Learning Sparse Representations in Reinforcement Learning with Sparse Coding

Lei Le*  
Dept. of Computer Science  
Indiana University  
Bloomington, IN, USA  
leile@indiana.edu

Raksha Kumaraswamy*  
Dept. of Computer Science  
Indiana University  
Bloomington, IN, USA  
rakkumar@indiana.edu

Martha White  
Dept. of Computer Science  
Indiana University  
Bloomington, IN, USA  
martha@indiana.edu

Abstract
A variety of representation learning approaches have been investigated for reinforcement learning; much less attention, however, has been given to investigating the utility of sparse coding. Outside of reinforcement learning, sparse coding representations have been widely used, with non-convex objectives that result in discriminative representations. In this work, we develop a supervised sparse coding objective for policy evaluation. Despite the non-convexity of this objective, we prove that all local minima are global minima, making the approach amenable to simple optimization strategies. We empirically show that it is key to use a supervised objective, rather than the more straightforward unsupervised sparse coding approach. We compare the learned representations to a canonical fixed sparse representation, called tile-coding, demonstrating that the sparse coding representation outperforms a wide variety of tile-coding representations.

1 Introduction
For tasks with large state or action spaces, where tabular representations are not feasible, reinforcement learning algorithms typically rely on function approximation. Whether they are learning the value function, policy or models, the success of function approximation techniques hinges on the quality of the representation. Typically, representations are hand-crafted, with some common representations including tile-coding, radial basis functions, polynomial basis functions and Fourier basis functions [Sutton 1996; Konidaris et al. 2011]. Automating feature discovery, however, alleviates this burden and has the potential to significantly improve learning.

Representation learning techniques in reinforcement learning have typically drawn on the large literature in unsupervised and supervised learning. Common approaches include feature selection, including $\ell_1$ regularization on the value function parameters [Loth et al. 2007; Kolter and Ng 2009; Nguyen et al. 2013] and matching pursuit [Parr et al. 2008; Painter-Wakefield and Parr 2012]; basis-function adaptation approaches [Menache et al. 2005; Whiteson et al. 2007]; instance-based approaches, such as locally weighted regression [Atkeson and Morimoto 2003], sparse distributed memories [Ratitch and Precup 2004], proto-value functions [Mahadevan and Maggioni 2007] and manifold learning techniques [Mahadevan 2009]; and neural network approaches, including more standard feedforward neural networks [Coulom 2002; Riedmiller 2005; Mnih et al. 2015] as well as random representations [Sutton and Whitehead 1993], linear threshold unit search [Sutton and Barto 2013], and evolutionary algorithms like NEAT [Stanley and Miikkulainen 2002].

Surprisingly, however, there has been little investigation into using sparse coding for reinforcement learning. Sparse coding approaches have been developed to learn MDP models for transfer learning [Ammar et al. 2012]; outside this work, however, little has been explored. Nonetheless, such sparse coding representations have several advantages, including that they naturally enable local models, are computationally efficient to use, are much simpler to train than more complicated models such as neural networks and are biologically motivated by the observed representation in the mammalian cortex [Olshausen and Field 1997].

In this work, we develop a principled sparse coding objective for policy evaluation. In particular, we formulate a joint optimization over the basis and the value function parameters, to provide a supervised sparse coding objective where the basis is informed by its utility for prediction. We highlight the importance of using the Bellman error or mean-squared return error for this objective, and discuss how the projected Bellman error is not suitable. We then show that, despite being a non-convex objective, all local minima are global minima, under minimal conditions. We avoid the need for careful initialization strategies needed for previous optimality results for sparse coding [Agarwal et al. 2014; Arora et al. 2015], using recent results for more general dictionary learning settings [Haefele and Vidal 2015] Le and White 2017], particularly by extending beyond smooth regularizers using $\Gamma$-convergence. Using this insight, we provide a simple alternating proximal gradient algorithm and demonstrate the utility of learning supervised sparse coding representations versus unsupervised sparse coding and a variety of tile-coding representations.

2 Background
In reinforcement learning, an agent interacts with its environment, receiving observations and selecting actions to maxi-
ize a scalar reward signal provided by the environment. This interaction is usually modeled by a Markov decision process (MDP). An MDP consists of $(S, A, P, R)$ where $S$ is the set of states; $A$ is a finite set of actions; $P : S \times A \times S \to [0, 1]$, the transition function, which describes the probability of reaching a state $s'$ from a given state and action $(s, a)$; and finally the reward function $R : S \times A \times S \to \mathbb{R}$, which returns a scalar value for transitioning from state-action $(s, a)$ to state $s'$. The state of the environment is said to be Markov if $P_r(s_{t+1}|s_t, a_t) = P_r(s_{t+1}|s_t, a_t, \ldots, s_0, a_0)$.

