Solving the Linear Bellman Equation via Dual Kernel Embeddings

Yunpeng Pan[1,2], **Xinyan Yan**[1,3], **Bo Dai**[4], **Le Song**[4], **Evangelos Theodorou**[1,2], **and Byron Boots**[1,3]
[1]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology
[2]School of Aerospace Engineering, Georgia Institute of Technology
[3]School of Interactive Computing, Georgia Institute of Technology
[4]School of Computational Science and Engineering, Georgia Institute of Technology
{ypan37,xyan43,bodai,evangelos.theodorou}@gatech.edu, {bboots,lsong}@cc.gatech.edu

Abstract

We introduce a data-efficient approach for solving the linear Bellman equation, which corresponds to a class of Markov decision processes (MDPs) and stochastic optimal control (SOC) problems. We show that this class of control problem can be cast as a stochastic composition optimization problem, which can be further reformulated as a saddle point problem and solved via dual kernel embeddings [1]. Our method is model-free and using only *one sample* per state transition from stochastic dynamical systems. Different from related work such as Z-learning [2, 3] based on temporal-difference learning [4], our method is an online algorithm following the true stochastic gradient. Numerical results are provided, showing that our method outperforms the Z-learning algorithm.

1 Introduction

Richard Bellman's "Principle of Optimality" is central to the theory of optimal control and Markov decision processes (MDPs). This principle is defined by the "Bellman optimality equation". Solving this equation can be very challenging and is known to suffer from the "curse of dimensionality". A more tractable class of control problems or MDPs has been derived based on an exponential transformation of the value function, which results in a linear Bellman equation. In control theory, the exponential transformation of the value function was introduced in [5, 6]. and has been explored in terms of path integral interpretations and theoretical generalizations [7, 8, 9, 10], discrete time formulations [3], and scalable reinforcement learning algorithms [11, 12, 13, 14, 15, 16]. The resulting computational frameworks are known as Path Integral (PI) control for continuous time, Kullback Leibler (KL) control for discrete time, or more generally linearly solvable optimal control [2, 3, 17]. In order to solve the linear Bellman equation, a model-free algorithm called Z-learning [2, 3] was developed and it outperforms the well-known Q-learning algorithm [4] significantly in terms of convergence speed.

Since Z-learning was developed based on the temporal-difference (TD) learning framework [4], it suffers from the limitations of TD learning. In particular, TD learning is not a true gradient descent algorithm and it does not converge to the minimum of the Bellman error in the on-policy setting. This issue has been addressed by introducing a new objective function [18]. However, to the best of our knowledge, a similar issue has not been addressed for solving the linear Bellman equation.

In this paper, we introduce a data-efficient approach for solving the linear Bellman equation via dual kernel embedding [1] and stochastic gradient descent [19]. Our method differs from Z-learning in various ways. In particular, our method updates the solution based on accumulated temporal differences, while Z-learning updates solution based on a single step temporal difference. Our method is data-efficient because it only requires *one sample* per state transition. For continuous state-action problems, our method avoids discretization of the state space [14] or random sampling from a known dynamics model [7], therefore it is scalable in terms of dimension as well. We compared our method to Z-learning [2, 3] and our method showed significant improvement in terms of approximation error.

2 Linear Bellman Equation

In the following we briefly introduce the linearization of the Bellman optimality equation.

