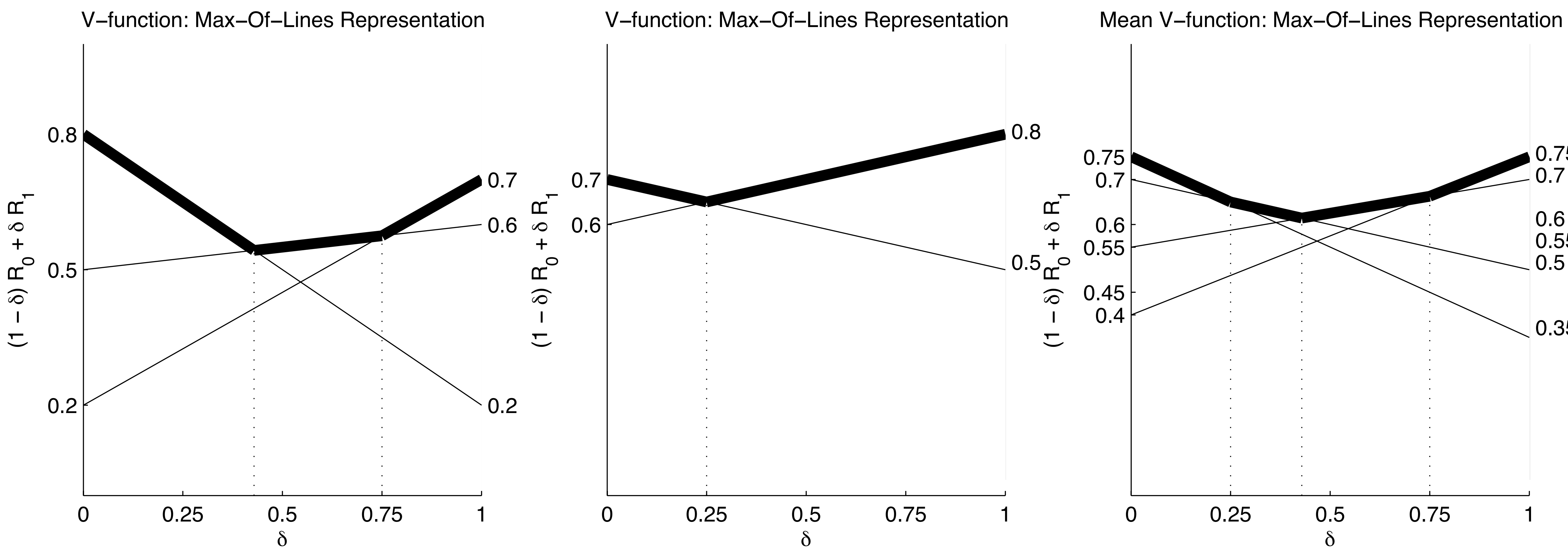
- Our value function representation “remembers” which actions are optimal over which intervals of delta