One important goal in reinforcement learning is policy evaluation: learning the value function for a policy. A value function $V^\pi : S \to \mathbb{R}$ approximates the expected return. The return $G_t$ from a state $s_t$ is the total discounted future reward, discounted by $\gamma \in [0, 1)$, for following policy $\pi : S \times A \to [0, 1]$,

$$G_t = \sum_{i=0}^{\infty} \gamma^i R_{t+i+1} = R_{t+1} + \gamma G_{t+1}$$

where $V^\pi(s_t)$ is the expectation of this return from state $s_t$. This value function can also be thought of as a vector of values $V^\pi \in \mathbb{R}^{|S|}$ satisfying the Bellman equation

$$V^\pi = r^\pi + \gamma P^\pi V^\pi$$

(1)

where $P^\pi(s, a') = \sum_{a \in A} P(s, a, a') \pi(s, a)$.

$$r^\pi(s) = \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s, a, s') R(s, a, s')$$

Given the reward function and transition probabilities, the solution can be analytically obtained: $V^\pi = (I - \gamma P^\pi)^{-1} r^\pi$.

In practice, however, we likely have a prohibitively large state space. The typical strategy in this setting is to use function approximation to learn $V^\pi(s)$ from a trajectory of samples: a sequence of states, actions, and rewards $s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, \ldots$, where $s_0$ is drawn from the start-state distribution, $s_{t+1} \sim P(\cdot | s_t, a_t)$ and $a_t \sim \pi(\cdot | s_t)$. Commonly, a linear function is assumed, $V^\pi(s) \approx \phi(s) \cdot w$ for $w \in \mathbb{R}^k$ a parameter vector and $\phi : S \to \mathbb{R}^k$ a feature function describing states. With this approximation, however, typically we can no longer satisfy the Bellman equation in (1), because there may not exist a $w$ such that $\Phi w$ equals $r^\pi + \gamma P^\pi \Phi w$ for $\Phi \in \mathbb{R}^{|S|} \times k$. Instead, we focus on minimizing the error to the true value function.

Reinforcement learning algorithms, such as temporal difference learning and residual gradient, therefore focus on finding an approximate solution to the Bellman equation, despite this representation issue. The quality of the representation is critical to accurately approximating $V^\pi$ with $\Phi w$, but also balancing compactness of the representation and speed of learning. Sparse coding, and sparse representations, have proven successful in machine learning and in reinforcement learning, particularly as fixed bases, such as tile coding, radial basis functions and other kernel representations. A natural goal, therefore, and the one we explore in this work, is to investigate learning these sparse representations automatically.

3 Sparse Coding for Reinforcement Learning

In this section, we formalize sparse coding for reinforcement learning as a joint optimization over the value function parameters and the representation. We introduce the true objective over all states, and then move to the sampled objective for the algorithm in the next section.

We begin by formalizing the representation learning component. Many unsupervised representation learning approaches consist of factorizing input observation $X \in \mathbb{R}^{|S|} \times d$ into a basis dictionary $B \in \mathbb{R}^{k \times d}$ and new representation $\Phi \in \mathbb{R}^{|S|} \times k$. The rows of $B$ form a set of bases, with columns in $\Phi$ weighting amongst these bases for each observation (column) in $X$. Though simple, this approach encompasses a broad range of models, including PCA, CCA, ISOMAP, locally linear embeddings and sparse coding [Singh and Gordon 2008; Le and White 2017]. The (unsupervised) sparse coding objective is [Aharon et al. 2006]

$$\min_{\Phi \in \mathbb{R}^{|S|} \times k, B \in \mathbb{R}^{k \times d}} \| \Phi B - X \|^2_D + \beta_B \| B \|^2_F + \beta_\phi \| \Phi \|^2_{D,1}$$

where $\| Y \|^2_F = \sum_{ij} Y_{ij}^2$ is the squared Frobenius norm; $B \in \mathbb{R}^{k \times d}$ is a learned basis dictionary; $\beta_B, \beta_\phi > 0$ determine the magnitudes of the regularizers; $D \in [0, 1]^{(|S|) \times (|S|)}$ is a diagonal matrix giving a distribution over states, corresponding to the stationary distribution of the policy $d^\pi : S \to [0, 1]$; and $\| z \|^2_D = z^\top D z$ is a weighted norm. The reconstruction error

$$\| \Phi B - X \|^2_D = \sum_{s \in S} d^\pi(s) \| \Phi(s, :) B - X(s, :) \|^2_2$$