2.1 Continuous Time Stochastic Optimal Control

We consider a stochastic optimal control problem in the first-exit setting with state $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ and control $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$. The goal is to construct a control policy $\mathbf{u} = \pi(\mathbf{x})$ that minimize the expected cumulative cost. The problem can be defined as follows

$$\begin{aligned} \underset{\pi}{\text{minimize}} \quad & v^\pi(\mathbf{x}(0)) = \mathbb{E}\bigg[q(\mathbf{x}(T)) + \int_0^T \mathcal{L}(\mathbf{x}(t), \pi(\mathbf{x}(t)))\mathrm{d}t\bigg], \\ \text{subject to} \quad & \mathrm{d}\mathbf{x} = \boldsymbol{\alpha}(\mathbf{x})\mathrm{d}t + \mathbf{B}(\mathbf{x})(\mathbf{u}\mathrm{d}t + \sigma\mathrm{d}\omega), \end{aligned} \tag{1}$$

where ω is Brownian motion. $T = \min(\{t \geq 0 | \mathbf{x} \in \mathcal{A}\})$ where $\mathcal{A}$ is the goal region. $\mathcal{L}$ is the cost rate function, defined as $\mathcal{L}(\mathbf{x}, \mathbf{u}) = q(\mathbf{x}) + \frac{1}{2\sigma^2}||\mathbf{u}||^2$ and $q(\mathbf{x})$ is the state cost. The constraints are dynamics constraints, where $\boldsymbol{\alpha}(\mathbf{x})$ is the drift term and $\mathbf{B}(\mathbf{x})$ is the diffusion matrix with σ a scale parameter. v^π is the value function under policy π. The optimal value function is $v(\mathbf{x}) = \min_\pi v^\pi(\mathbf{x})$ and the optimal controller has form of $\pi^*(\mathbf{x}) = -\sigma^2\mathbf{B}(\mathbf{x})^{\mathrm{T}} v_x(\mathbf{x})$, where v_x is the gradient. The optimal value function v has to satisfy the Hamilton-Jacobi-Bellman (HJB) equation (derivation is omitted):

$$0 = q(\mathbf{x}) + \mathcal{D}[v](\mathbf{x}) - \frac{1}{2} v_{\mathbf{x}}^{\mathrm{T}}(\mathbf{x})\Sigma(\mathbf{x})v_{\mathbf{x}}(\mathbf{x}), \tag{2}$$

and the linear differential operator $\mathcal{D}$ is defined as

$$\mathcal{D}[v](\mathbf{x}) = \boldsymbol{\alpha}(\mathbf{x})^{\mathrm{T}} v_{\mathbf{x}} + \frac{1}{2}\mathrm{tr}\bigg(\Sigma(\mathbf{x})v_{\mathbf{xx}}\bigg), \tag{3}$$

where $\Sigma(\mathbf{x})$ is the noise covariance matrix, defined as $\Sigma(\mathbf{x}) = \sigma^2\mathbf{B}(\mathbf{x})\mathbf{B}(\mathbf{x})^{\mathrm{T}}$. Note that for infinite horizon problems in the average cost-per-step setting, the left-hand side of (2) is the unknown average cost-per-step and v is the differential cost-to-go. It has been discovered that the HJB equation (2) takes a linear form under the exponential transformation $z(\mathbf{x}) = \exp(-v(\mathbf{x}))$. By exponentiating $v(\mathbf{x})$ we get

$$q(\mathbf{x})z(\mathbf{x}) = \mathcal{D}[z](\mathbf{x}). \tag{4}$$

Obviously the above transformed HJB equation is linear. Note that the expoenetial transformation of the value function appears first in [5]. Based on the linearized HJB equation, a class of computational schemes called path integral control has been developed over the last decade [7, 8, 9, 11, 12, 13, 15, 16] in the finite horizon setting.