Value Backup: Expectation Over Next State

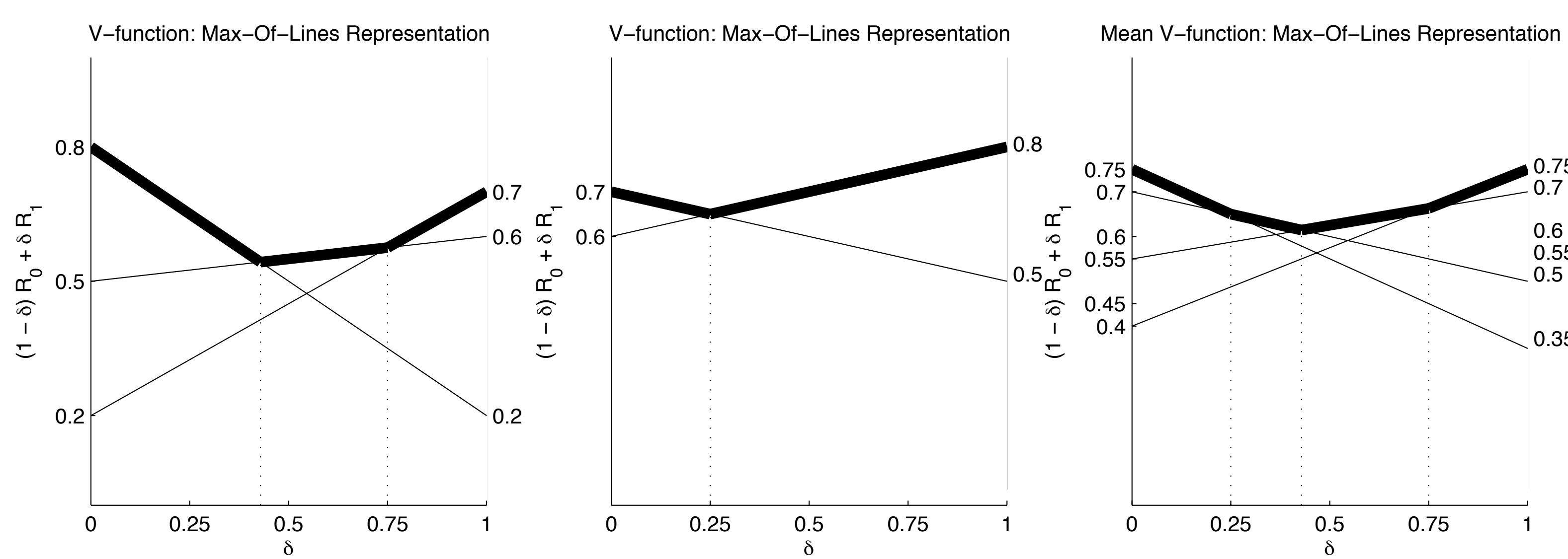
- $Q_{T-1}(s,a,\delta)$ is continuous and piecewise linear in δ
  - Average of $V_T(s',\delta)$ over trajectories with s,a,s' tuples.



