

Locally Weighted Least Squares Temporal Difference Learning

Matthew Howard and Yoshihiko Nakamura *

Dept. Mechano-informatics
University of Tokyo - Japan

Abstract. This paper introduces locally weighted temporal difference learning for evaluation of a class of policies whose value function is non-linear in the state. Least squares temporal difference learning is used for training local models according to a distance metric in state-space. Empirical evaluations are reported demonstrating learning performance on a number of strongly non-linear value functions, without the need for prior knowledge of features or a specific functional form.

1 Introduction

In recent years, a number of methods for solving optimal control problems have been proposed and applied successfully to a series of problems. Examples include variants on Differential Dynamic Programming (DDP) [1, 2], Iterative Linear Quadratic Regulator design (ILQR) [3, 4] and Path Integral Policy Iteration (PI²) [5]. A common strategy in these approaches is their use of local approximations of the system dynamics and the cost, in order to compute local optimal control laws along a nominal trajectory. This simplifies potentially high-dimensional, non-linear control problems within local regions of the state space, enabling their solution.

A disadvantage of such approaches is their reliance on model information, both in terms of the dynamics of the system, and the cost. In many situations, such information is unavailable, for example, when acting in unstructured environments or interacting with unfamiliar objects. In such situations, a data driven approach is desirable, in which controllers may be selected and evaluated according to the observations available from limited experience.

In this paper, we continue in the spirit of applying local approximation techniques to simplify non-linear optimal control problems. However, rather than relying on model knowledge, we explore the use of model-free reinforcement learning techniques, based on recent advances in Least Squares Temporal Difference learning [6, 7]. Our approach is inspired by local learning techniques [8, 9] that have been used for supervised learning, for example, for modelling the dynamics of robotic systems. In contrast to these approaches, however, here we directly learn the policy evaluation function from data, avoiding sources of error from inaccuracies in the dynamics model. We demonstrate our approach for learning value functions for a number of non-linear problems.

*M. Howard is supported by a JSPS Postdoctoral Fellowship and a Grant-in-Aid for Scientific Research (Kakenhi).

2 Problem Definition

We focus on the learning of state-action value functions for model-free evaluation of control policies in continuous state and action spaces. Denoting the state as $\mathbf{x} \in \mathbb{R}^P$ and the action as $\mathbf{u} \in \mathbb{R}^S$, the policy evaluation problem is to form an estimate of the function

$$Q^\pi(\mathbf{x}_t, \mathbf{u}_t) = \sum_{s=0}^{\infty} \gamma^s j(\mathbf{x}_{s+t}, \mathbf{u}_{s+t}) \quad (1)$$

that predicts the long-term return from state \mathbf{x}_t , when applying command \mathbf{u}_t , and acting according to policy $\mathbf{u} = \pi(\mathbf{x})$ thereafter [10]. Here, $\gamma \in [0, 1)$ is a discount factor and $j(\mathbf{x}, \mathbf{u})$ is the instantaneous cost received at state \mathbf{x} when applying command \mathbf{u} , and the state transitions are dictated by the (discrete time) system dynamics $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$.

We assume that the functions for the dynamics \mathbf{f} and the cost j are unknown, however, by acting in the environment, we may collect data with which to form our estimate of (1). Specifically, we assume that data of the form $D = \{\mathbf{x}_n, \mathbf{u}_n, \bar{\mathbf{x}}_n, j_n\}_{n=1}^N$ are given¹ (or may be collected) where $\bar{\mathbf{x}}_n$ is the state to which the system transitions when command \mathbf{u}_n is applied in state \mathbf{x}_n (i.e., $\bar{\mathbf{x}}_n = \mathbf{f}(\mathbf{x}_n, \mathbf{u}_n)$) and j_n is a measurement of the instantaneous cost of the transition. Note that, no assumption is made about the origin of these samples, thereby allowing on- or off-policy estimation of Q^π .

3 Least Squares Temporal Difference Learning

In order to estimate the state-action value function (1), least squares methods have been suggested [11, 6, 7] based on a linear approximation

$$Q^\pi(\mathbf{x}, \mathbf{u}) \approx \tilde{Q}^\pi(\mathbf{x}, \mathbf{u}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}, \mathbf{u}) \quad (2)$$

where $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^M$ is a vector of features and $\boldsymbol{\theta} \in \mathbb{R}^M$ is a vector of parameters. The features $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u}) = (\phi_1(\mathbf{x}, \mathbf{u}), \dots, \phi_M(\mathbf{x}, \mathbf{u}))^\top$ may be hand-selected for a given application, or consist of generic features (e.g., a set of polynomials, radial basis functions, etc. [11]).