2.2 Discrete Time Formulation

In the discrete time case, the stochastic dynamics are discretized and therefore $\mathbf{x}(k)$ in discrete time corresponds to $\mathbf{x}(k\mathrm{d}t)$, where k is time step. Let $\mathrm{p}^\pi(\cdot|\mathbf{x})$ denote the transition probability under the controlled dynamics, and $\mathrm{p}(\cdot|\mathbf{x})$ denote the transition probability under the uncontrolled dynamics. Then the cost rate function $\mathcal{L}$ can be formulated as $\mathcal{L}(\mathbf{x}, \mathrm{p}^\pi(\cdot|\mathbf{x})) = q(\mathbf{x})dt + \mathbb{KL}(\mathrm{p}^\pi(\cdot|\mathbf{x})||\mathrm{p}(\cdot|\mathbf{x}))$. The distribution under the optimal control law is $\mathrm{p}^{\pi*}(\mathbf{y}|\mathbf{x}) = \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})z(\mathbf{y})}{\mathcal{G}[z](\mathbf{x})}$ [3]. The term $\mathcal{G}$ is a linear integral operator defined as $\mathcal{G}[z](\mathbf{x}) = \int \mathrm{p}(\mathbf{y}|\mathbf{x})z(\mathbf{y})\mathrm{d}\mathbf{y}$. The minimized Bellman equation [3] can now be exponentiated and expressed in terms of z as follow

$$\exp\big(\mathrm{d}t q(\mathbf{x})\big)z(\mathbf{x}) = \mathcal{G}[z](\mathbf{x}). \tag{5}$$

The above equation is called the *linear Bellman equation*. Note that the min operator has been dropped and the solution to the linear Bellman equation is called the optimal *desirability* function.

2.3 Relationship between the Continuous and Discrete Cases

To make the connection with the continuous case we represent the passive dynamics $p(\mathbf{y}|\mathbf{x})$ as

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{x} + \mathrm{d}t\mathbf{a}(\mathbf{x}), \mathrm{d}t\boldsymbol{\Sigma}(\mathbf{x})). \tag{6}$$

Consider the generator of a stochastic process: $\lim_{\mathrm{d}t\to 0} \frac{\mathbb{E}[z(\mathbf{y})|\mathbf{y}(0)=\mathbf{x}]-z(\mathbf{x})}{\mathrm{d}t} = \mathcal{D}[z](\mathbf{x})$. Since $\mathcal{G}[z](\mathbf{x}) = \mathbb{E}[z(\mathbf{y})|\mathbf{y}(0)=\mathbf{x}]$ we will have

$$\mathcal{G}[z](\mathbf{x}) = z(\mathbf{x}) + \mathrm{d}t\mathcal{D}[z](\mathbf{x}) + o(\mathrm{d}t^2). \tag{7}$$

Substitute (7) into (5) results in (4). By solving the linear Bellman equation (5), we obtain the optimal z-function. The optimal control is computed as $\mathbf{u}^* = -\sigma^2\mathbf{B}(\mathbf{x})^T v_{\mathbf{x}}(\mathbf{x}) = \sigma^2\mathbf{B}(\mathbf{x})^T \frac{z_{\mathbf{x}}(\mathbf{x})}{z(\mathbf{x})}$, which is the solution to the original problem (1). In the next section we introduce a reformulation of the linear Bellman equation.

3 Problem Reformulation and Kernel Embedding

It can be seen in (5) that the optimal control problem has been reduced to a linear eigenvalue problem. This eigenvalue problem can be solved by various methods such as power iteration if the model of the MDP is available [2]. In this work we consider a model-free setting and all we have access to are samples $(\mathbf{x}, \mathbf{y}, q(\mathbf{x}))$. We parameterize the z-function as a function in the *reproducing kernel Hilbert space* (RKHS), i.e., $z \in \mathcal{Z}$ where $\mathcal{Z}$ is a RKHS induced by a kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{Z}}$ where $\phi(\cdot)$ is the feature map. Therefore we have the reproducing property $\langle z, \phi(\mathbf{x}) \rangle_{\mathcal{Z}} = z(\mathbf{x}), \forall z \in \mathcal{Z}$. First we rewrite Eq. 5 as

$$z(\mathbf{x}) = \exp\big(-\mathrm{d}tq(\mathbf{x})\big) \int \mathrm{p}(\mathbf{y}|\mathbf{x}) z(\mathbf{y}) \mathrm{d}\mathbf{y} = \mathbb{E}_{\mathbf{y}|\mathbf{x}}\Big[\exp\big(-\mathrm{d}tq(\mathbf{x})\big) z(\mathbf{y})\Big]. \tag{8}$$

