Linear Value Function Approximation and Linear Models

Ronald Parr[⋆] PARR@CS.DUKE.EDU
Gavin Taylor[⋆] GVTAYLOR@CS.DUKE.EDU
Christopher Painter-Wakefield[⋆] PAINT007@CS.DUKE.EDU
Lihong Li[†] LIHONG@CS.RUTGERS.EDU
Michael Littman[†] MLITTMAN@CS.RUTGERS.EDU

[⋆]Department of Computer Science, Duke University, Durham, NC 27708 USA
[†]Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA

1. Introduction

Broadly speaking, there are two approaches to learning value functions for reinforcement learning (RL): model-free and model-based. Model-free approaches typically use samples to learn a value function, while model-based approaches build a model of system dynamics from samples, and the model is used to compute a value function. We summarize recent work showing that the two approaches are equivalent for two classes of closely related RL methods, one based on least-squares approximation and the other kernelized linear approximation. We also briefly discuss a new insight about Bellman residual minimization arising from kernelized RL.

2. Formal Framework and Notation

This work concerns uncontrolled Markov processes, referred to as Markov reward processes (MRPs): $M = (S, P, R, \gamma)$. Given a state $s_i \in S$, the probability of a transition to a state s_j is given by P_{ij} and results in an expected reward of r_i.

We are concerned with finding value functions V that map each state s_i to the expected total γ-discounted reward for the process. In particular, we would like to find or closely approximate the solution to the Bellman equation:

$$V = R + \gamma PV.$$

For any matrix, A, we use A^T to indicate its transpose.

2.1. Linear Value Functions and Models

In cases where the value function cannot be represented exactly, it is common to use some form of parametric value-function approximation, such as a linear combination of features or basis functions:

$$\hat{V} = \sum_{i=1}^{k} w_i \phi_i,$$

where $\Phi = \{\phi_1, \ldots, \phi_k\}$ is a set of linearly independent basis functions of the state. Expressing the weights $\mathbf{w}$ as a column vector, we write $\hat{V} = \Phi\mathbf{w}$.

We can alternatively use features Φ with linear function approximation to predict rewards and *next* features. We define the linear reward model $\hat{R}$ as

$$\hat{R} = \Phi\mathbf{w}_R.$$

We similarly define a linear model $\widehat{\Phi}'$ on our expected next feature values $\Phi' = P\Phi$ as

$$\widehat{\Phi}' = \Phi\mathbf{W}_P.$$

The columns of $\widehat{\Phi}'$ can be viewed as linear predictors for the columns of Φ', with the columns of the $k \times k$ matrix $\mathbf{W}_P$ containing the corresponding approximation weights.

3. Linear Fixed Point Methods and Least-Squares Models

One group of related methods for finding a reasonable value function approximation weight vector $\mathbf{w}$ given Φ and a set of samples include linear TD (Sutton, 1988), LSTD (Bradtke & Barto, 1996) and LSPE (Yu & Bertsekas, 2006). We refer to this family of methods as *linear fixed-point* methods because they all solve for the fixed point

$$\hat{V} = \Phi\mathbf{w}_\Phi = \Pi(R + \gamma\Phi'\mathbf{w}_\Phi),$$

where Π is the L_2 projection operator into $span(\Phi)$. Solving for $\mathbf{w}_\Phi$ yields:

$$\mathbf{w}_\Phi = (\Phi^T\Phi - \gamma\Phi^T\Phi')^{-1}\Phi^T R. \quad (1)$$

The notion that linear fixed-point methods are implicitly computing some sort of model has been recognized in varying degrees for several years. For example, Boyan (1999) considered the intermediate calculations performed by LSTD in some special cases, and interpreted parts of the LSTD algorithm as computing a compressed model.

In recent work (Parr et al., 2008) we show that the linear fixed-point solution for features Φ is *exactly* the solution to

the linear model described by $\hat{R}$ and $\widehat{\Phi}'$ when these models are obtained via least-squares regression, i.e., when $\mathbf{w}_R = (\Phi^T\Phi)^{-1}\Phi^T R$ and $\mathbf{W}_P = (\Phi^T\Phi)^{-1}\Phi^T\Phi'$.

The Bellman equation for our approximate model has the fixed point $\hat{V} = \Phi\mathbf{w}$, where

$$\mathbf{w} = (I - \gamma\mathbf{W}_P)^{-1}\mathbf{w}_R. \tag{2}$$

This is the *linear model solution*. Expanding the expressions $\mathbf{w}_R$ and $\mathbf{W}_P$ in Eq. (2) gives, with simple algebraic manipulation, Eq. (1), completing the equivalence proof.

4. Kernelized Value Function Approximation and Kernel-Based Models

A special case of linear function approximation occurs when our features derive from a *kernel*. A *kernel* is a symmetric function k between two points, and a kernel matrix, $\mathbf{K}$, stores kernel values for all pairs in a dataset with $K_{ij} = K_{ji} = k(\mathbf{x}_i, \mathbf{x}_j)$.