With the linear model (2), the parameter $\boldsymbol{\theta}$ is learnt through a form of temporal difference learning [10] that aims to enforce self-consistency of the estimate based on the data. Specifically, the approximation that minimises the temporal difference error on each sample point, $\delta_n = \tilde{Q}_n^\pi - T^\pi[\tilde{Q}_n^\pi]$, is sought. In vector notation,

$$\boldsymbol{\theta} = \arg \min \|\tilde{\mathbf{Q}}^\pi - T^\pi[\tilde{\mathbf{Q}}^\pi]\| \quad (3)$$

where $\tilde{\mathbf{Q}}^\pi \in \mathbb{R}^N$ is the vector of model predictions (i.e., $\tilde{Q}_n^\pi = \tilde{Q}^\pi(\mathbf{x}_n, \mathbf{u}_n)$) and $T^\pi[\tilde{\mathbf{Q}}^\pi] \in \mathbb{R}^N$ are the model predictions under the Bellman operator

$$T^\pi[Q^\pi(\mathbf{x}, \mathbf{u})] = j(\mathbf{x}, \mathbf{u}) + \gamma Q^\pi(\bar{\mathbf{x}}, \pi(\bar{\mathbf{x}})). \quad (4)$$

¹Note that, if the data D consist of a continuous sequence of transitions (e.g., if D represents a single trajectory), then the subscript n may be replaced with the time index t , and $\bar{\mathbf{x}}_t = \mathbf{x}_{t+1}$. In general, however, this is not required and the individual tuples $\{(\mathbf{x}_1, \mathbf{u}_1, \bar{\mathbf{x}}_1, j_1), \dots, (\mathbf{x}_N, \mathbf{u}_N, \bar{\mathbf{x}}_N, j_N)\}$ may be drawn independently (e.g., if the data consist of (potentially disconnected) subsamples of a trajectory).

Geometrically, the vector of temporal difference errors $\delta = (\delta_1, \dots, \delta_N)^\top$ can be visualised as the black vector in Fig. 1.

As noted in [11], direct minimisation of (3) is difficult in the model-free setting, and experimentally we found it to be unstable in learning. An alternative is to use what is known as the Least Squares Fixed Point (LSFP) approximation [11] that minimises the projection of $T^\pi[\tilde{Q}^\pi]$ onto the column space of $\Phi \in \mathbb{R}^{N \times M}$, where $\Phi_{nm} := \phi_m(\mathbf{x}_n, \mathbf{u}_n)$. That is,

$$\Phi^\top (\tilde{Q}^\pi - T^\pi[\tilde{Q}^\pi]) = \mathbf{0} \quad (5)$$

is sought. Geometrically, this corresponds to the solution for which the red vector in Fig. 1 vanishes.

Substituting the model $\tilde{Q}^\pi = \Phi\theta$, and expanding the Bellman operator $T^\pi[\tilde{Q}^\pi] = \mathbf{j} + \gamma\bar{\Phi}\theta$ (where $\bar{\Phi}_{nm} := \phi_m(\bar{\mathbf{x}}_n, \pi(\bar{\mathbf{x}}_n))$ and $\mathbf{j} = (j_1, \dots, j_N)^\top$) it is straightforward to derive the optimal estimator

$$\theta = (\Phi^\top (\Phi - \gamma\bar{\Phi}))^{-1} \Phi^\top \mathbf{j}. \quad (6)$$

4 Locally Weighted Least Squares Temporal Difference Learning

In this paper, as an alternative to constructing a single, global estimator (6), we instead investigate the use of local learning techniques. Specifically, we compose our estimate of Q^π through a set of K local linear models, $\tilde{Q}_k^\pi(\mathbf{x}, \mathbf{u}) = \phi(\mathbf{x}, \mathbf{u})^\top \theta_k$, each of which is responsible for a local region of the state space.

Each local model is trained according to a weighted version of (3),

$$\theta_k = \arg \min \|\mathbf{W}_k(\tilde{Q}^\pi - T^\pi[\tilde{Q}^\pi])\| \quad (7)$$

where $\mathbf{W}_k \in \mathbb{R}^{N \times N}$ is a diagonal weighting matrix that controls the distribution of errors over the data for the k th model. In the proposed scheme, the weights \mathbf{W}_k are computed according to the distance from the model centre, i.e., $W_{k,nn} = \hat{w}_k(\mathbf{x}_n)$ where $\hat{w}_k(\mathbf{x}_n)$ is a normalised weighting kernel, such as a Gaussian or tricube function [8, 12].