is weighted by the stationary distribution $d^\pi$ because states are observed with frequency indicated by $d^\pi$. The weighted $\ell_1$

$$\| \Phi \|^2_{D,1} = \sum_{s \in S} d^\pi(s) \sum_{j=1}^{k} | \Phi(s, j) |$$

promotes sparsity on the entries of $\Phi$, preferring entries in $\Phi$ to be entirely pushed to zero rather than spreading magnitude across all of $\Phi$. The Frobenius norm regularizer on $B$ ensures that $B$ does not become too large. Without this regularizer, all magnitude can be shifted to $B$, producing the same $\Phi B$, but pushing $\| \Phi \|^2_{D,1}$ to zero and nullifying the utility of its regularizer. Optimizing this sparse coding objective would select a sparse representation $\phi$ for each observation $x$ such that $\phi B$ approximately reconstructs $x$.

Further, however, we would like to learn a new representation that is also optimized towards approximating the value function. Towards this aim, we need to jointly learn $\Phi$ and $w$, where $\Phi w$ provides the approximate value function. In this way, the optimization must balance between accurately recreating $X$ and approximating the value function $\Phi w$. For this, we must choose an objective for learning $w$.

We consider two types of objectives: fixed-point objectives and squared-error objectives. Two common fixed-point objectives are the mean-squared Bellman error (MSBE), also called the Bellman residual [Baird 1995]

$$\| \Phi w - (r^\pi + \gamma P^\pi \Phi w) \|^2_D$$

and mean-squared projected Bellman error (MSPEB) [Sutton et al. 2009]

$$\| \Phi w - \Pi(r^\pi + \gamma P^\pi \Phi w) \|^2_D$$

1This variable $X$ can also be a base set of features, on which the agent can improve or which the agent can sparsify.
where $D \in [0, 1]^{||S|| \times ||S||}$ is a diagonal matrix giving a distribution over states, corresponding to the stationary distribution of the policy; $||z||_2^2 = z^T D z$ is a weighted norm; and the projection matrix for linear value functions is $\Pi = \Phi (\Phi^T D \Phi)^{-1} \Phi^T D$. The family of TD algorithms converge to the minimum of the MSPBE, whereas residual gradient algorithms typically use the MSBE (see [Sun and Bagnell 2015] for an overview). Both have useful properties [Scherer 2010], though arguably the MSPBE is more widely used.

There are also two alternative squared-error objectives, that do not correspond to fixed-point equations: the mean-squared return error (MSRE) and the Bellman error (BE). For a trajectory of samples $\{(x_i, r_{i+1}, x_{i+1})\}_{i=0}^{t-1}$, BE is defined as

$$
\sum_{i=0}^{t-1} \|r_{i+1} + \gamma \phi_i^T w - \phi_i^T w\|_2^2
$$

and the MSRE as

$$
\sum_{i=0}^{t-1} \|g_{i+1} - \phi_i^T w\|_2^2
$$

where $g_{i+1} = \sum_{j=i}^{t-1} \gamma^{j-i} r_{j+1}$ is a sample return. In expectation, these objectives are, respectively,

$$
\sum_{s \in S} d^i(s) \mathbb{E}\left[\left(r(S_t, A_t, S_{t+1}) + \gamma \phi(S_{t+1})^T w - \phi(S_t)^T w\right)^2 | S_t = s\right]
$$

and

$$
\sum_{s \in S} d^i(s) \mathbb{E}\left[\sum_{i=0}^{t-1} \gamma^i r(S_{i+1}, A_{i+1}, S_{i+1}) - \phi(s)^T w \right]^2 | S_t = s\right]
$$

where the expectation is w.r.t. the transition probabilities and taking actions according to policy $\pi$.

These differ from the fixed-point objectives because of the placement of the expectation. To see why, consider the MSBE and BE. The expected value of the BE is the expected squared error between the prediction from this state and the reward plus the value from a possible next state. The MSBE, on the other hand, is the squared error between the prediction from this state and the expected reward plus the expected value for the next state. Though the MSPBE and MSBE constitute the most common objectives chosen for reinforcement learning, these squared-error objectives have also been shown to be useful particularly for learning online [Sun and Bagnell 2015].

For sparse coding, however, the MSPBE is not a suitable choice—compared to the MSBE, BE and MSRE—for two reasons. First, the MSBE, BE and MSRE are all convex in $\Phi$, whereas the MSPBE is not. Second, because of the projection onto the space spanned by the features, the MSPBE can be solved with zero error for any features $\Phi$. Therefore, because it does not inform the choice of $\Phi$, the MSPBE produces a two stage approach where features are learned in a completely unsupervised way and prediction performance does not influence $\Phi$.

