Protecting Against Evaluation Overfitting in Empirical Reinforcement Learning

Shimon Whiteson, University of Amsterdam
Brian Tanner, University of Alberta
Matthew E. Taylor, Lafayette College
Peter Stone, University of Texas at Austin

March 4, 2011

Evaluating Machine Learning Algorithms

- Subjective evaluations
  - Pros: leverage intuition
  - Cons: cannot expose fallacious assumptions
- Theoretical results
  - Pros: rigorous
  - Cons: not always obtainable; conditions may not apply
- Empirical evaluations
  - Pros: yields insights, spurs innovation
  - Cons: evaluation overfitting

The Problem

- **One common approach**: measure average cumulative reward across independent trials in a fixed benchmark environment
- Various design choices can yield an overfit method:
  - State representation
  - Initial value function
  - Learning rate, etc.
- **Extreme example**: 'learning algorithm' for Mountain Car that begins with optimal policy

Goal

Devise empirical methodologies that guard against overfitting in on-line reinforcement learning

The Problem

- **One common approach**: measure average cumulative reward across independent trials in a fixed benchmark environment
- Various design choices can yield an overfit method:
  - State representation
  - Initial value function
  - Learning rate, etc.
- **Extreme example**: 'learning algorithm' for Mountain Car that begins with optimal policy

Goal

Devise empirical methodologies that guard against overfitting in on-line reinforcement learning

Outline

- Evaluation Overfitting
  - Data vs. Environment Overfitting
  - Fitting vs. Overfitting
- Generalized Environments
  - Open Generalized Methodology
  - Secret Generalized Methodology
  - Meta-Generalized Methodology
  - Generalized Performance Measures
- Results

Evaluation Overfitting

Evaluation Process

A self-interested designer creates an agent with which an evaluator conducts independent trials yielding a score estimating some statistics, e.g., expected cumulative reward

- Scores implicitly represent performance on a target distribution
- In evaluation overfitting:
  - Evaluation yields a high score
  - Performance across target distribution is poor

Evaluation Overfitting

Evaluation Process

A self-interested designer creates an agent with which an evaluator conducts independent trials yielding a score estimating some statistics, e.g., expected cumulative reward

- Scores implicitly represent performance on a target distribution
- In evaluation overfitting:
  - Evaluation yields a high score
  - Performance across target distribution is poor

Data vs. Environment Overfitting

- In data overfitting:
  - Function agent produces is too customized to evaluation data
  - Poor generalization to new data from same environment
- In environment overfitting:
  - Agent is too customized to evaluation environment
  - Poor generalization to other environments in target distribution

While data overfitting is problematic in supervised learning, evaluation overfitting is problematic in reinforcement learning

Data vs. Environment Overfitting

- In data overfitting:
  - Function agent produces is too customized to evaluation data
  - Poor generalization to new data from same environment
- In environment overfitting:
  - Agent is too customized to evaluation environment
  - Poor generalization to other environments in target distribution

While data overfitting is problematic in supervised learning, evaluation overfitting is problematic in reinforcement learning

Fitting vs. Overfitting

- How broad should the target distribution be?
  - Broadly applicable agents are desirable
  - But specializing can give leverage
- Can environment overfitting be good?
  - No, but target distribution may be small
  - Fitting: customizing to target distribution at expense of others
  - Overfitting: customizing to evaluation setting at expense of target distribution

In reinforcement learning, target distributions need multiple environments in order to create reducible uncertainty

Fitting vs. Overfitting

- How broad should the target distribution be?
  - Broadly applicable agents are desirable
  - But specializing can give leverage
- Can environment overfitting be good?
  - No, but target distribution may be small
  - Fitting: customizing to target distribution at expense of others
  - Overfitting: customizing to evaluation setting at expense of target distribution

In reinforcement learning, target distributions need multiple environments in order to create reducible uncertainty

Generalized Environments

- Single-environment methodologies are not ideal
  - Invite environment overfitting
  - Still useful given a good-faith effort by designers
- **Simple solution**: formalize the target distribution in a generalized environment
  - $\mathcal{G} = \langle \Theta, \mu \rangle$, a distribution μ over a set of environments Θ
  - Score computed from multiple trials, each in a different environment sampled from Θ according to μ

Example: Helicopter Hovering in the RL Competition

Goal is to hover a helicopter in a fixed position; each trial has a different θ with an unknown wind velocity

Generalized Environments

- Single-environment methodologies are not ideal
  - Invite environment overfitting
  - Still useful given a good-faith effort by designers
- **Simple solution**: formalize the target distribution in a generalized environment
  - $\mathcal{G} = \langle \Theta, \mu \rangle$, a distribution μ over a set of environments Θ
  - Score computed from multiple trials, each in a different environment sampled from Θ according to μ