$V_T(s_1,\delta)$ $V_T(s_2,\delta)$ $Q_{T-1}(s,a,\delta) = E_{S'|s,a}[R_{T-1}+V_{t+1}(S',\delta)]$

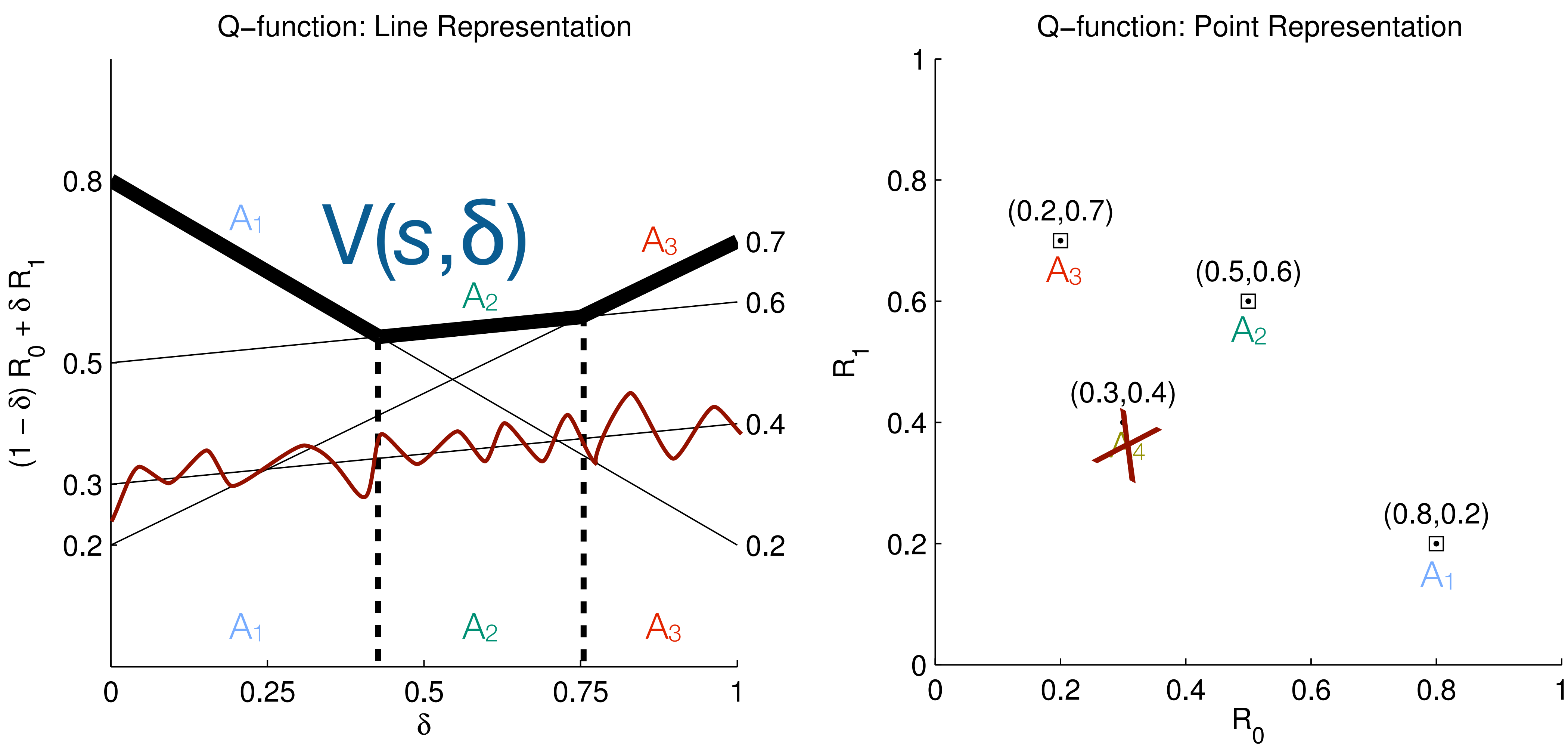
Value Backups: Discrete State (tabular)

