Efficient RL with Multiple Reward Functions for Randomized Clinical Trial Analysis

UNIVERSITY OF MICHIGAN

Daniel Lizotte, Michael Bowling, and Susan A. Murphy

danjl@umich.edu bowling@cs.ualberta.ca samurphy@umich.edu

UNIVERSITY OF ALBERTA

Motivation

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 (i.e. state)
3. Give the policy to a clinician

But really, a policy is too prescriptive. Our output is intended for an "agent" whose reward function we do not know.

In treatment of schizophrenia, one wants few symptoms but good functionality. Different people may have very different preferences about which to give up. Each has a different reward function.

We have a batch of trajectories like this: $o_0^i, a_1^i, o_1^i, a_2^i, o_2^i, \ldots, a_T^i, o_T^i$

The o_t^i include many measurements, including symptoms, side-effects, genetic markers, ...

We must define (much like in state-feature construction)

$$s_t^i = s_t^i(o_{0:t-1}^i, a_{1:t-1}^i)$$
$$r_t^i = r_t^i(o_{0:t}^i, a_{1:t}^i)$$
$$s_1^i, a_1^i, r_1^i, s_2^i, a_2^i, r_2^i, \ldots, s_T^i, a_T^i, r_T^i$$

Using these definitions, we can do fitted-Q iteration over the finite horizon, i.e. learn $Q_T(s_T, a_T) \approx E[R_T | s_T, a_T]$, then move backward through time:

$$Q_t(s_t, a_t) \approx E[R_t + \max_{a'_{t+1}} Q_{t+1}(S_{t+1}, a'_{t+1}) | s_t, a_t]$$

Consider combining a pair of important objectives into a single reward. 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,

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

Each δ identifies a specific reward function, and induces a corresponding $Q_t(\cdot,\cdot,\delta)$.

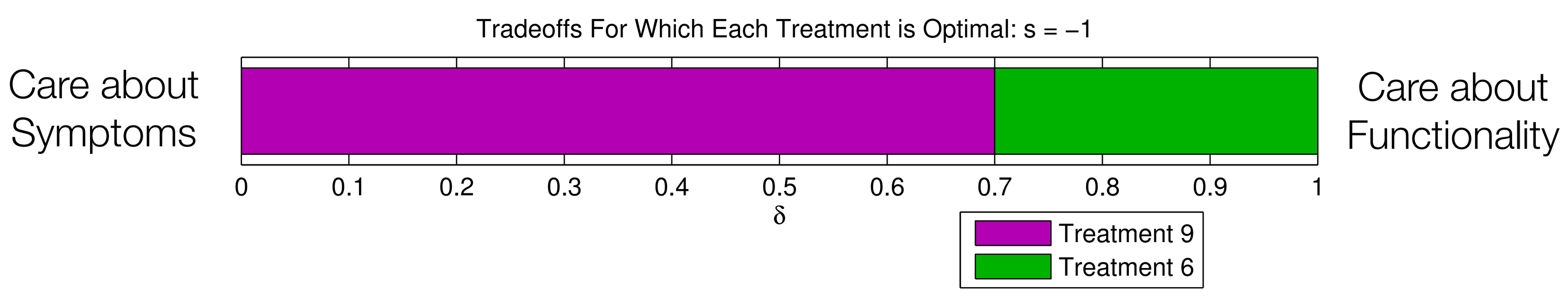
Depending on δ, the optimal policy "cares more" about $r_t^{(0)}$ or $r_t^{(1)}$.

Standard approach: "Preference Elicitation": Determine the decision-maker's value of δ.

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.



Say to patient in state s: "Take treatment 9 if you would trade up to 7 points of symptom reduction for 1 point of functionality improvement."

Goal: Represent $Q_t(s,a,\delta)$ and $V_t(s,\delta)$ in a way that reveals these preference ranges.