This problem seems to have been overlooked in two approaches for basis adaptation based on the MSPBE: adaptive bases algorithm for the projected Bellman error (ABPBE) [Di Castro and Mannor 2010] and mirror descent $Q(\lambda)$ with basis adaptation [Mahadevan et al. 2013]. For example, for ABPBE, it is not immediately obvious this would be a problem, because a stochastic approximation approach is taken. However, if written as a minimization over the basis parameters and the weights, one would obtain a

The final objective for loss $L(\Phi, w)$ set to either MSBE, BE or MSRE is

$$
\min_{w \in \mathbb{R}^d, \Phi \in \mathbb{R}^{||S|| \times d}, B \in \mathbb{R}^{k \times d}} L(\Phi, w) + \| \Phi B - X \|^2_D
$$

4 Algorithm for Sparse Coding

We now derive the algorithm for sparse coding for policy evaluation: SCoPE. We generally consider either the BE or MSRE. For a trajectory of samples $\{(x_i, r_{i+1}, x_{i+1})\}_{i=0}^{t-1}$, the objective is

$$
\min_{w \in \mathbb{R}^d, \Phi \in \mathbb{R}^{||S|| \times d}, B \in \mathbb{R}^{k \times d}} \frac{1}{t} \sum_{i=0}^{t-1} (y_i + \gamma \phi_i^T w - \phi_i^T w)^2 + \frac{1}{t} \sum_{i=0}^{t-1} \|\phi_i B - x_i\|^2_2 + \beta_B \|B\|^2_F + \beta_w \|w\|^2_2 + \beta_\Phi \|\Phi\|^p_2.
$$

for BE, $y_i = r_{i+1}$ and $\gamma = 1$ and for MSRE, $y_i = \sum_{j=i}^{t-1} \gamma^{j-i} r_{j+1}$ and $\gamma = 0$. We consider two possible powers for the $\ell_1$ norm $p = 1$ or 2, where the theory relies on using $p = 2$, but in practice we find they perform equivalently and $p = 1$ provides a slightly simpler optimization. The loss is averaged by $t$, to obtain a sample average, which in the limit converges to the expected value under $d^i$. This averaged loss is also more scale-invariant—in terms of the numbers of samples—to the choice of regularization parameters.

SCoPE consists of alternating amongst these three variables, $B$, $w$ and $\Phi$, with a proximal gradient update for the non-differentiable $\ell_1$ norm. The loss in terms of $B$ and $w$ is differentiable; to solve for $B$ (or $w$) with the other variables fixed, we can simply used gradient descent. To solve for $\Phi$ with the $B$ and $w$ fixed, however, we cannot use a standard gradient descent update because the $\ell_1$ regularizer is non-differentiable. The proximal update consists of stepping in the direction of the gradient for the smooth component of the objective—which is differentiable—and then projecting back to a sparse solution using the proximal operator: a soft thresholding operator. The convergence of this alternating minimization follows from results on block coordinate descent for non-smooth regularizers [Xu and Yin 2013].

To apply the standard proximal operator for the $\ell_1$ regularizer, we need to compute an upper bound on the Lipschitz constant for this objective. The upper bound is $2(1 + \gamma^2)\|w\|^2_2 + 2\|B\|^2_F$, computed by finding the maximum minimum error solution (i.e., error zero) immediately for any basis parameters. The basis parameters are considered to change on a slow timescale, and the weights on a fast timescale, which is a reflection of this type of separate minimization. [Menache et al. 2005] avoided this problem by explicitly using a two-stage approach, using MSPBE approaches for learning the parameters and using other score function, such as the squared Bellman error, to update the bases. This basis learning approach, however, is unsupervised.

Representation learning strategies for the MSPBE have been developed, by using local projections [Yu and Bertsekas 2009; Bhatnagar et al. 2009]. These strategies, however, do not incorporate sparse coding.
singular value of the Hessian of the objective w.r.t. \( \phi_i \) for each \( i \). We will provide additional details for this calculation, and implementation details, in a supplement.

### 4.1 Local Minima Are Global Minima

In this section, we show that despite nonconvexity, the objective for SC\text{OPE} has the nice property that all local minima are in fact global minima. Consequently, though there may be many different local minima, they are in fact equivalent in terms of the objective. This result justifies a simple alternating minimization scheme, where convergence to local minima ensures an optimal solution is obtained.

We need the following technical assumption. It is guaranteed to be true for a sufficiently large \( k \leq t \) (see [Haefele and Vidal, 2013], [Le and White, 2017]).

