A factorization perspective for learning representations in reinforcement learning

Martha White
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada

Abstract

Reinforcement learning is a general formalism for sequential decision-making, with recent algorithm development focusing on function approximation to handle large state spaces and high-dimensional, high-velocity (sensor) data. The success of function approximators, however, hinges on the quality of the data representation. In this work, we explore representation learning within batch reinforcement learning, with a focus on making the assumptions on the representation explicit and making the learning problem amenable to principled optimization techniques. We specify a reinforcement learning objective for value function learning that facilitates the addition of a regularized matrix factorization objective to specify the desired class of representations. The resulting joint optimization over the representation and value function parameters enables us to take advantages of recent advances in unsupervised learning and presents a general yet simple formalism for learning representations in reinforcement learning.

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 first define a representation set (implicitly or explicitly) and then optimize an objective or use heuristics to select a "good" representation from that set. For example, for feature selection, the set of representations is all possible subsets of the given features. There are numerous methods to find a representation from this set, such as ℓ_1 regularized least-squares temporal difference learning (LSTD) (Kolter and Ng 2009), sparse LSTD using LASSO (Loth et al. 2007), feature selection based on the Bellman error (Parr et al. 2008; Painter-Wakefield and Parr 2012) and online feature selection for model-based reinforcement learning (Nguyen et al. 2013). Another possible set of features is a subspace of the original feature space. One heuristic approach to find a representation in this set is to use random projections (Ghavamzadeh et al. 2010; Fard et al. 2013); another is an optimization approach that uses ℓ_2 regularized LSTD (Farahmand et al. 2008). Another approach is to optimize parameters of the commonly used basis functions and tile coding representations in reinforcement learning. Again, this involves heuristic approaches, such as adaptive tile coding (Whiteson et al. 2007), as well as explicit objectives, such as maximizing likelihood of parameters for basis functions (Menache et al. 2005).

The choice of set strongly influences the ability to optimally select the representation. Though some sets may be more powerful, such as neural network representations, the optimization can become more difficult. Heuristic approaches to find a representation in this set can be simple, such as random representations (Sutton and Whitehead 1993) and linear threshold unit search (Mahmood and Sutton 2013); others are computationally intensive optimizations of layered objectives, such as neural-Q iteration (Riedmiller 2005), evolutionary algorithms like NEAT (Stanley and Miikkulainen 2002) and deep reinforcement learning (Mnih et al. 2013). Similarly, the set of instance-based representations can be very powerful, since kernel representations are non-parametric and use a linear optimization to enable non-linear learning with respect to the original feature space. These approaches can have issues with storage of samples/states or choosing representative instances, such as in locally weighted regression (Atkeson and Morimoto 2003), sparse distributed memories (Ratitch and Precup 2004) and proto-value functions (Mahadevan and Maggioni 2007).

Regardless of the approach, it is key to (1) make the representation learning set explicit, so the algorithm target is clear, (2) connect the representation selection to learning performance and (3) facilitate selection of the representation from that set. We propose to look at representation learning as a matrix factorization: factorizing the features in a basis dictionary and new representation. Matrix factorization has been an important advance in unsupervised learning, because it unifies many unsupervised learning algorithms

into one framework (Xu et al. 2009; White and Schuurmans 2012; De la Torre 2012), including (exponential family) principal components analysis, k-means clustering, mixture model clustering, canonical correlation analysis and normalized graph cut. Moreover, there have been important advances in convex formulations for a restricted class of matrix factorization problems (Bach et al. 2008; Zhang et al. 2011; White et al. 2012), facilitating optimization for at least two important classes of representation learning: sparse coding and subspace learning.

In this work, we show how to extend Bellman residual minimization to include an unsupervised, matrix factorization component that ports these advances to reinforcement learning. Regularized matrix factorization clarifies the assumptions on the data distribution (from the chosen loss) and structure of the representation (from the chosen regularizer). In addition to making the representation set explicit and facilitating optimization, our proposed joint objective over the representation and value function function parameters connects the representation selection to prediction performance.