Algorithms for Value Backup

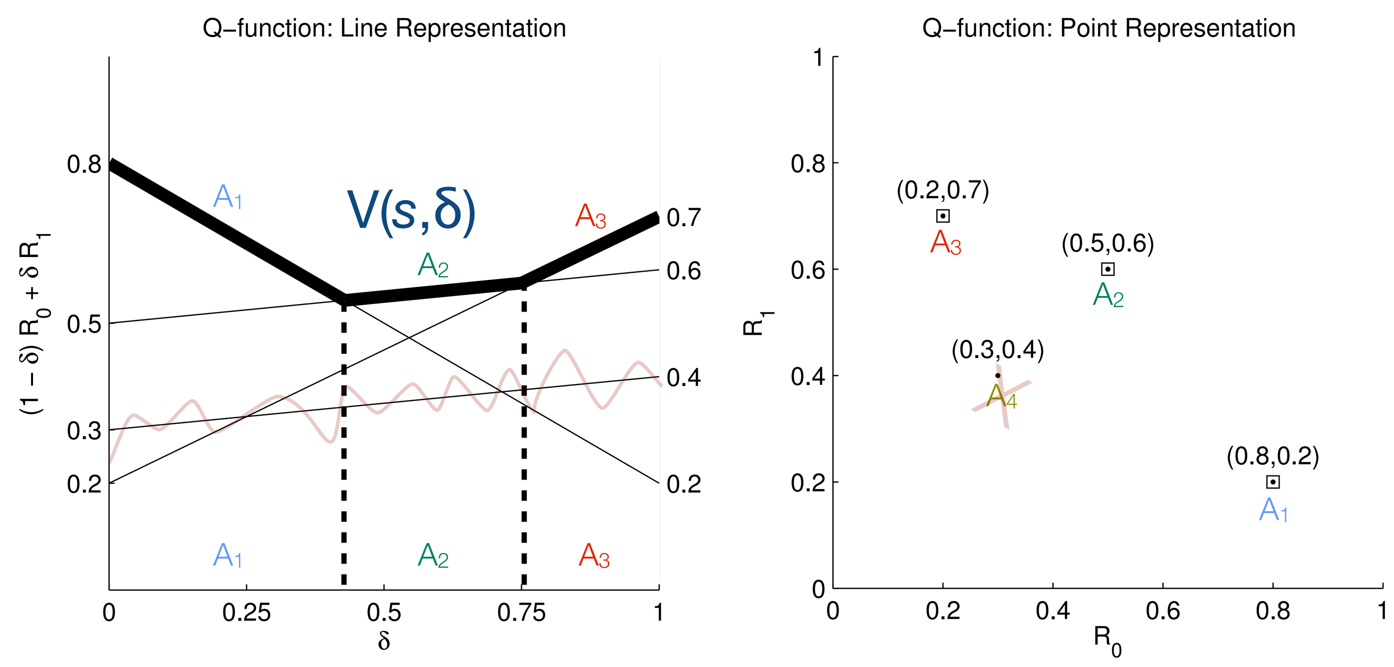
Discrete States

Max Over Actions

$Q_T(s,a,\delta)$ is linear in δ

$V_T(s,\delta)$ is continuous and piecewise linear in δ

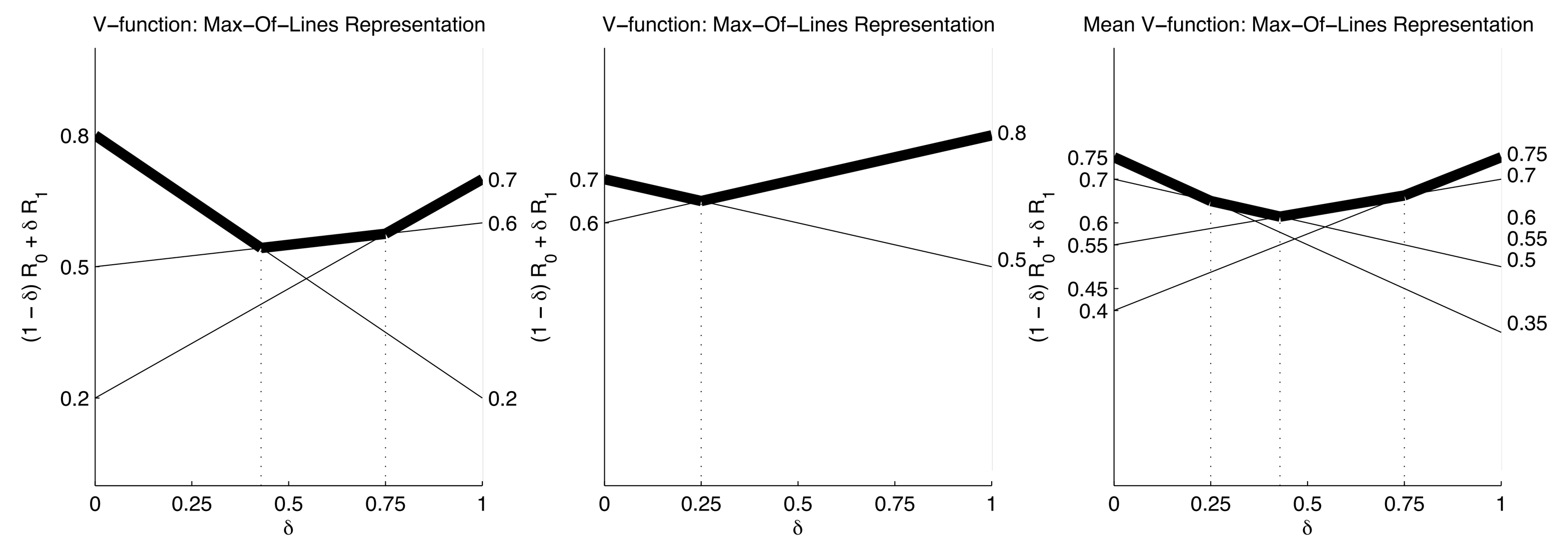
Knots introduced by pointwise max over a found by convex hull



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.