**Assumption 1.** For the given \( k \geq d \), the following function is convex in \( Z \in \mathbb{R}^{n \times d} \):

\[
\Phi[Z] = \Phi[B] w + \frac{\beta_n}{2} \|B\|_F^2 + \frac{\beta_m}{2} \|w\|_2^2 + \frac{\beta_\delta}{t} \sum_{i=0}^t \|\Phi_i\|_2^2
\]

where \( \Phi_i \) denotes the sequence of functions, \( |s\|_{\Phi_i} \) is the absolute value: \( |x|_i = \sqrt{\mu_i^2 + x_i^2} - \mu_i \). Let \( \Theta = (\Phi, B, w) \). The sequence of functions \( f_n \) are defined with \( \mu_n = 1/n \), as

\[
f_n(\Theta) = L(\Theta) + \beta_n \sum_{ij} \sqrt{\mu_i^2 + \Phi_{ij}^2} - \mu_n
\]

where \( L(\Theta) \) equals the equation in [3], but without the \( \ell_2^2 \) regularizer on \( \Phi \).

**Part 1:** All local minima of \( f_n \) for all \( n \) are global minima. To show this, we show each \( f_n \) satisfies the conditions of [Le and White, 2017] Theorem 10 and Proposition 11.

**Part 1.1** We can rewrite the loss in terms of \( \Phi[B w] \):

\[
\frac{1}{t} \sum_{i=0}^{t-1} \|y_i + \gamma \Phi_{i+1}^T w - \Phi_i^T w\|^2_2 + \frac{1}{t} \sum_{i=0}^{t-1} \|\phi_i B - x_i\|_2^2
\]

\[
= \frac{1}{t} \|X - \Phi B\|^2_2 + \frac{1}{t} \|y - (I_{0:t-1} - \gamma I_{1:t}) \Phi w\|_2^2
\]

where \( I_{1:t} \in \mathbb{R}^{t \times t} \) is a diagonal matrix of all ones with the first diagonal entry set to zero, and \( I_{0:t-1} \) with the last diagonal entry set to zero. This loss is convex in the joint variable \( \Phi[B w] \), but it is the composition of a convex function (squared norm) and an affine function (multiplication by \( \gamma I_{t+1:t} - I_{0:t-1} \) and addition of \( y \)).

**Part 1.2** The regularizer on \( [B w] \) must be a weighted Frobenius norm, with weightings on each column: here, we have weightings using regularization parameters \( \beta_B \) for the first \( d \) columns (corresponding to \( B \)) and regularization parameter \( \beta_m \) for the last column (corresponding to \( w \)).

**Part 1.3** The inner dimension \( k > d \), which is true by assumption and the common setting for sparse coding.

**Part 1.4** The pseudo-Huber loss, on the columns of \( \Phi \), is convex, centered and twice-differentiable.

**Part 2:** The sequence \( f_n \) converges uniformly to \( f \). To see why, recall the definition of uniform convergence. A sequence of functions \( \{f_n\} \) is uniformly convergent with limit \( f \) if for every \( \epsilon > 0 \), there exists \( N \in \mathbb{N} \) such that for all \( \Theta \in \Omega \) and all \( n \geq N \), \( |f_n(\Theta) - f(\Theta)| < \epsilon \). Further recall that for any complete metric space, if \( f_n \) is uniformly Cauchy, then it is uniformly convergent. The sequence is uniformly Cauchy if for all \( n, m \geq N \), \( |f_n(\Theta) - f_m(\Theta)| < \epsilon \). Take any \( \epsilon > 0 \) and let \( N = \lceil \frac{4(k+1)\beta_\delta}{\epsilon t} \rceil \). Then

\[
|f_n(\Theta) - f_m(\Theta)| \leq \frac{2(k+1)\beta_\delta}{t} |\mu_n - \mu_m| \leq \frac{2(k+1)\beta_\delta}{t} |\frac{1}{n} - \frac{1}{m}|
\]

\[
\leq \frac{4(k+1)\beta_\delta}{t N} \leq \epsilon.
\]

**Part 3:** Asymptotic equivalence of minimizers of \( f_n \) and \( f \). Because \( f \) is continuous, and so lower semi-continuous, and \( f_n \) uniformly converges to \( f \), we know that \( f_n \) \( \Gamma \)-converges to \( f: f_n \xrightarrow{\Gamma} f \) [Braides, 2013].

