Bellman Error Based Feature Generation Using Random Projections

Mahdi Milani Fard, Yuri Grinberg

Doina Precup, Joelle Pineau

Reasoning and Learning Laboratory

No Domain Knowledge

Given: continuous 5-dim trajectory of size ~ 1000 + rewards

Asked: policy evaluations

Typical Approach: use tile-coding

Tile-coding

$((2 \times 2 \times 2 \times 2 \times 2)\text{-grid}) * 4 \rightarrow 128$ features

- can train with ~ 1000 samples
- coarse-grained $\rightarrow$ bad approximation

$((3 \times 3 \times 3 \times 3 \times 3)\text{-grid}) * 4 \rightarrow 972$ features

- barely enough samples to train
- still coarse-grained $\rightarrow$ bad approximation

Tile-coding

$((10 \times 10 \times 10 \times 10 \times 10)\text{-grid}) * 10 \rightarrow 10^6$ features

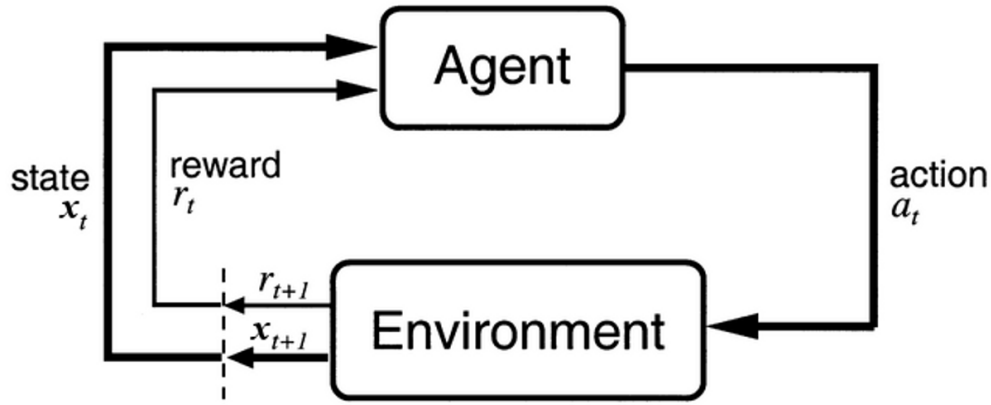
- 1000 times more features than samples
- cannot train

Tile-coding

$((10 \times 10 \times 10 \times 10 \times 10)\text{-grid}) * 10 \rightarrow 10^6$ features

- 1000 times more features than samples
- cannot train
- or can we?

Reinforcement Learning



Policy: $\pi(x)$

Transition kernel: $T(x,a)$

Value function: $V^{\pi}(x) \equiv \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$

Reinforcement Learning

Bellman Operator:

$$\mathcal{T}V(\mathbf{x}) = R(\mathbf{x}, \pi(\mathbf{x})) + \gamma \int V(\mathbf{y}) T(d\mathbf{y}|\mathbf{x}, \pi(\mathbf{x}))$$

Bellman Error:

$$e_V = \mathcal{T}V - V$$

$$(\mathcal{T} \ldots (\mathcal{T}(\mathcal{T}V))) \to V^{\pi}$$

Bellman Error Based Feature Generation

Simplified Version:

- Estimate: $\hat{e}_V \simeq \mathcal{T}V - V$
- Update: $V' = V + \hat{e}_V \simeq \mathcal{T}V$

Bellman Error Based Feature Generation

General algorithm (e.g. Parr et al. 2007):

- Given V_m built on a feature set $\{\psi_1, \ldots \psi_m\}$
- Generate BEBF $\psi_{m+1} \simeq \mathcal{T} V_m - V_m$
- Add ψ_{m+1} to the feature set
- Estimate the value V_{m+1} on the new feature space

Bellman Error Based Feature Generation

Convergence (Parr et al. 2007):

$$\psi_{m+1} = \text{Bellman Error} \quad \Rightarrow \quad \|V_{m+1} - V^{\pi}\| \leq \gamma \|V_m - V^{\pi}\|$$

$$\psi_{m+1} \simeq \text{Bellman Error} \quad \Rightarrow \quad \text{Error still contracts}$$

BEBFs in High Dimensional State Spaces

Problem:

Difficult to estimate e_V in high dimensional state spaces

Proposed Solution:

Use random projections to compress the space

Regress BEBF in the compressed space

Linear operator $\Phi^{D \times d}$ where $\Phi_{i,j} \sim \mathcal{N}(0, 1/d)$

Sparsity Assumption

States are in a k-sparse compact subspace of $\mathbb{R}^D$:

$$\mathcal{X} \triangleq \{\Psi \mathbf{z}, \text{ s.t. } \|\mathbf{z}\|_0 \leq k \text{ and } \|\mathbf{z}\| \leq 1\}$$

E.g. tile-coding, audio/image/video data, etc...

