Integrating Representation Learning and Temporal Difference Learning: A Matrix Factorization Approach

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 least-squares temporal difference learning (LSTD), with a focus on making the assumptions on the representation explicit and making the learning problem amenable to principled optimization techniques. We reformulate LSTD as a least-squares loss plus concave regularizer, facilitating 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, Osentoski, and Thomas 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, Davy, and Preux 2007), feature selection based on the Bellman error (Parr et al. 2008) 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, Ghavamzadeh, and Szepesvári 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, Taylor, and Stone 2007), as well as explicit objectives, such as maximizing likelihood of parameters for basis functions (Menache, Mannor, and Shimkin 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 al-

gorithms into one framework (Xu, White, and Schuurmans 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, Mairal, and Ponce 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 LSTD 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 contributions are

1. a novel formulation of LSTD as the combination of a least-squares loss and concave regularizer on the value function parameters (giving a new loss called the CRTD);
2. an explicit joint optimization over the value function parameters and the representation that is amenable to known optimization techniques.

The resulting approach removes the need for matrix inversion that can be a problem in LSTD, since the algorithm is a stochastic minimization of an objective function. Moreover, the representation learning component remains general, since matrix factorization encompasses many options for the representation depending on the chosen constraints.

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 (S, A, P, R) where S is the set of states; 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 $\pi : S \times A \to [0, 1]$ 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:

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

for $\mathbf{w} \in \mathbb{R}^k$ 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), 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, therefore focus on finding an approximate solution to the Bellman equation, despite this representation issue.

Least-Squares Temporal Difference Learning

LSTD finds the minimum of the mean-square projected Bellman error (MSPBE) (Sutton et al. 2009):

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

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

For simplicity, we first present LSTD and our representation learning extension assuming that we have the transition model and reward function. We describe how to move to a trajectory of samples in the last section.

The closed-form solution for this loss is the solution to the following linear system (Bradtke and Barto 1996):

$$\mathbf{w} = \left(\Phi^T D\Phi\right)^{-1} \Phi^T D\left(R + \gamma\Phi'\mathbf{w}\right)$$
$$\implies \underbrace{\Phi^T D\left(\Phi - \gamma P^\pi \Phi\right)}_{A} \mathbf{w} = \underbrace{\Phi^T DR}_{b}$$

Given samples, LSTD forms approximations to the matrices A and b and solves the system $w = A^{-1}b$.

Factorized representation learning for LSTD

We now show that LSTD corresponds to the minimization of a squared loss plus a concave regularizer.

$$\min_{\mathbf{w}} ||\Phi\mathbf{w} - R||_D^2 - 2\gamma\mathbf{w}^T\Phi^T DP^\pi\Phi\mathbf{w}$$
$$\nabla_{\mathbf{w}} = (\Phi^T D\Phi)\mathbf{w} - \Phi^T DR - \gamma\Phi^T DP^\pi\Phi\mathbf{w} = 0$$
$$\implies (\Phi^T D\Phi - \gamma\Phi^T DP^\pi\Phi)\mathbf{w} = \Phi^T DR$$
$$\implies \Phi^T D(\Phi - \gamma P^\pi\Phi)\mathbf{w} = \Phi^T DR \quad \text{(LSTD)}$$

We call $||\Phi\mathbf{w} - R||_D^2 - 2\gamma\mathbf{w}^T\Phi^T DP^\pi\Phi\mathbf{w}$ the Concave-Regularized-TD (CRTD) loss and distinguish it from the MSPBE because they are not equivalent, even though the minimization of the two losses results in the same solution (see the Appendix for the difference). Unfortunately, the minimization over $\mathbf{w}$ for this loss is not a convex optimization, since $f(\mathbf{w}) = \mathbf{w}^T\Phi^T DP^\pi\Phi\mathbf{w}$ is convex (making the negative of the function concave)[1].

This form does, however, facilitate specifying 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:

$$\min_{\Phi\in\mathbb{R}^{n\times k}, B\in\mathcal{B}} L(\Phi B, X) + \alpha||\Phi||$$

where L is any convex 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 is the weight on the regularizer. For example, X could be all cross products of the observations, and Φ could be a subset of these expanded features. The structure of the learned representation Φ, depends on the chosen regularizer, $||\cdot||$. For example, $||\Phi||_{1,1}$ imposes sparsity and $||\Phi||_{1,2}$ 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.

