Inverse Preference Elicitation for Dynamic Treatment Regimes

Simon Fraser University - 7 December 2010

Dan Lizotte
Postdoctoral Fellow
Department of Statistics

With Michael Bowling, Susan Murphy
University of Alberta, University of Michigan

UNIVERSITY OF MICHIGAN

Outline

- Motivation: Symptoms and Side-Effects in Schizophrenia
- Introduction: Dynamic Treatment Regimes, Q-Learning
- Contributions: Inverse Preference Elicitation
  - Inverse Preference Elicitation idea
  - More efficient algorithm for cell mean models
  - Novel algorithm for linear regression models
- Results: Exploratory Analysis of the CATIE Antipsychotic trial
- Discussion and Future Work:
  - Experimental evaluation using Mechanical Turk
  - Other extensions

Motivation:
Symptoms and Side-Effects in Schizophrenia

Motivation: Symptoms and Side-Effects in Schizophrenia

- Many treatments available for treating schizophrenia (dozens)
- Evidence-based medicine would look at outcomes, recommend a **sequence** of treatments: schizophrenia is chronic
- *At least* two important objectives:
  - Maximize symptom reduction, minimize weight gain

Motivation:
Symptoms and Side-Effects in Schizophrenia

- Many treatments available for treating schizophrenia (dozens)
- Evidence-based medicine would look at outcomes, recommend a **sequence** of treatments: schizophrenia is chronic
- *At least* two important objectives:
    - Maximize symptom reduction, minimize weight gain
- Treatments that provide the best symptom reduction induce the worst weight gain, and vice-versa

Motivation:
Symptoms and Side-Effects in Schizophrenia

- Many treatments available for treating schizophrenia (dozens)
- Evidence-based medicine would look at outcomes, recommend a **sequence** of treatments: schizophrenia is chronic
- *At least* two important objectives:
    - Maximize symptom reduction, minimize weight gain
- Treatments that provide the best symptom reduction induce the worst weight gain, and vice-versa
- Different doctors and patients have very different preferences about relative importance of outcomes

Motivation:
Symptoms and Side-Effects in Schizophrenia

- Many treatments available for treating schizophrenia (dozens)
- Evidence-based medicine would look at outcomes, recommend a **sequence** of treatments: schizophrenia is chronic
- *At least* two important objectives:
    - Maximize symptom reduction, minimize weight gain
- Treatments that provide the best symptom reduction induce the worst weight gain, and vice-versa
- Different doctors and patients have very different preferences about relative importance of outcomes
    - **How can we recommend a sequence of treatments that accommodates these preferences?**

Recommending Sequences of Treatment: Dynamic Treatment Regimes

Recommending Sequences of Treatment: Dynamic Treatment Regimes

- A dynamic treatment regime is a **sequence of rules for choosing a sequence of treatments for a patient**

Recommending Sequences of Treatment: Dynamic Treatment Regimes

- A dynamic treatment regime is a **sequence of rules for choosing a sequence of treatments for a patient**
  - Input: patient's current "**state**," "features," or "covariates," e.g. previous treatments, response to those previous treatments, genetic markers, family history, ...

Recommending Sequences of Treatment: Dynamic Treatment Regimes

- A dynamic treatment regime is a **sequence of rules for choosing a sequence of treatments for a patient**
  - Input: patient's current "**state**," "features," or "covariates," e.g. previous treatments, response to those previous treatments, genetic markers, family history, ...
  - Output: treatment choice or "**action**" for that point in time

Recommending Sequences of Treatment: Dynamic Treatment Regimes

- A dynamic treatment regime is a **sequence of rules for choosing a sequence of treatments for a patient**
  - Input: patient's current "**state**," "features," or "covariates," e.g. previous treatments, response to those previous treatments, genetic markers, family history, ...
  - Output: treatment choice or "**action**" for that point in time
- Each rule in the sequence uses the most up-to-date state information

Recommending Sequences of Treatment: Dynamic Treatment Regimes

- A dynamic treatment regime is a **sequence of rules for choosing a sequence of treatments for a patient**
  - Input: patient's current "**state**," "features," or "covariates," e.g. previous treatments, response to those previous treatments, genetic markers, family history, ...
  - Output: treatment choice or "**action**" for that point in time
- Each rule in the sequence uses the most up-to-date state information
- In an optimal DTR, actions are chosen to maximize the patient's total expected outcome or "**reward**."

Learning a Dynamic Treatment Regime From Data

Learning a Dynamic Treatment Regime From Data

- (S_1,A_1,S_2,A_2,R) for each individual
    - S_j - “State” - Patient covariates (previous txts, response,...)
    - A_j - “Action” - Treatment offered to the patient
    - R - “Reward” - Clinical outcome

Learning a Dynamic Treatment Regime From Data

- (S_1,A_1,S_2,A_2,R) for each individual
  - S_j - “State” - Patient covariates (previous txts, response,...)
  - A_j - “Action” - Treatment offered to the patient
  - R - “Reward” - Clinical outcome
- Actions A_j have known randomization probability

Learning a Dynamic Treatment Regime From Data

- (S_1, A_1, S_2, A_2, R) for each individual
  - S_j - “State” - Patient covariates (previous txts, response,...)
  - A_j - “Action” - Treatment offered to the patient
  - R - “Reward” - Clinical outcome
- Actions A_j have known randomization probability
- The Proposed DTR,

$$\pi = \{\pi_1 : S_1 \to A_1,\ \pi_2 : S_2 \to A_2\},$$

should have high **value** $V^\pi = E^\pi[R]$. (π stands for “Policy”)

Learning a Dynamic Treatment Regime From Data

Learning a Dynamic Treatment Regime From Data

- Q-Learning
  - Generalization of regression to multiple stages
  - Backwards induction (dynamic programming)
  - Conditional expectations approximated using regression

Learning a Dynamic Treatment Regime From Data

- Q-Learning
  - Generalization of regression to multiple stages
  - Backwards induction (dynamic programming)
  - Conditional expectations approximated using regression
- Development
  - Computing Science: Watkins (1989), Sutton & Barto (1998), Ernst (2005),...
  - Operations Research: Bertsekas & Tsitsiklis (1996),...
  - Statistics: Murphy (2003), Zhao et al.(2009), Robins (2004),..

Learning a Dynamic Treatment Regime From Data

- Q-Learning
  - Generalization of regression to multiple stages
  - Backwards induction (dynamic programming)
  - Conditional expectations approximated using regression
- Development
  - Computing Science: Watkins (1989), Sutton & Barto (1998), Ernst (2005),...
  - Operations Research: Bertsekas & Tsitsiklis (1996),...
  - Statistics: Murphy (2003), Zhao et al.(2009), Robins (2004),..
- One of many methods that are part of “Reinforcement Learning”

Q-Learning for Two Stages

Q-Learning for Two Stages

- For two stages, the optimal value $V^* = \max_\pi V^\pi$ can be written as
  - $V^* = \mathrm{E}[\ \max_{a_1} \mathrm{E}[\ \max_{a_2} \mathrm{E}[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$

Q-Learning for Two Stages

- For two stages, the optimal value $V^* = \max_\pi V^\pi$ can be written as
  - $V^* = E[\ \max_{a1} E[\ \max_{a2} E[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- Define Stage 2 Q-function $Q_2(S_2,A_2) = E[\ R \mid S_2, A_2\ ]$
  - $V^* = E[\ \max_{a1} E[\ \max_{a2} Q_2(S_2,a_2) \mid S_1, A_1\ ]]$

Q-Learning for Two Stages

- For two stages, the optimal value $V^* = \max_\pi V^\pi$ can be written as
    - $V^* = E[\ \max_{a_1} E[\ \max_{a_2} E[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- Define Stage 2 Q-function $Q_2(S_2,A_2) = E[\ R \mid S_2, A_2\ ]$
    - $V^* = E[\ \max_{a_1} E[\ \max_{a_2} Q_2(S_2,a_2) \mid S_1, A_1\ ]]$
- Define Stage 1 Q-function $Q_1(S_1,A_1) = E[\ \max_{a_2} Q_2(S_2,a_2) \mid S_2, A_2\ ]$
    - $V^* = E[\ \max_{a_1} Q_1(S_1,a_1)\ ]$

Q-Learning for Two Stages

- For two stages, the optimal value $V^* = \max_\pi V^\pi$ can be written as
  - $V^* = E[\ \max_{a_1} E[\ \max_{a_2} E[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- Define Stage 2 Q-function $Q_2(S_2,A_2) = E[\ R \mid S_2, A_2\ ]$
  - $V^* = E[\ \max_{a_1} E[\ \max_{a_2} Q_2(S_2,a_2) \mid S_1, A_1\ ]]$
- Define Stage 1 Q-function $Q_1(S_1,A_1) = E[\ \max_{a_2} Q_2(S_2,a_2) \mid S_2, A_2\ ]$
  - $V^* = E[\ \max_{a_1} Q_1(S_1,a_1)\ ]$
- Plan: Estimate $Q_2(S_2,a_2)$ and $Q_1(S_1,a_1)$, use argmax to estimate π^*

Q-Learning at Stage 2

Q-Learning at Stage 2

- $V^* = \mathrm{E}[\ \max_{a_1} \mathrm{E}[\ \max_{a_2} \mathrm{E}[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$

Q-Learning at Stage 2

- $V^* = \mathrm{E}[\ \max_{a_1} \mathrm{E}[\ \max_{a_2} \mathrm{E}[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- S_{21}, S_{22} are features of S_2, $A_2 \in \{-1,1\}$

Q-Learning at Stage 2

- $V^* = \text{E}[\ \max_{a_1} \text{E}[\ \max_{a_2} \text{E}[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- S_{21}, S_{22} are features of S_2, $A_2 \in \{-1,1\}$
- Regress R on S_{21}, S_{22}, A_2, giving

$$\hat{Q}_2(S_2, A_2) = \hat{\beta}_{21}^{\mathsf{T}} S_{21} + \hat{\beta}_{22}^{\mathsf{T}} S_{22} A_2$$

$$\hat{\pi}_2(S_2) = \operatorname{argmax}_{a_2} \hat{Q}_2(S_2, a_2)$$

Q-Learning at Stage 2

- $V^* = \text{E}[\ \max_{a_1} \text{E}[\ \max_{a_2} \text{E}[\ R \mid S_2, A_2 = a_2\ ] \mid S_1, A_1 = a_1\ ]]$
- S_{21}, S_{22} are features of S_2, $A_2 \in \{-1,1\}$

- Regress R on S_{21}, S_{22}, A_2, giving

$$\hat{Q}_2(S_2, A_2) = \hat{\beta}_{21}^{\mathsf{T}} S_{21} + \hat{\beta}_{22}^{\mathsf{T}} S_{22} A_2$$

$$\hat{\pi}_2(S_2) = \text{argmax}_{a_2}\ \hat{Q}_2(S_2, a_2)$$

- Notice $\hat{V}_2(S_2) = \max_{a_2} \hat{Q}_2(S_2, a_2)$ is an estimator of $\max_{a_2} Q_2(S_2, a_2)$

Q-Learning at Stage 1

Q-Learning at Stage 1

- $V^* = E[\ \max_{a_1} E[\ \max_{a_2} Q_2(S_2, a_2) \mid S_1, A_1\ ]]$
  - Plug in $\hat{V}_2$ for $\max_{a_2} Q_2$

Q-Learning at Stage 1

- $V^* = E[\ \max_{a_1} E[\ \max_{a_2} Q_2(S_2,a_2)\ |\ S_1, A_1\ ]]$
    - Plug in $\hat{V}_2$ for $\max_{a_2} Q_2$

- S_{11}, S_{12} are features of S_1, $A_1 \in \{-1,1\}$
- Regress $\hat{V}_2(S_2)$ on S_{11}, S_{12}, A_1, giving

$$\hat{Q}_1(S_1, A_1) = \hat{\beta}_{11}^{\mathsf{T}} S_{11} + \hat{\beta}_{12}^{\mathsf{T}} S_{12} A_1$$

$$\hat{\pi}_1(S_1) = \operatorname{argmax}_{a_1} \hat{Q}_1(S_1, a_1)$$

Q-Learning at Stage 1

- $V^* = \mathrm{E}[\ \max_{a1}\ \mathrm{E}[\ \max_{a2}\ Q_2(S_2, a_2)\ |\ S_1, A_1\ ]]$
  - Plug in $\hat{V}_2$ for $\max_{a2}\ Q_2$

- S_{11}, S_{12} are features of S_1, $A_1 \in \{-1,1\}$
- Regress $\hat{V}_2(S_2)$ on S_{11}, S_{12}, A_1, giving

$$\hat{Q}_1(S_1, A_1) = \hat{\beta}_{11}^{\mathsf{T}} S_{11} + \hat{\beta}_{12}^{\mathsf{T}} S_{12} A_1$$

$$\hat{\pi}_1(S_1) = \mathrm{argmax}_{a_1}\ \hat{Q}_1(S_1, a_1)$$

- $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$ is our estimate of the optimal DTR

“Definition” of R

“Definition” of R

- Idealized picture presented earlier:
  - (S_1,A_1,S_2,A_2,R) for each individual, find π to maximize $V^\pi = E^\pi[R]$.

“Definition” of *R*

- Idealized picture presented earlier:
  - (S_1,A_1,S_2,A_2,R) for each individual, find π to maximize $V^\pi = \mathrm{E}^\pi[R]$.
- More truthful picture:
  - (O_1,A_1,O_2,A_2,O_3) for each individual
  - Must **define** $S_t = S_t(O_{1:t},A_{1:t-1})$, $R = R(O_1,O_2,O_3)$

“Definition” of R

- Idealized picture presented earlier:
    - (S_1,A_1,S_2,A_2,R) for each individual, find π to maximize $V^\pi = E^\pi[R]$.
- More truthful picture:
    - (O_1,A_1,O_2,A_2,O_3) for each individual
    - Must **define** $S_t = S_t(O_{1:t},A_{1:t-1})$, $R = R(O_1,O_2,O_3)$
- $S_t(O_{1:t},A_{1:t-1})$ may be chosen by expert knowledge, or by model selection techniques (e.g. Gunter (2009), Qian (2010))

“Definition” of R

- Idealized picture presented earlier:
  - (S_1, A_1, S_2, A_2, R) for each individual, find π to maximize $V^\pi = E^\pi[R]$.
- More truthful picture:
  - $(O_1, A_1, O_2, A_2, O_3)$ for each individual
  - Must **define** $S_t = S_t(O_{1:t}, A_{1:t-1})$, $R = R(O_1, O_2, O_3)$
- $S_t(O_{1:t}, A_{1:t-1})$ may be chosen by expert knowledge, or by model selection techniques (e.g. Gunter (2009), Qian (2010))
- $R(O_1, O_2, O_3)$, in many cases, is not obvious

“Definition” of R

- Idealized picture presented earlier:
  - (S_1, A_1, S_2, A_2, R) for each individual, find π to maximize $V^\pi = E^\pi[R]$.
- More truthful picture:
  - $(O_1, A_1, O_2, A_2, O_3)$ for each individual
  - Must **define** $S_t = S_t(O_{1:t}, A_{1:t-1})$, $R = R(O_1, O_2, O_3)$
- $S_t(O_{1:t}, A_{1:t-1})$ may be chosen by expert knowledge, or by model selection techniques (e.g. Gunter (2009), Qian (2010))
- $R(O_1, O_2, O_3)$, in many cases, is not obvious
- The “correct” R may depend on individual preferences
  - May not correspond to a single “measurement”

From Dynamic Treatment Regime to Clinical Decision Support

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system, doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system,
  doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system,
  doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:
  - Estimation error in $\hat{Q}_t(S_t, A_t)$

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system,
  doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:
  - Estimation error in $\hat{Q}_t(S_t, A_t)$
    - Recommend a **set** of actions depending on confidence, i.e. eliminate bad actions

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system, doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:
  - Estimation error in $\hat{Q}_t(S_t, A_t)$
    - Recommend a **set** of actions depending on confidence, i.e. eliminate bad actions
  - **Definition of *R* does not reflect desired outcome**

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system, doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:
  - Estimation error in $\hat{Q}_t(S_t, A_t)$
    - Recommend a **set** of actions depending on confidence, i.e. eliminate bad actions
  - Definition of *R* does not reflect desired outcome
    - Consider a set of *R* during Q-learning

From Dynamic Treatment Regime to Clinical Decision Support

- Q-learning gives us $\hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- As part of a Clinical Decision Support system, doctor provides a patient's s_t, and $\hat{\pi}_t$ recommends an action
- Some causes for concern:
  - Estimation error in $\hat{Q}_t(S_t, A_t)$
    - Recommend a **set** of actions depending on confidence, i.e. eliminate bad actions
  - Definition of *R* does not reflect desired outcome
    - Consider a set of *R* during Q-learning
    - "Inverse Preference Elicitation"

Considering a Set of Reward Definitions

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
    - $R^{(0)}$ reflects symptoms
    - $R^{(1)}$ reflects weight control

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
  - $R^{(0)}$ reflects symptoms
  - $R^{(1)}$ reflects weight control
- Transform both to percentiles w.r.t. baseline population

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
    - $R^{(0)}$ reflects symptoms
    - $R^{(1)}$ reflects weight control
- Transform both to percentiles w.r.t. baseline population
- Convex set of reward definitions:

$$\{R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)} \mid 0 \leq \delta \leq 1\}$$

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
    - $R^{(0)}$ reflects symptoms
    - $R^{(1)}$ reflects weight control
- Transform both to percentiles w.r.t. baseline population
- Convex set of reward definitions:

$$\{R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)} \mid 0 \leq \delta \leq 1\}$$