By the fundamental theorem of \( \Gamma \)-convergence, if the \( \{f_n\} \) is an equi-coercive family of functions, then the minimizers of \( f_n \) converge to minimizers of \( f \). A sequence of functions \( \{f_n\} \) is equi-coercive iff there exists a lower semi-continuous coercive function \( \psi: \Theta \to \mathbb{R} \cup \{-\infty, \infty\} \) such that \( f_n \geq \psi \) on \( \Theta \) for every \( n \in \mathbb{N} \) [Dal Maso, 2012] Proposition 7.7. A function is coercive if \( \psi(\Theta) \to \infty \) as \( \|\Theta\| \to \infty \). For \( \psi(\Theta) = f(\Theta) \), it is clear that \( \psi \) is coercive, as well as lower semi-continuous (since it is continuous). Further, \( f_n(\Theta) \geq f(\Theta) \) because the regularizer on \( \Phi \) is non-negative. Therefore, the family \( \{f_n\} \) is equi-coercive, and so the minimizers of \( f_n \) converge to minimizers of \( f \).

For the other direction, if a local minimum \( \Theta \) of \( f \) is an isolated local minimum, then there exists a sequence \( \Theta_n \to \Theta \) with \( \Theta_n \) a local minimizer of \( f_n \) for \( \mu_n \) sufficiently small [Braides, 2013] Theorem 5.1]. Because we have Frobenius norm regularizers on \( B, w \), which are strongly convex, the objective is strictly convex with respect to \( B, w \). Further, because \( X \) is full rank, \( \| \Phi B - X \|_F^2 \) is a strictly convex function
with respect to $\Phi$. Therefore, locally the objective is strictly convex with respect to $\theta$. We therefore know that local minima of $f$ are isolated, and so there exists an $N$ such that for all $n > N$, $\theta_n$ are local minimizers of $f_n$. Since these local minimizers are global minimizers, and they converge to $\theta$, this means $\theta$ is a global minimum of $f$.

For the second statement, we use [Haeffele and Vidal 2015] Theorem 15. Because we already showed above that our loss can be cast as factorization, it is clear our loss and regularizers are positively homogenous, of order 2. A minimum is guaranteed to exist for our objective, because the loss function is continuous, bounded below (by zero) and goes to infinity as the parameters go to $\pm \infty$.

5 Experimental Results

We aim to address the question: can we learn useful representations using SCoPE? We therefore tackle the setting where the representation is first learned, and then used, to avoid confounding incremental estimation and the utility of the representation. We particularly aim to evaluate estimation accuracy, as well as qualitatively understanding the types of sparse representations learned by SCoPE.

Domains. We conducted experiments in three benchmark RL domains - Mountain Car, Puddle World and Acrobot [Sutton 1996]. All domains are episodic, with discount set to 1 until termination. The data in Mountain Car is generated using the standard energy-pumping policy policy with 10% randomness. The data in Puddle World is generated by a policy that chooses to go North with 50% probability, and East with 50% probability on each step, with the starting position in the lower-left corner of the grid, and the goal in the top-right corner. The data in Acrobot is generated by a near-optimal policy.

Evaluation. We measure value function estimation accuracy using mean absolute percentage value error (MAPVE), with rollouts to compute the true value estimates. MAPVE $= \frac{1}{t_{test}} \sum_{s \in X_{test}} \frac{|V(s) - \hat{V}(s)|}{|V^*(s)|}$, where $X_{test}$ is the set of test states, $t_{test} = 5000$ is the number of samples in the test set, $\hat{V}(s)$ is the estimated value of state $s$ and $V^*(s)$ is the true value of state $s$ computed using extensive rollouts. Errors are averaged over 50 runs.

Algorithms. We compare to using several fixed tile-coding (TC) representations. TC uses overlapping grids on the observation space. It is a sparse representation that is well known to perform well for Mountain Car, Puddle World, and Acrobot. We varied the granularity of the grid-size $N$ and number of tilings $D$, where $D$ is the number of active features for each observation. The grid is either $N \times N$ for Mountain Car and Puddle World or $N^4$ for Acrobot. We explore $(D=4, N=4), (D=4, N=8), (D=16, N=4), (D=16, N=8), (D=32, N=4), (D=32, N=8)$; a grid size of 16 performed poorly, and so is omitted. For Mountain Car and Puddle World the number of features respectively are 64, 256, 256, 1024, 512, 2048, then hashed to 1024 dimensions; for Acrobot, the number of features are 1024, 16384, 4096, 65536, 8192, 131072, then hashed to 4096. Both of these hashed sizes are much larger than our chosen $k = 100$.