We obtain the following Factorized-Representation CRTD (FR-CRTD) optimization, where we overload f, for convenience:

$$\min_{\mathbf{w},\Phi,B\in\mathcal{B}} ||\Phi\mathbf{w} - R||_D^2 - \gamma f(\mathbf{w},\Phi) + L(\Phi B, X) + \alpha||\Phi||$$

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 LSTD.

Improved optimization for FR-CRTD

Let $U = [\mathbf{w}\ B] \in \mathcal{U}$, where $\mathcal{U}$ is a constraint set on U. We use a change of variables, $Z_1 = \Phi\mathbf{w}, Z_2 = \Phi B$ to obtain a simpler optimization.

$$\min_{U=[\mathbf{w}\ B]\in\mathcal{U},\Phi} \begin{array}{r} ||\Phi\mathbf{w} - R||_D^2 - \gamma\mathbf{w}^T\Phi^T DP^\pi\Phi\mathbf{w} \\ +L(\Phi B, X) + \alpha||\Phi|| \end{array} \quad (3)$$

$$\equiv \min_{Z=[Z1\ Z2]} ||Z_1 - R||_D^2 - \gamma Z_1^T DP^\pi Z_1 + L(Z_2, X) + \alpha\|\!|Z|\!\|^*$$

where

$$\|\!|Z|\!\|^* = \min_{U\in\mathcal{U}} \min_{\Phi:\Phi U=Z} ||\Phi||$$

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

$$\min_Z L(Z, Y) - \gamma\mathbf{e}_1^T Z^T DP^\pi Z\mathbf{e}_1 + \alpha\|\!|Z|\!\|^* \quad (4)$$

where L now contains both the loss between ΦB and X and the loss between $\Phi\mathbf{w}$ and R.

Recent advances in (semi-supervised) matrix factorization (Bach, Mairal, and Ponce 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, $\mathcal{U}$. See the Appendix for a list of efficiently computable convex induced regularizers on Z. Though this list is currently quite restricted, FR-CRTD 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-CRTD.

The resulting optimization over $Z = [Z1\ Z2]$ is the addition of a convex problem with a concave regularizer. Though this optimization seems difficult, minimization of concave-convex problems has been studied, which we leverage in the next section to find an efficient optimization approach.

Once we obtain Z, we can use a boosting procedure to recover the parameters U and Φ (Zhang, Yu, and Schuurmans 2012). For certain settings, it is more simple; for example, for $\mathcal{U} = \{U : ||U_{i,:}||_2 \le 1\}$ and $||\Phi||_{1,2}$ the recovery is simply a singular value decomposition: for $Z = Q\Sigma M^T$ with Q and M orthonormal and Σ a diagonal matrix of singular values, $U = M^T$ and $\Phi = Q\Sigma$. Since the recovery procedures rely only on the regularizer on Φ, they apply to this setting despite the addition of the concave component.

Minimizing the concave-convex FR-CRTD loss

Once we have this concave-convex form, there are several optimization strategies that we can explore. First, we can use the concave-convex procedure (CCCP) (Yuille and Rangarajan 2002). CCCP linearizes the concave component on each iteration, to make the problem convex (Sriperumbudur and Lanckriet 2009). In this setting, the CCCP algorithm that minimizes (4) would be:

$$Z^{(l+1)} \in \arg\min_Z L(Z, Y) - \mathbf{1}^T Z \circ \nabla R(Z^{(l)})\mathbf{1}$$

where $R(Z) = \gamma\mathbf{e}_1^T Z^T DP^\pi Z\mathbf{e}_1$, $\circ$ is the component-wise product and the inner product with $\mathbf{1}$ on either sides sums all the entries. Because energy functions can be decomposed into a convex plus concave function (Yuille and Rangarajan 2002), there may also be more specific algorithms available to more efficiently solve our concave-convex problem.

Another potential option is to reformulate the objective to enable use of global solution methods that minimize a concave objective with convex constraints (Hoffman 1981). This would require formulating the equivalent constrained

[1] $f(\mathbf{w})$ is convex since it is the composition of a linear function, $\Phi\mathbf{w}$, and a convex function, the ℓ_2 norm.

optimization, which moves the convex component of the loss to a constraint: $\min_{Z:L(Z,Y)+\alpha\|Z\|^*-c\leq 0} -R(Z)$. The effectiveness of this optimization approach is left for future research, but could be a promising avenue for convex, generalized representation learning for reinforcement learning.