- δ identifies a reward definition, gives $\{\hat{Q}_1, \hat{Q}_2\}, \hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
  - $R^{(0)}$ reflects symptoms
  - $R^{(1)}$ reflects weight control
- Transform both to percentiles w.r.t. baseline population
- Convex set of reward definitions:

$$\{R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)} \mid 0 \leq \delta \leq 1\}$$

- δ identifies a reward definition, gives $\{\hat{Q}_1, \hat{Q}_2\}, \hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- Depending on δ, $\hat{\pi}$ "cares more" about optimizing $R^{(0)}$ or $R^{(1)}$

Considering a Set of Reward Definitions

- Identify a pair of important rewards (Coded higher is better)
    - $R^{(0)}$ reflects symptoms
    - $R^{(1)}$ reflects weight control
- Transform both to percentiles w.r.t. baseline population
- Convex set of reward definitions:

$$\{R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)} \mid 0 \leq \delta \leq 1\}$$

- δ identifies a reward definition, gives $\{\hat{Q}_1, \hat{Q}_2\}, \hat{\pi} = \{\hat{\pi}_1, \hat{\pi}_2\}$
- Depending on δ, $\hat{\pi}$ “cares more” about optimizing $R^{(0)}$ or $R^{(1)}$
- For $\delta = 0.5$, $\hat{\pi}$ “cares” equally about both

Inverse Preference Elicitation

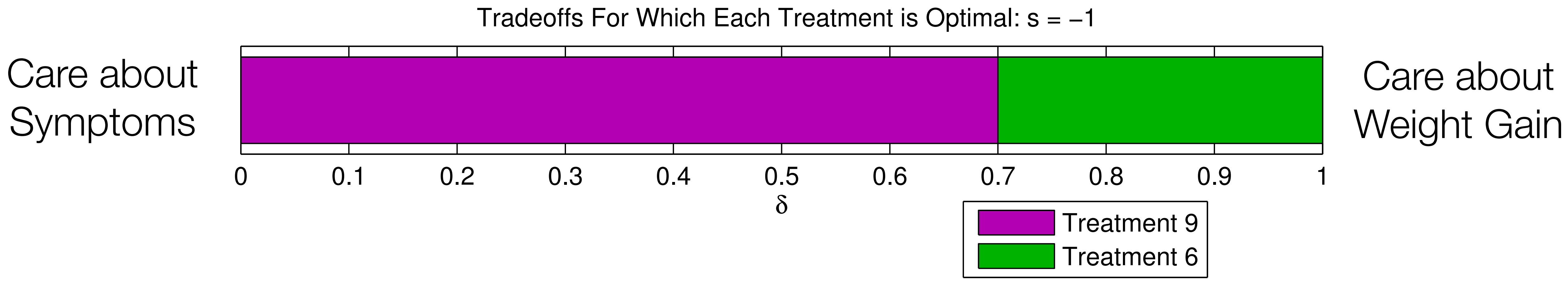
Inverse Preference Elicitation

- One approach: “Preference Elicitation”
    - Try to determine the decision-maker’s true value of δ via time tradeoff, standard gamble, visual analog scales,...
    - Use Q-learning, suggest an action based on state and elicited δ
    - There is much debate about how well this works
    - Says **nothing** about pros and cons of available treatments

Inverse Preference Elicitation

- One approach: “Preference Elicitation”
    - Try to determine the decision-maker’s true value of δ via time tradeoff, standard gamble, visual analog scales,...
    - Use Q-learning, suggest an action based on state and elicited δ
    - There is much debate about how well this works
    - Says **nothing** about pros and cons of available treatments
- Our approach: “Inverse Preference Elicitation”
    - Given state, report, for each action, the range of δ for which that action is optimal
    - Patient/clinician selects an action using this information
    - **“This is what your choice of action says about your preferences.”**

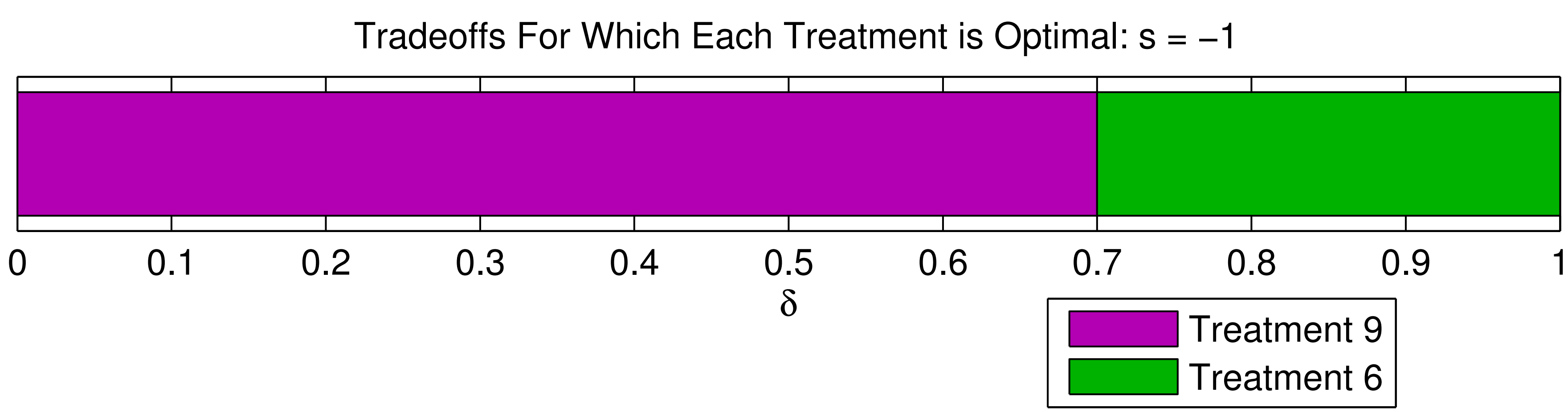
Example Output:
Inverse Preference Elicitation



Example Output:
Inverse Preference Elicitation

- Take $R(\delta) \equiv (1 - \delta)\cdot R^{(0)} + \delta\cdot R^{(1)}$
- Use Q-learning to find optimal actions given every possible δ, i.e. estimate $\hat{Q}_t(S_t, A_t, \delta), \hat{\pi}_t(S_t, \delta)$ **for all** $\delta \in [0, 1]$.
- Given state, report, for each action, the range of δ for which that action is optimal.

Care about Symptoms

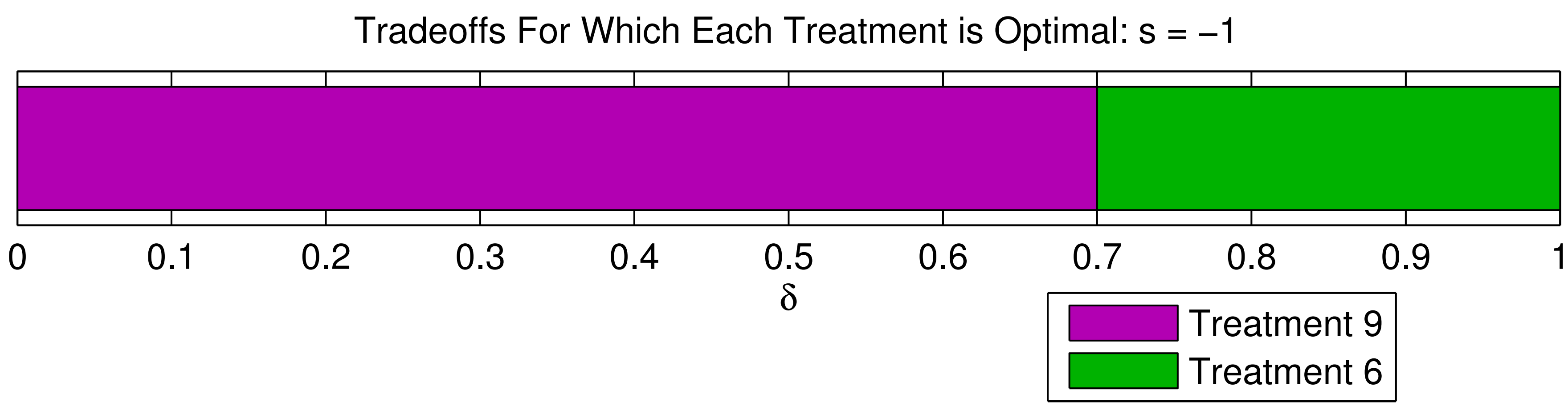


Care about Weight Gain

Example Output: Inverse Preference Elicitation

- Take $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- Use Q-learning to find optimal actions given every possible δ, i.e. estimate $\hat{Q}_t(S_t, A_t, \delta), \hat{\pi}_t(S_t, \delta)$ **for all** $\delta \in [0, 1]$.
- Given state, report, for each action, the range of δ for which that action is optimal.

Care about Symptoms



Care about Weight Gain

- Fortunately we don't need to explicitly solve for every δ.

Algorithm for Cell Mean Models: Preview

Algorithm for Cell Mean Models: Preview

- No smoothing of estimated Q: S_t are discrete
- $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- $\hat{Q}_2(s_2, a_2, \delta)$ is sample mean of $R(\delta)$ over tuples where $S_2 = s_2, A_2 = a_2$

Algorithm for Cell Mean Models: Preview

- No smoothing of estimated Q: S_t are discrete
- $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- $\hat{Q}_2(s_2, a_2, \delta)$ is sample mean of $R(\delta)$ over tuples where $S_2 = s_2, A_2 = a_2$
- Therefore $\hat{Q}_2(S_2, A_2, \delta)$ is linear in δ

Algorithm for Cell Mean Models: Preview

- No smoothing of estimated Q: S_t are discrete
- $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- $\hat{Q}_2(s_2, a_2, \delta)$ is sample mean of $R(\delta)$ over tuples where $S_2 = s_2, A_2 = a_2$
- Therefore $\hat{Q}_2(S_2, A_2, \delta)$ is linear in δ
    - $\hat{V}_2(S_2, \delta)$ is continuous and piecewise linear in δ by pointwise max

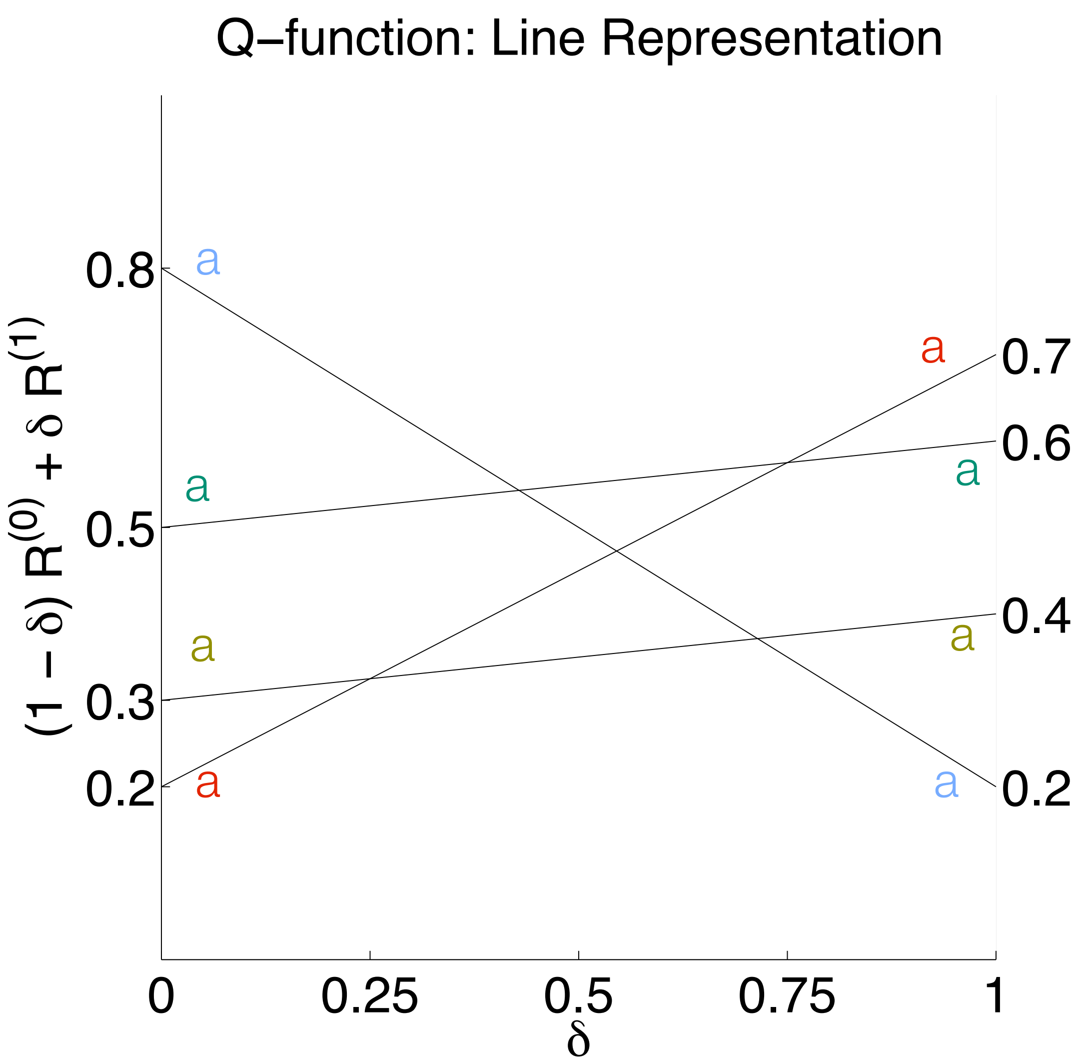
Algorithm for Cell Mean Models: Preview

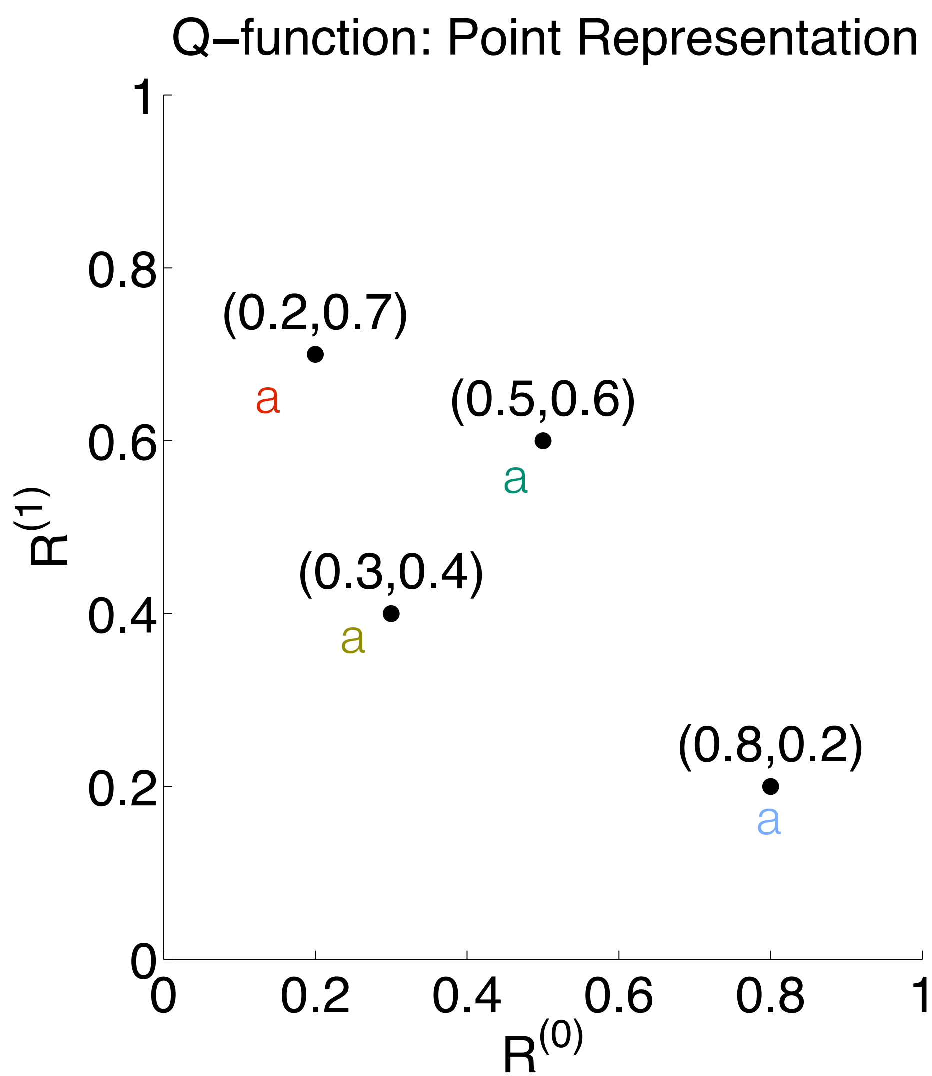
- No smoothing of estimated Q: S_t are discrete
- $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- $\hat{Q}_2(s_2, a_2, \delta)$ is sample mean of $R(\delta)$ over tuples where $S_2 = s_2, A_2 = a_2$
- Therefore $\hat{Q}_2(S_2, A_2, \delta)$ is linear in δ
  - $\hat{V}_2(S_2, \delta)$ is continuous and piecewise linear in δ by pointwise max
  - $\hat{\pi}_2(S_2, \delta)$ is piecewise constant in δ

Algorithm for Cell Mean Models: Preview

- No smoothing of estimated Q: S_t are discrete
- $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$
- $\hat{Q}_2(s_2, a_2, \delta)$ is sample mean of $R(\delta)$ over tuples where $S_2 = s_2, A_2 = a_2$
- Therefore $\hat{Q}_2(S_2, A_2, \delta)$ is linear in δ
  - $\hat{V}_2(S_2, \delta)$ is continuous and piecewise linear in δ by pointwise max
  - $\hat{\pi}_2(S_2, \delta)$ is piecewise constant in δ
- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in δ
  - Average of $\hat{V}_2(S_2, \delta)$ over tuples where $S_1 = s_1, A_1 = a_1$