Our main contribution is an explicit joint optimization over the value function parameters and the representation that is amenable to known optimization techniques, including convex reformulation techniques. In most previous representation learning approaches with regularization, only the weights are regularized, with the representation remaining fixed. In the below approach, however, the representation itself is imputed, enabling more general properties to be placed on the representation. In particular, for certain forms, such as subspace and sparse representations, there are known convex reformulations (Zhang et al. 2011; White et al. 2012) that guarantee a globally optimal pair of value function parameters and representation. We indicate the required restrictions on the chosen objective for learning the value function parameters and the representation structure that enables a convex reformulation. Moreover, we show that the mean-squared project Bellman error is not suitable for joint imputation of the value function parameters and the representation. We develop the optimization using the Bellman residual; the approach, however, is general and could use other reinforcement learning objectives for the value function parameters.

Background

In reinforcement learning, an agent interacts with its environment, receiving observations and selecting actions to maximize a scalar reward signal provided by the environment. This interaction is usually modeled by a Markov decision process (MDP). An MDP consists of $(\mathcal{S}, \mathcal{A}, P, R)$ where $\mathcal{S}$ is the set of states; $\mathcal{A}$ is a finite set of actions; P, 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 $R(s')$, 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 $Pr(s_{t+1}|s_t, a_t) = Pr(s_{t+1}|s_t, a_t, \ldots, s_0, a_0)$.

Learning a Value Function

One important goal in reinforcement learning is to learn the *value function* for a policy. A value function approximates the expected total discounted future reward for following policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ from a given state s_t:

$$V^\pi(s_t) = E\left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}) \;\middle|\; s_i \sim P(\cdot|s_{i-1}, a_{i-1}), a_i \sim \pi(\cdot|s_i)\right]$$

This value function satisfies the Bellman equation

$$V^\pi(s) = R(s) + \gamma \sum_a \pi(a|s) \sum_{s'} P(s'|s, a) V^\pi(s') \quad (1)$$

For a finite number of states and actions, this formula can be re-expressed in terms of matrices and vectors for each state

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

where $V^\pi, R \in \mathbb{R}^n$ are vectors of state values and rewards, and $P^\pi \in \mathbb{R}^{n \times n}$ is the probability of transitioning between two states under policy π

$$P^\pi_{i,j} = \sum_a \pi(a|s = i) P(s' = j|s = i, a)$$

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

In practice, however, we likely have a prohibitively large state-action 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 $s_0, a_0, r_0, s_1, a_1, r_1, s_2, r_2, a_2 \ldots$, 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:

$$\hat{V}^\pi(s) = \phi^\top(s)\mathbf{w}$$

for $\mathbf{w} \in \mathbb{R}^k$ a parameter vector and $\phi : \mathcal{S} \to \mathbb{R}^k$ a feature function describing states. With this approximation, however, typically we can no longer satisfy the Bellman equation in (1), since solving for $\Phi\mathbf{w} = R + \gamma P^\pi \Phi\mathbf{w}$ with $\Phi \in \mathbb{R}^{n \times k}$ may not be defined if Φ is not invertible. Reinforcement learning algorithms, such as LSTD and Bellman residual minimization, therefore focus on finding an approximate solution to the Bellman equation, despite this representation issue.

Factorized representation learning

We now specify a joint optimization over these value function parameters and the representation. Let $L_v(\Phi, \mathbf{w})$ be the chosen objective for learning the value function parameters, $\mathbf{w}$. For example, $L_v(\Phi, \mathbf{w}) = \text{MSPBE}(\Phi, \mathbf{w})$ or $L_v(\Phi, \mathbf{w}) = \text{BR}(\Phi, \mathbf{w})$, described in the next section. In particular, for convex reformulations, we will require that $L_v(\Phi, \mathbf{w})$ is convex in each parameter; in general, however, it can be any loss for which a gradient is computable for Φ.