Example: Helicopter Hovering in the RL Competition

Goal is to hover a helicopter in a fixed position; each trial has a different θ with an unknown wind velocity

Generalized Environments

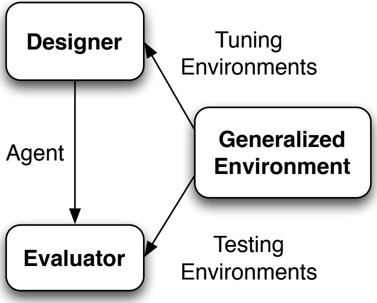
- Single-environment methodologies are not ideal
  - Invite environment overfitting
  - Still useful given a good-faith effort by designers
- **Simple solution**: formalize the target distribution in a generalized environment
  - $\mathcal{G} = \langle \Theta, \mu \rangle$, a distribution μ over a set of environments Θ
  - Score computed from multiple trials, each in a different environment sampled from Θ according to μ

Example: Helicopter Hovering in the RL Competition

Goal is to hover a helicopter in a fixed position; each trial has a different θ with an unknown wind velocity

Open Generalized Methodology

- $\mathcal{G}$ is known to designer
- In tuning phase, designer samples θ's freely from $\mathcal{G}$
- In test phase, evaluator samples new θ from $\mathcal{G}$ for each trial
- Protects against both data and environment overfitting



Secret Generalized Methodology

- Open methodology creates uncertainty about θ but not $\mathcal{G}$
- $\mathcal{G}$ may only approximate true target distribution
- In uncertainty overfitting, the agent is customized to $\mathcal{G}$ at the expense of other possible true target distributions
- In secret generalized methodology:
  - $\mathcal{G}$ is hidden
  - Designer receives only a fixed set of θ's sampled from $\mathcal{G}$
  - Agent is tested on independent θ's sampled from $\mathcal{G}$
- Pros and cons:
  - Protects against data, environment, and uncertainty overfitting
  - Does not require formalizing $\mathcal{G}$
  - Requires secrecy: limited to one-shot settings

Secret Generalized Methodology

- Open methodology creates uncertainty about θ but not $\mathcal{G}$
- $\mathcal{G}$ may only approximate true target distribution
- In uncertainty overfitting, the agent is customized to $\mathcal{G}$ at the expense of other possible true target distributions
- In secret generalized methodology:
  - $\mathcal{G}$ is hidden
  - Designer receives only a fixed set of θ's sampled from $\mathcal{G}$
  - Agent is tested on independent θ's sampled from $\mathcal{G}$
- Pros and cons:
  - Protects against data, environment, and uncertainty overfitting
  - Does not require formalizing $\mathcal{G}$
  - Requires secrecy: limited to one-shot settings

Meta-Generalized Methodology

- Avoid trade-offs with a meta-generalized environment
- $\mathcal{H} = \langle \Gamma, \tau \rangle$, a distribution τ over a set of generalized environments Γ
- In meta-generalized methodology:
  - In tuning, designer samples freely from $\mathcal{H}$
  - In testing, each meta-trial, involves a series of trials on environments sampled from a fixed $\mathcal{G}_i$ sampled from $\mathcal{H}$
- Pros and cons
  - Protects against data, environment, and uncertainty overfitting
  - No secrecy required
  - Requires formalizing $\mathcal{H}$ and conducting many trials

Meta-Generalized Methodology

- Avoid trade-offs with a meta-generalized environment
- $\mathcal{H} = \langle \Gamma, \tau \rangle$, a distribution τ over a set of generalized environments Γ
- In meta-generalized methodology:
  - In tuning, designer samples freely from $\mathcal{H}$
  - In testing, each meta-trial, involves a series of trials on environments sampled from a fixed $\mathcal{G}_i$ sampled from $\mathcal{H}$
- Pros and cons
  - Protects against data, environment, and uncertainty overfitting
  - No secrecy required
  - Requires formalizing $\mathcal{H}$ and conducting many trials

Generalized Performance Measures

Example: Averaging Temperatures from Different Scales

The statement "the average of $\langle -32°\mathrm{C}, 130°\mathrm{F}\rangle$ is greater than that of $\langle -10°\mathrm{C}, 100°\mathrm{F}\rangle$" is true but not meaningful: converting the $°\mathrm{F}$ measurements to $°\mathrm{C}$ makes it false.

- Reward scales in reinforcement learning are often arbitrary
- Averages across differently scaled environments can mislead
- Many other performance measures are possible
- The sign test counts how many times one agent outperforms another in a series of matched trials.

