Bellman Error Based Feature Generation Using Random Projections

Mahdi Milani Fard MMILAN1@CS.MCGILL.CA
Yuri Grinberg YGRINB@CS.MCGILL.CA
Joelle Pineau JPINEAU@CS.MCGILL.CA
Doina Precup DPRECUP@CS.MCGILL.CA
School of Computer Science, McGill University

1. Introduction

The accuracy of parametrized policy evaluation depends on the quality of the features used for estimating the value function. Hence, feature generation/selection in reinforcement learning (RL) has received a lot of attention (Di Castro and Mannor, 2010). We focus on methods that aim to generate features in the direction of the Bellman error of the current value estimates (Bellman Error Based, or BEBF, features). Successive addition of exact BEBFs has been shown to reduce the error of a linear value estimator at a rate similar to value iteration (Parr et al., 2007). However, unlike fitted value iteration (Boyan and Moore, 1995), which works with a fixed feature set, iterative BEBF generates new features and does not diverge, as long as the error in the generation does not cancel out the contraction effect of the Bellman operator (Parr et al., 2007).

A number of methods have been introduced in RL to generate features related to the Bellman error, with a fair amount of success (Geramifard et al., 2011; Di Castro and Mannor, 2010; Manoonpong et al., 2010; Parr et al., 2007; Keller et al., 2006). In this work, we use the idea of applying random projections specifically in very large and sparse feature spaces. In short, we iteratively project the original features into exponentially smaller-dimensional spaces and apply linear regression of Bellman residuals to approximate BEBFs. We carry out a finite sample analysis that helps determine the optimal size of the projections and the number of iterations.

We focus on spaces that are large, bounded and k-sparse: at any state only k of the features are non-zero, in some known or unknown basis. Such spaces occur both naturally (e.g. image, audio and video signals) and also from most discretization-based methods (e.g. tile-coding). For simplicity, we assume that regardless of the current estimate of the value function $\hat{V}$, the Bellman error is always linearly representable in the feature space. This seems like a strong assumption, but is true, for example, in virtually any tile-coded space (the number of representable states is smaller than the number of features).

2. Random Projections Preserve Linearity

Our first contribution (tightening the bound in Fard et al. (2012)) shows that if a function is linear in a sparse space, it is almost linear in an exponentially smaller projected space:

Theorem 1 *Let $\Phi^{D\times d}$ be a random projection according to: $\phi_{i,j} \sim \mathcal{N}(0, 1/d)$. Let $\mathcal{X}$ be a D-dimensional k-sparse space. Then for any fixed w and $\epsilon > 0$:*

$$\forall x \in \mathcal{X} : \left|\langle \Phi^T w, \Phi^T x\rangle - \langle w, x\rangle\right| \leq \epsilon \|w\|\|x\|, \tag{1}$$

fails with probability less than $(4D + 2)e^{-d\epsilon^2/48k}$.

Hence, projections of size $O(k \log D)$ preserve the linearity up to an arbitrary constant. Note that this is a worst-case bound on the entire space.

3. Compressed Linear BEBFs

We assume a finite MDP, with bounded stochastic reward of mean $\mathbf{R}$, and transition matrix $\mathbf{P}$ for a fixed policy. Let $\mathbf{V}$ be the vector of values assigned to the states. Let $\mathcal{T}$ be the Bellman operator: $\mathcal{T}\mathbf{V} \stackrel{\text{def}}{=} \mathbf{R} + \gamma\mathbf{P}\mathbf{V}$. The Bellman error is defined as the difference between the value function and the result of the Bellman operator applied on the value: $BE(\mathbf{V}) \stackrel{\text{def}}{=} \mathcal{T}\mathbf{V} - \mathbf{V}$.

Linear function approximators can be used to estimate the value of a given state. Let $\hat{\mathbf{V}}_m$ be an estimated value function described in a linear space defined by a feature set $\{\xi_1, \ldots \xi_m\}$. Parr et al. (2007) show that if we add a new BEBF $\xi_{m+1} = BE(\hat{\mathbf{V}}_m)$ to the feature set, (with mild assumptions) the approximation error on the new linear space shrinks by a factor of γ. They also show that if we can estimate the Bellman error within a constant angular error, $\cos^{-1}(\gamma)$, the error will still shrink. In light of this result, we propose the following (simplified) algorithm:

Algorithm 1: Compressed BEBFs

Input: Trajectory $\mathbf{x}_1, r_1, \mathbf{x}_2, r_2 \ldots$, where $\mathbf{x}_t$ is the observation received at time t, and r_t is the reward; Number of BEBFs: m; Projection size schedule: $d_1, d_2, \ldots, d_m$

Output: $\mathbf{V}$, the approximated value function

$\mathbf{V} \leftarrow 0$;

for $i \leftarrow 1$ **to** m **do**

Generate random projection $\Phi^{D \times d_i}$;

Approximate the Bellman residuals by the TD-errors, δ_t, computed at each step;

Let $\mathbf{y}_t$ be the result of ordinary least-squares regression using $\Phi^T \mathbf{x}_t$ as inputs and δ_t as outputs.

Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{y}_t$;

end

Theorem 1 suggests that if the Bellman error is linear in the original features, the bias due to the projection can be bounded within a fixed angular error with logarithmic size projections. With proper mixing assumptions, the on-sample variance of the estimator can also be bounded if the trajectory size is on the order of the projected dimension size, using Azuma's inequality on the martingale induced by Bellman residuals (like in Ghavamzadeh et al. (2010) and Grinberg et al. (2011)). If the Markov chain "forgets" exponentially fast, one can even bound the off-sample *worst-case* variance part of the error by a constant angular error with similar sizes of sampled transitions (Samson, 2000). In the poster we will also discuss empirical results using this method.

References

J. Boyan and A.W. Moore. Generalization in reinforcement learning: Safely approximating the value function. *NIPS*, pages 369–376, 1995.

D. Di Castro and S. Mannor. Adaptive bases for reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pages 312–327, 2010.

M.M. Fard, Y. Grinberg, and D. Pineau, J.and Precup. Compressed least-squares regression on sparse spaces. In *AAAI*, 2012.

A. Geramifard, F. Doshi, J. Redding, N. Roy, and J.P. How. Online discovery of feature dependencies. In *ICML*, pages 881–888, 2011.

M. Ghavamzadeh, A. Lazaric, O.A. Maillard, and R. Munos. LSTD with random projections. In *NIPS*, 2010.

Y. Grinberg, M.M. Fard, and J. Pineau. LSTD on sparse spaces. In *NIPS Workshop on New Frontiers in Model Order Selection*, 2011.

P.W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *ICML*, pages 449–456, 2006.

P. Manoonpong, F. Wörgötter, and J. Morimoto. Extraction of reward-related feature space using correlation-based and reward-based learning methods. *Neural Information Processing. Theory and Algorithms*, pages 414–421, 2010.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value-function approximation. In *ICML*, pages 737–744, 2007.

P.M. Samson. Concentration of measure inequalities for markov chains and phi-mixing processes. *The Annals of Probability*, 28(1):416–461, 2000.