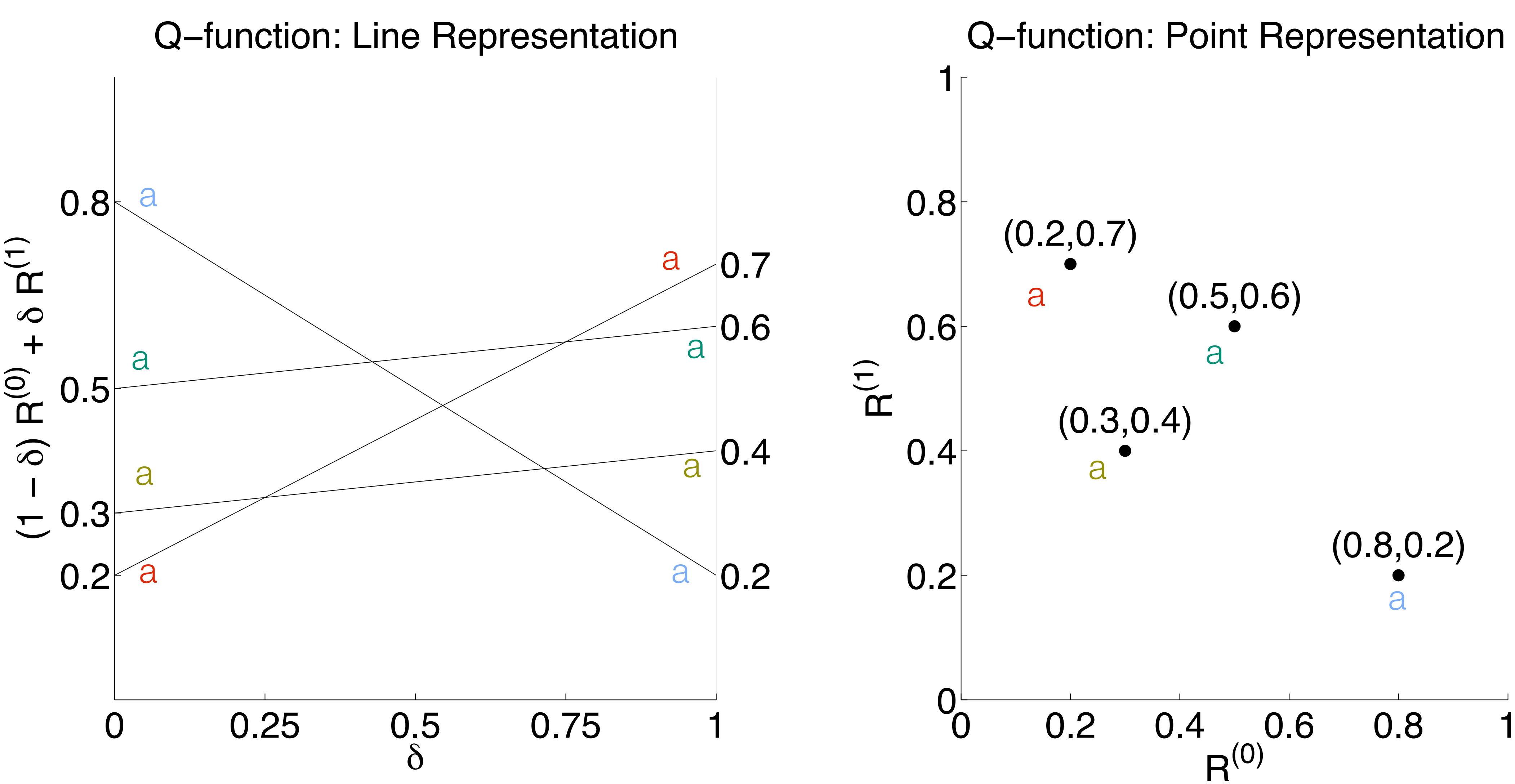
Pointwise Maximum Over Actions





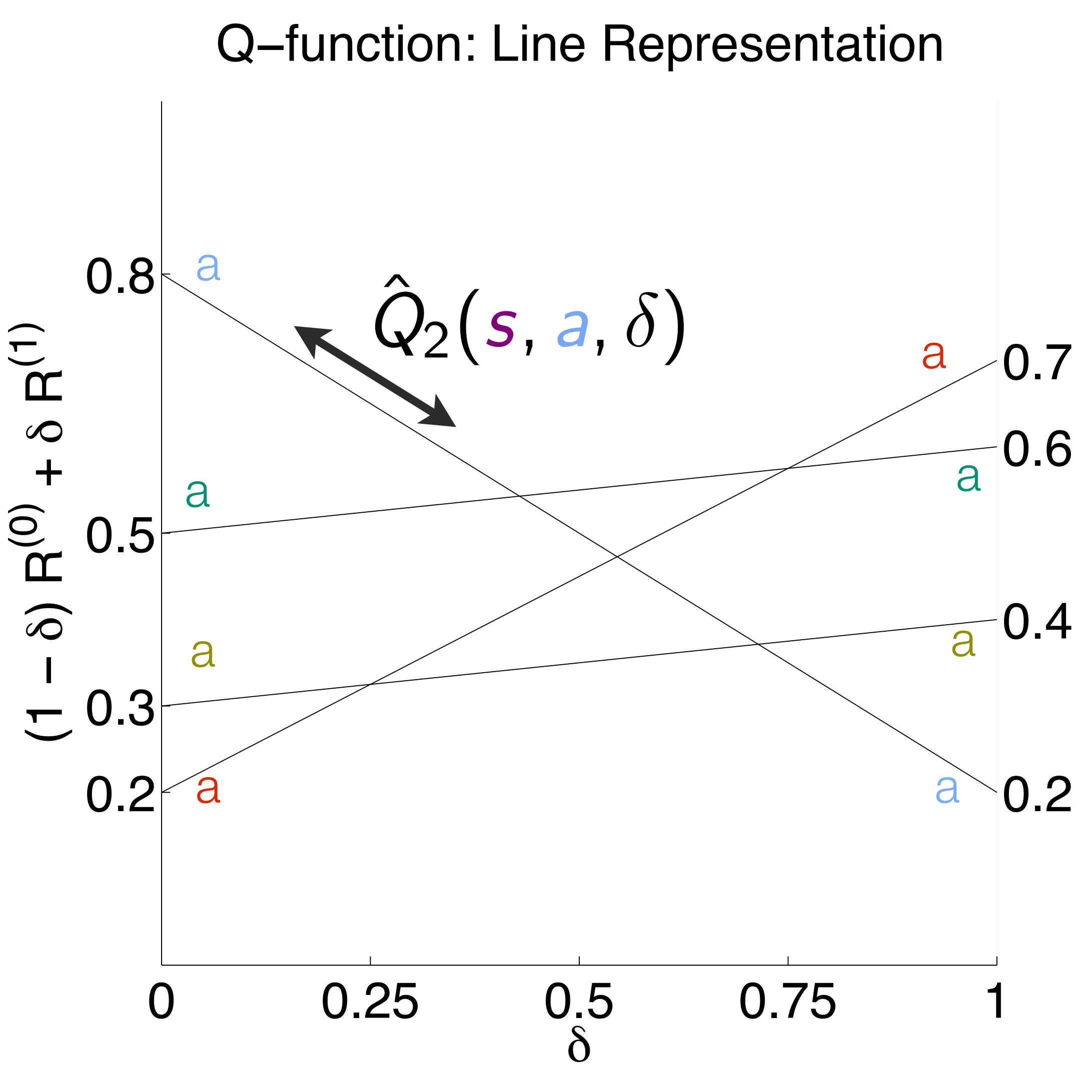
Pointwise Maximum Over Actions

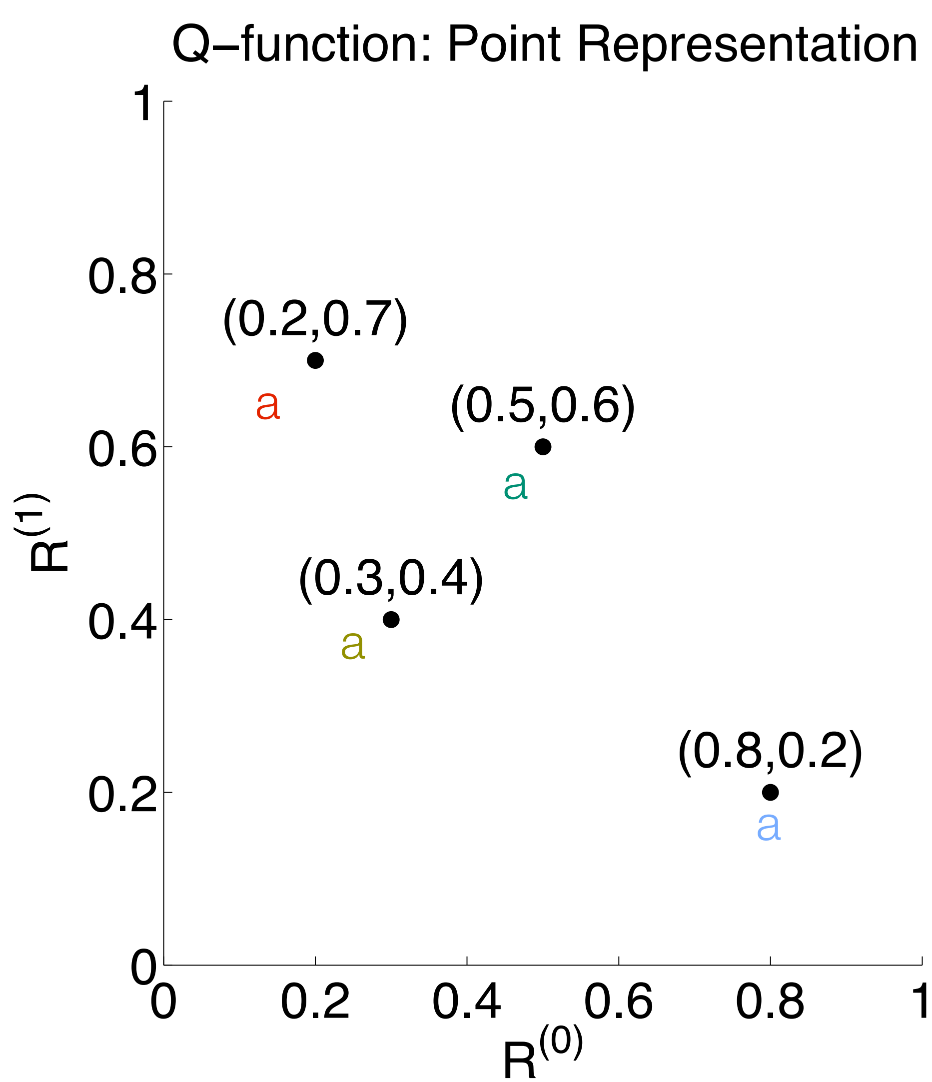
- $\hat{Q}_2(s_2, a_2, \delta)$ is linear in δ, represented by pair of sample means
- Two “representations”: Line representation, point representation
- Each “cell mean” is a function of δ



Pointwise Maximum Over Actions

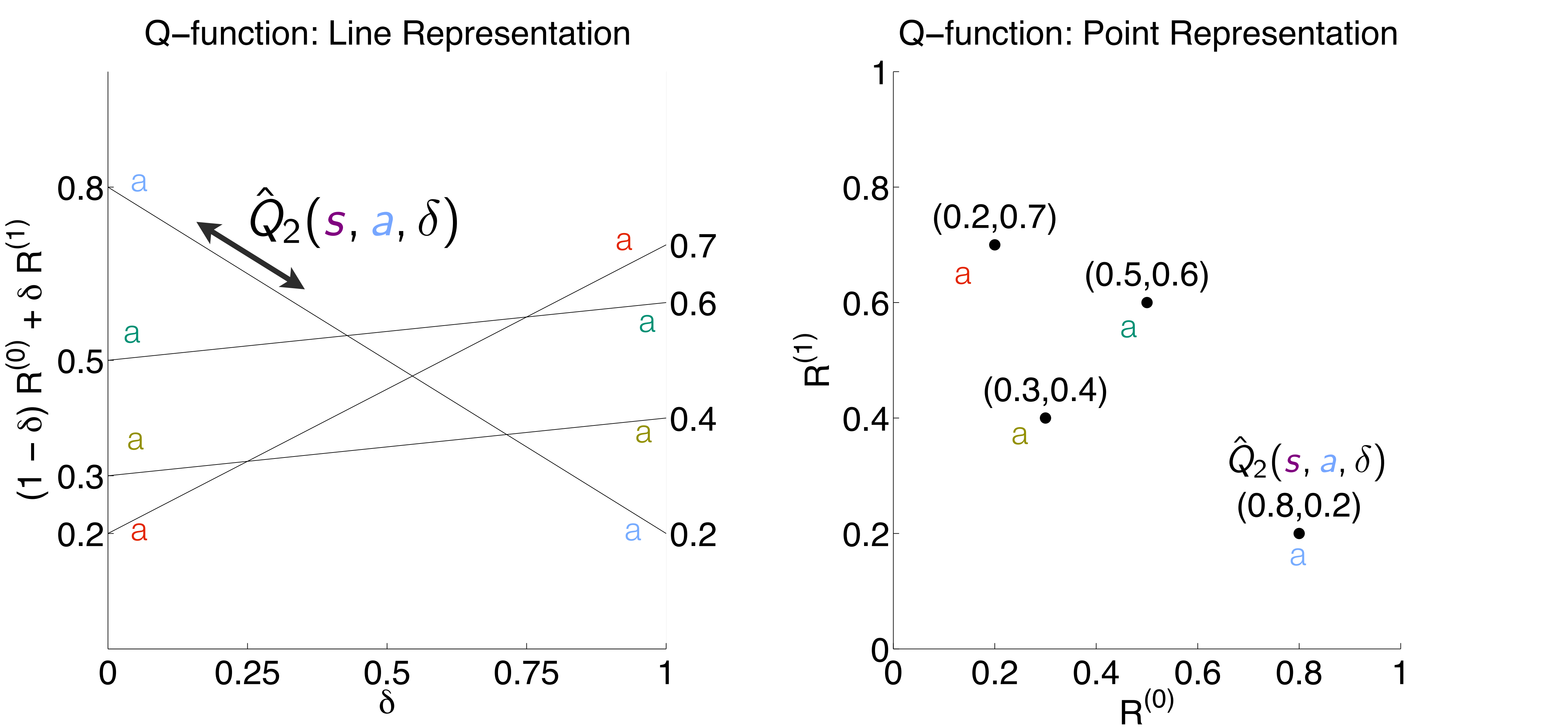
- $\hat{Q}_2(s_2, a_2, \delta)$ is linear in δ, represented by pair of sample means
- Two "representations": Line representation, point representation
- Each "cell mean" is a function of δ