- $Q_T(s,a,\delta)$ is piecewise linear, with $O(|A|)$ pieces
- $E_{S'|s,a}[V_T(S',\delta)]$ is piecewise linear, with $O(|S||A|)$ pieces
- At $T\text{-}t$, for each s, $V(s,\delta)$ has $O(|S|^{T-t}|A|^{T-t+1})$ pieces
- Can compute in $O(|S|^{T-t+1}|A|^{T-t+1})$ time by operating directly on the piecewise linear functions
  Previous work took $O(|S|^{2(T-t)+1}|A|^{2(T-t)+1}\log |S|^{2(T-t)+1}|A|^{2(T-t)+1})$ time
- $V_{t-1}(s,\delta)$ is convex linear combination of $V_t(s',\delta)$, therefore stays convex in δ



Dominated Actions

- Some actions are not optimal for any δ



- Some actions are not optimal for any (δ,s)!
  Can enumerate s to check this.

Overview

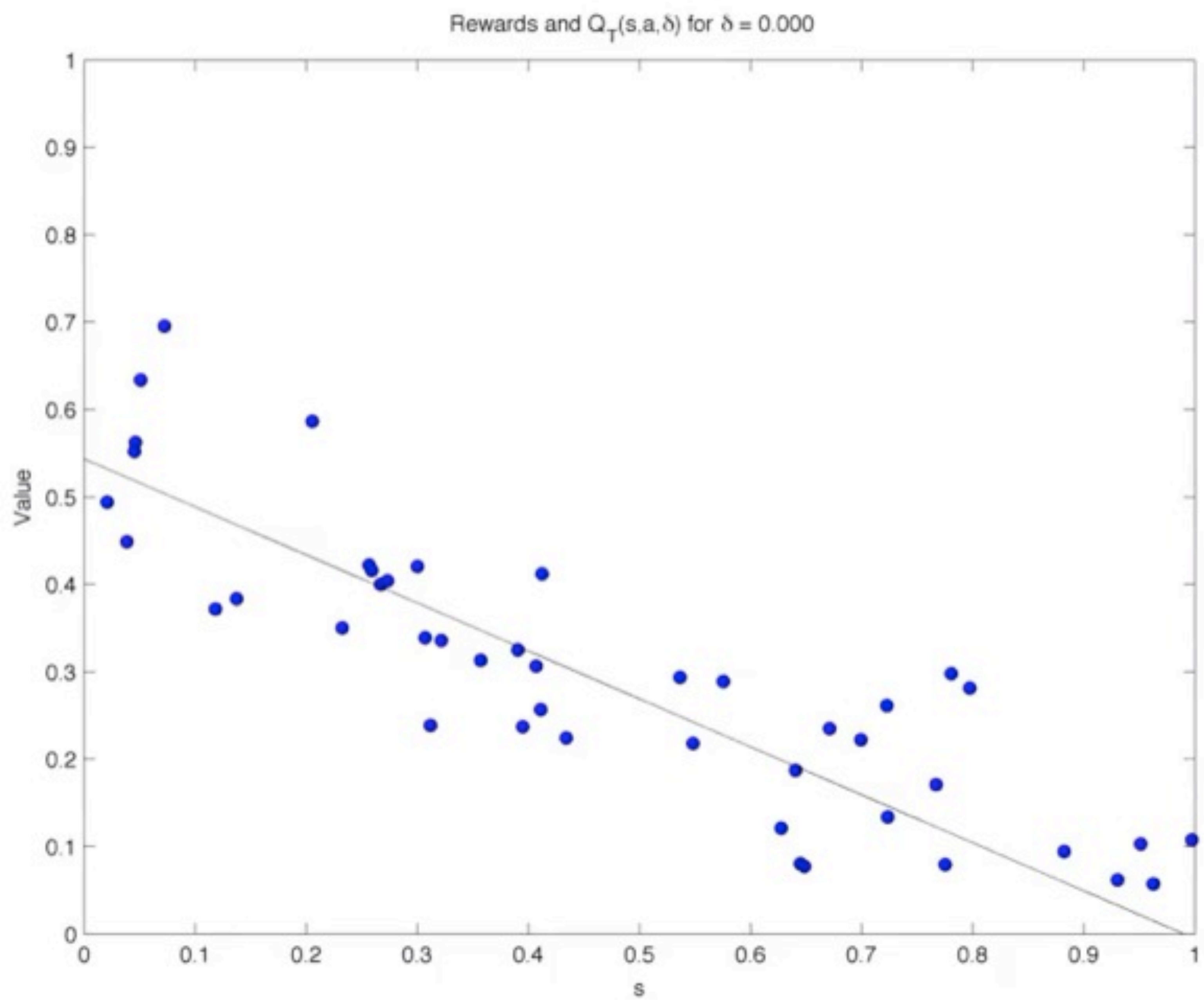
- Why are we interested in multiple reward functions?
  - Medical research applications for RL
- Algorithm for discrete (tabular) case
  - More efficient than previous approaches
- **Algorithm for linear function approximation**
  - Previous approaches not applicable