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)

$$\bar{\Phi} \equiv \begin{bmatrix} \phi(s_1)^T \\ \phi(s_2)^T \\ \vdots \\ \phi(s_t)^T \end{bmatrix}, \bar{\Phi}' \equiv \begin{bmatrix} \phi(s'_1)^T \\ \phi(s'_2)^T \\ \vdots \\ \phi(s'_t)^T \end{bmatrix}, \bar{R} \equiv \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_t \end{bmatrix}$$

For LSTD, we can express the closed form solution in terms of these approximate matrices:

$$\mathbf{w} = \left(\bar{\Phi}^T\bar{\Phi}\right)^{-1}\bar{\Phi}^T\left(\bar{R} + \gamma\bar{\Phi}'\mathbf{w}\right)$$

Importantly, it has been shown that, as $t \to \infty$, the fixed point for this approximate problem converges to the fixed point of the original problem (2), with probability one (Bradtke and Barto 1996).

Moving to samples is more complicated for FR-CRTD, since we cannot use the matrix of next features, Φ'. To avoid learning Φ', we need $\hat{P}^\pi$ such that $\hat{P}^\pi\hat{\Phi} = \hat{\Phi}'$. Fortunately, this linear transformation is quite simple in practice, since $\hat{\Phi}'$ is $\hat{\Phi}$ shifted by one index. Of course, we do not have access to the last vector in $\hat{\Phi}'$, but we can simply drop that last sample as a reasonable approximation to the loss.

Define

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

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

$$\min_Z L(Z, \hat{Y}) - \gamma\mathbf{e}_1^T Z^T \hat{P}^\pi Z\mathbf{e}_1 + \alpha\|Z\|^*$$

The samples, unfortunately, will not always be perfectly aligned or in order, such as is the case when multiple episodes or trajectories are obtained. Again, we can ignore constraints across boundaries, but in general, the problem of estimating $\hat{P}^\pi$ is an important avenue for future work.

Discussion

Several interesting questions arise from viewing LSTD and representation learning under the FR-CRTD 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 (3). 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 LSTD by generalizing the least-squares loss on the reward prediction to any convex loss in Equation (4). If we choose a Bregman divergence, for example, this generalization suggests certain distributional assumptions on the reward (White and Schuurmans 2012). The relationship to the original fixed-point problem, however, becomes unclear and requires further exploration.

Second, it is important to notice that FR-CRTD maintains the fixed-point interpretation of LSTD. A complaint about the sparse LASSO approach to LSTD (Loth, Davy, and Preux 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 inner optimization, we are simply doing an LSTD outer optimization.

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-CRTD, 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-CRTD a practical option.

Finally, viewing LSTD as a least-squares loss plus concave regularizer provides a new intuition. Maximizing the inner product corresponds to finding vectors pointing in the same direction. LSTD, therefore, is balancing minimizing the angle between the current and next state values and predicting the reward for the current state.

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*.

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*.

Hoffman, K. L. 1981. A method for globally minimizing concave functions over convex sets. *Mathematical Programming*.

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 Twenty-Fifth AAAI Conference on Artificial Intelligence*.

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*.

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 Twenty-Fifth 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*.

Sriperumbudur, B. K., and Lanckriet, G. 2009. On the convergence of the concave-convex procedure. In *Advances in Neural Processing Systems*.

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 Tenth 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*.

Yuille, A. L., and Rangarajan, A. 2002. The concave-convex procedure (CCCP) . In *Advances in Neural Information Processing Systems*.

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

Relationships between the CRTD and the MSPBE

Even though the minimization of the FR-CRTD is equivalent to the minimization of the MSPBE, they are themselves not equivalent. The difference becomes clear when we explicitly write out the MSPBE loss. Recall that the projection operator in the MSPBE is $\Pi = \Phi(\Phi^T D\Phi)^{-1}\Phi^T D$, the Bellman operator is $TV = (R+\gamma PV)$ and the $MSPBE(V) = ||V-\Pi TV||_D^2$ (Sutton et al. 2009). We will use the following simplifications

$$D\Phi(\Phi^T D\Phi)^{-1}\Phi^T D = D \qquad \text{since } \Phi^T(D\Phi(\Phi^T D\Phi)^{-1}\Phi^T D) = \Phi^T D \tag{5}$$

$$\Phi^T D\Pi = \Phi^T D \qquad \text{since } \Phi^T D(\Phi(\Phi^T D\Phi)^{-1}\Phi^T D) = \Phi^T D \tag{6}$$

