Learning Linear Models from Demonstration

Clement Gehring

April 27th 2016
10th Barbados Workshop on RL

MIT CSAIL

Demonstrations for fast learning

Learn Fast!

- Demonstrations from mentors provide "useful" data

Possible problems:

- Demonstrations need not cover all actions in all states
- Won't ever get "all the data"

The case of the missing data

- What do I predict when I have seen no examples?

Exploration vs. Exploitation

~~Exploration vs.~~ Exploitation

~~Exploration vs.~~ Exploitation

"Don't leave where the data is":

- e.g., guided policy search[1], DAGGER[2].

[1] Levine and Koltun 2013.

[2] Ross, Gordon, and Bagnell 2010.

~~Exploration vs.~~ Exploitation

"Don't leave where the data is":

- e.g., guided policy search[1], DAGGER[2].

Are those approach the right thing to do?

- They address an important question
- Avoid the question!

[1] Levine and Koltun 2013.

[2] Ross, Gordon, and Bagnell 2010.

Linear Expectation Models

Let $\phi : \mathcal{S} \to \mathbb{R}^n$ be a set of expressive, non-linear features. A linear expectation model for action a is

$$F_a = \arg\min_F \mathbb{E}\left[||F\phi(S_t) - \phi(S_{t+1})|| \mid A_t = a\right],$$

with linear approximation r of the reward:

$$r = \arg\min_r \mathbb{E}[r^\top \phi(S_t) - R_t]$$

Linear Expectation Models

Why?

- Composable models:

$$\mathbb{E}\left[R_{t+2} \mid A_t = a, A_{t+1} = b, S_t = s\right] = r^\top F_b F_a \phi(s)$$

- Computationally efficient (if n is not too big)
- easy to fit

Solving for the models

Example states transition Φ_a, Φ'_a for action a:

$$\Phi_a = \begin{bmatrix} \phi(s_1)^\top \\ \phi(s_2)^\top \\ \vdots \end{bmatrix}$$

$$F_a = \Phi'^\top \Phi (\Phi^\top \Phi + \lambda \mathbb{I})^{-1},$$

Solving for the models

Example states transition Φ_a, Φ'_a for action a:

$$\Phi_a = \begin{bmatrix} \phi(s_1)^\top \\ \phi(s_2)^\top \\ \vdots \end{bmatrix}$$

$$F_a = \Phi'^\top \Phi(\Phi^\top \Phi + \lambda \mathbb{I})^{-1},$$

or, kernel formulation:

$$F_a = \Phi'^\top (\Phi \Phi^\top + \lambda \mathbb{I})^{-1} \Phi$$

The case of the missing data

- What do I predict when I have seen no examples?

Regularized to zero

Predicting zero where no similar states have been seen before. Let $k(x, y) = \phi(x)^\top \phi(y)$, and $\mathbf{X}, y, \mathbf{X}'$ be your sampled data, then predict with:

Regularized to zero

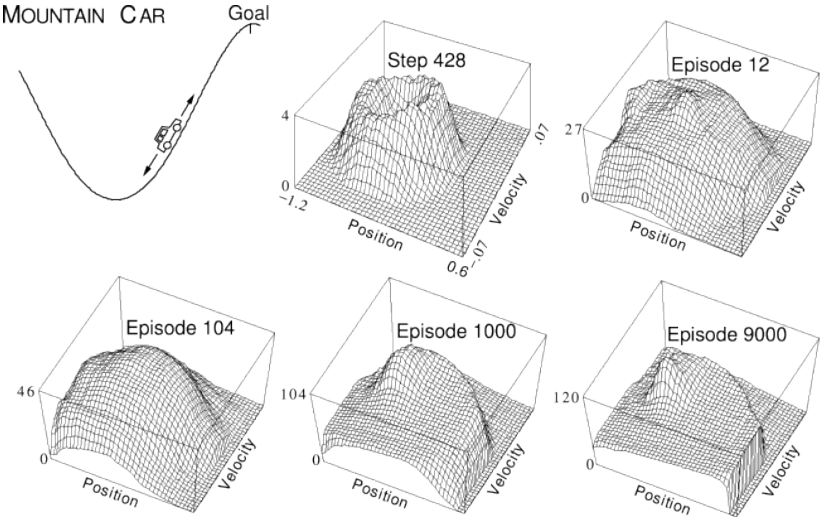
Predicting zero where no similar states have been seen before. Let $k(x, y) = \phi(x)^\top \phi(y)$, and $\mathbf{X}, y, \mathbf{X}'$ be your sampled data, then predict with:

$$D = (k(\mathbf{X}, \mathbf{X}) + \lambda \mathbb{I})^{-1}$$

$$\mathbb{E}\left[R_{t+1} \mid A_t = a, S_t = s\right] = y^\top D\, k(\mathbf{X}, \mathbf{X}') D\, k(\mathbf{X}, s).$$

This observation applies to the explicit feature view too.

Mountain Car[3]



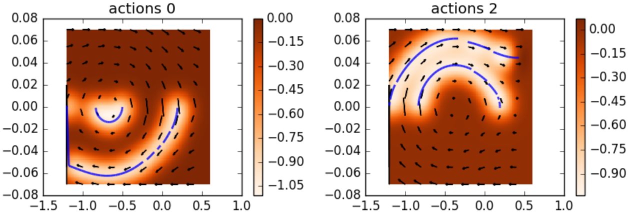
[3]Sutton and Barto 1998.