Note that the above expression coincides with the well-known *Feynman-Kac formula* with a single time step. Next we formulate the problem as minimizing the mean-square error

$$\begin{aligned} & \min_{z\in\mathcal{Z}} \mathbb{E}_{\mathbf{x}}\left[\left(z(\mathbf{x}) - \mathbb{E}_{\mathbf{y}|\mathbf{x}}\Big[\exp\big(-\mathrm{d}tq(\mathbf{x})\big) z(\mathbf{y})\Big]\right)^2\right] \\ \Longrightarrow \quad & \min_{z\in\mathcal{Z}} \mathbb{E}_{\mathbf{x}}\Bigg[\underbrace{\left(\mathbb{E}_{\mathbf{y}|\mathbf{x}}\Big[\underbrace{z(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big) z(\mathbf{y})}_{\xi(\mathbf{x},\mathbf{y})}\Big]\right)^2}_{f(\xi)}\Bigg] \end{aligned} \tag{9}$$

The above nested expectation minimization problem corresponds to compositional stochastic programming [1, 20], an emerging topic in the field of machine learning. Since $f(\xi)$ is convex, we exploit [1] that uses the Fenchel dual representation of (9) resulting in a saddle point problem. Specifically, let ψ be the dual variable, and $f^*(\psi) = \max_\xi\{\psi\xi - f(\xi)\} = \frac{1}{2}\psi^2$ be the Fenchel conjugate of $f(\xi) = \xi^2$. The problem can be reformulated as

$$\begin{aligned} & \min_{z\in\mathcal{Z}} \mathbb{E}_{\mathbf{x}}\left[\max_{\psi}\left[\mathbb{E}_{\mathbf{y}|\mathbf{x}}\Big[z(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big) z(\mathbf{y})\Big] \cdot \psi(\mathbf{x}) - f^*\big(\psi(\mathbf{x})\big)\right]\right] \\ \Longrightarrow \quad & \min_{z\in\mathcal{Z}} \max_{\psi\in\mathcal{S}(\mathcal{X})} \mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[\big(z(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big) z(\mathbf{y})\big) \cdot \psi(\mathbf{x})\Big] - \frac{1}{2}\mathbb{E}_{\mathbf{x}}\Big[\big(\psi(\mathbf{x})\big)^2\Big] \end{aligned} \tag{10}$$

The above objective function is concave in the dual variable ψ for fixed z, and convex in z for fixed ψ. Therefore it is a convex-concave saddle point problem. Note that in (10) we switched the $\max$ and $\mathbb{E}_{\mathbf{x}}$ operators, which seems problematic. However, since the dual variable ψ is a function over $\mathcal{X}$, this reformulation can be justified. If $\mathrm{p}(\mathbf{y}|\mathbf{x})$ is continuous in $\mathbf{y}$ for any $\mathbf{x}$, then the optimal dual function is unique and continuous [1]. Therefore we approximate the dual function space $\mathcal{S}(\mathcal{X})$ by a RKHS $\mathcal{H}$ induced by kernel $\tilde{k}(\mathbf{x}, \mathbf{y}) = \langle \tilde{\phi}(\mathbf{x}), \tilde{\phi}(\mathbf{y}) \rangle_{\mathcal{H}}$. We employ the dual kernel embedding [1] and rewrite the saddle point problem as

$$\min_{z\in\mathcal{Z}} \max_{\psi\in\mathcal{H}} J(z,\psi) = \mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[\Big\langle z, \phi(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big)\phi(\mathbf{y})\Big\rangle_{\mathcal{Z}} \cdot \langle \psi, \tilde{\phi}(\mathbf{x})\rangle_{\mathcal{H}}\Big] - \frac{1}{2}\mathbb{E}_{\mathbf{x}}\Big[\big(\psi(\mathbf{x})\big)^2\Big]. \tag{11}$$