Generalized Performance Measures

Example: Averaging Temperatures from Different Scales

The statement "the average of $\langle -32°\mathrm{C}, 130°\mathrm{F} \rangle$ is greater than that of $\langle -10°\mathrm{C}, 100°\mathrm{F} \rangle$" is true but not meaningful: converting the $°\mathrm{F}$ measurements to $°\mathrm{C}$ makes it false.

- Reward scales in reinforcement learning are often arbitrary
- Averages across differently scaled environments can mislead
- Many other performance measures are possible
- The sign test counts how many times one agent outperforms another in a series of matched trials.

Experimental Approach

- Devise an intuitively useful adaptive function approximator
- Show that generalized methodologies can validate it but single-environment methodologies cannot
- Evaluate the methodology, not the learning algorithm

Range-Adaptive Tile Coding

- Tile coding requires knowledge of state value ranges
- Instead, dynamically spread fixed memory over observed values
- When values outside range occur, transplant to a larger range

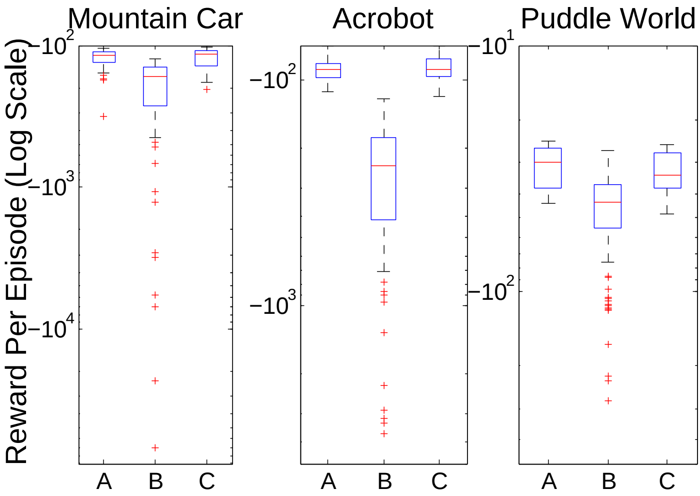
Algorithm 1 TRANSPLANT

```
for i := 0 ... numTiles do
    c := getCenterOfTile(i,oldInputRanges)
    k := getTileForState(c,newInputRanges)
    newWeights[k] := newWeights[k] + oldWeights[i]
    newWeightCounts[k] := newWeightCounts[k] + 1
end for
for i := 0 ... numTiles do
    newWeights[i] := newWeights[i]/newWeightCounts[i]
end for
```

Generalizations and Methods

- Environments:
  - Mountain Car
  - Acrobot
  - Puddle World
- Generalizations:
  - Action effects randomly perturbed
  - Observations scaled, inverted, translated, trigonometric nonlinearities applied
  - Initial state fixed or random
- Methods:
  - **Adaptive (A)**: range-adaptive tile coder
  - **Baseline (B)**: smallest range sufficient for all environments
  - **Cheater (C)**: perfect environment-specific range info
- Each method is tuned to each generalized environment

Generalized Methodology Results



Using the Sign Test

- Tuned agents selected via Copeland's method are the same (except for Puddle World)
- Comparisons between A, B, and C are the same for each generalized environment
- Different story on union task:
  - Cannot distinguish A and C with averaging or sign test metrics
  - Tuned adaptive agent selected via Copeland's method is better

Conclusions

- Generalized methodologies for reinforcement learning
  - Protect against environment overfitting
  - Enable fairer comparisons between agents
  - Make explicit what environment generality is desired
  - Incentivize adaptable algorithms
- Form of methodology depends on purpose of evaluation
  - One-shot settings: secret methodologies protect against uncertainty overfitting
  - Otherwise: open methodologies do not need secrecy
- Performance measure depends on generalized environment
  - Averaging for similar, well-understood environments
  - Sign tests for disparate environments with arbitrary scales

Conclusions

- Generalized methodologies for reinforcement learning
  - Protect against environment overfitting
  - Enable fairer comparisons between agents
  - Make explicit what environment generality is desired
  - Incentivize adaptable algorithms
- Form of methodology depends on purpose of evaluation
  - One-shot settings: secret methodologies protect against uncertainty overfitting
  - Otherwise: open methodologies do not need secrecy
- Performance measure depends on generalized environment
  - Averaging for similar, well-understood environments
  - Sign tests for disparate environments with arbitrary scales

Conclusions

- Generalized methodologies for reinforcement learning
  - Protect against environment overfitting
  - Enable fairer comparisons between agents
  - Make explicit what environment generality is desired
  - Incentivize adaptable algorithms
- Form of methodology depends on purpose of evaluation
  - One-shot settings: secret methodologies protect against uncertainty overfitting
  - Otherwise: open methodologies do not need secrecy
- Performance measure depends on generalized environment
  - Averaging for similar, well-understood environments
  - Sign tests for disparate environments with arbitrary scales