We augment this optimization to specify representation learning in terms of regularization strategies used in unsupervised learning. In particular, we can add a regularized matrix factorization loss to find a representation:

$$\min_{\Phi \in \mathcal{F} \subset \mathbb{R}^{n \times k}, B \in \mathcal{B}} L_r(\Phi B, X) + \alpha_2 \text{Reg}(\Phi)$$

where L_r is any loss, $X \in \mathbb{R}^{n\times d}$ is the default (expanded) feature set, $B \in \mathcal{B} \subset \mathbb{R}^{k\times d}$ is a learned basis dictionary and α_2 is the weight on the regularizer. For example, X could be all cross products of the observations, or a default set of randomly generated basis functions. To obtain binary features, set $\mathcal{F} = \{\Phi \in \{0,1\}^{n\times k}\}$, or for probabilistic features, $\mathcal{F} = \{\Phi \in [0,1]^{n\times k}\}$. The structure of the learned representation Φ, depends on the chosen regularizer. For example, $\text{Reg}(\Phi) = ||\Phi||_{1,1}$ imposes sparsity and $||\Phi^\top||_{2,1}$ imposes a subspace structure to reduce the dimension of the representation. Both of these forms can be useful for dealing with high-dimensional, high-volume data. Note that we could also include a regularizer on B; for simplicity in presentation, however, we omit this addition.

We obtain the following Factorized-Representation RL (FR-RL) optimization,

$$\min_{\mathbf{w},\Phi,B\in\mathcal{B}} L_v(\Phi,\mathbf{w}) + \alpha_1 L_r(\Phi B, X) + \alpha_2 \text{Reg}(\Phi)$$

where α_1 is included to enable control on the importance of all three components in the objective. This new joint optimization combines a supervised and unsupervised loss, directing representation learning based both on the desired structure and on prediction performance. For a fixed representation, Φ, the optimization reduces to learning the value function parameters for the learned representation. An out-of-the-box optimization approach, therefore, is simply to alternate between the variables using gradient descent algorithms or to use gradient descent on the representation variables Φ and B and solve each inner optimization over $\mathbf{w}$ on each gradient step. Though in general, this approach may be the only option for certain choices of constraint sets $\mathcal{F}$ and $\mathcal{B}$, regularizers and loss functions, in the next two section, we show settings in which we can reformulate FR-RL as a convex optimization.

Objective functions for Factorized Representations

Two widely used objective in reinforcement learning are the the Bellman residual (BR) (Baird 1995):

$$\min_{\mathbf{w}\in\mathbb{R}^k} \|\Phi\mathbf{w} - T(\Phi\mathbf{w})\|_D^2 = \min_{\mathbf{w}\in\mathbb{R}^k} \|\Phi\mathbf{w} - (R + \gamma P^\pi \Phi\mathbf{w})\|_D^2$$

mean-squared projected Bellman error (MSPBE) (Sutton et al. 2009):

$$\min_{\mathbf{w}\in\mathbb{R}^k} \|\Phi\mathbf{w} - \Pi(R + \gamma P^\pi \Phi\mathbf{w})\|_D^2$$

where $D \in [0,1]^{n\times n}$ is a diagonal matrix giving the distribution over states, $||\mathbf{z}||_D^2 = \mathbf{z}^\top D\mathbf{z}$ and the projection matrix for linear value functions is $\Pi = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D$.

Though both have useful properties (Scherrer 2010), we can see that the MSPBE is not a suitable choice for this joint optimization because it can be solved with zero error for each set of features. Assuming that we the transition model and reward function are given, the closed form LSTD solution to the MSPBE is (Bradtke and Barto 1996):

$$\begin{aligned}\mathbf{w} &= \left(\Phi^\top D\Phi\right)^{-1}\Phi^\top D\left(R + \gamma P^\pi\Phi\mathbf{w}\right)\\ \implies \Phi\mathbf{w} &= \Pi\left(R + \gamma P^\pi\Phi\mathbf{w}\right) = \Pi T(\Phi\mathbf{w})\end{aligned}$$

Therefore, setting $L_v = \text{MSPBE}$ produces a two stage approach, where features are learned in a completely unsupervised way and prediction performance does not influence Φ.

