Online Kernel Selection for Bayesian RL

Joseph Reisinger, Peter Stone and Risto Miikkulainen
The University of Texas at Austin
7/6/08
• **Quick summary**: In **Gaussian Process RL**, the choice of kernel is important for performance. How can we choose the kernel efficiently online?
Reinforcement Learning

States  $x \in S$
Actions  $a \in A$
Reward  $r \in \mathbb{R}$

Transition probabilities
$\mathbf{x}_{t+1} \sim p(\cdot|x_t, a_t)$

Want to find a policy $\mu : S \times A \rightarrow [0, 1]$ that maximizes
$V^\mu(x) = \mathbb{E}_\mu[D(x)]$  the “expected discounted return”

$D(x) = \sum_{i=0}^{\infty} \gamma^i R(x_i)|x_0 = x \quad x_{i+1} \sim p^\mu(\cdot|x_i)$

(Sutton and Barto 1998)
Value Function

- Can compute a policy from a value function
- How is the value function represented?
- Generalization without “approximation” is possible!
  ➔ e.g. with Gaussian processes
Gaussian Processes

- Don’t make complexity assumptions without seeing the data. **Nonparametric inference** allows the number of parameters to scale with the size of the data set.

- But what about overfitting? Use a **Bayesian** method: i.e. regularization through the prior.
Gaussian Processes

- Don’t make complexity assumptions without seeing the data. **Nonparametric inference** allows the number of parameters to scale with the size of the data set.

- But what about overfitting? Use a **Bayesian** method: i.e. regularization through the prior.
Gaussian Processes

- Don’t make complexity assumptions without seeing the data. **Nonparametric inference** allows the number of parameters to scale with the size of the data set

- But what about overfitting? Use a **Bayesian** method: i.e. regularization through the prior
Gaussian Processes

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^N$ be the observed (labeled) data.

Assume that the random variables $y$ are distributed

$$y \sim \mathcal{N}(0, K) \quad \text{where} \quad [K]_{ij} \overset{\text{def}}{=} k(x_i, x_j)$$

Using the data, we can infer an unknown value $y^*$ at a test pt $x^*$

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K & k \\ k^\top & \sigma^* \end{bmatrix}\right)$$

where, $k \overset{\text{def}}{=} (k(x_0, x^*), \ldots, k(x_n, x^*))^\top$. This has posterior moments

$$\mathbb{E}[y^* | y] = k^\top K^{-1} y$$

$$\text{Var}[y^* | y] = \sigma^* - k^\top K^{-1} k$$

i.e. we can do prediction given the covariance.

(Rasmussen and Williams 2006)
Gaussian Processes

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^N$ be the observed (labeled) data.

Assume that the random variables $y$ are distributed

$y \sim \mathcal{N}(0, K)$ where $[K]_{ij} \overset{\text{def}}{=} k(x_i, x_j)$

A Gaussian process is completely defined by the data and the kernel

- Straightforward adaptation to RL:
  - value function model
  - sparsification: don’t save all of the data
  - $O(n)$ online updates

(Engel 2005, Rasmussen and Williams 2006)
Gaussian Processes

Properties of learned function depends on the choice of kernel (e.g. smoothness)
Kernel Selection

- Kernel notion is very powerful, defining a metric on $S \times A$
- How can we choose kernels?
- Typically some model selection step, e.g. cross-validation... also Bayesian model-averaging
- ... but these methods weren’t designed to work online
Online Kernel Selection

\( \theta_i \) kernel instantiation \( \theta_i \in \Theta \) the model space

\( \{ \theta_i^{(0)} \}_{i=1}^n \sim p(\theta) \)

Calculate \( \{ w_i^{(t)} \}_{i=1}^n \)

\( w_i \overset{\text{def}}{=} \frac{p(\mathcal{D}|\theta_i^{(t)})p(\theta_i^{(t)})}{\sum_m p(\mathcal{D}|\theta_m^{(t)})p(\theta_m^{(t)})} \)

likelihood of the data given the kernel

simplification: use average reward as a surrogate
Online Kernel Selection

\[ \theta_i \text{ kernel instantiation} \quad \theta_i \in \Theta \text{ the model space} \]

\[ \{ \theta_i^{(0)} \}_{i=1}^{n} \sim p(\theta) \]

Calculate \[ \{ w_i^{(t)} \}_{i=1}^{n} \]

Resample \[ \{ \tilde{\theta}_i^{(t+1)} \}_{i=1}^{n} \]

Dictionary of observations \( \mathcal{D} \) can be “inherited” here, unlike e.g., NEAT+Q
Online Kernel Selection

\( \theta_i \) kernel instantiation  \( \theta_i \in \Theta \) the model space

\( \{ \theta_i^{(0)} \}_{i=1}^n \sim p(\theta) \)

Calculate \( \{ w_i^{(t)} \}_{i=1}^n \)

Resample \( \{ \tilde{\theta}_i^{(t+1)} \}_{i=1}^n \)

Transition kernel
Experimental Setup

- Three methods:
  - **GPSARSA** - standard model-free GPRL + grid search over kernel parameters
  - **RKRL** - SMC kernel selection
  - **EP-RKRL** - RKRL + dictionary of training points is inherited

\[
k(x, x') = \exp \left[ -\frac{||x - x'||^2}{\sigma^2} \right]
\]

\[
k(x, x') = \exp \left[ -\sum_i w_i (x_i - x'_i)^2 \right]
\]
Results: Mountain Car

\[ x = (\dot{x}, x) \in \mathbb{R}^2 \]

\[ a \in \{-1, 0, 1\} \]

100 eps/eval

<table>
<thead>
<tr>
<th>Function</th>
<th>GPSARSA</th>
<th>RKRL</th>
<th>EP-RKRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>-175</td>
<td>-350</td>
<td>-525</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0</td>
<td>-175</td>
<td>-350</td>
</tr>
<tr>
<td>Tanh norm</td>
<td>-700</td>
<td>-525</td>
<td>-350</td>
</tr>
<tr>
<td>Tanh dot</td>
<td>-700</td>
<td>-525</td>
<td>-350</td>
</tr>
</tbody>
</table>

Whiteson (2007): -52
White (2007): -53.92 (± 0.37) 

(Figure from Singh and Sutton, 1996)
Results: Sailboat Steering

$$\mathbf{x} = (\theta, \dot{\theta}, \dot{x}) \in \mathbb{R}^3$$

$$\mathbf{a} \in [-90, 90] \times [-1, 2] \subset \mathbb{R}^2$$

discretized actions
(3 degrees, 0.5 thrust)

1000 steps/episode
Results: Capture Go

$x \in \{-1, 0, 1\}^{25}$

$a \in [0, 25]$
Conclusion

- Introduced an online kernel selection procedure: RKRL
- RKRL significantly improves the performance of Gaussian process reinforcement learning
- RKRL is practical online even with many parameters
Thanks!
Questions?
Acknowledgements

- Thanks to Yaakov and Matthew Taylor, Bryan Silverthorn for helpful discussions.

- This work was supported by an NSF Graduate Research Fellowship and NSF CAREER award #11S-0237699
# Parameters

<table>
<thead>
<tr>
<th>Basic RL Parameters</th>
<th>GPRL Parameters</th>
<th>RKRL Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon = 0.01$</td>
<td>$\sigma = 1.0$</td>
<td>$N = 25$</td>
</tr>
<tr>
<td>$\gamma = 1.0$</td>
<td>$\nu = 0.0$</td>
<td>$\mu = 0.01$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\tau = 0.5$</td>
</tr>
</tbody>
</table>