For consistency, once the SCoPE representation is learned, we use the same batch gradient descent update on the MSRE for all the algorithms, with line search to select step-sizes. The regularization weights $\beta_B$ are chosen from $\{1^{-5}, \ldots, 1^{-1}, 0\}$, based on lowest cumulative error. For convenience, $\beta_w$ is fixed to be the same as $\beta_B$. For learning the SCoPE representations, regularization parameters were chosen using 5-fold cross-validation on 5000 training samples, with $\beta_\phi = 0.1$ fixed to give a reasonable level of sparsity. This data is only used to learn the representation; for the learning curves, the weights are learned from scratch in the same way they are learned for TC. The dimension $k = 100$ is set to be smaller than for tile coding, to investigate if SCoPE can learn a more compact sparse representation. We tested unsupervised sparse coding, but the error was poor (approximately 10 times worse). We discuss the differences between the representations learned by supervised and unsupervised sparse coding below.

Learning curves. We first demonstrate learning with increasing number of samples, in Figure 1. The weights are recom-
Across domains, SCoPE results in faster learning and, in
Mountain Car and Acrobot, obtains lowest final error. Match-
ing the performance of TC is meaningful, as TC is well-
understood and optimized for these domains. For Acrobot,
it’s clear a larger TC is needed resulting in relatively poor
performance, whereas SCoPE can still perform well with a
compact, learned sparse representation. These learning curves
provide some insight that we can learn effective sparse repre-
sentations with SCoPE, but also raise some questions. One
issue is that SCoPE is not as effective in Puddle World as some
of the TC representations, namely 4-4 and 16-4. The reason
for this appears to be that we optimize MSRE to obtain the
representation, which is a surrogate for the MAPVE. When
measuring MSRE instead of MAPVE on the test data, SCoPE
consistently outperforms TC. Optimizing both the representa-
tion and weights according to MSRE may have overfitting
issues; extensions to MSBE or BE, or improvements in select-
ing regularization parameters, may alleviate this issue.

**Learned representations.** We also examine the learned rep-
resentations, both for unsupervised sparse coding and SCoPE,
shown in Figure 2. We draw two conclusions from these re-
results: the structure in the observations is not sufficient for
unsupervised sparse coding, and the combination of supervised
and unsupervised losses sufficiently constrain the space to ob-
tain discriminative representations. For these two-dimensional
and four-dimensional observations, it is relatively easy to re-
construct the observations by using only a small subset of
dictionary atoms (row vectors of $B$ in equation (2)). The unsu-
supervised representations, even with additional non-negativity
constraints to narrow the search space, are less distributed,
with darker and thicker blocks, and more frequently pick less
features. For the supervised sparse coding representation,
however, the sparsity pattern is smoother and more distributed:
more features are selected by at least one sample, but the level
of sparsity is similar. We further verified the utility of super-
vised sparse coding, by only optimizing the supervised loss
(MSRE), without including the unsupervised loss; the result-
ing representations looked similar to the purely unsupervised
representations. The combination of the two losses, therefore,
much more effectively constrains or regularizes the space of
feasible representations and improves discriminative power.

The learning demonstrated for SCoPE here is under ideal
conditions. This was intentionally chosen to focus on the
question: can we learn effective sparse representations using
the SCoPE objective? With the promising results here, future
work needs to investigate the utility of jointly estimating the
representation and learning the value function, as well as pro-
viding incremental algorithms for learning the representations
and setting the regularization parameters.

### 6 Conclusion

In this work, we investigated sparse coding for policy evalu-
ation in reinforcement learning. We proposed a supervised
sparse coding objective, for joint estimation of the dictionary,
sparse representation and value function weights. We pro-
vided a simple algorithm that uses alternating minimization
on these variables, and proved that this simple and easy-to-
use approach is principled. We finally demonstrate results on
three benchmark domains, Mountain Car, Puddle World and
Acrobot, against a variety of configurations for tile coding.

This paper provides a new view of using dictionary learning
techniques from machine learning in reinforcement learning.
It lays a theoretical and empirical foundation for further in-
vestigating sparse coding, and other dictionary learning
approaches, for policy evaluation and suggests that they show
some promise. Formalizing representation learning as a dic-
tionary learning problem facilitates extending recent and upcom-
ing advances in unsupervised learning to the reinforcement
learning setting. For example, though we considered a batch
gradient descent approach for this first investigation, the sparse
coding objective is amenable to incremental estimation, with
several works investigating effective stochastic gradient de-
cent algorithms [Mairal et al. 2009, 2010, Le and White
2017]. The generality of the approach and easy to understand
optimization make it a promising direction for representation
learning in reinforcement learning.

### References

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth
Netrapalli. Learning sparsely used overcomplete dictionaries via
alternating minimization. In *Ann. Conf. on Learning Theory*, 2014.
Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 2006.