The Bellman residual, however, does result in an interesting optimization. Moreover, it is convex in both Φ and $\mathbf{w}$, which is also not the case MSPBE. We first present optimization approaches to our factorized representation objective with the Bellman residual assuming we have access to the transition model and reward function; we describe how to move to a trajectory of samples in the last section.

Improved optimization for FR-BRM

Let $C = [\mathbf{w}\ B] \in \mathcal{C}$, where $\mathcal{C}$ is a constraint set on C. We use a change of variables, $Z_1 = \Phi\mathbf{w}, Z_2 = \Phi B$ to obtain a simpler optimization. For current convex reformulations, we need to assume a norm regularizer, $\text{Reg}(\Phi) = \|\Phi\|$. Set

$$L_v(\Phi,\mathbf{w}) = \|(I - \gamma P^\pi)\Phi\mathbf{w} - R\|_D^2 = \text{BR}(\Phi,\mathbf{w})$$

then we get the following reformulation of the FR-BRM

$$\begin{aligned}\min_{C=[\mathbf{w}\ B]\in\mathcal{C},\Phi} \quad & \|(I - \gamma P^\pi)\Phi\mathbf{w} - R\|_D^2 \\ + \quad & \alpha_1 L(\Phi B, X) + \alpha_2 ||\Phi||\end{aligned} \tag{2}$$

$$\equiv \min_{Z=[Z_1\ Z_2]} \|(I - \gamma P^\pi)Z_1 - R\|_D^2 + L_r(Z_2, X) + \alpha_2 |\!|\!| Z |\!|\!|$$

where

$$|\!|\!| Z |\!|\!| = \min_{C\in\mathcal{C}} \min_{\Phi:\Phi C = Z} ||\Phi||$$

is the induced norm given the norm on Φ. We can simplify further using $Y = [R\ X]$ and $\mathbf{e}_1 = [1\ 0\ldots 0]$, giving

$$\min_Z L(Z,Y) + \alpha_2 |\!|\!| Z |\!|\!| \tag{3}$$

where L now contains both the loss between ΦB and X and the loss between $\Phi\mathbf{w}$ and R. Note that $\|(I - \gamma P^\pi)Z_1 - R\|_D^2$ is convex in Z_1 since multiplying by a positive matrix maintain convexity: $(I - \gamma P^\pi)$ simply puts weights on instances.

Recent advances in (semi-supervised) matrix factorization (Bach et al. 2008; Zhang et al. 2011; White et al. 2012) indicate that the induced regularizer $|\!|\!| \cdot |\!|\!|$ is convex as long as the regularizer on Φ sums over all latent features, i.e. $\sum_{i=1}^k ||\Phi_{:,i}||$ where $1 \le k \le \infty$ and for a restricted class of constraint sets, $\mathcal{C}$. See the Appendix for a list of efficiently computable convex induced regularizers on Z. Though this list is currently quite restricted, FR-BRM does not rely on the above set and can advance as more efficiently computable induced regularizers are discovered. The ability to benefit from advances in the large field of unsupervised learning is a strong benefit of FR-BRM.

Once we obtain Z, we can use a boosting procedure to recover the parameters C and Φ (Zhang et al. 2012). For certain settings, it is more simple; for example, for $\mathcal{C} = \{C : ||C_{i,:}||_2 \le 1\}$ and $||\Phi^\top||_{2,1}$ the recovery is simply a singular value decomposition: for $Z = Q\Sigma M^\top$ with Q and M orthonormal and Σ a diagonal matrix of singular values, $C = M^\top$ and $\Phi = Q\Sigma$.

Learning from samples

To practically deal with real-world streams of data and large state-spaces, we cannot assume we have explicit knowledge of the (large) transition model P^π and R. Though these could be learned, it is often desirable to be able to solve the parameters without needing to find these models.