Random Projections and Sparse Spaces

Random projections of size $O(k \log D)$ preserve linearity for sparse spaces. (*Fard et al. 2012*)

For any fixed $\mathbf{w} \in \mathbb{R}^D$ and any k-sparse space $\mathcal{X}$:

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

fails with probability $< e^{O(\log(D) - d\epsilon^2/k)}$

Simplified Compressed BEBF algorithm

Input: Number of BEBFs: m; Schedule: $d_1, d_2, \ldots, d_m$

Output: $\mathbf{w} \in \mathbb{R}^D$, the linear coefficient of the approximator

$\mathbf{w} \leftarrow 0$;

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

Calculate TD-errors: $\delta_t = r_t + \gamma \mathbf{x}_{t+1}^T \mathbf{w} - \mathbf{x}_t^T \mathbf{w}$;

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

Apply OLS in the compressed space: $\mathbf{w}_{\text{ols}}^{(\Phi)} = (\mathbf{X}\Phi)^{\dagger}\delta$;

Update $\mathbf{w} \leftarrow \mathbf{w} + \Phi \mathbf{w}_{\text{ols}}^{(\Phi)}$;

end

Finite Sample Analysis

Under Review

Proper mixing with stationary dist. ρ

Bellman errors linear in features and bounded by $e_{\max}$

TD errors bounded by $\delta_{\max}$

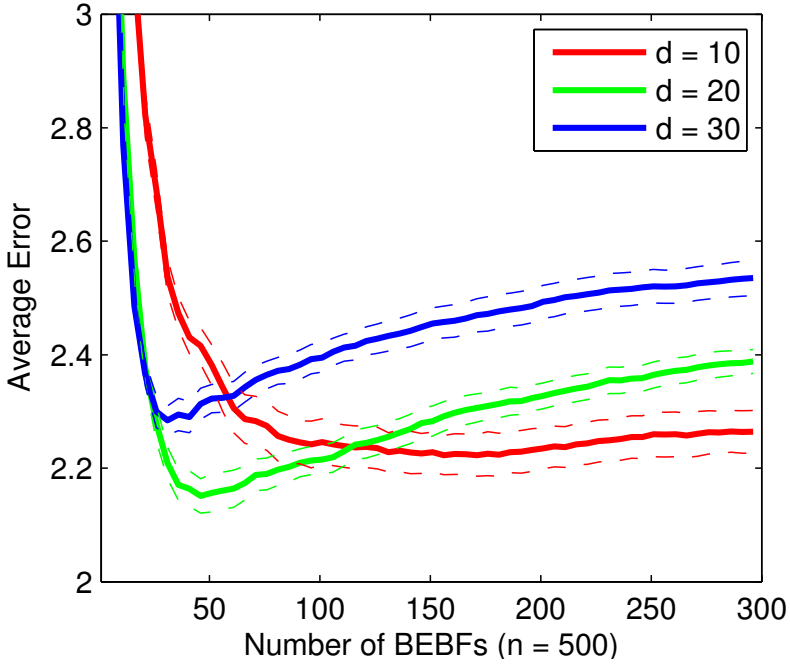
$$\left\| \mathbf{x}^T \Phi \mathbf{w}_{\text{ols}}^{(\Phi)} - e_V(\mathbf{x}) \right\|_\rho \leq \tilde{O}\left(\sqrt{k \log D} \left(e_{\max} \sqrt{\frac{1}{d}} + \delta_{\max} \sqrt{\frac{d}{n}} \right) \right)$$

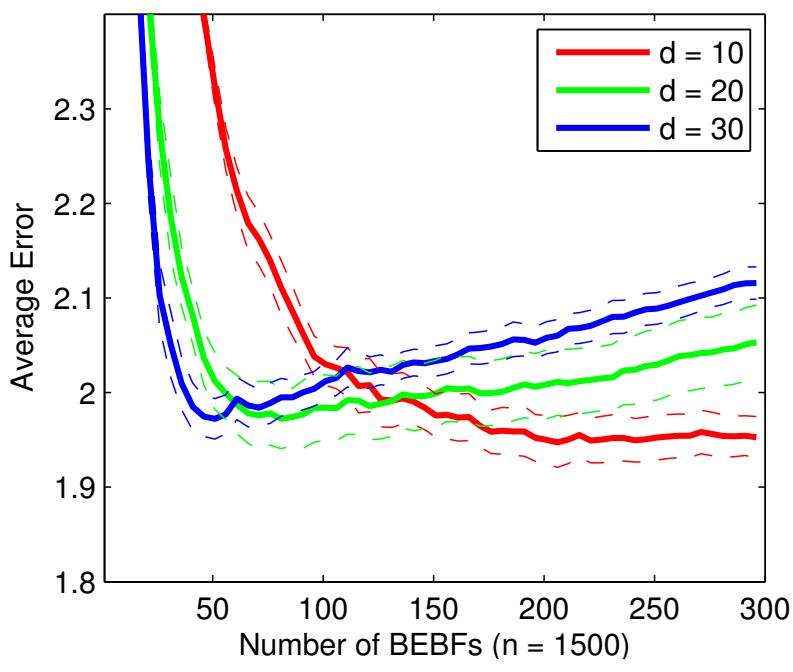
Empirical Results

- **Neurostimulation** domain: Apply direct electrical neurostimulation to suppress epileptic seizures
- **Generative model** from real-world data, stimulated at 1Hz (Bush et al. 2009)
- 6 continuous features, **tile-coded** into approx. half a million binary features, $(k = 10)$-sparse.
- Evaluate Return Prediction (RP) error on a test set
- Return U: observed discounted sum of rewards

