Model-Selection for Non-Parametric Function Approximation: A Case Study in a Smart Energy System

Daniel Urieli   Peter Stone

Department of Computer Science
The University of Texas at Austin
{urieli,pstone}@cs.utexas.edu

ECML 2013

UT Austin Learning Agents Research Group

Motivation

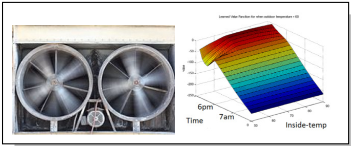
A smart energy problem:

Controlling a thermostat for reducing energy consumption in an HVAC[a] system while maintaining comfort requirements

[a] Heating, Ventilation and Air-Conditioning

General Motivation

Applying value-function based reinforcement learning (RL) to discrete-time, continuous-control problems



UT Austin Learning Agents Research Group

Discrete-Time, Continuous Control Problems

- System's state-space is continuous
- Control actions are taken at discrete times
- Further assuming that action-set is small and discrete
- Examples:


Value-Function based RL

- In theory, value-function based RL can solve such problems optimally
- In practice, it is often unclear how to approximate the value function well enough
- Indeed, recent successes used direct policy search

Value-Function based RL

- In theory, value-function based RL can solve such problems optimally
- In practice, it is often unclear how to approximate the value function well enough
- Indeed, recent successes used direct policy search

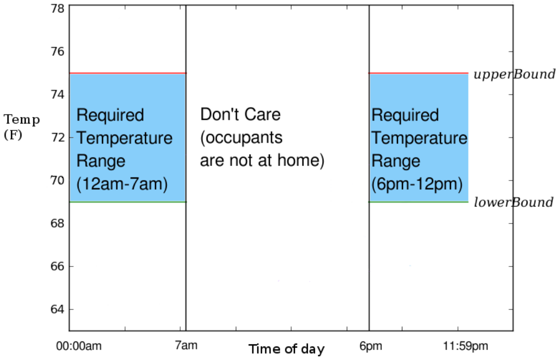
Value-Function based RL

- In theory, value-function based RL can solve such problems optimally
- In practice, it is often unclear how to approximate the value function well enough
- Indeed, recent successes used direct policy search

Value-Function based RL

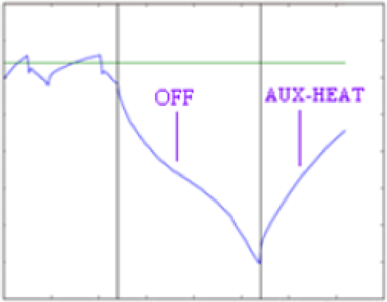
- Still, value-function based RL has desirable advantages:
  - Aiming for global optimum
  - Bootstrapping $\Longrightarrow$ less interactions with the real-world

Case Study: Smart Thermostat Control



- Minimize energy consumption while satisfying this comfort specification

Case Study: Smart Thermostat Control



- Straightforward turn-off strategy fails to satisfy both requirements

Smart Thermostat Control as an MDP

We model the problem as an MDP:

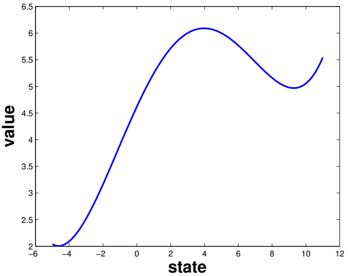
- **S**: $\{\langle T_{in}, T_{out}, Time\rangle\}$
- **A**: $\{\text{COOL}, \text{OFF}, \text{HEAT}, \text{AUX}\}$
- **P**: computed by the simulator, initially unknown
- **R**: $-energyConsumedByLastAction - C_{6pm}$
- **T**: $\{s \in S | s.time == \text{23:59pm}\}$

Plan

For the value-function (VF) approximation part, we need to:

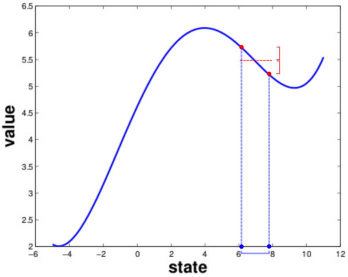
1. Choose a function approximator
2. Choose an algorithm to compute the approximate VF
3. Tune the function approximator's parameters through model-selection

The Challenge of Value-Function Approximation



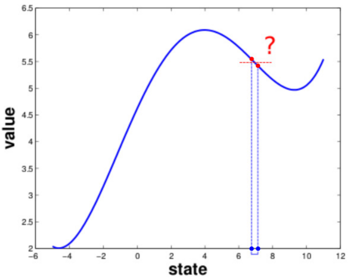
- Must differentiate optimal from suboptimal action
- Non-trivial with "small" action effects + smooth value function $\Longrightarrow$ losses accumulate over time

The Challenge of Value-Function Approximation



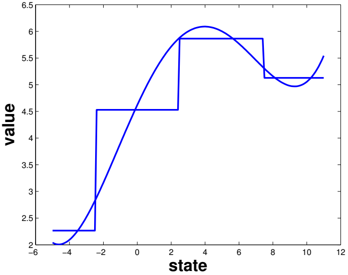
- Must differentiate optimal from suboptimal action
- Non-trivial with "small" action effects + smooth value function $\Longrightarrow$ losses accumulate over time

The Challenge of Value-Function Approximation



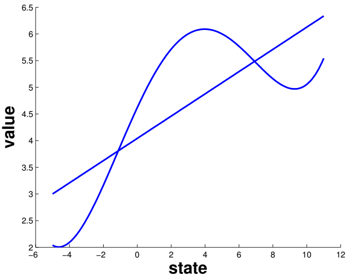
- Must differentiate optimal from suboptimal action
- Non-trivial with "small" action effects + smooth value function $\Longrightarrow$ losses accumulate over time

Function Approximation Methods



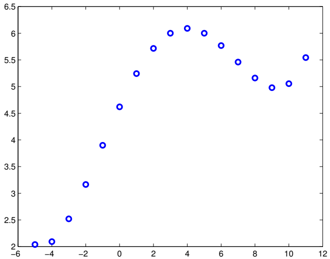
- Discretization
- Suffers from the curse of dimensionality at the required resolution levels

Function Approximation Methods



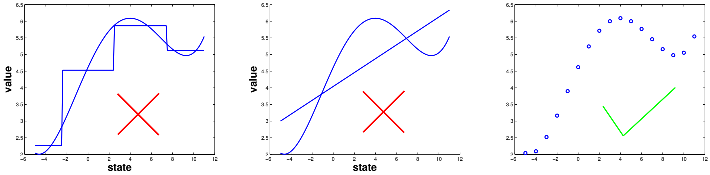
- Linear Function Approximation
- Depends on choosing good features
- Frequently not clear how to do that

Function Approximation Methods



- Non-Parametric: can represent any function
- Using lots of data...

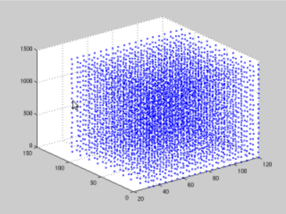
Non-Parametric Value Function Approximation



- To minimize the assumptions about the VF representation we use a smooth, non-parametric function approximator: Locally Weighted Linear Regression (LWR)

Compute an Approximate VF Using FVI

- To compute the approximate VF, we use Fitted Value Iteration (FVI):



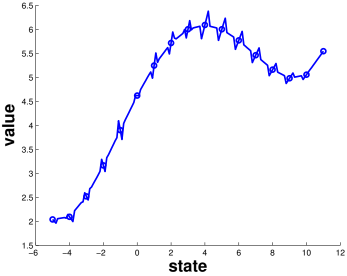
S_FVI $:= \{s^{(1)}, s^{(2)}, \ldots, s^{(m)}\}$