To avoid using the models, we define matrices approximated from sampled quartets (s_i, a_i, r_i, s'_i)

$$\bar{X} \equiv \begin{bmatrix} \mathbf{f}(s_1)^\top \\ \mathbf{f}(s_2)^\top \\ \vdots \\ \mathbf{f}(s_t)^\top \end{bmatrix}, \bar{R} \equiv \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_t \end{bmatrix}$$

where $\mathbf{f} : \mathcal{S} \to \mathbb{R}^d$ is the feature function for the initial set of features, such as the observations. Unlike previous LSTD and BRM sampled approaches, however, we cannot sample both Φ and $\Phi' = P^\pi \Phi$, because the features are being imputed. Instead, we must directly approximate P^π to get the corresponding instance weights in the loss. Fortunately, this linear transformation is quite simple in practice, since $\hat{\Phi}'$ is $\hat{\Phi}$ shifted by one index. For example, if X is a sequential stream of data, then we define

$$\hat{P}^\pi = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 \end{bmatrix}$$

In general, the samples might not always be perfectly aligned or in order, such as for offline trajectories or episode ends; in such cases, the permutation matrix $\hat{P}^\pi$ would have to be defined to take these discontinuities into account.

The resulting model-free FR-BRM optimization for $\hat{Y} = [\hat{R}\ \hat{X}]$ can now be stated as:

$$\begin{aligned} \min_{C=[\mathbf{w}\ B] \in \mathcal{C}, \Phi} & \left\| (I - \gamma \hat{P}^\pi)\Phi\mathbf{w} - \hat{R} \right\|_D^2 \\ & + \alpha_1 L_r(\Phi B, \hat{X}) + \alpha_2 ||\Phi|| \\ \equiv \min_{Z=[Z_1\ Z_2]} & \left\| (I - \gamma \hat{P}^\pi) Z_1 - R \right\|_D^2 \\ & + L_r(Z_2, X) + \alpha_2 \|\!| Z |\!\| \end{aligned}$$

Discussion

Several questions arise from viewing representation learning for reinforcement learning under the FR-BRM optimization.

The first natural question is about the generality of this approach. Because the set of regularizers on Φ to obtain a convex formulation is limited, this suggests few structures can be chosen. If we do not require convexity, however, we can use a wider class of regularizers in Equation (2). For example, if we wanted to learn a representation similar to tile coding, we could add the constraint that $\Phi \in [0, 1]$ and use a large regularizer weight on a sparsity regularizer to push most entries to zero. This optimization is no longer convex, but we can still optimize the non-convex objective over the variables $\mathbf{w}$, B and Φ.

In addition, we can notice an interesting generalization of Bellman residual minimization by generalizing the least-squares loss on the reward prediction to any convex loss in Equation (3). If we choose a Bregman divergence, for example, this generalization suggests certain distributional assumptions on the reward (White and Schuurmans 2012). The relationship to fixed-point interpretations, however, becomes unclear and requires further exploration.

Second, it is important to notice that FR-BRM maintains the original properties of the value-function learning objective. A complaint about the sparse LASSO approach to LSTD (Loth et al. 2007) was that the fixed-point interpretation was lost after adding a sparse regularizer. In this situation, however, if we compute and fix the representation in the optimization, we revert to learning the value function parameters according to the chosen reinforcement learning objective, such as the Bellman residual or MSPBE.

Third, we need to consider computational complexity, which is typically a large consideration for high-velocity, high-dimensional data that occurs in realistic sequential decision-making tasks. The types of representations the formalism specifies, such as sparse or subspace representations, is key for high-dimensional data. The current algorithms for this objective, however, have poor computational complexity. One strategy is to develop an online approach for optimizing FR-BRM, which has been possible for several regularized matrix factorization problems (Warmuth and Kuzmin 2008; Mairal et al. 2010). Generally, however, there has been little development of online algorithms for regularized matrix factorization; this is likely the most crucial research direction for making FR-BRM a practical option. I would argue, however, that the computational complexity of the representation learning component itself is not as crucial as producing and using the representation. Representation learning could be viewed as a difficult, "life-long" problem: a representation for an observation must be produced quickly for the value function learner to make predictions quickly, but the representation basis/function itself can be improved more slowly in the background. This *representation hypothesis* does not negate the need for tractable approaches, but does suggest a direction to focus computational efforts.