Continuous State Space, Linear Function Approx.

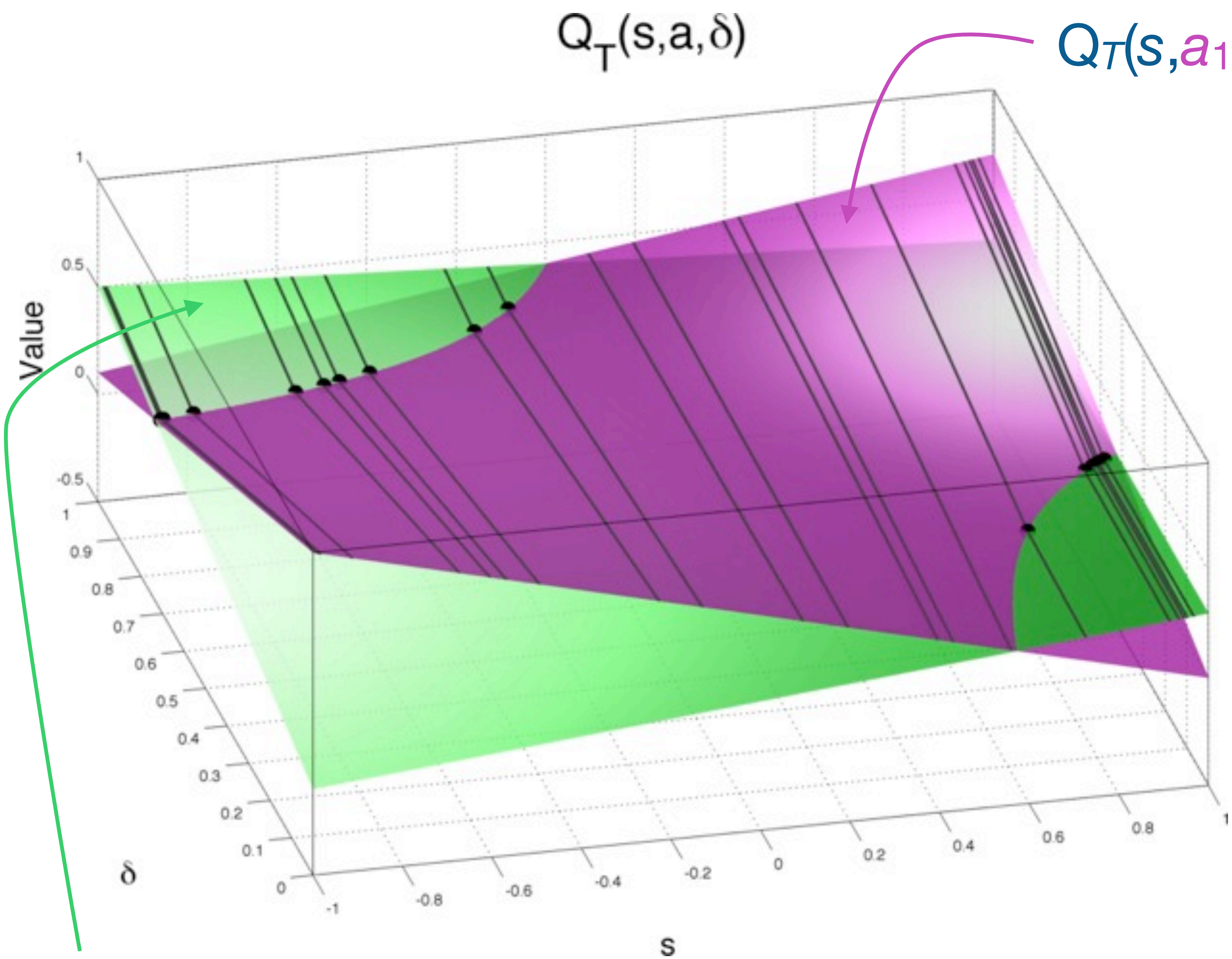
- What if we want to generalize across state?
- Estimate $Q_T(s,a,0)$, $Q_T(s,a,1)$ using linear regression rather than sample averages. Use these to compute $Q_T(s,a,\delta)$ for other δ
- Recall: $r_T(s,a,\delta) \equiv (1 - \delta)\cdot r_T^{(0)}(\mathrm{s,a}) + \delta\cdot r_T^{(1)}(\mathrm{s,a})$
- Construct design matrices S_a ($n_a \times p$), targets $\boldsymbol{r}_a(\delta)$ ($n_a \times 1$) from our data set
- $Q_T(s,a,\delta;\beta) = \beta_a(\delta)^{\mathrm{T}}s$, $\beta_a(\delta) = (S_a^{\mathrm{T}}S_a)^{-1}S_a^{\mathrm{T}}\boldsymbol{r}_a(\delta)$
  - $Q_T(s,a,\delta;\beta)$ linear in β, each element of β linear in r, and r linear in δ
- $Q_T(s,a,\delta;\beta) = ((1 - \delta)\cdot\beta_a(0) + \delta\cdot\beta_a(1))^{\mathrm{T}}\,[1,\,\mathrm{s}]^{\mathrm{T}}$

Movie of Q_T

- $Q_T(s,a,\delta;\beta) = ((1 - \delta) \cdot \beta_a(0) + \delta \cdot \beta_a(1))^T [1, s]^T$