If regularized least-squares regression is re-derived using the kernel trick, we arrive at the dual (kernelized) form of linear least-squares regression (Bishop, 2006),

$$y(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T\left(\mathbf{K} + \mathbf{\Sigma}\right)^{-1}\mathbf{t}, \tag{3}$$

where $\mathbf{t}$ represents the target values of the sampled points, and $\mathbf{k}(\mathbf{x})$ is a column vector with elements $k_i(\mathbf{x}) = k(\mathbf{x}_i, \mathbf{x})$. $\mathbf{\Sigma}$ is a generic regularization term. Frequently $\mathbf{\Sigma} = \lambda\mathbf{I}$, but as in general Tikhonov regression, non-zero off-diagonal terms are possible.

4.1. Equivalence of Kernelized Reinforcement Learning Methods

Taylor and Parr (2009) have formulated a model-based RL algorithm using linear transition and reward model approximations derived from kernelized regression. The value function resulting from these approximate models is

$$\hat{V}(s) = \mathbf{k}(s)^T\left[\left(\mathbf{K} + \mathbf{\Sigma}_R\right) - \gamma\left(\mathbf{K} + \mathbf{\Sigma}_R\right)\left(\mathbf{K} + \mathbf{\Sigma}_P\right)^{-1}\mathbf{K}'\right]^{-1}R, \tag{4}$$

with separate regularization terms $\mathbf{\Sigma}_R$ and $\mathbf{\Sigma}_P$ for the reward model and transition model approximations. Analogous to Φ' in Section 2.1, $\mathbf{K}' = P\mathbf{K}$ is the matrix of *next* kernel values.

For particular choices of regularization parameters $\mathbf{\Sigma}_R$ and $\mathbf{\Sigma}_P$, they show that the value function obtained by solving the approximate model is identical to that of Kernelized LSTD (Xu et al., 2005), and the means returned by Gaussian Process Temporal Difference Learning (Engel et al., 2005) and Gaussian Processes in Reinforcement Learning (Rasmussen & Kuss, 2004).

5. Kernelized Value Function, Linear Fixed Point, and Bellman Residual Minimization

In this section we compare the form of the kernelized value function (Eq. (4)) with the linear fixed point solution (Eq. (1)) and the solution obtained by Bellman residual minimization (BRM). We show that, when using the columns of a kernel matrix $\mathbf{K}$ as our feature set, both the linear fixed point and BRM solutions are instances of the kernelized value function.

The BRM solution is the approximate value function that minimizes the Bellman residual $R + \gamma\Phi'\mathbf{w} - \Phi\mathbf{w}$. Minimizing the Bellman residual in a least squares sense,

$$\begin{aligned}\mathbf{w}_{BRM} &= \operatorname*{argmin}_{\mathbf{w}} \left\|R + \gamma\Phi'\mathbf{w} - \Phi\mathbf{w}\right\|^2 \\ &= \left((\Phi - \gamma\Phi')^T(\Phi - \gamma\Phi')\right)^{-1}(\Phi - \gamma\Phi')^T R,\end{aligned}$$

we can see immediately that the BRM solution is equivalent to least-squares regression on target vector R with features $(\Phi - \gamma\Phi')$. Substituting $\mathbf{K}$ for Φ in the BRM solution, and solving for the value of one state only, we have

$$\begin{aligned}\hat{V}(s) &= \mathbf{k}(s)^T\left((\mathbf{K} - \gamma\mathbf{K}')^T(\mathbf{K} - \gamma\mathbf{K}')\right)^{-1}(\mathbf{K} - \gamma\mathbf{K}')^T R \\ &= \mathbf{k}(s)^T(\mathbf{K} - \gamma\mathbf{K}')^{-1}R,\end{aligned}$$

which is Eq. (4) with $\mathbf{\Sigma}_R = \mathbf{\Sigma}_P = \mathbf{0}$. Note that the final step is allowed because $\mathbf{K} - \gamma\mathbf{K}'$ is square. The linear fixed point solution with the substitution of $\mathbf{K}$ for Φ can similarly be shown to reduce to Eq. (4) with no regularization.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boyan, J. A. (1999). Least-squares temporal difference learning. *ICML-99*.

Bradtke, S., & Barto, A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, *2*.

Engel, Y., Mannor, S., & Meir, R. (2005). Reinforcement learning with Gaussian processes. *Machine Learning-International Workshop then Conference* (pp. 201–208).

Parr, R., Li, L., Taylor, G., Painter-Wakefield, C., & Littman, M. (2008). An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. *International Conference of Machine Learning* (pp. 752–759).

Rasmussen, C. E., & Kuss, M. (2004). Gaussian processes in reinforcement learning. *Advances in Neural Information Processing Systems* (pp. 751–759).

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*.

Taylor, G., & Parr, R. (2009). Kernelized value function approximation for reinforcement learning. *International Conference of Machine Learning*.

Xu, X., Xie, T., Hu, D., & Lu, X. (2005). Kernel least-squares temporal difference learning. *International Journal of Information Technology*, *11*, 54–63.

Yu, H., & Bertsekas, D. (2006). *Convergence results for some temporal difference methods based on least squares* (Technical Report LIDS-2697). MIT.