Finally, it is important to consider issues with using the MSPBE for learning or selecting representations. Previous work combined the MSPBE and ℓ_1 regularization on the value function parameters (Kolter and Ng 2009; Painter-Wakefield and Parr 2012). This regularization, however, is not balanced with prediction quality, since MSPBE prediction quality is not tied to the representation. In fact, the optimal choice of value function parameters zeros all but one entry, since the ℓ_1 loss will be minimal and minimizing the MSPBE for the single feature gives zero loss.

Overall, formalizing representation learning as a matrix factorization facilitates extending recent and upcoming advances in unsupervised learning to the reinforcement learning setting. The generality of the approach and easy to understand optimization make it a promising direction for representation learning in reinforcement learning.

Acknowledgements

This work was supported by grants from Alberta Innovates Technology Futures and the National Science and Engineering Research Council of Canada.

References

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

Bach, F.; Mairal, J.; and Ponce, J. 2008. Convex sparse matrix factorizations. *arXiv.org*.

Baird, L. 1995. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*.

Bradtke, S. J., and Barto, A. G. 1996. Linear least-squares algorithms for temporal difference learning. *Machine Learning*.

De la Torre, F. 2012. A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Farahmand, A. M.; Ghavamzadeh, M.; and Szepesvári, C. 2008. Regularized policy iteration. In *Advances in Neural Information Processing Systems*.

Fard, M. M.; Grinberg, Y.; Farahmand, A. m.; Pineau, J.; and Precup, D. 2013. Bellman error based feature generation using random projections on sparse spaces. *Advances in Neural Information Processing Systems*.

Ghavamzadeh, M.; Lazaric, A.; Maillard, O. A.; and Munos, R. 2010. LSTD with random projections. In *Advances in Neural Information Processing Systems*.

Kolter, J., and Ng, A. 2009. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Konidaris, G.; Osentoski, S.; and Thomas, P. S. 2011. Value function approximation in reinforcement learning using the Fourier basis. In *Proceedings of the 25th international conference on Machine learning*.

Loth, M.; Davy, M.; and Preux, P. 2007. Sparse temporal difference learning using LASSO. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*.

Mahadevan, S., and Maggioni, M. 2007. Proto-value functions: a Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning*.

Mahmood, A. R., and Sutton, R. 2013. Representation search through generate and test. In *Proceedings of the AAAI Workshop on Learning Rich Representations from Low-Level Sensors*.

Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2010. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*.

Menache, I.; Mannor, S.; and Shimkin, N. 2005. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with deep reinforcement learning. *arXiv.org*.

Nguyen, T.; Li, Z.; Silander, T.; and Yun Leong, T. 2013. Online feature selection for model-based reinforcement learning. *Journal on Machine Learning*.

Painter-Wakefield, C., and Parr, R. 2012. Greedy algorithms for sparse reinforcement learning. *Proceedings of the 29th International Conference on Machine Learning*.

Parr, R.; Li, L.; Taylor, G.; and Painter-Wakefield, C. 2008. An analysis of linear models linear value function approximation and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*.

Ratitch, B., and Precup, D. 2004. Sparse distributed memories for on-line value-based reinforcement learning. In *Machine Learning: ECML 2004*.

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

Scherrer, B. 2010. Should one compute the Temporal Difference fix point or minimize the Bellman Residual? The unified oblique projection view. In *Proceedings of the 27th International Conference on Machine Learning*.