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

Continuous States

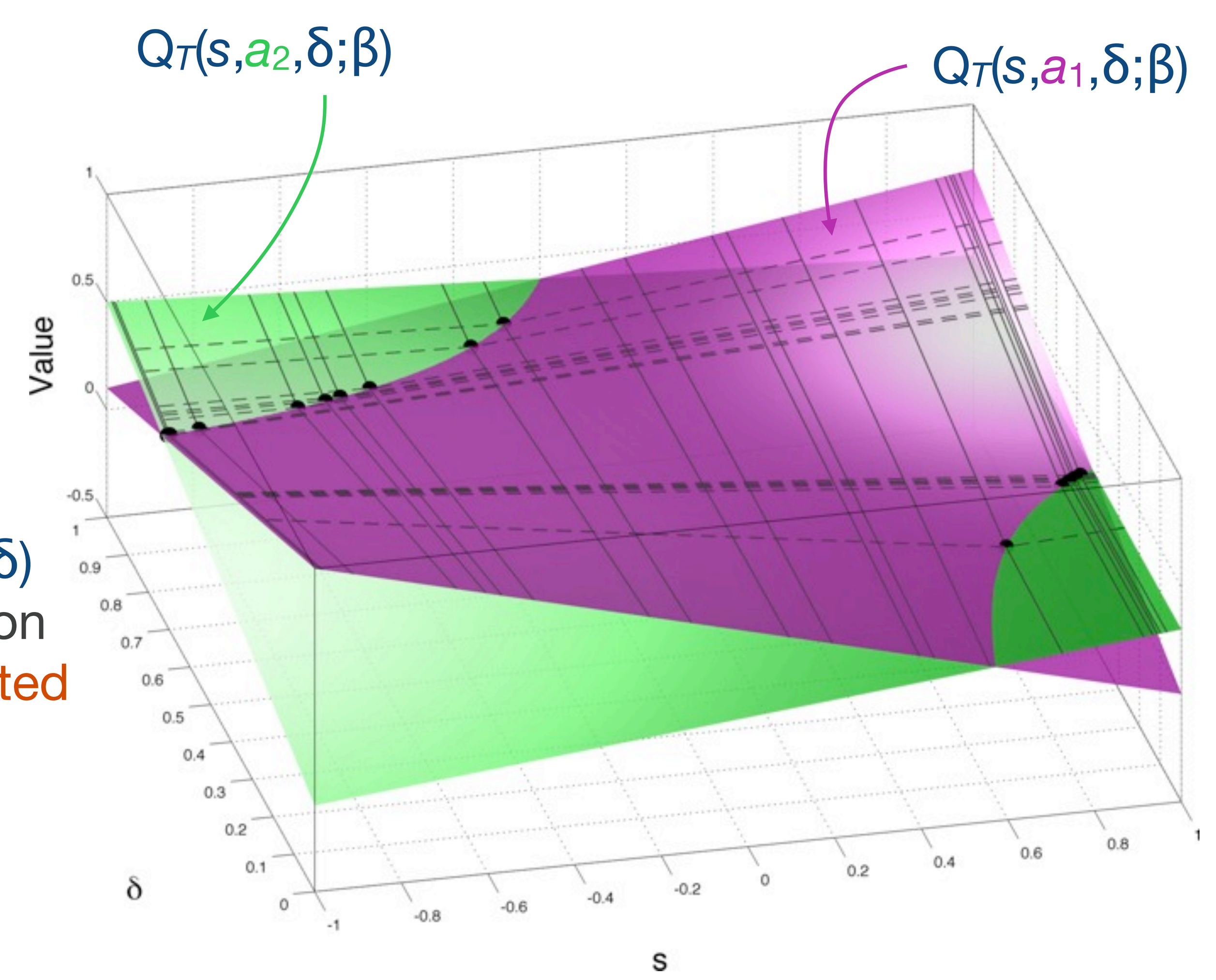
Estimate $Q_T(s,a,0)$, $Q_T(s,a,1)$ using linear regression rather than sample averages.

Use these to compute $Q_T(s,a,\delta)$ for other δ. Expectations for backups also use regression.

Max Over Actions

$Q_T(s,a,\delta)$ 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



Regression Over Next State

While learning, we only evaluate $V_T(s,\delta)$ at s_i we have in our dataset. Regression 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.

Complexity

Worst case, at time $T-t$, there could be $O(n^{T-t}|A|^{T-t})$ knots. In practice, there are far fewer. Empirical studies on typical clinical trial dataset sizes ($n = 1290$, $|A| = 3$, $T = 3$) induce ~3000 knots when worst case bounds indicate $1.5\cdot 10^7$ knots. Runtime: 6.55 seconds.

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