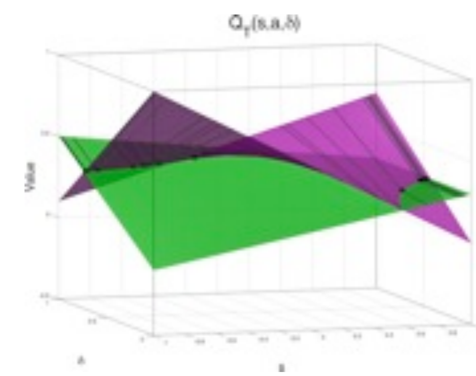


Maximization Over Actions



$Q_T(s,a_1,\delta;\beta)$

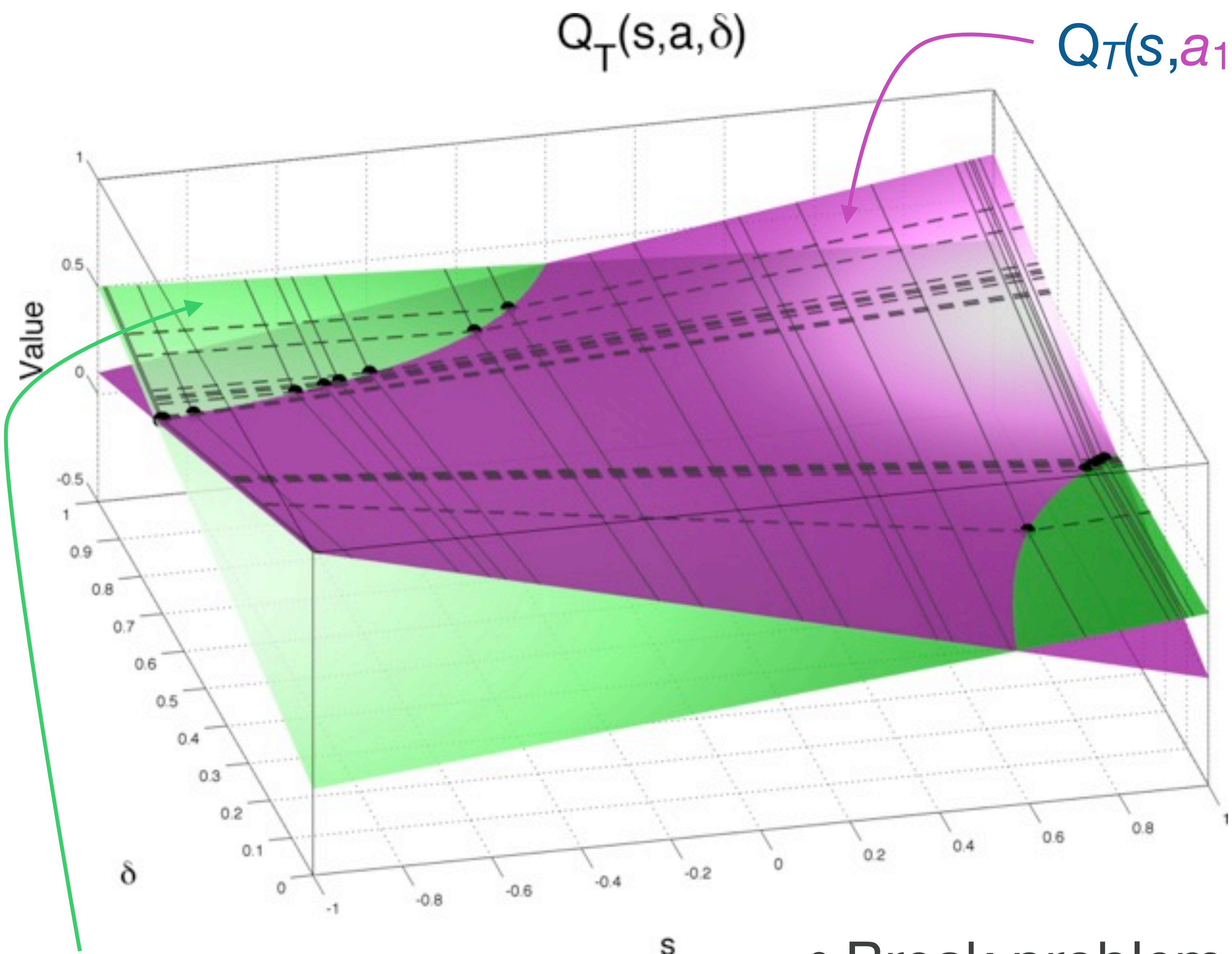
$Q_T(s,a_2,\delta;\beta)$



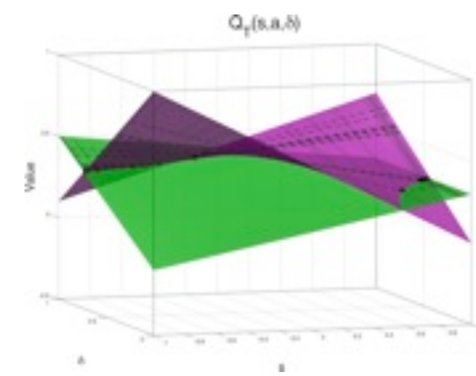
- $V_T(s,\delta)$ is continuous and piecewise linear in both δ and s
- While learning, we only evaluate $V_T(s,\delta)$ at s_i we have in our dataset
- Knots found by convex hull