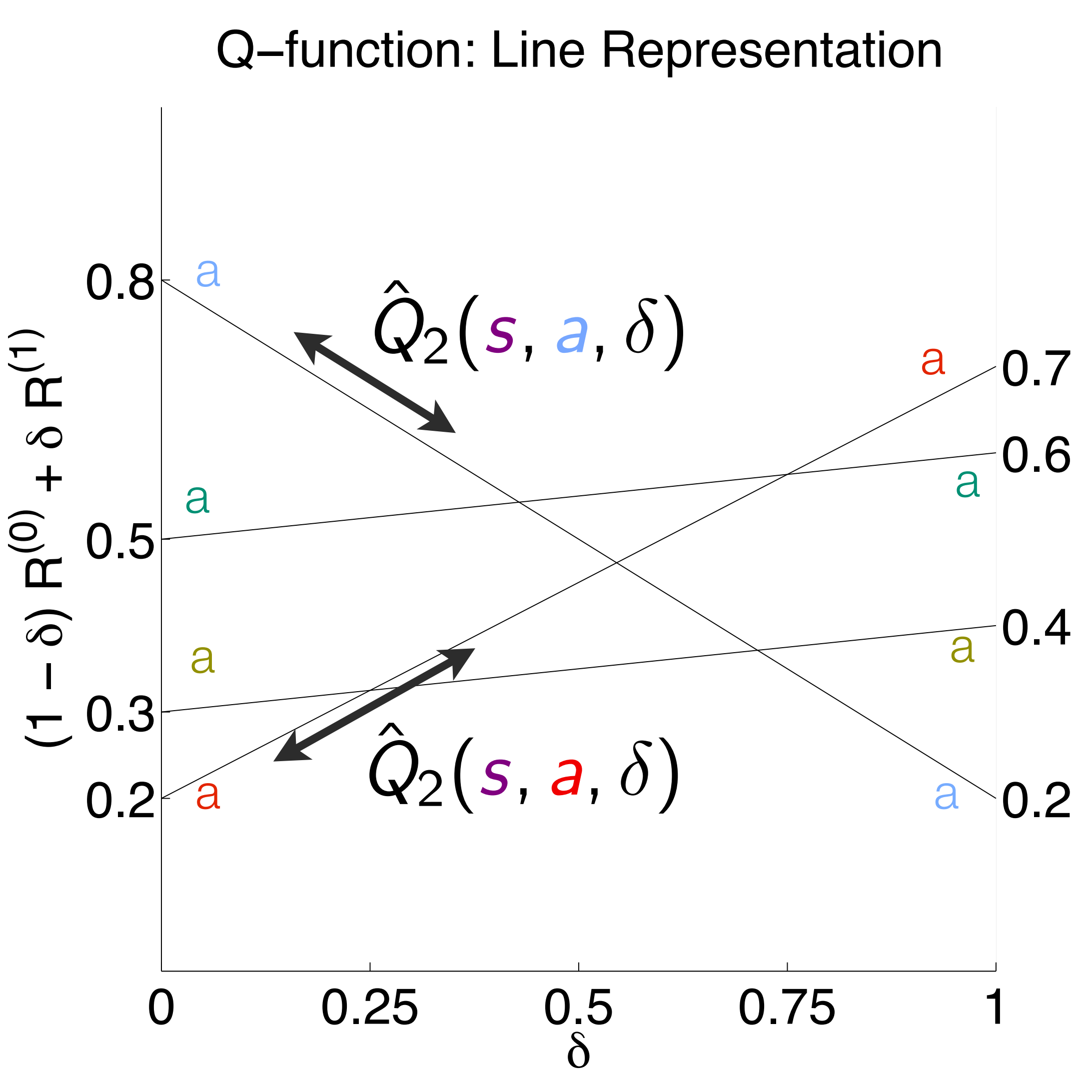
Pointwise Maximum Over Actions

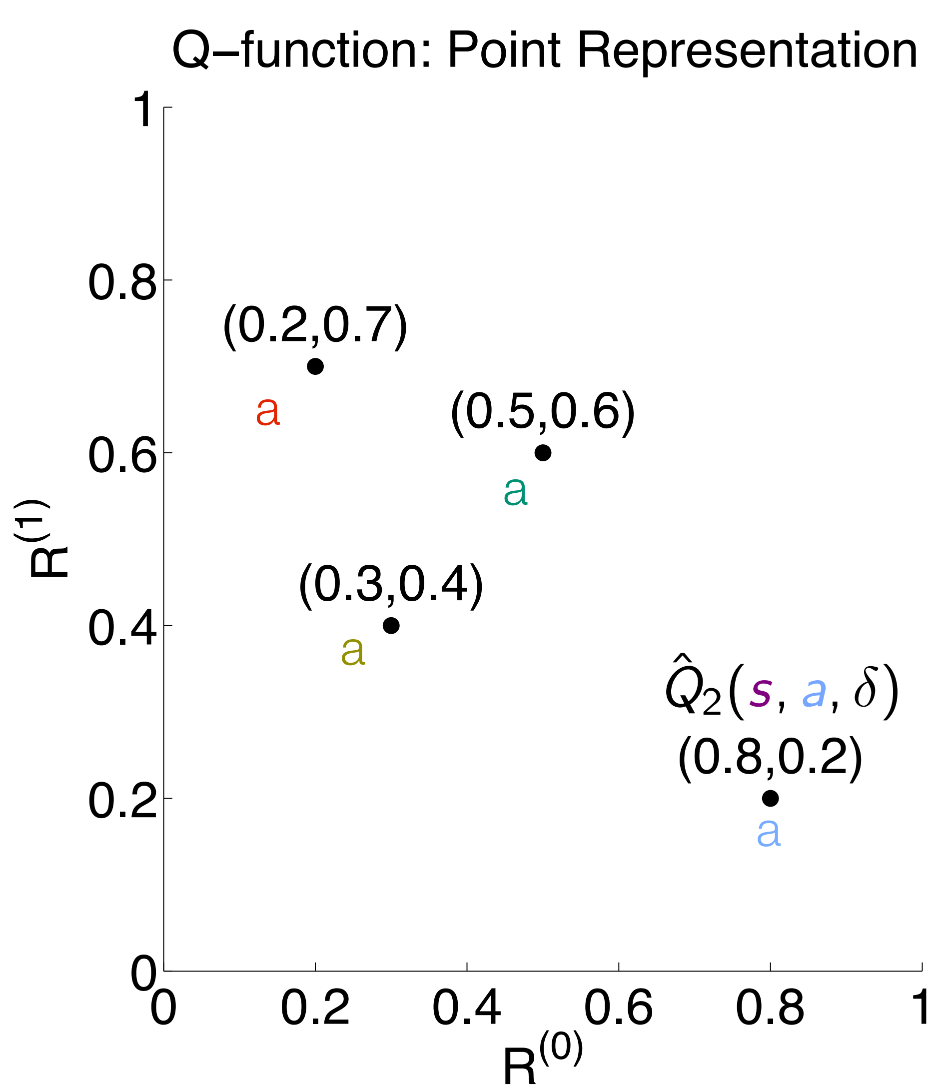
- $\hat{Q}_2(s_2, a_2, \delta)$ is linear in δ, represented by pair of sample means
- Two "representations": Line representation, point representation
- Each "cell mean" is a function of δ



Pointwise Maximum Over Actions

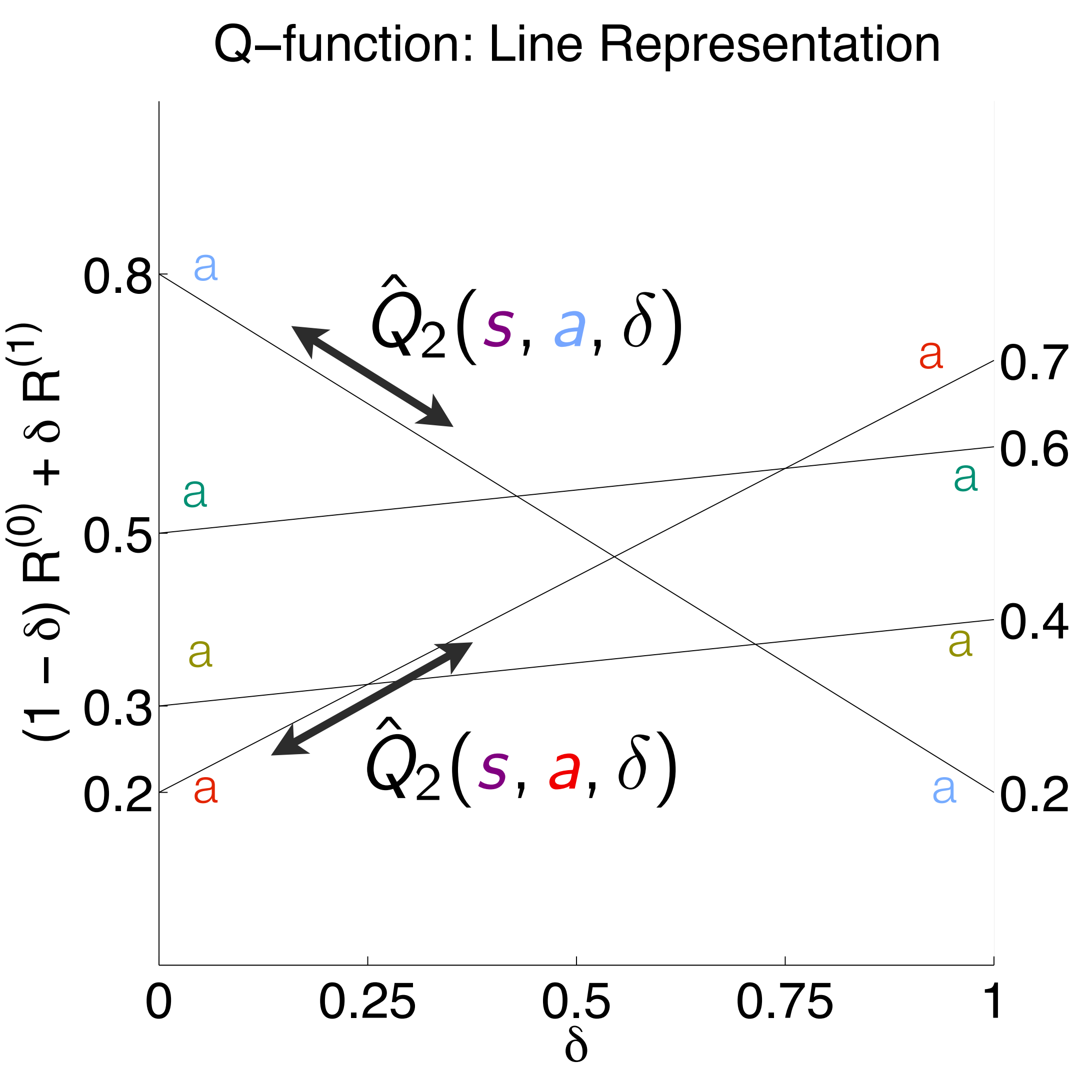
- $\hat{Q}_2(s_2, a_2, \delta)$ is linear in δ, represented by pair of sample means
- Two "representations": Line representation, point representation
- Each "cell mean" is a function of δ

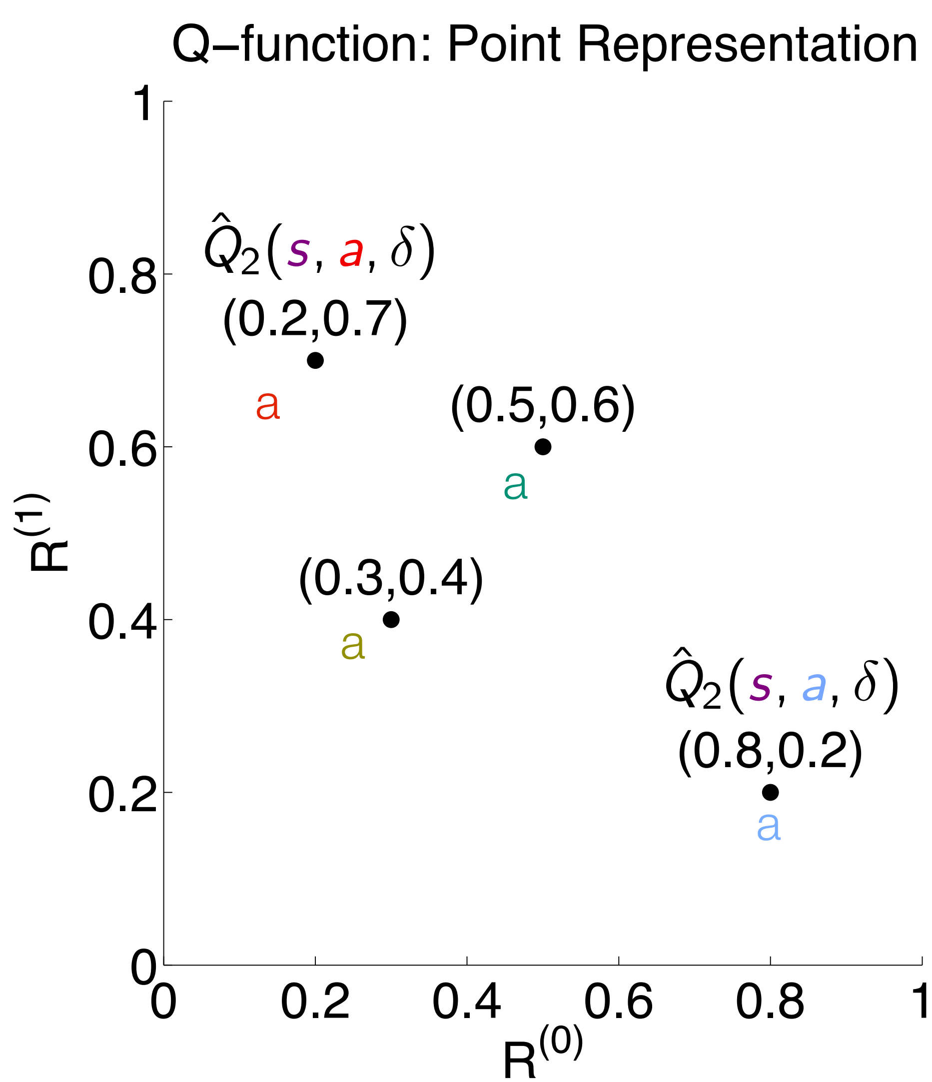




Pointwise Maximum Over Actions

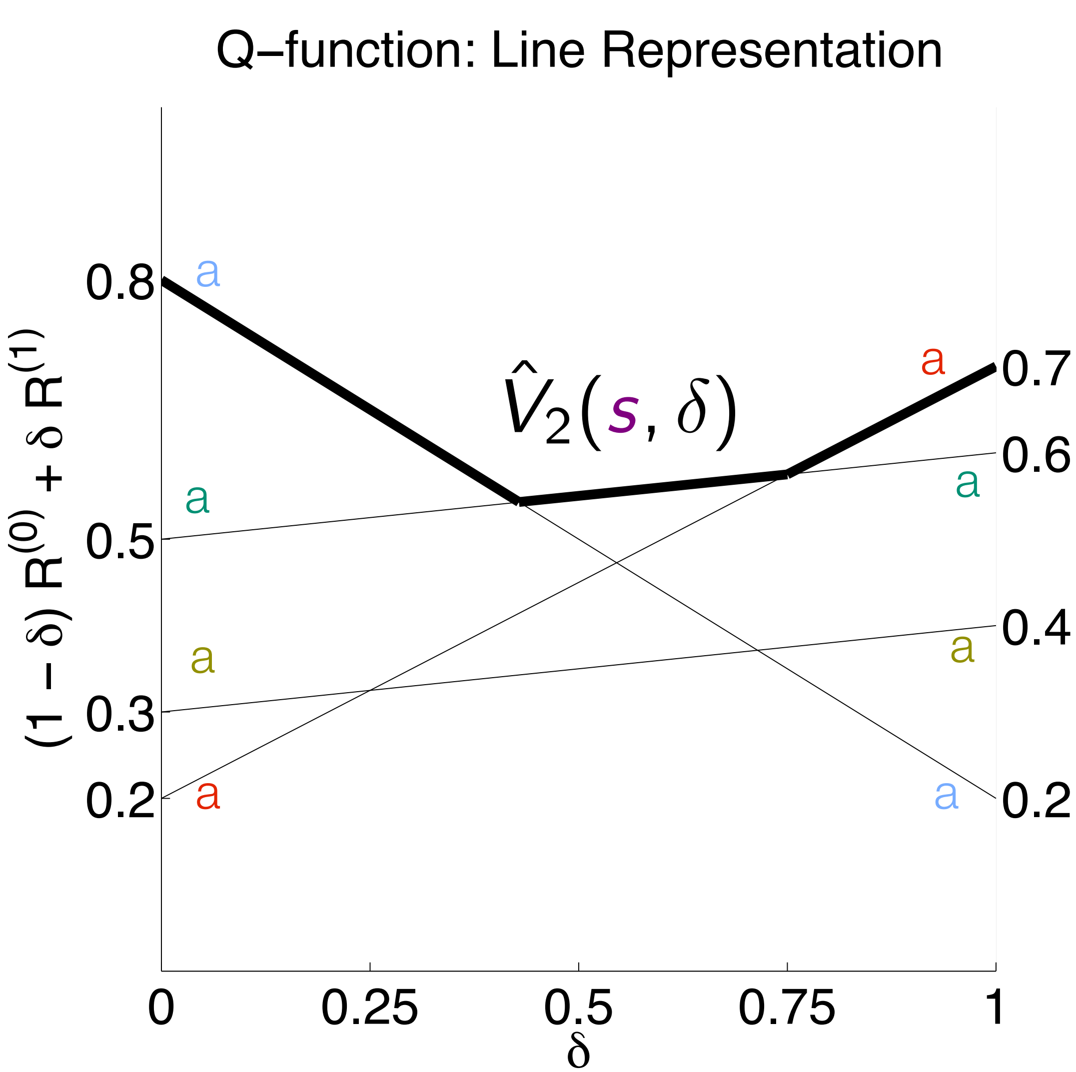
- $\hat{Q}_2(s_2, a_2, \delta)$ is linear in δ, represented by pair of sample means
- Two "representations": Line representation, point representation
- Each "cell mean" is a function of δ