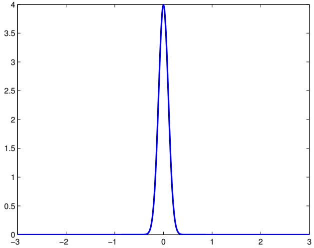
$$
\begin{aligned}
&RepeatUntilConvergence\{\\
&\quad \forall i \in 1, \ldots, m\\
&\quad y^{(i)} := max_a \left(R(s^{(i)}, a) + \gamma E_{[s' \mid s^{(i)} a]} [\hat{V}^{\pi^*}(s')] \right)\\
&\quad \hat{V}^{\pi^*}(s) := LWR \left(\{ \langle s^{(i)}, y^{(i)} \rangle \mid i \in 1, \ldots, m \} \right)\\
&\}
\end{aligned}
$$

UT Austin Learning Agents Research Group

Model-Selection for LWR

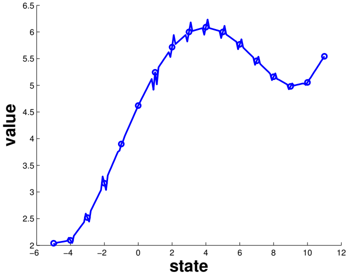
- LWR needs tuning, for instance the kernel bandwidth in 1-d:

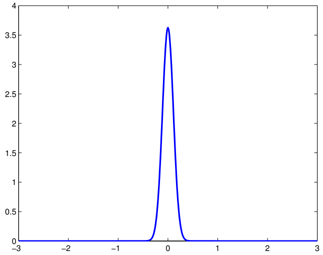




Model-Selection for LWR

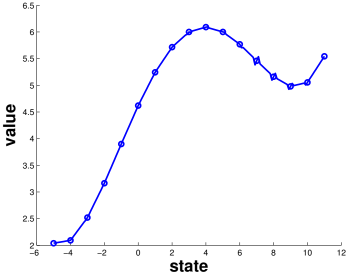
- LWR needs tuning, for instance the kernel bandwidth in 1-d:

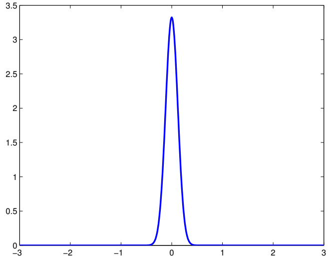




Model-Selection for LWR

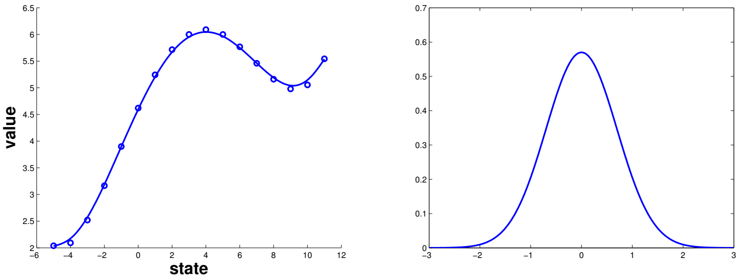
- LWR needs tuning, for instance the kernel bandwidth in 1-d:





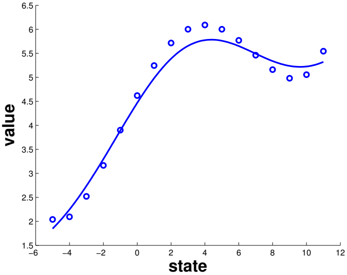
Model-Selection for LWR

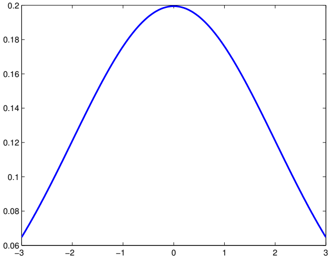
- LWR needs tuning, for instance the kernel bandwidth in 1-d:



Model-Selection for LWR

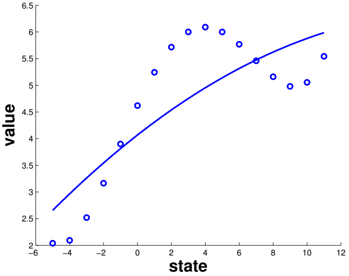
- LWR needs tuning, for instance the kernel bandwidth in 1-d:

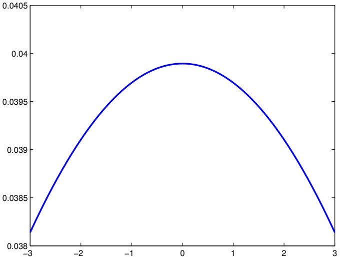




Model-Selection for LWR

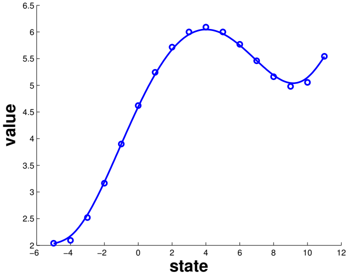
- LWR needs tuning, for instance the kernel bandwidth in 1-d:

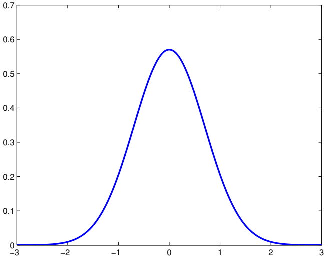




Model-Selection for LWR

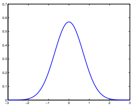
- LWR needs tuning, for instance the kernel bandwidth in 1-d:

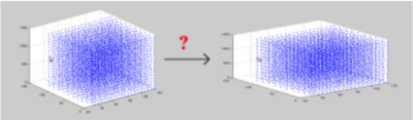




Model-Selection for LWR in N-dimensions

- In N dimensions, it is common to tune N+1 parameters:
  - 1 bandwidth parameter: τ
  - N attribute-scaling parameters: $c_1, \ldots, c_n$



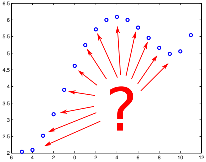


- Tuning these parameters is a form of model-selection

Model Selection - How to Evaluate A Model?

Model-evaluation measure?

- In supervised learning: prediction performance on held-out sets
- In reinforcement learning?



- We don't have the true values (labels) of states

```
for i = 1 → numModels do
    run agent for 1 year with model?
    Record Total Reward
end for
choose best model
```

- Performance is accumulated reward - often too expensive to evaluate

Model Selection - How to Evaluate A Model?

- We use the fact that the optimal value function must satisfy Bellman's optimality equation:

$$\hat{V} \equiv V^{\pi^*} \iff \forall s \in S : BE_{\hat{V}(s)} = 0$$

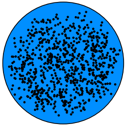
where

$$BE_{\hat{V}}(s) := |\hat{V}(s) - max_a(R(s,a) + \gamma E_{[s'|sa]}[\hat{V}(s')])|$$

- It already holds for $s \in S_{FVI}$ (FVI's convergence condition).
- But not necessarily for $s \notin S_{FVI}$

The Resulting Model Evaluation Measure

- Therefore, to evaluate a model, we:
  1. Sample random states $\mathcal{T} := \{t^{(1)}, ..., t^{(m')}\}$, $t_i \notin S_{FVI}$, $|\mathcal{T}| >> |S_{FVI}|$