Reward and behaviour

The reward function will impact how trajectories are perceived depending on proximity to the data.

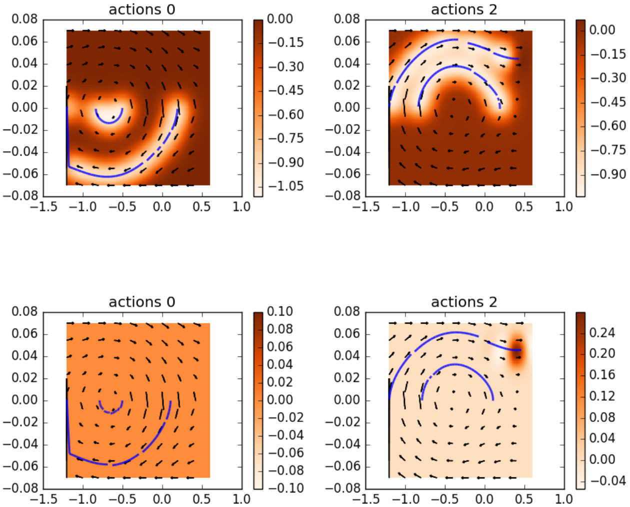
Kernel regression case

Gaussian Kernel



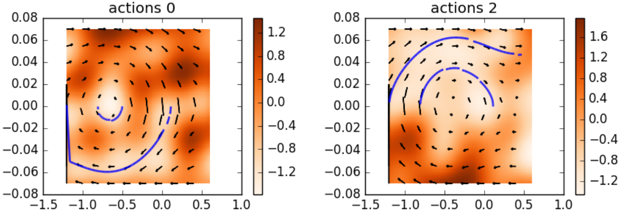
Kernel regression case

Gaussian Kernel



Random Fourier Features[4]

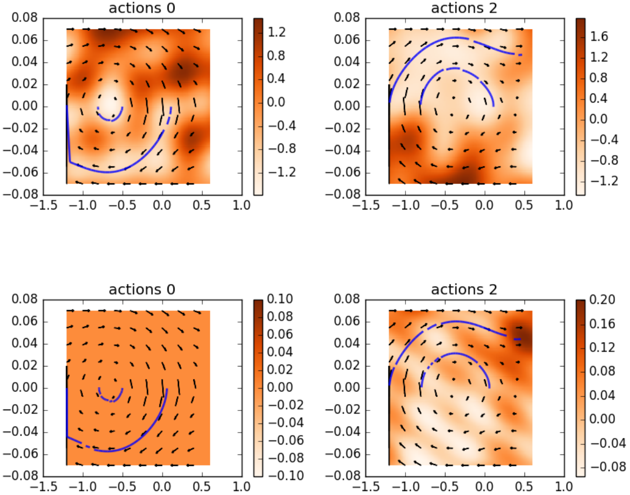
20 Random Fourier Features



[4] Rahimi and Recht 2007.

Random Fourier Features[4]

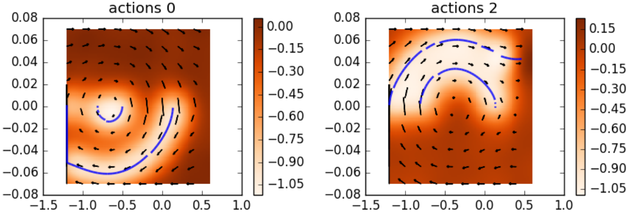
20 Random Fourier Features



[4] Rahimi and Recht 2007.

Random Fourier Features[5], More!

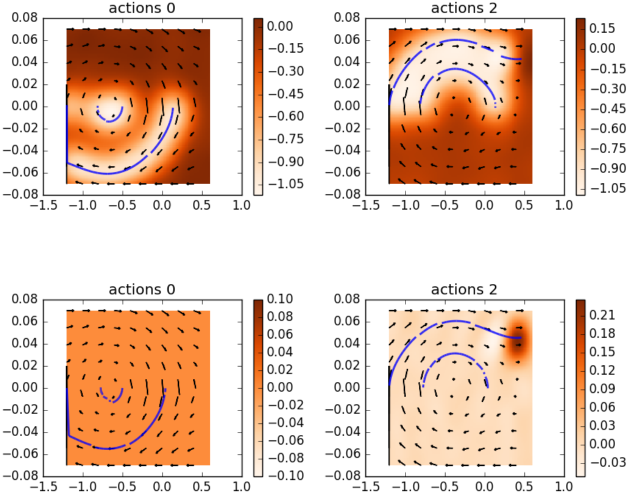
10000 Random Fourier Features



[5]Rahimi and Recht 2007.

Random Fourier Features[5], More!

10000 Random Fourier Features



[5] Rahimi and Recht 2007.

Thank you!

Sergey Levine and Vladlen Koltun. "Guided policy search". In: *Proceedings of The 30th International Conference on Machine Learning*. 2013, pp. 1–9.

Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2007, pp. 1177–1184.

Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning". In: *arXiv preprint arXiv:1011.0686* (2010).

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.