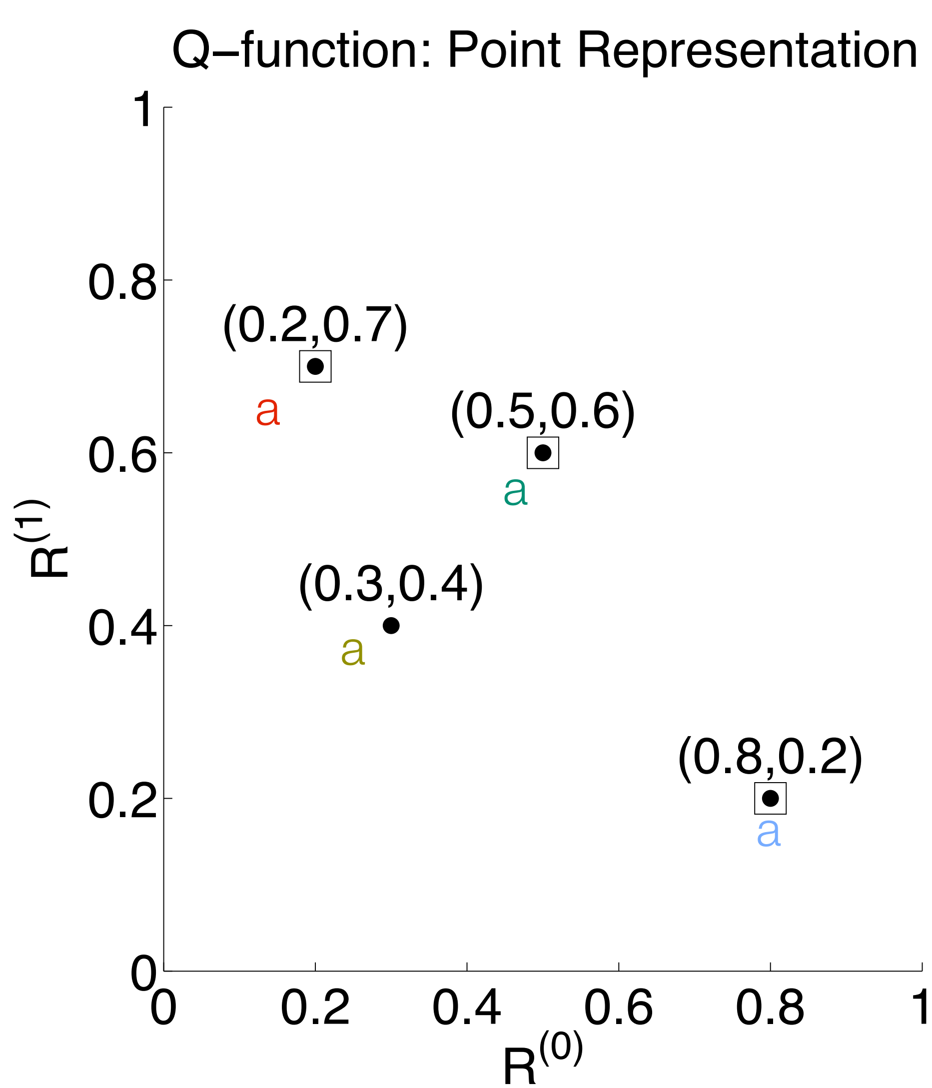




Pointwise Maximum Over Actions

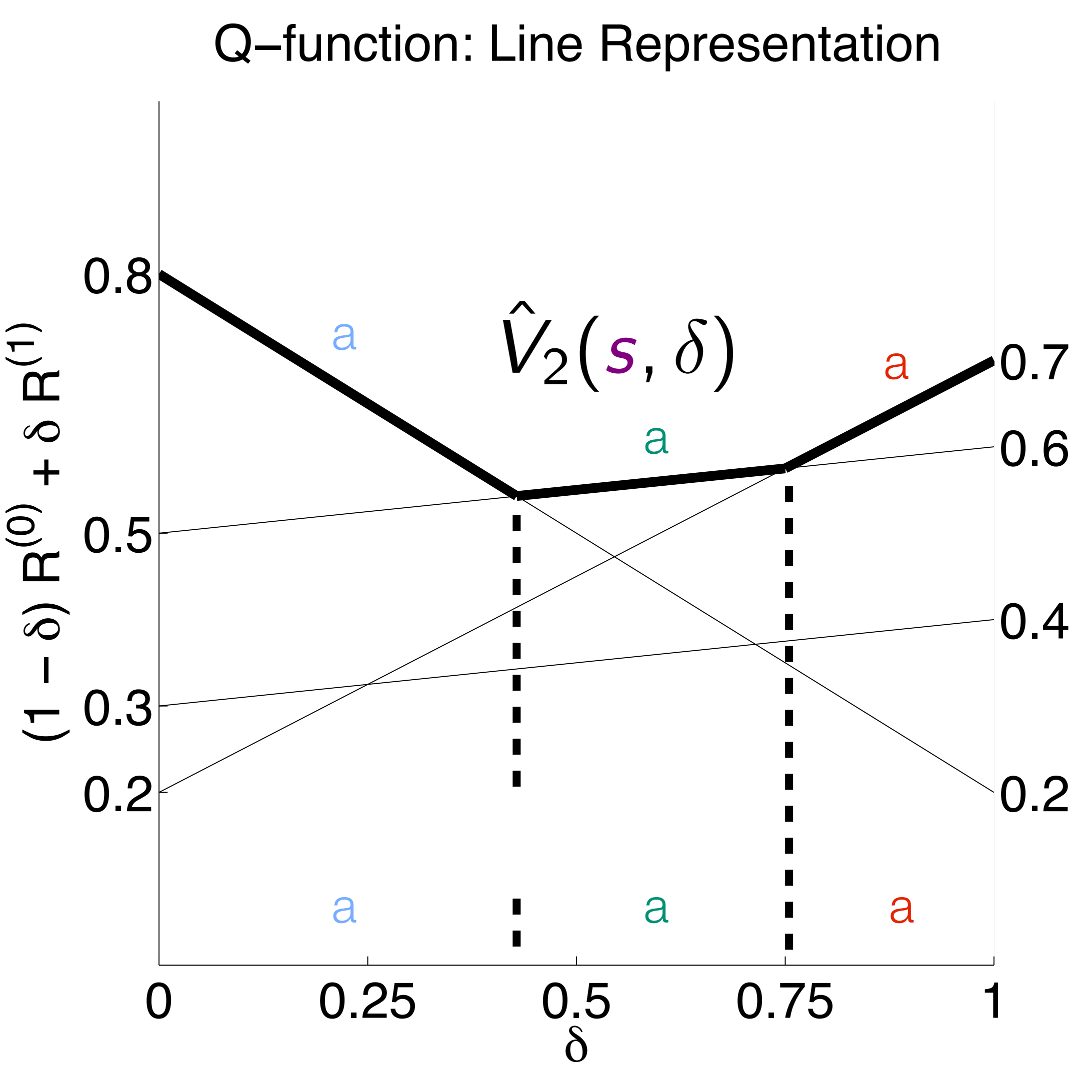
- $\hat{V}_2(s_2, \delta)$ is continuous and piecewise linear in δ
- Point-based representation has computational advantage
  - Knots identified by convex hull

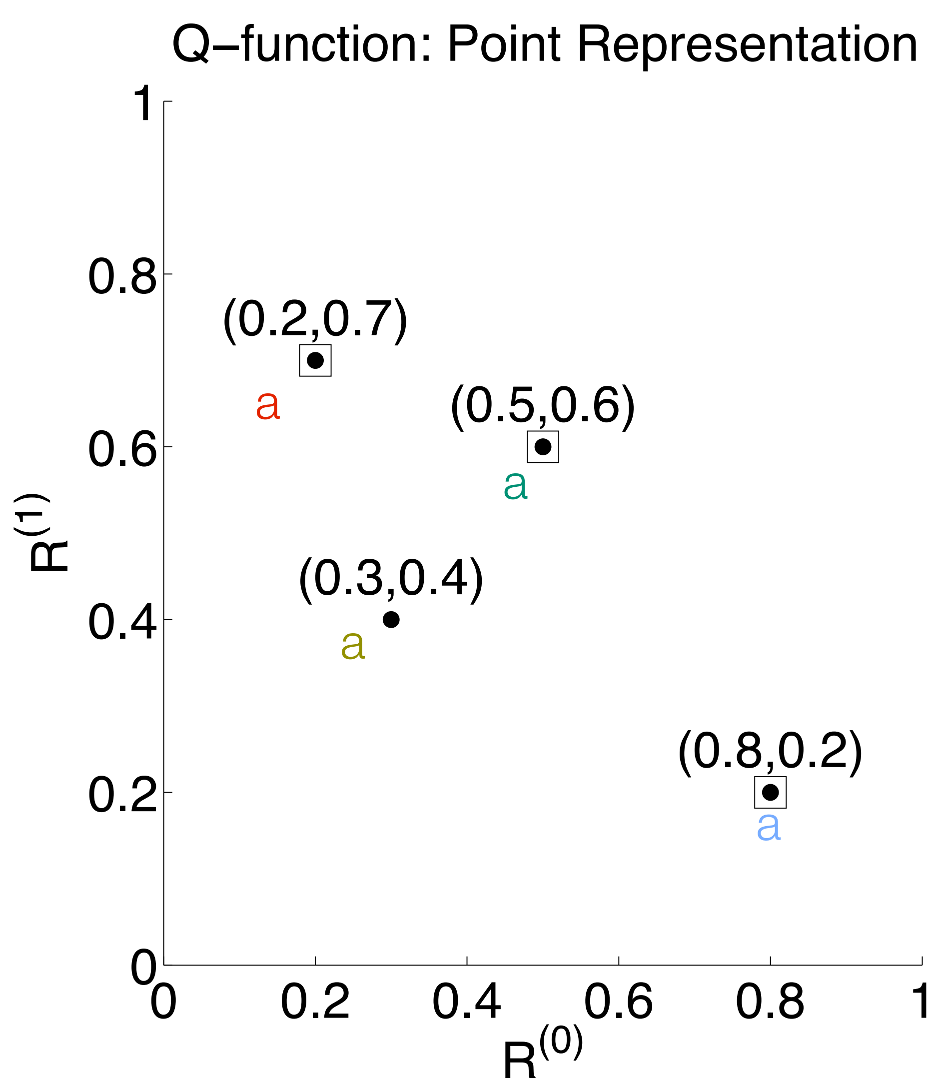




Pointwise Maximum Over Actions

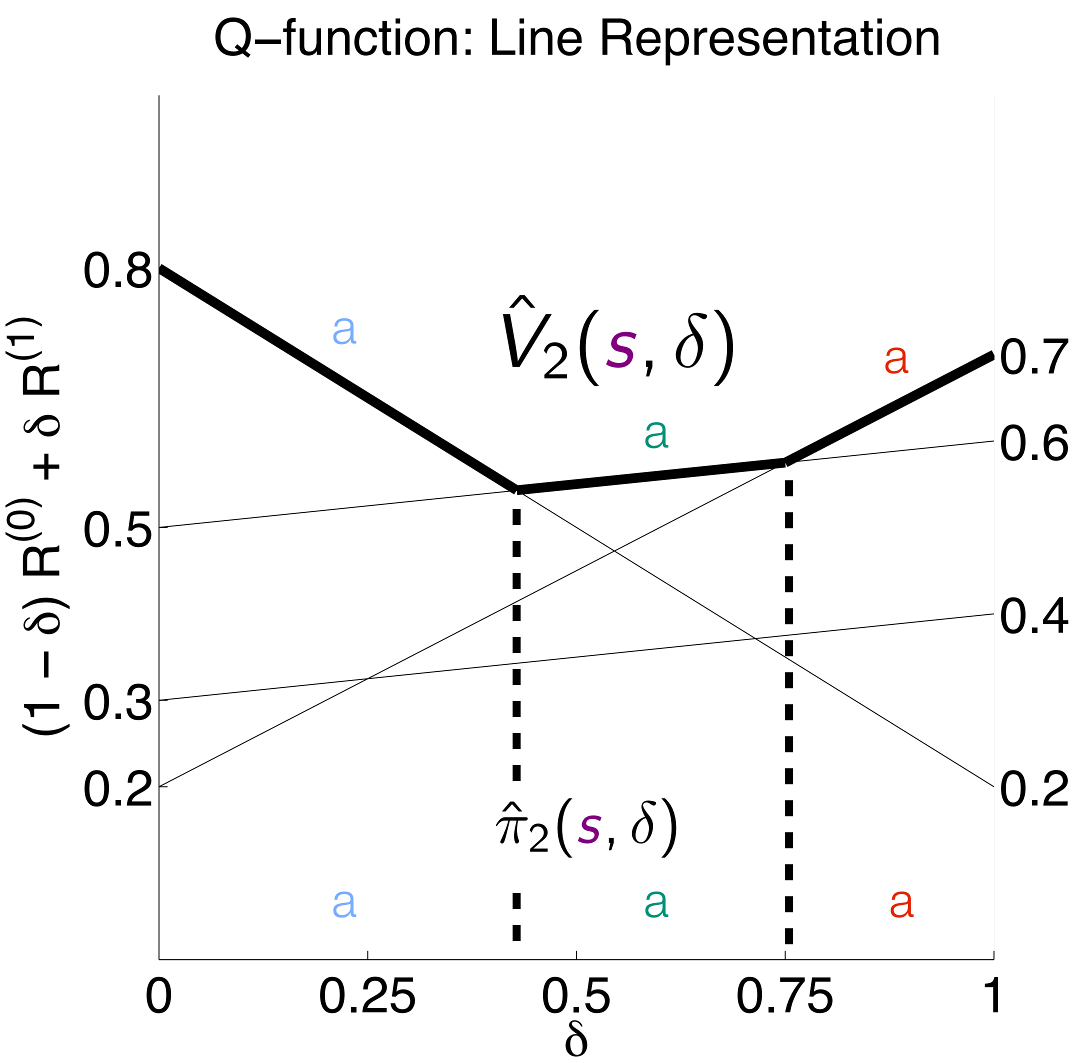
- When we take the max, also “remember” the argmax
- This gives

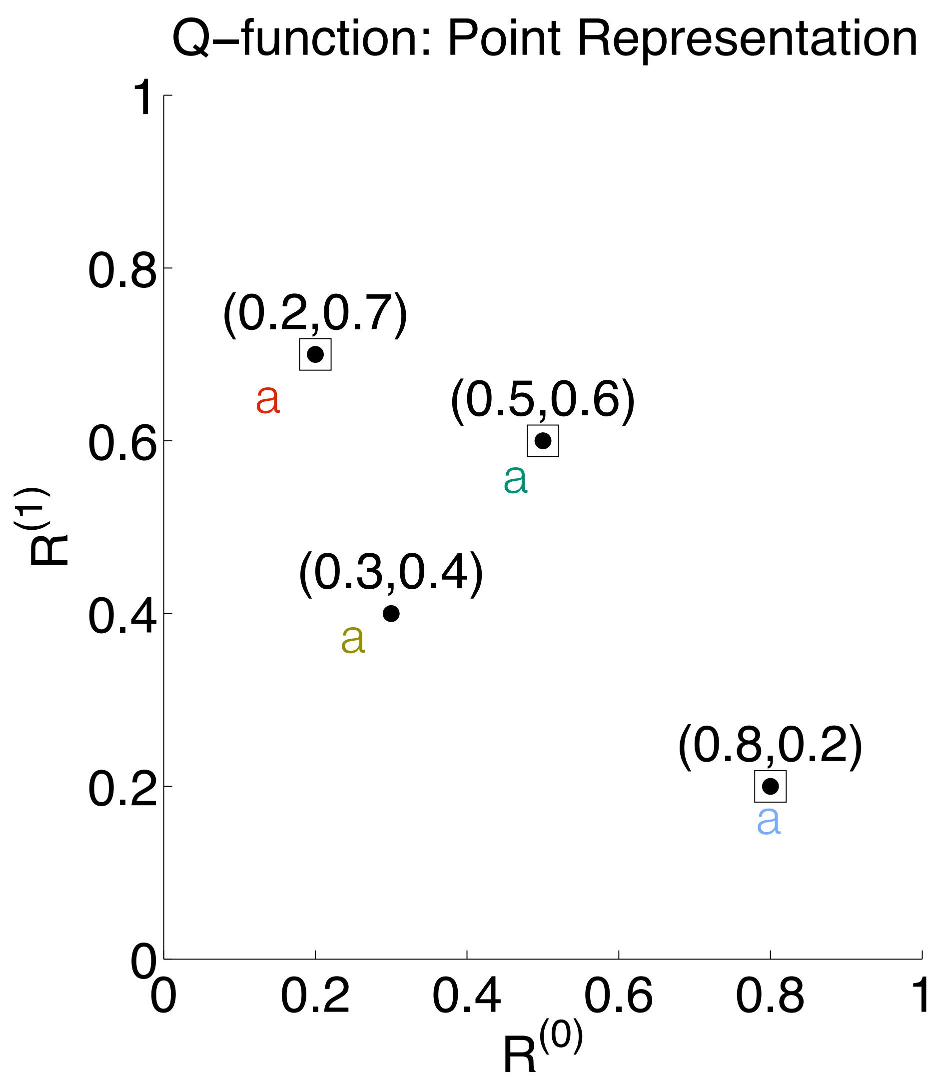




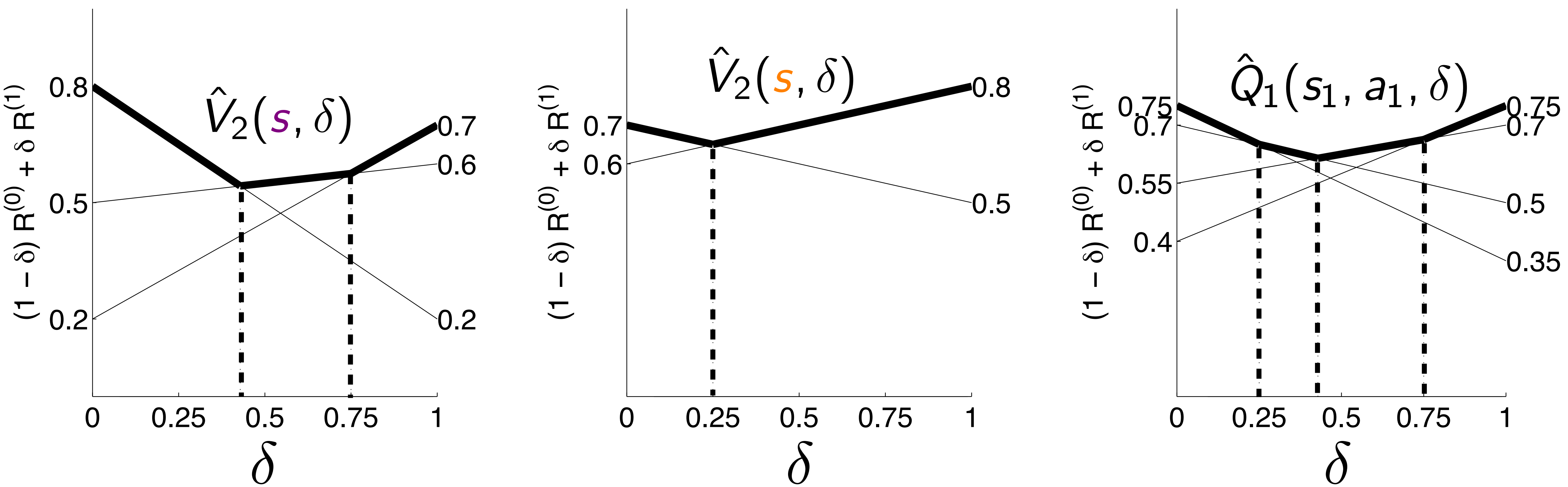
Pointwise Maximum Over Actions

- When we take the max, also “remember” the argmax
- This gives $\hat{\pi}_2(S_2, \delta)$