Under the linear model, the LSFP approximation can be found by solving

$$\Phi^\top \mathbf{W}_k (\Phi \theta_k - \mathbf{j} + \gamma \bar{\Phi} \theta_k) = \mathbf{0} \quad (8)$$

from which we obtain the weighted least squares estimator

$$\theta_k = (\Phi^\top \mathbf{W}_k (\Phi - \gamma \bar{\Phi}))^{-1} \Phi^\top \mathbf{W}_k \mathbf{j}. \quad (9)$$

For prediction, we then combine the set of local models into one global model

$$\tilde{Q}^\pi(\mathbf{x}, \mathbf{u}) = \sum_{k=1}^K \hat{w}_k(\mathbf{x}) \tilde{Q}_k^\pi(\mathbf{x}, \mathbf{u}). \quad (10)$$

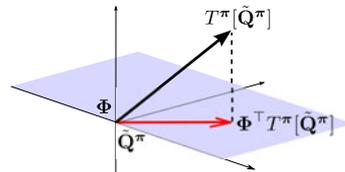


Fig. 1: Orthogonal projection of $T^\pi[\tilde{Q}^\pi]$ onto the column space of Φ (blue area). In the LSFP approximation, the parameter θ is chosen such that the error vector (red) is driven to zero.

Note that, the global model (10) can potentially represent any non-linear function Q^π with appropriate choice of the local models $\tilde{Q}_k^\pi(\mathbf{x}, \mathbf{u})$ (or, more specifically, basis functions $\phi(\mathbf{x}, \mathbf{u})$) [12]. Ideally, to avoid the need for prior knowledge of the form of Q^π , a generic set of models with simple features, such as linear or polynomial functions of \mathbf{x}, \mathbf{u} , is preferable. In our experiments, we investigate learning performance using features forming a second-order polynomial basis.

5 Empirical Evaluation

In this section, we assess learning performance of the proposed approach for problems involving non-linear Q^π , in the absence of prior knowledge of its functional form. For this, we use data sampled from a second-order, discrete time linear dynamic system (with time step $\delta t = 0.02 s$)

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{b}u_t \quad (11)$$

where $\mathbf{x} = (q, \dot{q})^\top$ represents generalised position and velocity, and $u = \tau$ represents a generalised force.

Under these dynamics, non-linearities in Q^π may result from non-linearity either in the control policy or the cost function. To test the robustness of learning, we therefore apply our approach to learning from different combinations of (i) a linear policy $\pi(\mathbf{x}) = -4q - \sqrt{4}\dot{q}$ and (ii) a non-linear policy $\pi(\mathbf{x}) = 4\sin(2q)\cos(2\dot{q})$, and from (i) a quadratic cost function $j(\mathbf{x}, \mathbf{u}) = q^2 + 0.1\dot{q}^2 + 0.1u^2$ and (ii) a non-quadratic cost $j(\mathbf{x}, \mathbf{u}) = \sin(3q)\cos(3\dot{q}) + 0.01u^2$ (both discounted with $\gamma = 0.75$). As illustrated in Fig. 2, the value function associated with these is strongly non-linear in the state.

As training data, we used 750 data points, sampled uniform-randomly across the state-action space $q \sim \mathcal{U}[-1, 1]$, $\dot{q} \sim \mathcal{U}[-1, 1]$, $\tau \sim \mathcal{U}[-10, 10]$, and a further 750 points were reserved as unseen test data. For learning, we used a model consisting of a 5×5 grid of local models with Gaussian weighting functions $\hat{w}(\cdot)$ and widths $\sigma^2 = 0.1$. For evaluation, it should be noted that, even for these relatively simple problems, the true Q^π is not known. We therefore use Monte-Carlo (MC) sampling on these data (using trajectories with $T = 4 s$, i.e., 200 steps) as our ground truth in order to estimate the approximation error. The experiment was repeated for 50 trials on different data sets.

The results are given in Fig. 2 and Table 1. Looking at Fig. 2, we see that there is a close fit between the Q^π estimated with the proposed approach, and the ground truth (MC) estimate. This is confirmed by the figures in Table 1, where we see low normalised mean squared error (NMSE) and low variance across data sets. These levels are comparable to those obtained with least squares regression on the MC data (not reported here). In the case of learning from a linear policy, and quadratic cost function, the NMSE is of the order of machine precision.

To further test the performance, we repeated this experiment with varying quantities of data, and with varying levels of noise. In the case of the latter, the state data $\mathbf{x}_n, \bar{\mathbf{x}}_n$ and the cost data j_n were corrupted with zero-mean Gaus-