  2. Use $||BE_{\hat{V}}(\mathcal{T})||_\infty$ as model evaluation measure
- Model-Selection becomes minimizing

$$F : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$$

where

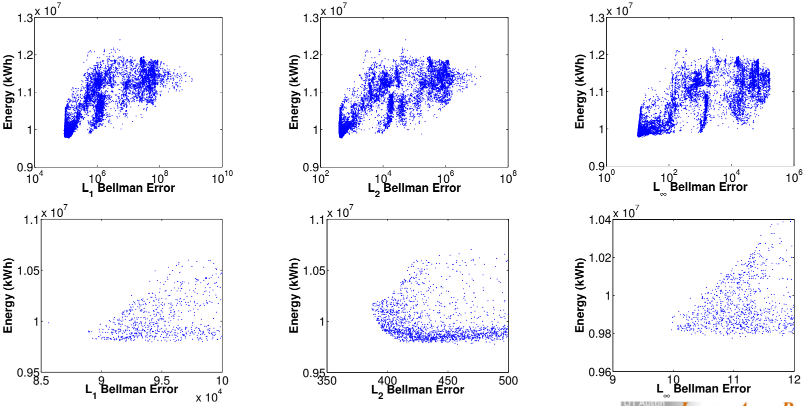
$$(c_1, \ldots, c_n, \tau) \mapsto ||BE_{\hat{V}}(\mathcal{T})||_\infty$$

- No need to evaluate an agent in the environment

Practical Model-Selection: 2 conditions

- To have a practical model-selection algorithm we need to show that:
  1. Bellman Error is correlated with actual performance
  2. Finding the minimum can be done efficiently

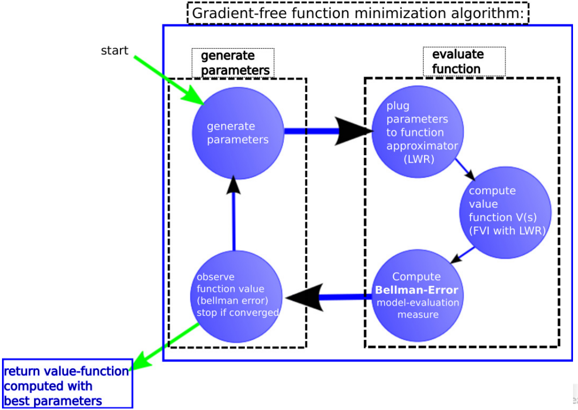
Correlation Between the Bellman Errors and Performance



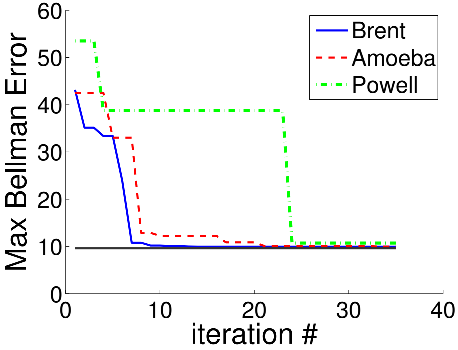
UT Austin Learning Agents Research Group

The MSNP Algorithm

We use these two assumptions to define the following model-selection algorithm, named MSNP:

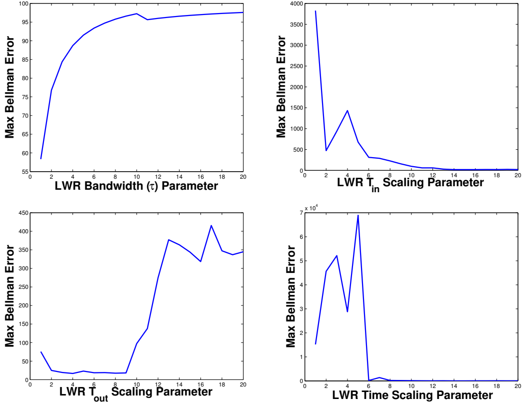


Efficiently Optimizing the Bellman Errors

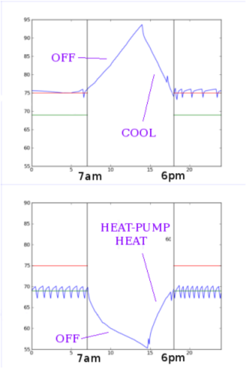


Basins of Convergence of the Max Bellman Error

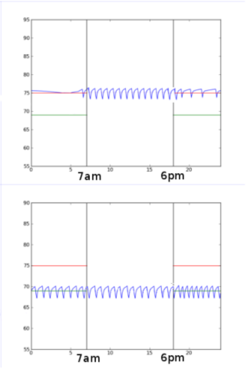
Plotting $p_i \mapsto ||BE_{\hat{V}}(\mathcal{T})||_\infty$, for each $p_i \in \{c_1, c_2, c_3, \tau\}$ (for $j \neq i$, p_j are held fixed at default values)