Given a sample triple $(\mathbf{x}, \mathbf{y}, q(\mathbf{x}))$, we can construct an unbiased stochastic estimate of the objective function's gradient $\widehat{\nabla}_z J(z,\psi) = \big(\phi(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big)\phi(\mathbf{y})\big)\psi(\mathbf{x})$ and $\widehat{\nabla}_\psi J(z,\psi) = z(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big)z(\mathbf{y})\big)\phi(\mathbf{x}) - \psi(\mathbf{x})$, respectively. Based on these gradients we can solve (11) by stochastic gradient descent algorithm[19].

4 Proposed Algorithm

Based on the saddle-point formulation, we perform stochastic gradient descent given samples from joint distribution $(\mathbf{x}, \mathbf{y}) \sim \mathrm{p}(\mathbf{x}, \mathbf{y})$. At each iteration, we only need one sample for each conditional distribution $\mathrm{p}(\mathbf{y}|\mathbf{x})$. This is particularly useful when solving continuous state-action problems since most transition samples are collected from trajectories. In this case only one sample is available for each conditional distribution. At iteration $i+1$, we sample a triple $(\mathbf{x}, \mathbf{y}, q(\mathbf{x}))$, and update the primal and dual function as follow

$$\begin{aligned} z^{i+1} &= z^i - \gamma^i \Big(\exp\big(-\mathrm{d}tq(\mathbf{x})\big) k(\mathbf{y}, \cdot) - k(\mathbf{x}, \cdot) \Big) \psi^i(\mathbf{x}), \\ \psi^{i+1} &= \psi^i + \gamma^i \Big(z^i(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big) z^i(\mathbf{y}) - \psi^i(\mathbf{x}) \Big) \tilde{k}(\mathbf{x}, \cdot). \end{aligned} \tag{12}$$

where γ^i is the learning rate and $i = 1, ..., t$. Under the assumption that the variance of stochastic gradient estimate is bounded, the algorithm features $O(\frac{1}{\sqrt{t}})$ convergence rate [1], which is unimprovable for traditional stochastic optimization with general convex loss function [19]. In contrast, the Z-learning algorithm [2, 3] performs the following functional update

$$z^{i+1} = z^i - \gamma^i \Big(z^i(\mathbf{x}) - \exp\big(-\mathrm{d}tq(\mathbf{x})\big) z^i(\mathbf{y}) \Big) k(\mathbf{x}, \cdot). \tag{13}$$

Note that we parameterize the z-function in RKHS for Z-learning in order to make comparison. Intuitively, the above Z-learning algorithm updates the solution based on the temporal difference of one triple $(\mathbf{x}, q(\mathbf{x}), \mathbf{y})$. While our method (12) updates the solution based on the temporal differences accumulated over i iterations. Note that this algorithm can be easily extended to finite-basis function parameterization of the z-function, e.g., approximating the kernel function with random features [21, 22, 1]. If we parameterize the z-function with finite basis functions for both methods, i.e., $z(\mathbf{x}) = \alpha^{\mathsf{T}} \varphi_{\mathbf{x}}$ where $\alpha, \varphi_{\mathbf{x}} \in \mathbb{R}^N$, the relationship between our method and Z-learning can be interpreted by the relationship between the gradient-TD2 [18] and conventional TD algorithm [4] in the on-policy setting, even though we are solving a fundamentally different problem here.