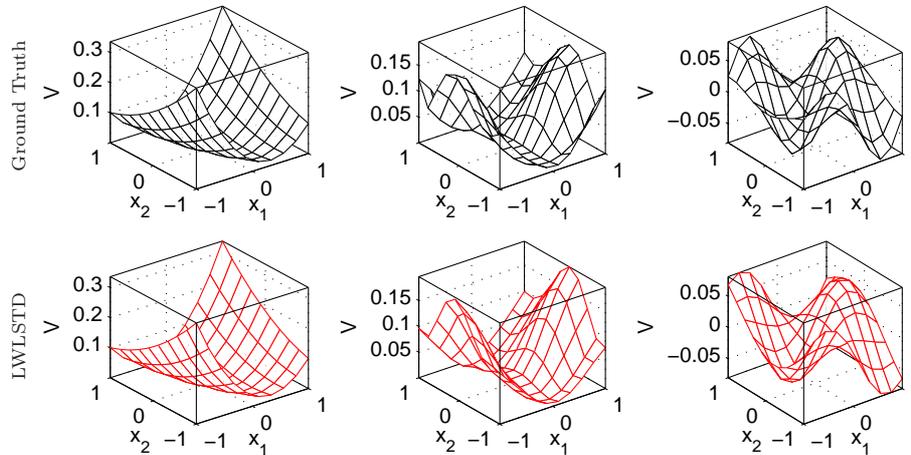


Fig. 2: $V^\pi(\mathbf{x}) = Q^\pi(\mathbf{x}, \pi(\mathbf{x}))$ for different combinations of linear/non-linear policies and quadratic/non-quadratic cost. From left to right: (i) Linear policy, quadratic cost, (ii) linear policy, non-quadratic cost, (iii) non-linear policy, quadratic cost.

Cost	Policy	NMSE	
		Train	Test
quadratic	sinusoidal	0.014 ± 0.001	0.015 ± 0.002
sinusoidal	linear	0.055 ± 0.004	0.057 ± 0.006

Table 1: Normalised mean squared error in predicting non-linear Q^π -functions (mean \pm s.d. over 50 trials).

sian noise with variance proportional to the scale of the data². The results are provided in Fig. 3. As can be seen, increasing amounts of data result in gradually decreasing NMSE, while increasing noise results in a graceful degradation of performance.

6 Conclusion

By exploiting locally weighted learning techniques, this study presents an extension to Least Squares Temporal Difference learning that enables non-linear value functions to be learnt in the model-free setting. Empirical evaluations illustrate the effectiveness of the approach for learning a number of non-linear value functions from data, and characterised the data requirements and susceptibility to noise. Notably, the presented approach avoids the need for hand specification of features based on domain knowledge. This is valuable in the model-free setting, where lack of knowledge about the system dynamics makes specification of such

²Note that, this matches the situation that is likely to be encountered in a real learning situation: information about the state, and the cost are likely to come from noisy sensor data. The \mathbf{u}_n , however, are likely to be exactly known, since these are given from the controller.

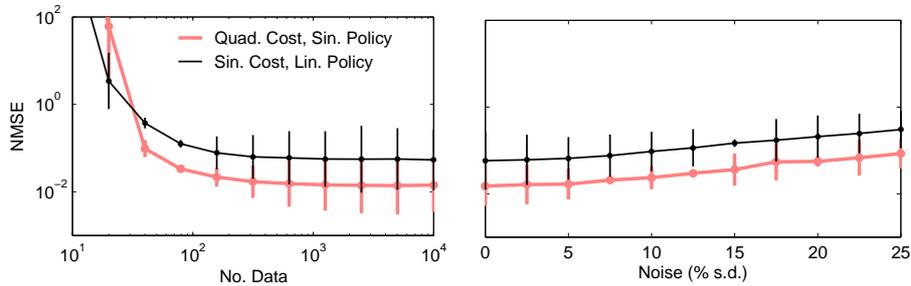


Fig. 3: Effect of varying amounts of data (left) and noise (right). Shown are normalised mean squared error (mean \pm s.d. over 50 trials) on unseen test data.

features non-trivial.

In future work, the extension of this approach to online, incremental learning will be explored, in which streaming data is used to evaluate multiple policies simultaneously. Furthermore, the application of this approach to model-free learning of control policies will also be investigated.

References

- [1] C. G. Atkeson and B. J. Stephens. Random sampling of states in dynamic programming. *IEEE Trans. Sys., Man, Cybernetics*, 38:924–929, 2008.
- [2] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.
- [3] D. Mitrovic, S. Klanke, and S. Vijayakumar. Adaptive optimal feedback control with learned internal dynamics models. In *From Motor Learning to Interaction Learning in Robots*. Springer, 2010.
- [4] W. Li and E. Todorov. Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Int. Conf. Informatics in Control, Automation and Robotics*, 2004.
- [5] E. Theodorou, A. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *J. Machine Learning Research*, 11:3137–3181, 2010.
- [6] J. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.
- [7] S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996.
- [8] S. Vijayakumar, A. D’Souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.
- [9] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1998.
- [10] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [11] M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *J. Machine Learning Research*, 4:1107–1149, 2003.
- [12] W. S. Cleveland and S. J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *J. American Statistical Association*, 83(403):596–610, 1988.