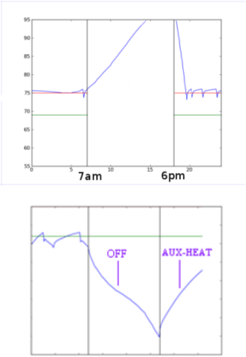
Temperature Graphs



- MSNP



- Default



- Turn-off

Performance of MSNP

- Comparing Yearly energy consumption (lower is better)
- **Default**: default strategy that is deployed in practice
- **MSNP**: our model-selection algorithm is
  1. better than **LargeSample**
  2. close to **CMA-ES**

City	Default (kWh)	LargeSample (kWh)	MSNP **(kWh)**	CMA-ES (kWh)	% Energy-Savings
New York City	11084.8	10923.5	**9859.3**	9816.3	**11.0**%
Boston	12277.1	12480.7	**11433.6**	11052.8	**6.9**%
Chicago	15172.5	14778.2	**14186**	13778.4	**6.5**%

Performance of MSNP

- Comparing Yearly energy consumption (lower is better)
- **Default**: default strategy that is deployed in practice
- **MSNP**: our model-selection algorithm is
  1. better than **LargeSample**
  2. close to **CMA-ES**

City	Default (kWh)	LargeSample (kWh)	MSNP **(kWh)**	CMA-ES (kWh)	% Energy-Savings
New York City	11084.8	10923.5	**9859.3**	9816.3	**11.0**%
Boston	12277.1	12480.7	**11433.6**	11052.8	**6.9**%
Chicago	15172.5	14778.2	**14186**	13778.4	**6.5**%

Performance of MSNP

- Comparing Yearly energy consumption (lower is better)
- **Default**: default strategy that is deployed in practice
- **MSNP**: our model-selection algorithm is
  1. better than **LargeSample**
  2. close to **CMA-ES**

City	Default (kWh)	LargeSample (kWh)	MSNP **(kWh)**	CMA-ES (kWh)	% Energy-Savings
New York City	11084.8	10923.5	**9859.3**	9816.3	**11.0**%
Boston	12277.1	12480.7	**11433.6**	11052.8	**6.9**%
Chicago	15172.5	14778.2	**14186**	13778.4	**6.5**%

Performance of MSNP

- Comparing Yearly energy consumption (lower is better)
- **Default**: default strategy that is deployed in practice
- **MSNP**: our model-selection algorithm is
  1. better than **LargeSample**
  2. close to **CMA-ES**

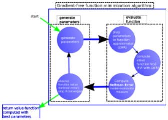
City	Default (kWh)	LargeSample (kWh)	MSNP **(kWh)**	CMA-ES (kWh)	% Energy-Savings
New York City	11084.8	10923.5	**9859.3**	9816.3	**11.0**%
Boston	12277.1	12480.7	**11433.6**	11052.8	**6.9**%
Chicago	15172.5	14778.2	**14186**	13778.4	**6.5**%

Related Work

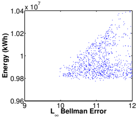
- Bellman error for generalized policy iteration (Antos et al 2008, Lagoudakis and Parr 2003)
- Bellman error for tuning basis functions in linear architectures (Keller et al 2006, Menache et al 2005, Parr et al 2007)
- LWR Model selection for learning a transition-function (Ng et al 2004)
- Abstract model-selection algorithm for RL (Farahmand and Szepesvari 2011)

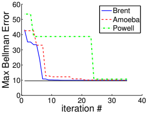
Summary

- Introduced MSNP - practical model selection algorithm for RL



- MSNP is based on two main ideas:





- Value-function based RL for thermostat control
- Outlook
  - Theoretical analysis, Bellman Error's basin of convergence
  - High-dimensional problems