$$e_{\mathrm{RP}} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(U(\mathbf{x}_i) - V(\mathbf{x}_i)\right)^2}$$

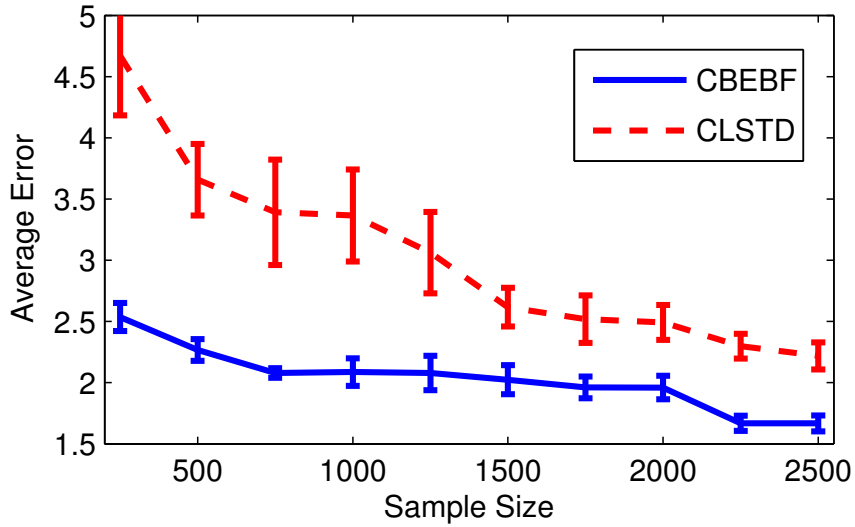
Empirical Results





Empirical Results

LSTD with random projections (CLSTD)
(Ghavamzadeh et al. 2010)



Discussion

Compressed BEBF Algorithm:

- Fast, robust to the choice of parameters
- Empirically outperforms other methods

Future work:

- Optimal projection size schedule
- Evaluate on different environments
- Apply to online RL (e.g. Policy Iteration)

Take Home Message

No Domain Knowledge

- Blow up your feature space
- Think about effective dimension
- Regularize you model

Questions?

References:

R. Parr, C. Painter, L. Li, M. Littman. Analyzing feature generation for value-function approximation. *ICML 2007*.

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

K. Bush, J. Pineau, A. Guez, B. Vincent, G. Panuccio, M. Avoli. Dynamic Representations for Adaptive Neurostimulation Treatment of Epilepsy. *IWSP 2009.*

M. M. Fard, Y. Grinberg, J. Pineau, D. Precup. Compressed Least-Squares Regression on Sparse Spaces, *AAAI 2012*

Theorem

If $e_V(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$, *with probability* $\geq 1 - (\xi_1 + \xi_2 + \xi_3 + \xi_4 + \xi_5)$:

$$\begin{aligned}
\left\|\mathbf{x}^T\Phi\mathbf{w}_{ols}^{(\Phi)} - e_V(\mathbf{x})\right\|_{\rho(\mathbf{x})} \leq \quad & \tilde{O}(n^{-2}) \\
& + \epsilon_{prj}^{(\xi_1)} m_{\max} \|\mathbf{w}\| \left(2 + m_{\max}\left\|(\mathbf{X}\Phi)^\dagger\right\| \sqrt[4]{c_1 n \log\frac{c_2}{\xi_2}}\right) \\
& + \frac{\delta_{\max} m_{\max}}{n}\left\|\Sigma_\Phi^{-1}\right\| \|\mathbf{X}\Phi\| \sqrt{2k\log\frac{2D}{\xi_3}} \\
& + \delta_{\max} m_{\max}^3 \sqrt{\frac{d^3}{n^3}}\left\|\Sigma_\Phi^{-1}\right\|^2 \|\mathbf{X}\Phi\| \sqrt{2c_3 \log\frac{c_4 d^2}{\xi_4}\log\frac{2d}{\xi_5}},
\end{aligned}$$

where $\epsilon_{prj}^{(\xi_1)} = \sqrt{\frac{48k}{d}\log\frac{4D}{\xi_1}}$, $m_{\max} = \max_{\mathbf{z}\in\mathcal{X}}\left\|\mathbf{z}^T\Phi\right\|$ *and* Σ_Φ *is the feature covariance matrix under measure* ρ.