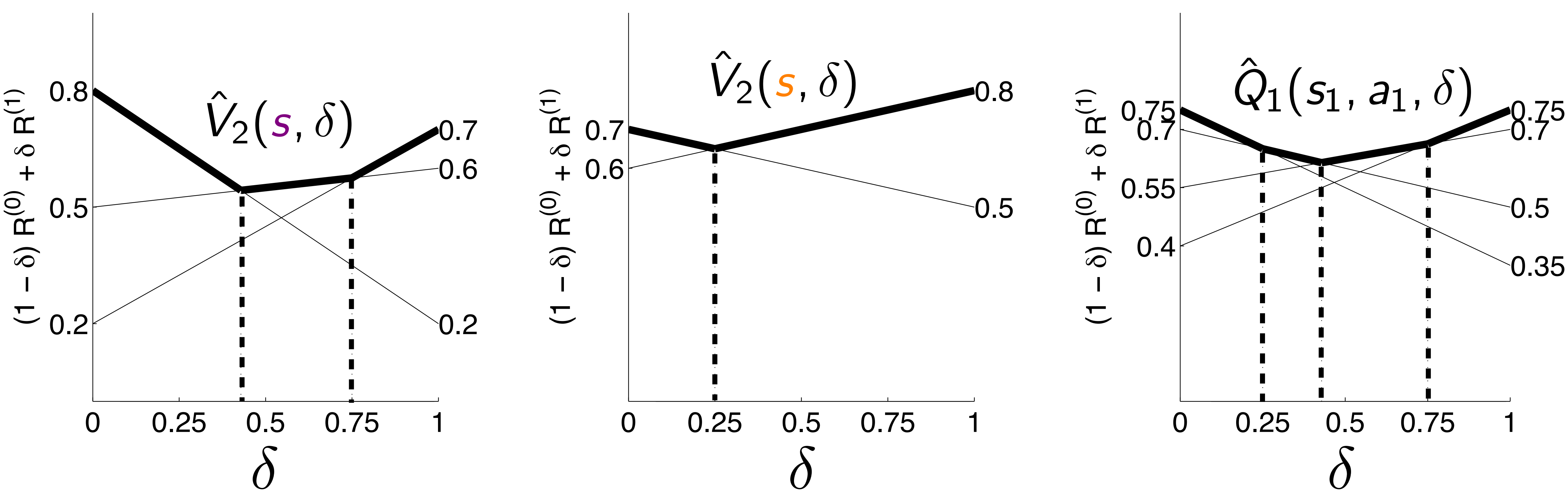


Pointwise Average Over Next State



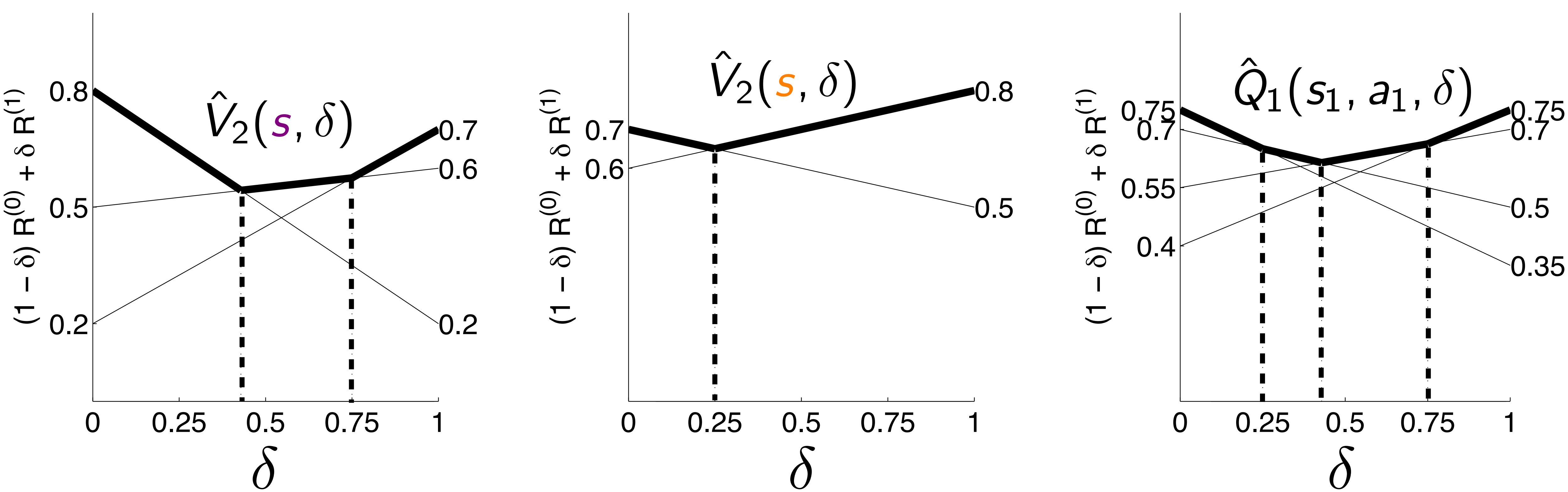
Pointwise Average Over Next State

- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in **δ**
  - Average of $\hat{V}_2(S_2, \delta)$ over tuples where $S_1 = s_1, A_1 = a_1$



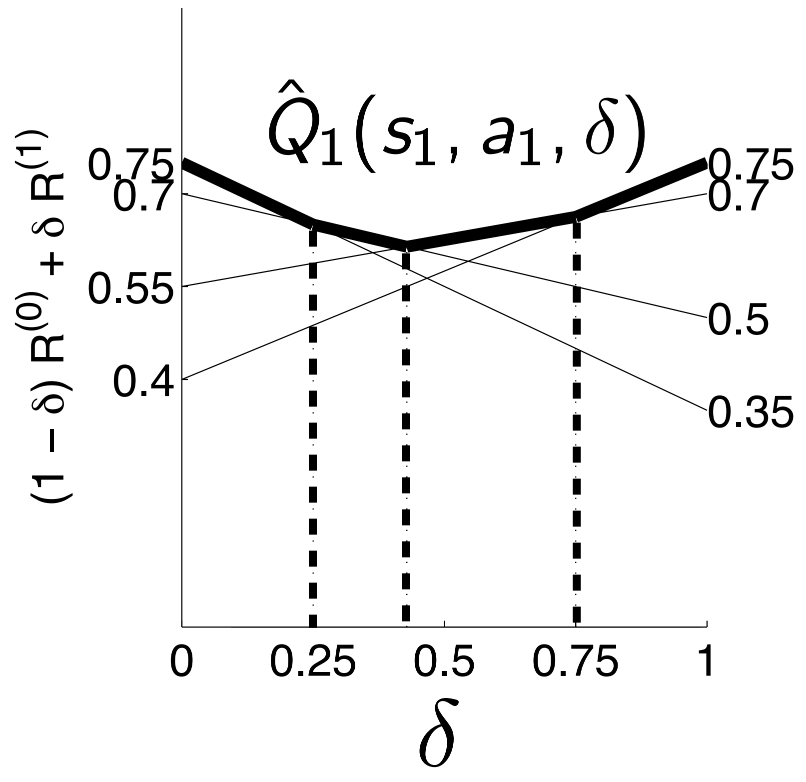
Pointwise Average Over Next State

- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in δ
  - Average of $\hat{V}_2(S_2, \delta)$ over tuples where $S_1 = s_1, A_1 = a_1$
- Line-based representation has computational advantage
  - Identify regions where $\hat{V}_2(s, \delta)$, $\hat{V}_2(s, \delta)$, ... are *simultaneously* linear
  - Compute averages at knots between regions



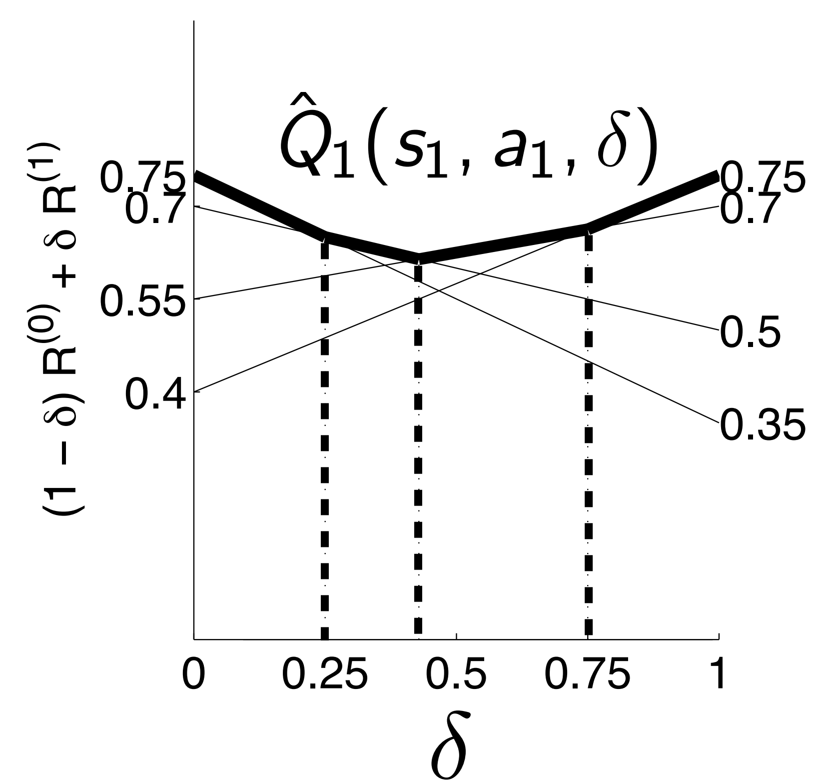
Pointwise Maximum Over Actions

$$\hat{Q}_1(s, a, \delta), \hat{Q}_1(s, a, \delta), \ldots$$



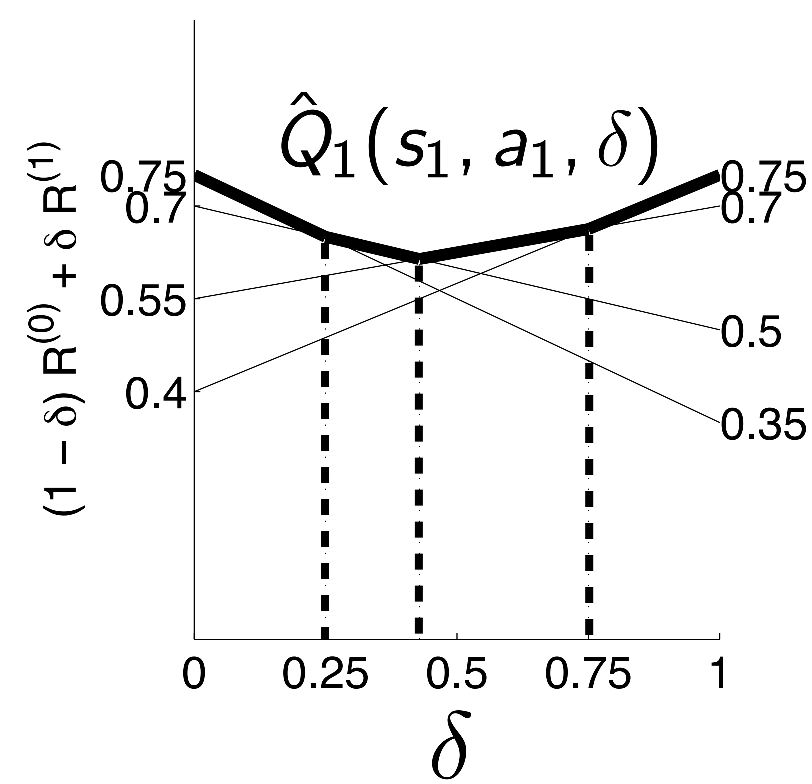
Pointwise Maximum Over Actions

- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in δ
  - We know where the pieces are
  - Identify regions where $\hat{Q}_1(s, a, \delta)$, $\hat{Q}_1(s, a, \delta)$, ... are simultaneously linear



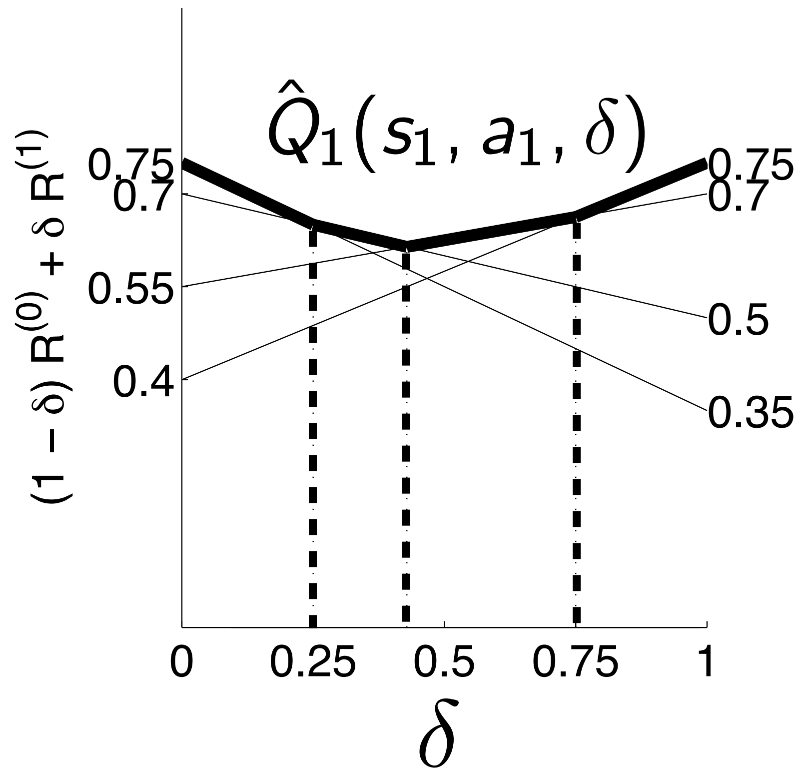
Pointwise Maximum Over Actions

- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in **δ**
  - We know where the pieces are
  - Identify regions where $\hat{Q}_1(s, a, \delta)$, $\hat{Q}_1(s, a, \delta)$, ... are simultaneously linear
  - We know how to take pointwise argmax of linear functions

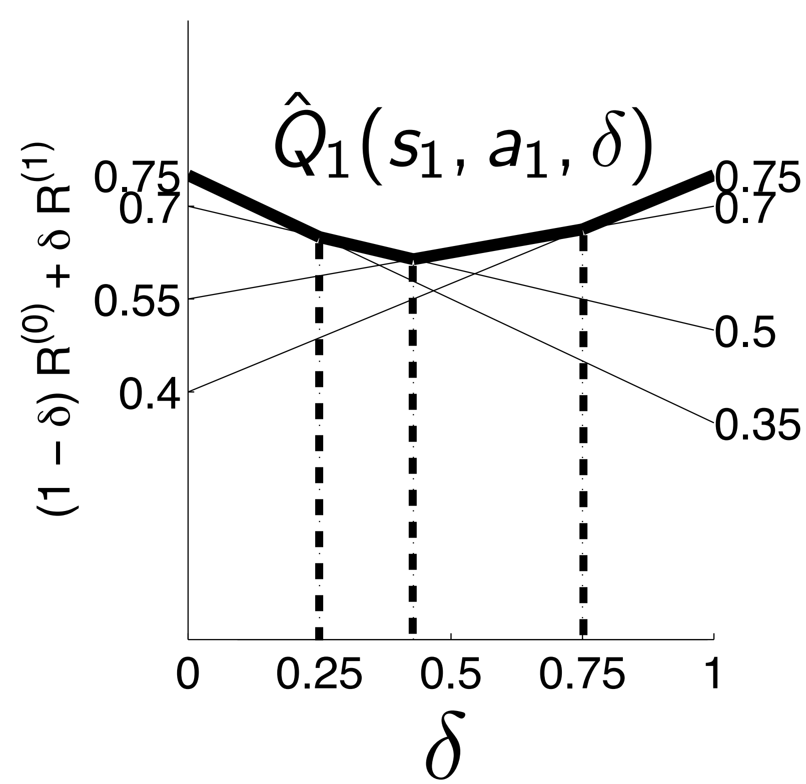


Pointwise Maximum Over Actions

- $\hat{Q}_1(s_1, a_1, \delta)$ is continuous and piecewise linear in δ
  - We know where the pieces are
  - Identify regions where $\hat{Q}_1(s, a, \delta)$, $\hat{Q}_1(s, a, \delta)$, ... are simultaneously linear
  - We know how to take pointwise argmax of linear functions
- This yields $\hat{\pi}_1(s_1, \delta)$. Done!