$$\Pi^T D\Phi = D\Phi \qquad \text{since } \Pi^T D\Phi = (\Phi^T D\Pi)^T = (\Phi^T D)^T = D\Phi \tag{7}$$

$$\Pi^T D\Pi = D \qquad \text{since } \Pi^T = D\Phi(\Phi^T D\Phi)^{-1}\Phi^T \text{ giving} \tag{8}$$

$$\Pi^T D\Pi = D\Phi(\Phi^T D\Phi)^{-1}\Phi^T D\Pi = D\Phi(\Phi^T D\Phi)^{-1}\Phi^T D = D$$

$$\mathbf{w}^T\Phi^T D(T\Phi\mathbf{w}) = (T\Phi\mathbf{w})^T D\Phi\mathbf{w} \qquad \text{since this in an inner product} \tag{9}$$

Now we can write out the MSPBE with some simplifications

$$\begin{aligned}
MSPBE(\mathbf{w}) &= ||\Phi\mathbf{w} - \Pi(T\Phi\mathbf{w})||_D^2 \\
&= (\mathbf{w}^T\Phi^T - (T\Phi\mathbf{w})^T\Pi^T)D(\Phi\mathbf{w} - \Pi T\Phi\mathbf{w}) \\
&= \mathbf{w}^T\Phi^T D\Phi\mathbf{w} - \mathbf{w}^T\underbrace{\Phi^T D\Pi}_{\Phi^T D}T\Phi\mathbf{w} - (T\Phi\mathbf{w})^T\underbrace{\Pi^T D\Phi}_{D\Phi}\mathbf{w} + (T\Phi\mathbf{w})^T\underbrace{\Pi^T D\Pi}_{D}T\Phi\mathbf{w} \\
&= \mathbf{w}^T\Phi^T D\Phi\mathbf{w} - 2\mathbf{w}^T\Phi^T D(R+\gamma P\Phi\mathbf{w}) + (R^T + \gamma\mathbf{w}^T\Phi^T P^T)D(R+\gamma P\Phi\mathbf{w}) \\
&= \mathbf{w}^T\Phi^T D\Phi\mathbf{w} - 2\mathbf{w}^T\Phi^T DR - 2\gamma\mathbf{w}^T\Phi^T DP\Phi\mathbf{w} + R^T DR \\
&\quad + \gamma R^T DP\Phi\mathbf{w} + \gamma\mathbf{w}^T\Phi^T P^T DR + \gamma^2\mathbf{w}^T\Phi^T P^T DP\Phi\mathbf{w}
\end{aligned}$$

Notice that the first part of the last equality is actually the CRTD! So, we can write

$$MSPBE(\mathbf{w}) = CRTD(\mathbf{w}) + 2\gamma\mathbf{w}^T\Phi^T P^T DR + \gamma^2\mathbf{w}^T\Phi^T P^T DP\Phi\mathbf{w}$$

Since the minimization of CRTD leads to the same solution as minimizing the MSPBE (as their minimum both corresponds to the LSTD solution), this last factor should not influence the chosen $\mathbf{w}$. Interestingly, these last components are on the next state value (i.e. $P\Phi\mathbf{w}$), so intuitively, it is possible that they may not be needed for choosing the best params for this state value.

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, Mairal, and Ponce 2008).

Regardless of the choice, the list of tractable induced norms remains the same. The following constitute some of the known (and used) regularizer 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 $\mathcal{U} = \{U : ||U_{i,:}||_q \leq 1\}$, the induced norm is $||Z^T||_{q,1}$. For $\mathcal{U} = \{[\mathbf{w}\ B] : ||\mathbf{w}||_{q_1} \leq 1, ||B_{i,:}||_{q_2} \leq \beta\}$, 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 lead to trivial vector quantization solutions (Zhang et al. 2011); with the concave regularizer, this may no longer be the case.
2. The regularizer $||\Phi||_{1,2}$ is chosen for subspace learning. For $\mathcal{U} = \{U : ||U_{i,:}||_2 \leq 1\}$, the induced norm is $||Z||_{\text{tr}}$. For $\mathcal{U} = \{[\mathbf{w}\ B] : ||\mathbf{w}||_2 \leq 1, ||B_{i,:}||_2 \leq \beta\}$, the induced norm on Z is $\max_{0\leq\eta\leq 1}||Z||_{\text{tr}}$.
3. The regularizer $||\Phi||_{1,p}$ 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 $\mathcal{U} = \{U : ||U_{i,:}||_1 \leq 1\}$, the induced norm is $||Z||_{p,1}$