Stanley, K. O., and Miikkulainen, R. 2002. Efficient evolution of neural network topologies. In *Proceedings of the 2002 Congress on Evolutionary Computation*.

Sutton, R., and Whitehead, S. 1993. Online learning with random representations. In *Proceedings of the 10th International Conference on Machine Learning*.

Sutton, R.; Maei, H.; Precup, D.; and Bhatnagar, S. 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. *Proceedings of the 26th International Conference on Machine Learning*.

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

Warmuth, M. K., and Kuzmin, D. 2008. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*.

White, M., and Schuurmans, D. 2012. Generalized optimal reverse prediction. In *Proceedings of the 15th International Conference on Artifical Intelligence and Statistics*.

White, M.; Yu, Y.; Zhang, X.; and Schuurmans, D. 2012. Convex multi-view subspace learning. In *Advances in Neural Information Processing Systems*.

Whiteson, S.; Taylor, M. E.; and Stone, P. 2007. Adaptive tile coding for value function approximation. Technical report, University of Texas at Austin.

Xu, L.; White, M.; and Schuurmans, D. 2009. Optimal reverse prediction: a unified perspective on supervised, unsupervised and semi-supervised learning. In *Proceedings of the 26th International Conference on Machine Learning*.

Zhang, X.; Yu, Y.; White, M.; Huang, R.; and Schuurmans, D. 2011. Convex sparse coding, subspace learning, and semi-supervised extensions. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*.

Zhang, X.; Yu, Y.; and Schuurmans, D. 2012. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems*.

Appendix

List of known convex induced regularizers

The introduced matrix factorization approach for representation learning formalized the approach using a constraint set on B and a regularizer on Φ. Interestingly, it can equivalently be formulated without constraints and instead regularizers on both parameters (Bach et al. 2008).

Regardless of the choice, the list of tractable induced norms remains the same. The following constitute the known list of regularizers and constraint set options that result in an efficient, closed-form induced regularizer on Z:

1. The regularizer $||\Phi||_{1,1}$ is chosen for sparsity. For $\mathcal{C} = \{C : ||C_{i,:}||_q \leq 1\}$, the induced norm is $||Z^T||_{q,1}$. For $\mathcal{C} = \{[\mathbf{w}\ B] : ||\mathbf{w}||_{q_1} \leq 1, ||B_{i,:}||_{q_2} \leq \beta\}$, the induced norm on Z is $\sum_j \max\left(||Z_1^T||_{1,q_1}, \frac{1}{\beta}||Z_2^T||_{1,q_2}\right)$. Previously, these induced norms led to trivial vector quantization solutions (Zhang et al. 2011); this issue needs to be further understood if this regularizer is chosen for FR-RL.
2. The regularizer $||\Phi^\top||_{2,1}$ is chosen for subspace learning. For $\mathcal{C} = \{C : ||C_{i,:}||_2 \leq 1\}$, the induced norm is $||Z||_{\text{tr}}$. For $\mathcal{C} = \{[\mathbf{w}\ B] : ||\mathbf{w}||_2 \leq \beta_1, ||B_{i,:}||_2 \leq \beta 2\}$, the induced norm on Z is $\max_{0\leq\eta\leq 1} ||ZE_\eta^{-1}||_{\text{tr}}$ where
$$E_\eta := \begin{bmatrix} \beta_1/\sqrt{\eta} & 0 \\ 0 & \beta_2/\sqrt{1-\eta}\, I_n \end{bmatrix}.$$
.
3. The regularizer $||\Phi^\top||_{p,1}$ can be useful to push down large values. The ℓ_∞ norm is used to bound maximum values, and as p gets larger, ℓ_p approaches the ℓ_∞ norm. For $1 < p < 2$, we could also imagine some blended behaviour between $p = 1$ and $p = 2$. In general, however, $p \neq 1, 2, \infty$ is not commonly used. If it is chosen, then for $\mathcal{C} = \{C : ||C_{i,:}||_1 \leq 1\}$, the induced norm is $||Z^\top||_{p,1}$