Gradient Boosting for RL in Complex Domains

David Abel\textsuperscript{2}, Alekh Agarwal\textsuperscript{1}, Fernando Diaz\textsuperscript{1}, Akshay Krishnamurthy\textsuperscript{1}, Robert Schapire\textsuperscript{1}

\textsuperscript{1}Microsoft Research
\textsuperscript{2}Brown University

ICML RL and Abstraction Workshop 2016
Goal

Develop simple and scalable Reinforcement Learning (RL) techniques that can solve high dimensional problems.
Minecraft
MALMO: Minecraft AI Testbed

MALMO: an API for developing agents in Minecraft

Gridworld

MALMO: Minecraft AI Testbed

Gridworld

Goal

difficulty

Build 32 bit ALU
Key Components

Developed an RL agent for Minecraft-scale problems:
Key Components

Developed an RL agent for Minecraft-scale problems:

1) A *vision system* capable of real-time RL in Minecraft.
Developed an RL agent for Minecraft-scale problems:

1) A vision system capable of real-time RL in Minecraft.

2) A new lightweight function approximator for RL.

Gradient Boosting  [Friedman 2001, Mason 1999]
Developed an RL agent for Minecraft-scale problems:

1) A *vision system* capable of real-time RL in Minecraft.

2) A *new lightweight function approximator* for RL.

3) An *exploration* technique for model-free RL (but: preliminary experiments are inconclusive).
Gradient Boosting for RL

*Treat RL as a Regression problem for the $Q$-function*
Gradient Boosting for RL

1) Fix an $\epsilon$-greedy policy with respect to $\hat{Q}$
Gradient Boosting for RL

1) Fix an $\epsilon$-greedy policy with respect to $\hat{Q}$

2) Run an episode —> receive a dataset:

*Treat RL as a Regression problem for the $Q$-function*
Gradient Boosting for RL

1) Fix an $\varepsilon$-greedy policy with respect to $\hat{Q}$

2) Run an episode $\rightarrow$ receive a dataset:

$$D = \langle (s_1, a_1, r_1), \ldots, (s_N, a_N, r_N) \rangle$$

state  reward  action

Treat RL as a Regression problem for the $Q$-function
Gradient Boosting for RL

1) Fix an $\epsilon$-greedy policy with respect to $\hat{Q}$

2) Run an episode $\rightarrow$ receive a dataset:

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

*Treat RL as a Regression problem for the $Q$-function*
Gradient Boosting for RL

1) Fix an $\varepsilon$-greedy policy with respect to $\hat{Q}$

2) Run an episode $\rightarrow$ receive a dataset:

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[ h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2$$

**Treat RL as a Regression problem for the $Q$-function**
Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[ h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2$$

- new weak learner
- previous estimate
- Bellman residual
Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[ h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2$$

Where:

$$\hat{Q}(s, a) = \sum_{e=1}^{E} h_e(s, a)$$

*Treat RL as a Regression problem for the $Q$-function*
Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[ h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2$$

Where:

$$\hat{Q}(s, a) = \sum_{e=1}^{E} h_e(s, a)$$

Treat RL as a Regression problem for the $Q$-function
Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$
\min_h \sum_{i=1}^{N} \left[ h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2
$$

We solve this using regression trees as the weak learner.

*Treat RL as a Regression problem for the $Q$-function*
High Level

episode 1

observation: 
reward: a number

agent

world

action
High Level

Episode 1

Observation:

Reward: a number

\[ \hat{Q} = \sum ( \text{agent} ) \]
Observation: a number

Reward: a number

\[ \hat{Q} = \sum \left( \langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle \right) \]
High Level

episode 1

observation:

reward: a number

world

action

agent

\[ \mathcal{D}_1 \]

\[ \langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle \]

\[ \hat{Q} = \sum ( ) \]
High Level

\[ \hat{Q} = \sum (h_1(s, a)) \]

episode 1

observation: world

reward: a number

agent

\[ \mathcal{D}_1 \]

\[ \langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle \]

Loss

Tree

\[ = h_1(s, a) \]
High Level

episode 1

world

observation:

reward: a number

agent

action

\( \langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle \)

\( \mathcal{D}_1 \)

Loss

\[ \hat{Q} = \sum_{\mathcal{D}_1} (h_1) \]
episode 1

observation: 

reward: a number

world

tagent

\( \hat{Q} = \sum_{D_1} h_1 \)
episode 1

observation: a picture

reward: a number

agent

world

episode 2

\hat{Q} = \sum_{D_1} \left( h_1 \right)

\mathcal{D}_2

\langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle

\text{Loss}

Tree

\mathcal{D}_2

h_2(s, a)
High Level

\[ \hat{Q} = \sum \left( \begin{array}{c} h_1 \\ h_2 \end{array} \right) \]
High Level

observation:

reward: a number

\[ \hat{Q} = \sum (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8) \]

\[ D_1 \quad D_2 \quad \ldots \quad \ldots \quad \ldots \quad D_8 \]
Intuitively Nice Properties

• Non-parametric

• Simple, easy to implement, minimal hand-engineering

• Interleaved data collection

• Rich theoretical literature, room for analysis.

• Only need to store one episode’s worth of data.
Experiments: Baselines

• Baseline 1  (Linear Approximator)
• Baseline 2  (Random Forest Approximator)
• Baseline 3  (Batch Boost Approximator)
Experiments: Baselines

- Baseline 1  
  - (Linear Approximator)

- Baseline 2  
  - (Random Forest Approximator)

- Baseline 3  
  - (Batch Boost Approximator)

Similar to Fitted Q-iteration [Ernst et al. 2005]
Experiments: Visual Grid
Visual Grid: Results

Key

Gradient Booster  Batch Booster  Linear  Forest
Experiments: Hillclimbing
Visual Hill Climb: Results

Key

Gradient Booster  Batch Booster  Linear  Forest
Next Steps

• Investigate relevant exploration techniques inspired by Gradient Boosting.

• Use rich foundation of theory on gradient boosting to inspire analysis of this approach.

• Further experimentation.
Acknowledgments

A big thank you to The MALMO team!

David Bignell, Katja Hofmann, Tim Hutton, Matthew Johnson, Pushmeet Kohli, Nate Kushman, Ewa Luger, Bhaskar Mitra, Jamie Shotton, Evelyne Viegas.