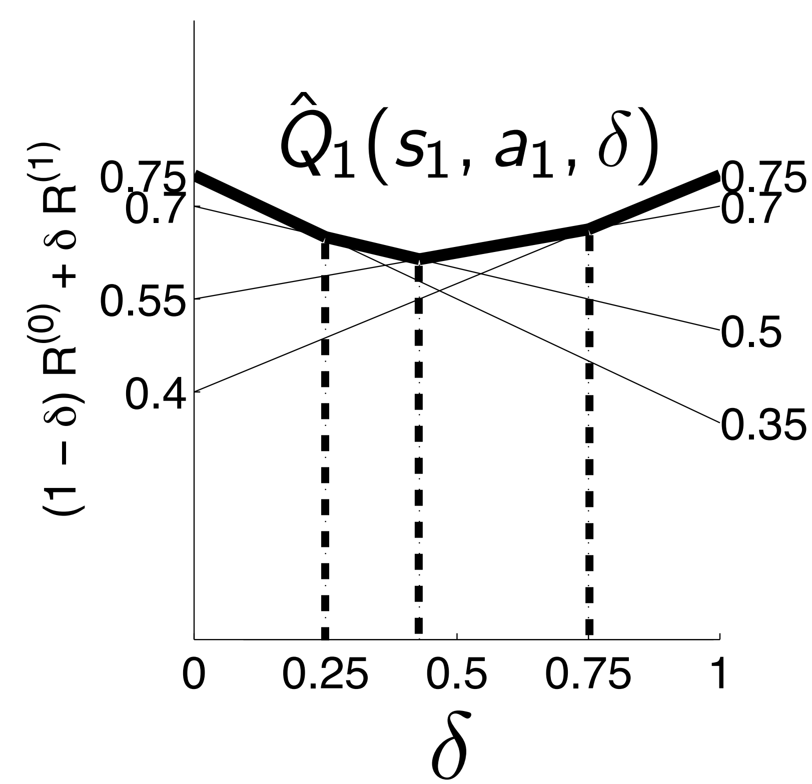


Summary:
Q-Learning with Multiple Reward Definitions



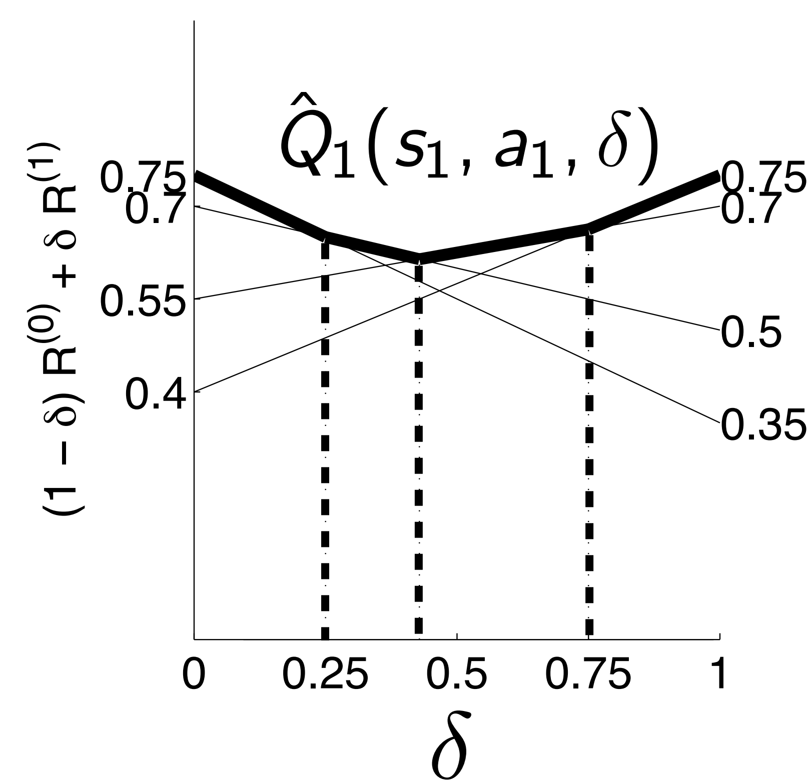
Summary:
Q-Learning with Multiple Reward Definitions

- Summary:



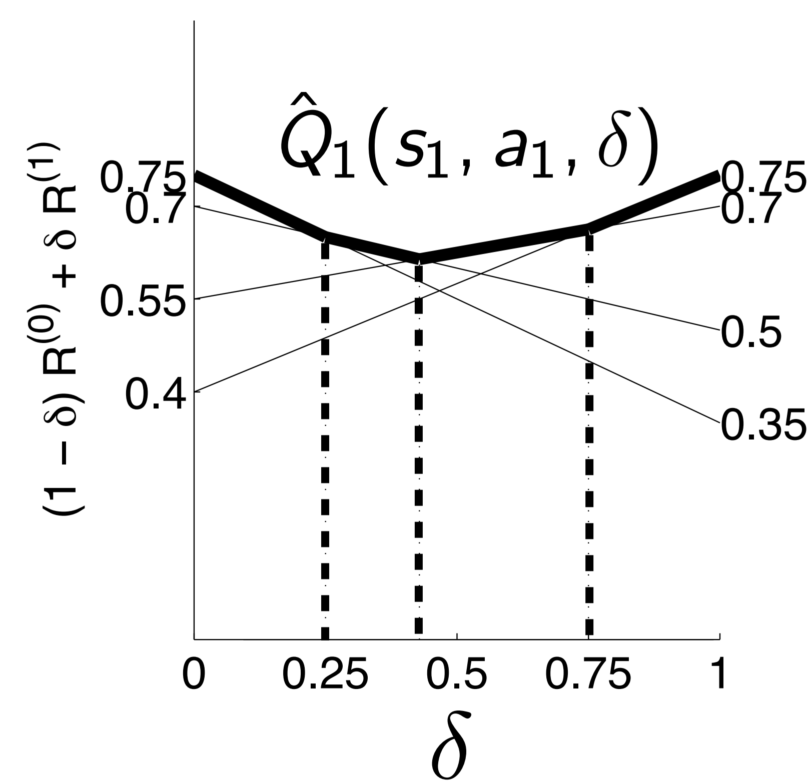
Summary:
Q-Learning with Multiple Reward Definitions

- Summary:
  - Pointwise max over actions turns $\hat{Q}_2$ into $\hat{V}_2$
    - Use point representation, convex hull



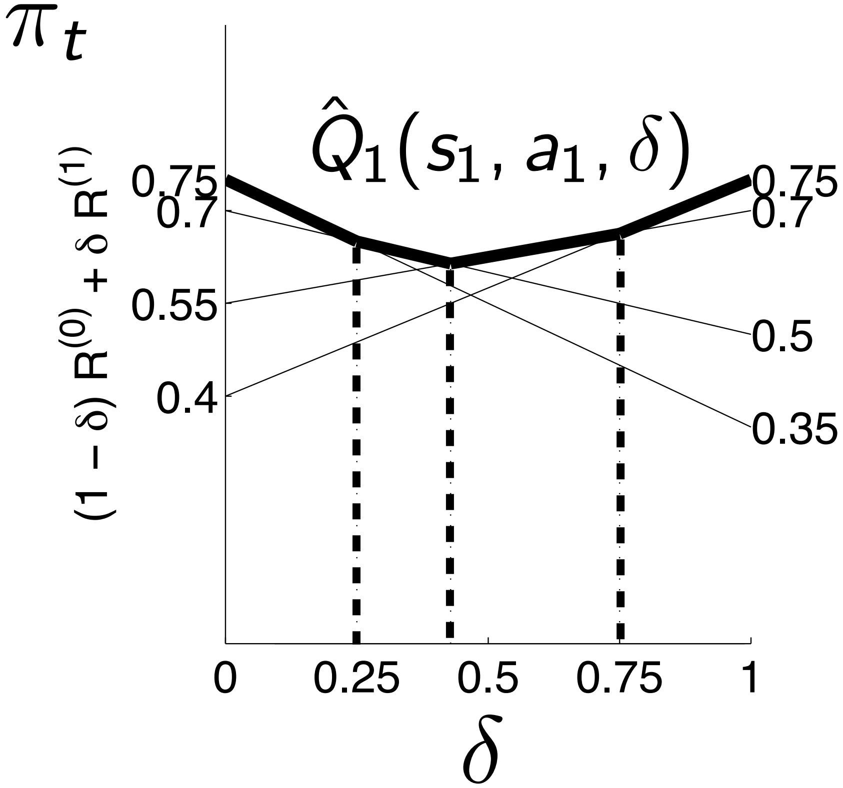
Summary:
Q-Learning with Multiple Reward Definitions

- Summary:
  - Pointwise max over actions turns $\hat{Q}_2$ into $\hat{V}_2$
    - Use point representation, convex hull
  - Pointwise average over state turns $\hat{V}_2$ into $\hat{Q}_1$
    - Use line representation, average at knots



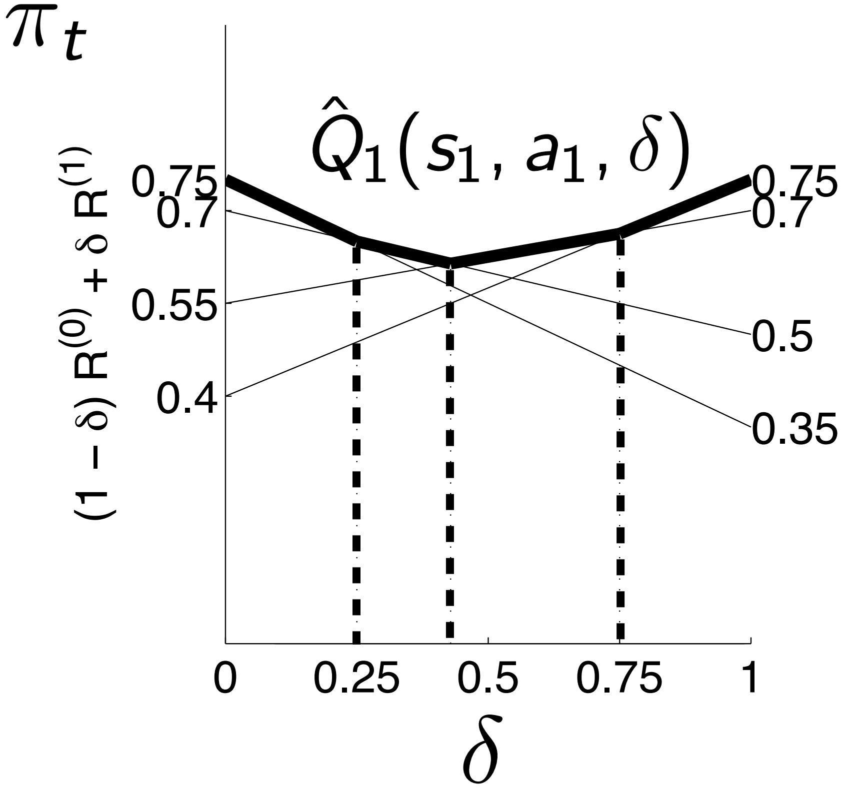
Summary:
Q-Learning with Multiple Reward Definitions

- Summary:
  - Pointwise max over actions turns $\hat{Q}_2$ into $\hat{V}_2$
    - Use point representation, convex hull
  - Pointwise average over state turns $\hat{V}_2$ into $\hat{Q}_1$
    - Use line representation, average at knots
  - Can take pointwise argmax of $\hat{Q}_t$ to get $\hat{\pi}_t$



Summary:
Q-Learning with Multiple Reward Definitions

- Summary:
  - Pointwise max over actions turns $\hat{Q}_2$ into $\hat{V}_2$
    - Use point representation, convex hull
  - Pointwise average over state turns $\hat{V}_2$ into $\hat{Q}_1$
    - Use line representation, average at knots
  - Can take pointwise argmax of $\hat{Q}_t$ to get $\hat{\pi}_t$
- Works for arbitrary number of stages



Computational Complexity

Computational Complexity

- How complex are the functions?
- $\hat{V}_2(S_2, \delta)$ is cts. and piecewise linear in δ, with $O(|\mathcal{A}|)$ pieces
- $\hat{Q}_1(s_1, a_1, \delta)$ is cts. and piecewise linear in δ with $O(|\mathcal{S}||\mathcal{A}|)$ pieces
- *T* stages: At stage *t*, $\hat{V}t(S_t, \delta)$ has $O(|\mathcal{S}|^{T-t}|\mathcal{A}|^{T-t})$ pieces

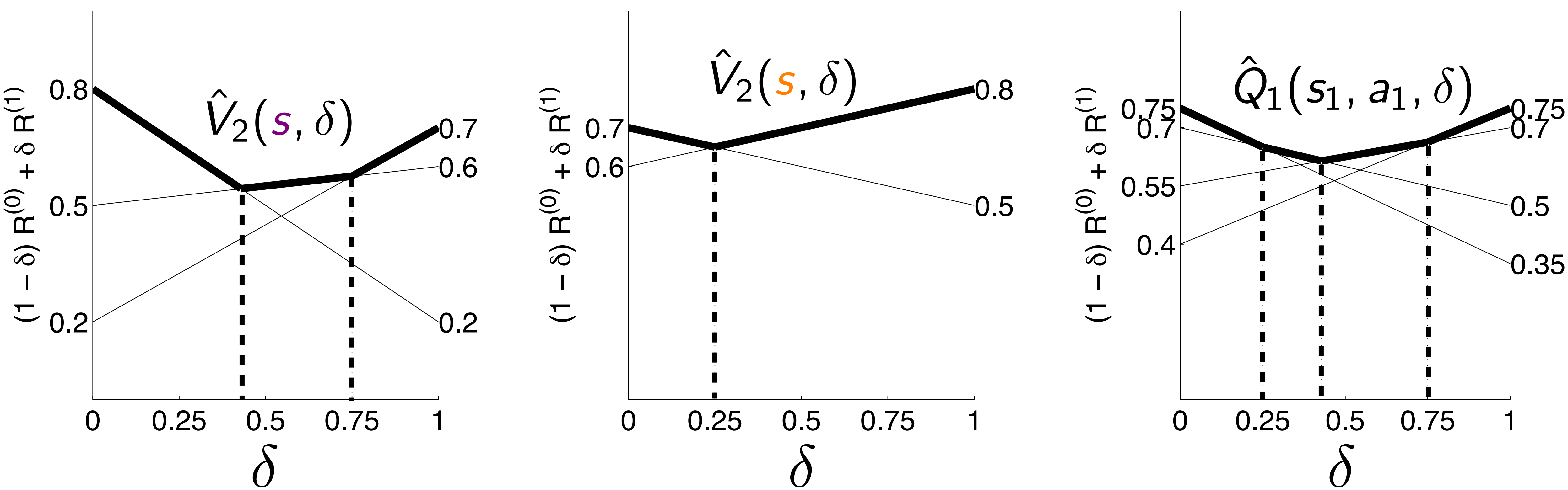
Computational Complexity

- How complex are the functions?
- $\hat{V}_2(S_2, \delta)$ is cts. and piecewise linear in δ, with $O(|\mathcal{A}|)$ pieces
- $\hat{Q}_1(s_1, a_1, \delta)$ is cts. and piecewise linear in δ with $O(|\mathcal{S}||\mathcal{A}|)$ pieces
- T stages: At stage t, $\hat{V}t(S_t, \delta)$ has $O(|\mathcal{S}|^{T-t}|\mathcal{A}|^{T-t})$ pieces

- To compute $\hat{Q}_{t-1}(s_{t-1}, a_{t-1}, \delta)$
    - using the line representation takes $O(|\mathcal{S}|^{T-t}|\mathcal{A}|^{T-t} \cdot |\mathcal{S}||\mathcal{A}|)$
    - using the point representation takes $\tilde{O}((|\mathcal{S}|^{T-t}|\mathcal{A}|^{T-t})^2 \cdot |\mathcal{S}||\mathcal{A}|)$
        - point based approach by Barret & Narayanan 2008

Computational Complexity

- Previous work: took $\tilde{O}((|\mathcal{S}|^{T-t}|\mathcal{A}|^{T-t})^2 \cdot |\mathcal{S}||\mathcal{A}|)$ time using pt. rep.
  - Relies on convexity in δ of $\hat{Q}_t(S_t, A_t, \delta)\ \forall t$
- Our algorithm is faster, does not require convexity
  - Can be used with linear regression models



Algorithm for Linear Regression: Executive Summary

Algorithm for Linear Regression: Executive Summary

- At each timepoint t, define $\hat{Q}_t(S_t, A_t, \delta; \hat{\beta}_t(\delta)) = c_{s_t, a_t}^{\mathsf{T}} \hat{\beta}_t(\delta)$

Algorithm for Linear Regression: Executive Summary

- At each timepoint t, define $\hat{Q}_t(S_t, A_t, \delta; \hat{\beta}_t(\delta)) = c_{s_t,a_t}^{\mathsf{T}} \hat{\beta}_t(\delta)$
- In least-squares regression, each **coefficient** is linear **in the targets**
  - $\hat{\beta} = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y$

Algorithm for Linear Regression: Executive Summary

- At each timepoint t, define $\hat{Q}_t(S_t, A_t, \delta; \hat{\beta}_t(\delta)) = c_{s_t,a_t}^{\mathsf{T}} \hat{\beta}_t(\delta)$
- In least-squares regression, each **coefficient** is linear **in the targets**
  - $\hat{\beta} = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y$
- For $t = T$, targets are $R(\boldsymbol{\delta}) \equiv (1 - \boldsymbol{\delta}) \cdot R^{(0)} + \boldsymbol{\delta} \cdot R^{(1)}$, which is linear in $\boldsymbol{\delta}$
  - $\hat{\beta}_T(\delta)$ is linear in $\boldsymbol{\delta}$

Algorithm for Linear Regression: Executive Summary

- At each timepoint t, define $\hat{Q}_t(S_t, A_t, \delta; \hat{\beta}_t(\delta)) = c_{s_t, a_t}^{\mathsf{T}} \hat{\beta}_t(\delta)$
- In least-squares regression, each **coefficient** is linear **in the targets**
  - $\hat{\beta} = (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}y$
- For $t = T$, targets are $R(\delta) \equiv (1 - \delta) \cdot R^{(0)} + \delta \cdot R^{(1)}$, which is linear in δ
  - $\hat{\beta}_T(\delta)$ is linear in δ
- For $t < T$, targets are $\hat{V}_{t+1}(S_{t+1}, \delta)$, which is piecewise linear in δ
  - $\hat{\beta}_t(\delta)$ is piecewise linear in δ
  - **But** not necessarily convex, so previous method would not work