Haitham B Ammar, Karl Tuyls, Matthew E Taylor, Kurt Driessens, and Gerhard Weiss. Reinforcement learning transfer via sparse coding. In *Inter. Conf. on Autonomous Agents and Multiagent Systems*, 2012.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. arXiv:1503.00778v1 [cs.LG], 2015.

Christopher G Atkeson and Jun Morimoto. Nonparametric representation of policies and value functions: a trajectory-based approach. In *Advances in Neural Information Processing Systems*, 2003.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Inter. Conf. on Mach. Learning*, 1995.

Shalabh Bhatnagar, Doina Precup, David Silver, Richard S Sutton, Hamid R Maei, and Csaba Szepesvári. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems*, 2009.

Andrea Braides. Local minimization, variational evolution and Γ-convergence. *Lecture Notes in Mathematics*, 2013.

Rémi Coulom. Reinforcement learning using neural networks, with applications to motor control. PhD thesis, INPG, 2002.

Gianni Dal Maso. *An Introduction to Γ-Convergence*. Springer, 2012.

Dotan Di Castro and Shie Mannor. Adaptive bases for Q-learning. In *Conference on Decision and Control*, 2010.

Kimon Fountoulakis and Jacek Gondzio. A second-order method for strongly convex ℓ1-regularization problems. *Math. Prog.*, 2013.

Benjamin D Haefele and Rene Vidal. Global Optimality in Tensor Factorization. Deep Learning, and Beyond. arXiv.org, 2015.

J Zico Kolter and Andrew Y Ng. Regularization and feature selection in least-squares temporal difference learning. In *Inter. Conf. on Mach. Learning*, 2009.

George Konidaris, Sarah Osentoski, and Philip Thomas. Value function approximation in reinforcement learning using the Fourier basis. In *Inter. Conf. on Mach. Learning*, 2011.

Lei Le and Martha White. Global optimization of factor models using alternating minimization. arXiv.org:1604.04942v3, 2017.

Manuel Loth, Manuel Davy, and Philippe Preux. Sparse temporal difference learning using LASSO. In *Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: a Laplacian framework for learning representation and control in Markov decision processes. *J. Machine Learning Research*, 2007.

Sridhar Mahadevan, Stephen Giguere, and Nicholas Jacek. Basis Adaptation for Sparse Nonlinear Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 2013.

Sridhar Mahadevan. Learning Representation and Control in Markov Decision Processes: New Frontiers. *Foundations and Trends® in Machine Learning*, 2009.

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, 2009.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Machine Learning Res.*, 2010.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 2005.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Trung Nguyen, Zhiqun Li, Toshiyuki Saito, and Tze Yun Leong. Online feature selection for model-based reinforcement learning. *J. Machine Learning Research*, 2013.

Bruno Ollshausen and David Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 1997.

Christopher Painter-Wakefield and Ronald Parr. Greedy algorithms for sparse reinforcement learning. In *Inter. Conf. on Mach. Learning*, 2012.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L Littman. An analysis of linear models linear value function approximation and feature selection for reinforcement learning. In *Inter. Conf. on Mach. Learning*, 2008.

Bohdana Ratitch and Doina Precup. Sparse distributed memories for on-line value-based reinforcement learning. In *ECML PKDD*, 2004.

Martin Riedmiller. Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML PKDD*, 2005.

Bruno Scherrer. Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. In *Inter. Conf. on Mach. Learning*, 2010.

Ajit Singh and Geoffrey Gordon. A unified view of matrix factorization models. In *ECML PKDD*, 2008.

Kenneth O Stanley and Risto Miikkulainen. Efficient evolution of neural network topologies. In *CEC*, 2002.

Wen Sun and J Andrew Bagnell. Online Bellman Residual Algorithms with Predictive Error Guarantees. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

Richard S Sutton and Andrew G Barto. Representation search through generate and test. In *Proceedings of the AAAI Workshop on Learning Rich Representations from Low-Level Sensors*, 2013.

Richard S Sutton and Steven D Whitehead. Online learning with random representations. In *Inter. Conf. on Mach. Learning*, 1993.

Richard S Sutton, Hamid 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 *Inter. Conf. on Mach. Learning*, 2009.

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, 1996.

Shimon Whiteson, Matthew E Taylor, and Peter Stone. Adaptive tile coding for value function approximation. Technical report, 2007.

Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to non-negative tensor factorization and completion. *SIAM J. Imaging Sciences*, 2013.

Huizhen Yu and Dimitri P Bertsekas. Basis function adaptation methods for cost approximation in MDP. In *Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2009.