Regression Over Next State

$Q_T(s,a_1,\delta;\beta)$



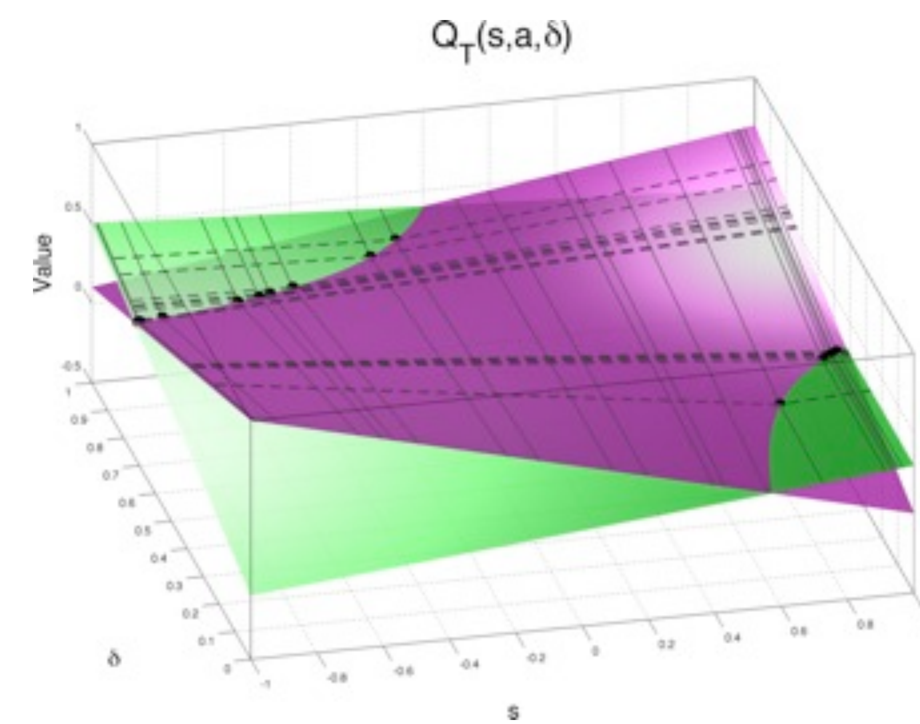
$Q_T(s,a_2,\delta;\beta)$



- While learning, we only evaluate $V_T(s,\delta)$ at s_i we have in our dataset
- Reg. coefficients for $Q_{T-1}(s,a,\delta;\beta)$ are **weighted** sums of $V_T(s_i,\delta)$
- Break problem into regions of δ-space where $V_T(s_i,\delta)$ are simultaneously linear

Value Backups: Continuous State

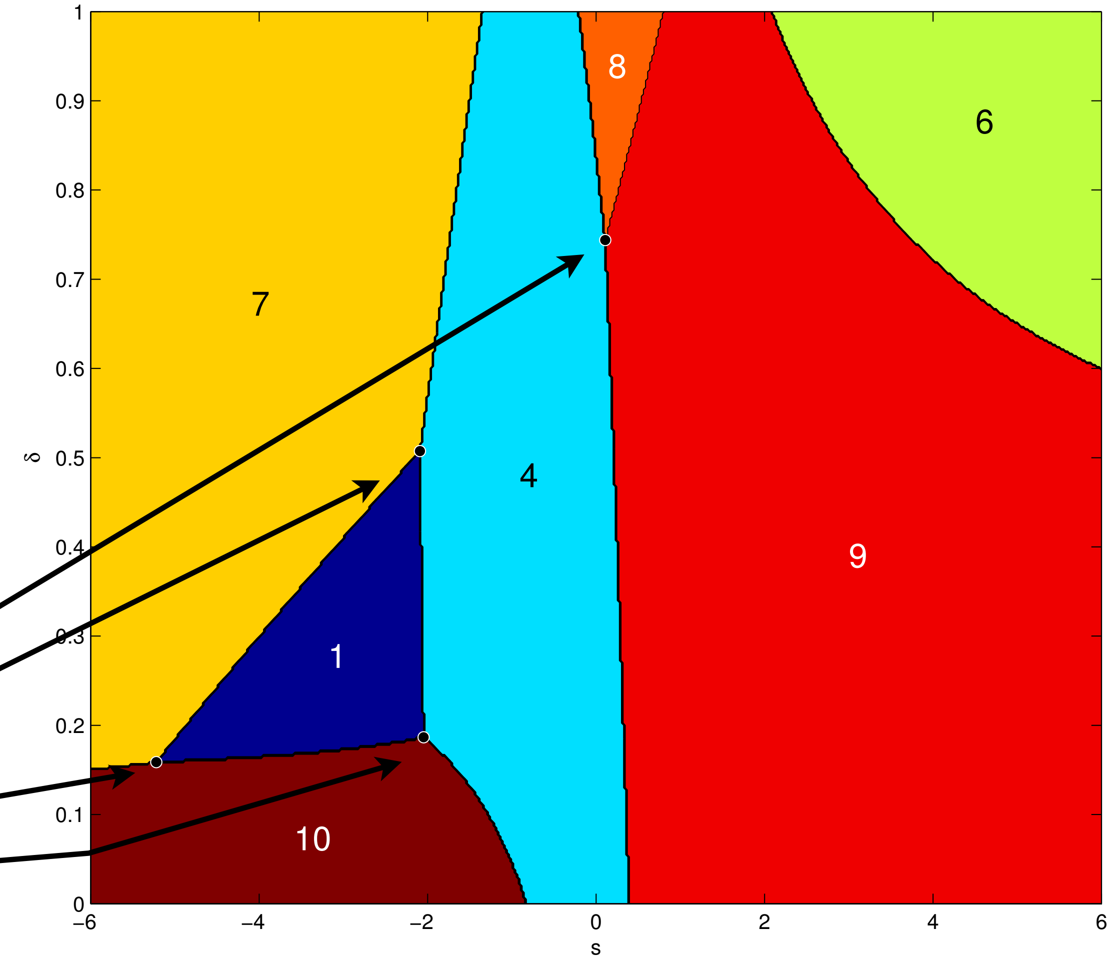
- $Q_T(s,a,\delta;\beta) = ((1 - \delta)\cdot\beta_a(0) + \delta\cdot\beta_a(1))^T\,[1, s]^T$
  - $V_T(s,\delta) = \max_a Q_T(s,a,\delta;\beta)$ is *piecewise linear* in δ
- $Q_{T-1}(s,a,\delta;\beta) = \beta_a(\delta)^T s,\ \ \beta_a(\delta) = (S_a^T S_a)^{-1} S_a^T\, \mathbf{V}_T(s',\delta)$ **is not convex in** δ
- Each element of $\mathbf{V}_T(s',\delta)$ is piecewise linear in δ
- Where $\mathbf{V}_T(s',\delta)$ are simultaneously linear, elements of $\beta_a(\delta)^T$ are linear. Must compute $\beta_a(\delta)^T$ at knot δs between linear regions