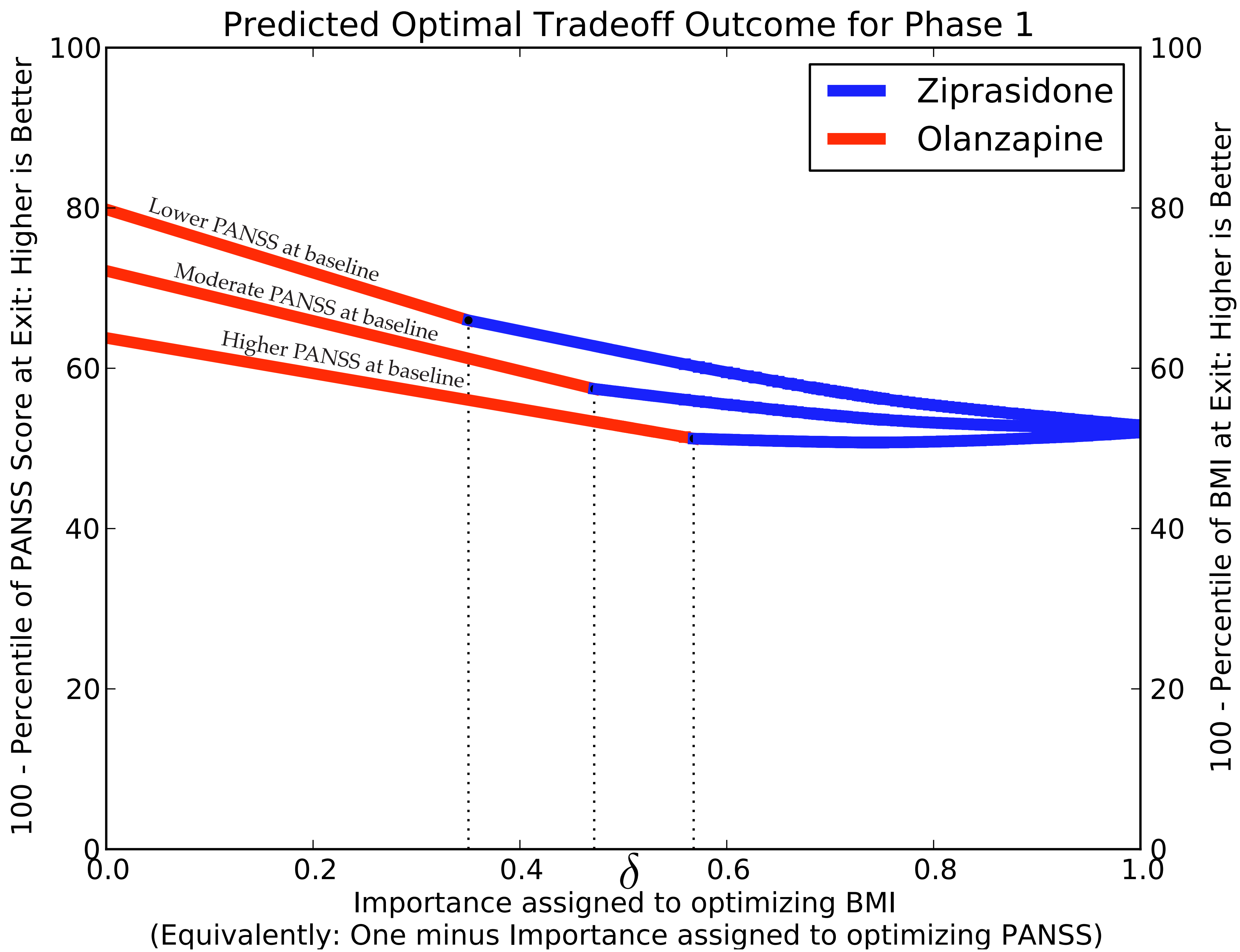
Algorithm for Linear Regression: Executive Summary

- At each timepoint t, define $\hat{Q}_t(S_t, A_t, \delta; \hat{\beta}_t(\delta)) = c_{s_t,a_t}^{\mathsf{T}} \hat{\beta}_t(\delta)$
- In least-squares regression, each **coefficient** is linear **in the targets**
  - $\hat{\beta} = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y$
- For $t = T$, targets are $R(\boldsymbol{\delta}) \equiv (1 - \boldsymbol{\delta}) \cdot R^{(0)} + \boldsymbol{\delta} \cdot R^{(1)}$, which is linear in $\boldsymbol{\delta}$
  - $\hat{\beta}_T(\delta)$ is linear in $\boldsymbol{\delta}$
- For $t < T$, targets are $\hat{V}_{t+1}(S_{t+1}, \delta)$, which is piecewise linear in $\boldsymbol{\delta}$
  - $\hat{\beta}_t(\delta)$ is piecewise linear in $\boldsymbol{\delta}$
  - **But** not necessarily convex, so previous method would not work
- Time complexity to compute $\hat{Q}_{t-1}(S_{t-1}, A_{t-1}, \delta; \hat{\beta}_{t-1}(\delta))$ from $\hat{V}_t(S_t, \delta)$ is $O(n^{T-t}|\mathcal{A}|^{T-t} \cdot n|\mathcal{A}|)$

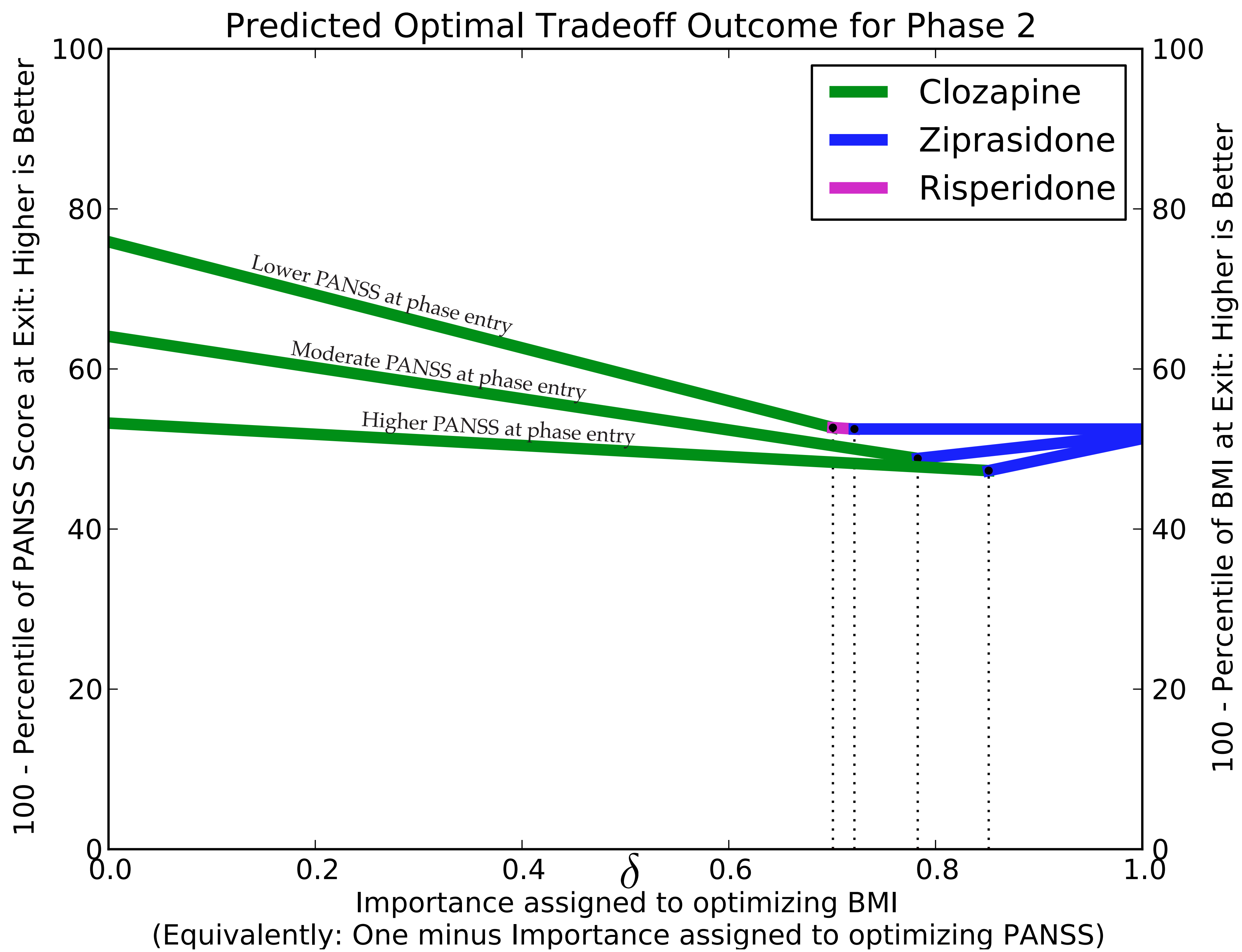
Example: CATIE

- Large (n = 1460) comparative effectiveness trial funded by NIMH
- Compares medications for treatment of schizophrenia
- Most patients randomized two times:
    - First to one of 5 actions
    - Then, if desired, to one of 5 different actions
- Details are quite complicated
- Following is a *highly* simplified analysis
- Overall, the results are consistent with what is known in the literature
- Rewards: PANSS (symptoms) versus BMI (weight gain side-effect)

Example: CATIE Exploratory Analysis



Example: CATIE Exploratory Analysis



Example: CATIE-based Decision Aid

- One possibility for a decision aid is a very coarse version of the plots:

Recommendation given State and Preference	Strong Preference for Symptom Relief over Weight Control	Mild Preference for Symptom Relief over Weight Control	Mild Preference for Weight Control over Symptom Relief	Strong Preference for Weight Control over Symptom Relief
Lower PANSS at Entry to Phase 1	Olanzapine	Olanzapine or Ziprasidone	Ziprasidone	Ziprasidone
Moderate PANSS at Entry to Phase 1	Olanzapine	Olanzapine or Ziprasidone	Ziprasidone	Ziprasidone
Higher PANSS at Entry to Phase 1	Olanzapine	Olanzapine	Olanzapine or Ziprasidone	Ziprasidone
Lower PANSS at Entry to Phase 2	Clozapine	Clozapine	Clozapine, Risperidone, or Ziprasidone	Ziprasidone
Moderate PANSS at Entry to Phase 2	Clozapine	Clozapine	Clozapine	Clozapine or Ziprasidone
Higher PANSS at Entry to Phase 2	Clozapine	Clozapine	Clozapine	Clozapine or Ziprasidone

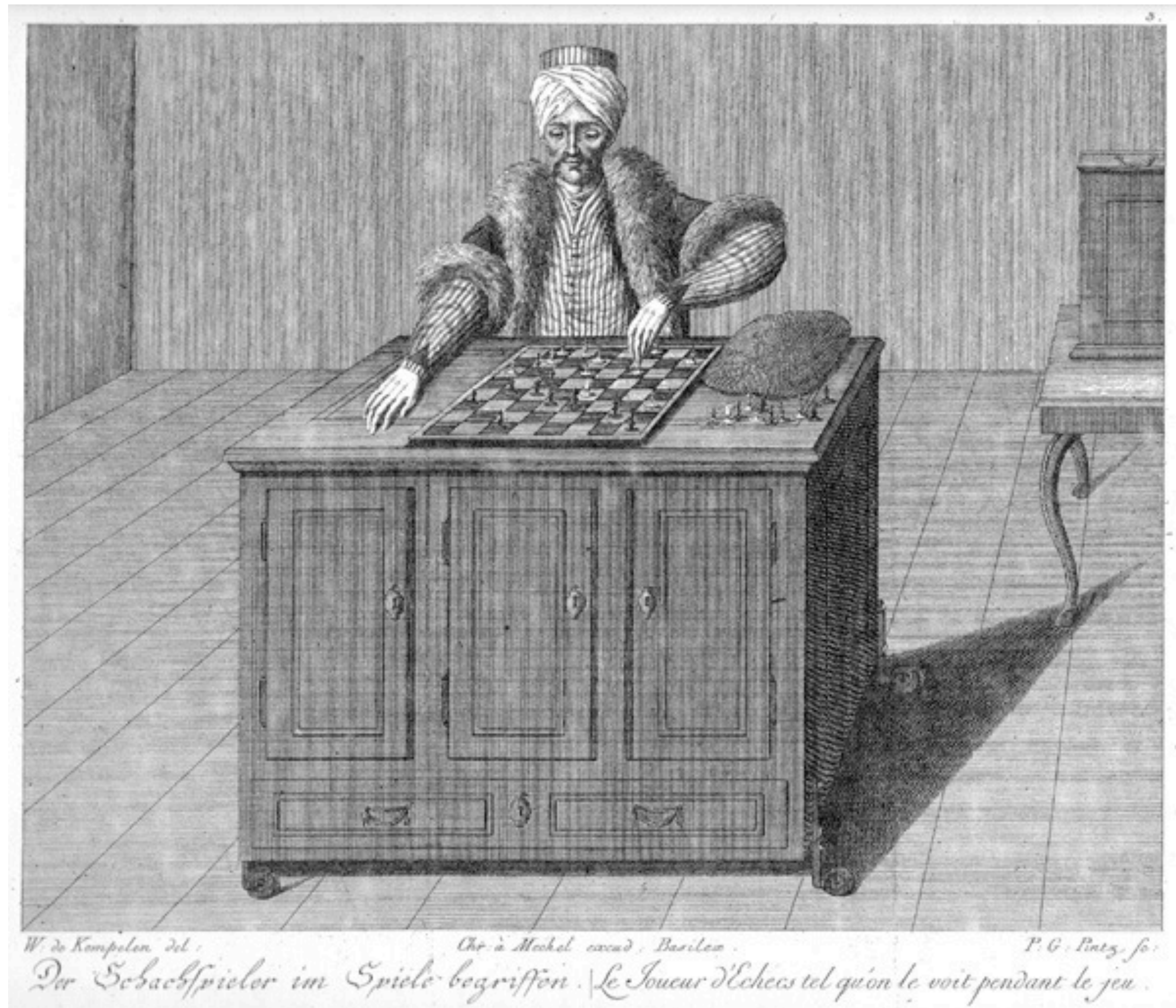
- Thanks to: Holly Wittemann, Brian Zikmund-Fisher for this idea

Future Work

- Evaluating the “Inverse Preference Elicitation” Idea
  - MTurk Evaluation
- The Algorithms and Methods
  - Measures of Uncertainty
  - More flexible models / Approximation algorithms
  - More reward definitions
- Clinical Science Applications

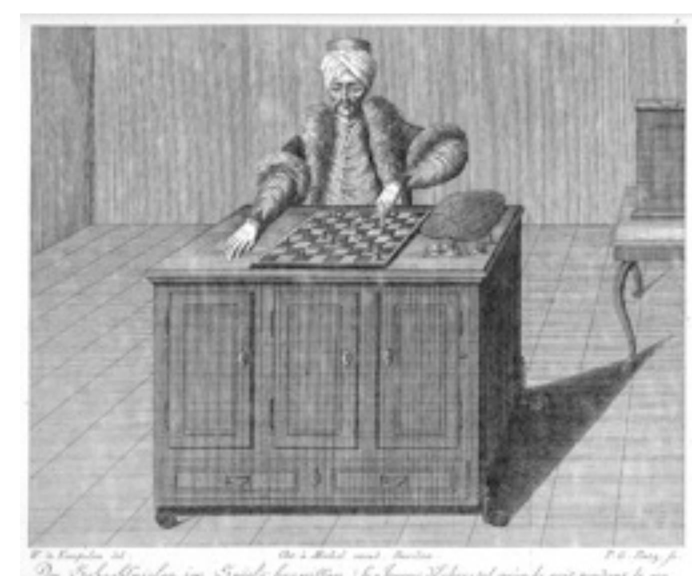
Amazon Mechanical Turk

- Mechanism for recruiting and paying users to do “Human Intelligence Tasks” - HITs
- Popular for running survey experiments (demographics at least as good as undergrads [Paolacci, Chandler, Ipeirotis 2010])



Amazon Mechanical Turk

- Our experiment will compare **eliciting δ using a slider** with directly **eliciting an action using a decision aid**.
- User will perform one of four different (similar and boring) sub-tasks, each one with different payoff and time required
- The choice of action determines the sub-task, *and also* affects the workload of all the subsequent subtasks - myopic decision making is sub-optimal.
- Competing preferences:
  - Save time vs. Make money
- We will compare the appeal of the two methods
- Plan to go live January 2011



Future Work - Measures of Uncertainty

- Optimal policies for fixed δ do not reflect possible estimation error in $\hat{\pi}_t(S_t, \delta)$, or equivalently, uncertainty about $\hat{Q}_t(S_t, A_t, \delta)$
- Even for fixed δ, constructing confidence intervals for $\hat{Q}_t(S_t, A_t, \delta)$ requires care when $t < T$
  - Because of the max operator used in Q-learning, estimators $\hat{Q}_t(S_t, A_t, \delta)$ are non-regular at earlier time points
  - Work in progress by Laber, Lizotte, Qian, Murphy addresses this
- Presentation of uncertainty information requires more thought

Future Work - More Reward Definitions

- For backups: Allowing 3 reward definitions is feasible using methods from computational geometry (have already implemented)
- Representing non-convex continuous piecewise linear functions in high dimensions is difficult
- Making use of a three-reward analysis for decision making will be more complex

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. Type 2 Diabetes
   - Future disease complications versus drug side-effects

Questions

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

- Daniel J. Lizotte, Michael Bowling, and Susan A. Murphy. *Efficient Reinforcement Learning with Multiple Reward Functions for Randomized Clinical Trial Analysis*. Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML), 2010.

- Related work:

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