5 Numerical Example

In order to test the proposed algorithm, we focus on an inverted pendulum swing-up and balancing task, which is a continuous state-action problem. Let $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2]$. The system has passive dynamics $\boldsymbol{\alpha}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 \\ \sin(\mathbf{x}_2) \end{bmatrix}$. The state cost function $q(\mathbf{x}) = (\mathbf{x}_1 - \pi)^2 + \mathbf{x}_2^2$. We used 3000 random training samples in the region $0 \leq \mathbf{x}_1 \leq 2\pi$ and $-8 \leq \mathbf{x}_2 \leq 8$. The testing samples were drawn from the same region. For comparison we applied Todorov's Z-learning algorithm [2, 3] with the same z-function parameterization as in our method. Results are shown in Fig.1. Our method significantly outperforms Z-learning in terms of approximation error. As mentioned in the last section, our method updates the solution based on accumulated temporal differences while Z-learning performs update using one-step temporal difference.

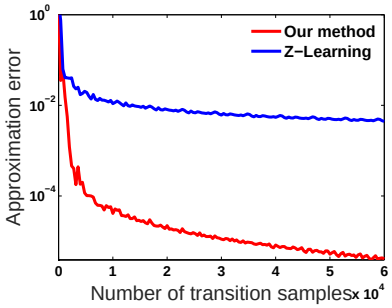

Figure 1: Learning performance. The approximation error defined in (9) is a measure of how closely the approximated z-function satisfies the linear Bellman equation.

6 Conclusion

We have presented a stochastic gradient descent algorithm for solving the linear Bellman equation, which is central to the theory of linearly solvable optimal control [2, 3], path integral control [7], etc. We reformulate the control problem as a stochastic composite optimization problem, and solve it by leveraging dual kernel embeddings [1] and stochastic gradient descent algorithm [19]. Our method is data-efficient because only *one sample* per conditional distribution of state transition is needed to estimate the unbiased stochastic gradient at each iteration. Compared to Z-learning [2, 3], our algorithm converges to better solutions. Future work will focus on using the proposed method to solve challenging control problems based on trajectory rollouts.

References

[1] Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual kernel embeddings. *arXiv preprint arXiv:1607.04579*, 2016.

[2] Emanuel Todorov. Linearly-solvable markov decision problems. In *Advances in neural information processing systems*, pages 1369–1376, 2006.

[3] E. Todorov. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483, 2009.

[4] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[5] W.H. Fleming. Exit probabilities and optimal stochastic control. *Applied Math. Optim*, 9:329–346, 1971.

[6] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*. Applications of mathematics. Springer, New York, 1st edition, 1993.

[7] H. J. Kappen. Linear theory for control of nonlinear stochastic systems. *Phys Rev Lett*, 95:200–201, 2005.

[8] H. J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 11:P11011, 2005.

[9] H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. *AIP Conference Proceedings*, 887(1), 2007.

[10] S. Thijssen and H. J. Kappen. Path integral control and state-dependent feedback. *Phys. Rev. E*, 91:032104, Mar 2015.

[11] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 11:3137–3181, 2010.

[12] F. Stulp and O. Sigaud. Path integral policy improvement with covariance matrix adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 281–288. ACM, 2012.

[13] K. Rawlik, M. Toussaint, and S. Vijayakumar. Path integral control by reproducing kernel hilbert space embedding. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 1628–1634, 2013.

[14] Yunpeng Pan and Evangelos A Theodorou. Nonparametric infinite horizon kullback-leibler stochastic control. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–8. IEEE, 2014.

[15] V. Gómez, H.J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Machine Learning and Knowledge Discovery in Databases*, pages 482–497. Springer, 2014.

[16] Yunpeng Pan, Evangelos Theodorou, and Michail Kontitsis. Sample efficient path integral control under uncertainty. In *Advances in Neural Information Processing Systems*, pages 2314–2322, 2015.

[17] K. Dvijotham and E Todorov. Linearly solvable optimal control. *Reinforcement learning and approximate dynamic programming for feedback control*, pages 119–141, 2012.

[18] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009.

[19] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[20] Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, pages 1–31, 2014.

[21] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[22] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M. Balcan, and L. Song. Scalable kernel methods via doubly stochastic gradients. In *NIPS*, 2014.