- $O(n|A|)$ knots, n is number of trajectories. At time T-t, there could be $O(n^{T-t}|A|^{T-t})$ knots.

Dominated Actions: Continuous State

- Set of dominated actions can be determined analytically in $O(|A|^3 \cdot \#knots)$ time for 1D state and 1D tradeoff
- Algorithm based on properties of boundaries between regions where different actions are optimal
- E.g., top-down view of Q-function, actions 2,3,5 are not optimal for any (s,δ).



Approach:
Identify 'triple points'

Reality Check

- Must compute $\beta_a(\delta)^T$ at knot δs between linear regions. At time T-t, there could be $O(n^{T-t}|A|^{T-t})$ knots, in the worst case.
- Is this even feasible? Consider 1000 randomly generated datasets, n = 1290, |A| = 3, T = 3, parameters similar to real data
- Maximum time for 1 simulation run is 6.55 seconds on 8 procs.

	Worst-case #knots	Observed Min	Observed Med	Observed Max
t=2	3870	687	790	910
t=1	$1.5 \cdot 10^7$	2814	3160	3916

Overview

- Why are we interested in multiple reward functions?
  - Medical research applications for RL
- Algorithm for discrete (tabular) case
  - More efficient than previous approaches
- Algorithm for linear function approximation
  - Previous approaches not applicable

Future Work - Computing Science, Statistics

- Allow more state variables
  - For backups: Easy! Each element of β is piecewise linear in δ
  - When checking for dominated actions, 2 reward functions plus 2 state variables is feasible. (Or 3 reward functions + 1 state variable.)
- Allow more reward functions
  - For backups: 3 reward functions feasible.
    Representing non-convex continuous piecewise linear functions in high dimensions appears difficult.
- Approximations, now that we know what we are approximating.
- **Measures of uncertainty for preference ranges**

Future Work - Clinical Science

1. Schizophrenia
   - Symptom reduction versus functionality, or weight gain
2. Major Depressive Disorder
   - Symptom reduction versus weight gain, other side-effects
3. Diabetes
   - Disease complications versus drug side-effects

Thanks!

- Supported by National Institute of Health grants R01 MH080015 and P50 DA10075

- Questions?

- Related work:

  Barrett, L. and Narayanan, S. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, 2008.