An Analysis of Linear Models, Linear Value-Function Approximation, and Feature Selection for Reinforcement Learning

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman

DUKE COMPUTER SCIENCE

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

Preliminaries

- Markov Reward Process (MRP)
  - States – $S=[s_1 \ldots s_n]$
  - Reward – $R:S \rightarrow \mathbb{R}$
  - Transition matrix – $P[i,j]=P(s_i|s_j)$
  - Discount – $0 \le \gamma < 1$
  - Value – V(s) = expected, discounted value of state s
- True Value Function: $V^*=(I-\gamma P)^{-1}R$

Linear Value Function Approximation

- |S| typically quite large
- Pick linearly independent features $\Phi=(\phi_1 \ldots \phi_k)$ (basis functions)
- Desire weights $\mathbf{w}=w_1 \ldots w_k$, s.t.

$$V^*(s) \approx \hat{V}(s) = \sum_{i=1}^{k} w_i \phi_i(s)$$

$$\hat{V} = \Phi \mathbf{w}$$

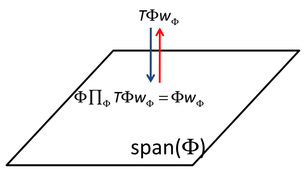
Bellman Operator

$$TV = R + \gamma PV$$

- Used in, e.g., value iteration: $V_{i+1} = R + \gamma PV_i$
- Defines fixed point: V*=TV*
- Bellman error (residual): $BE(V) = TV - V$
- BE bounds actual error: $\|V^* - V\|_\infty \le \frac{\|BE(V)\|_\infty}{1-\gamma}$

Linear Fixed Point

- Π_ΦV=weights of projection of V into span(Φ)



- LSTD, linear TD, etc. solve for the linear fixed point:

$$w_\Phi = \Pi_\Phi T \Phi w_\Phi$$

Linear Model Approximation

- Linearly independent features $\Phi=(\phi_1 \ldots \phi_k)$ (n×k)
- Want r_Φ = reward model (k×1) w/smallest L_2 error:

$$\Phi R_\Phi \approx R$$

$$R_\Phi = \Pi_\Phi R$$

- Want P_Φ = feature × feature model (k×k) w/ smallest L_2 error

$$\Phi P_\Phi \approx P\Phi$$ ← Expected next feature values (n×k)

$$P_\Phi = \Pi_\Phi P \Phi$$

Value Function of the Linear Model

- Value function is in span(Φ)
- → Can express value functions as Φw
- If V is bounded, then:

$$\underbrace{w}_{(k\times 1)} = (I - \gamma \underbrace{P_\Phi}_{(k\times k)})^{-1} \underbrace{R_\Phi}_{(k\times 1)}$$

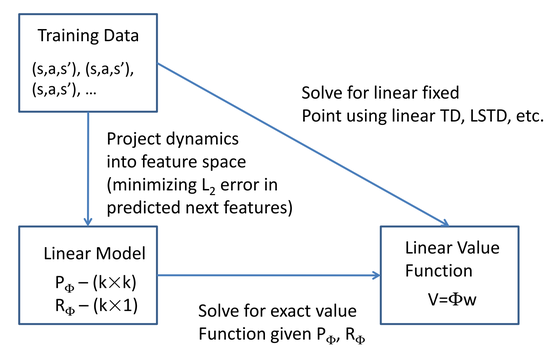
- Note similarity to conventional solution:

$$\underbrace{V^*}_{(n\times 1)} = (I - \gamma \underbrace{P}_{(n\times n)})^{-1} \underbrace{R}_{(n\times 1)}$$

Analysis

Linear Model, Linear Fixed Point Equivalence

Given: Linearly independent features $\Phi=(\phi_1 \ldots \phi_k)$



Model Error

- Linearly independent features $\Phi=(\phi_1 \ldots \phi_k)$
- Error in reward: $\Delta R = R - \Phi R_\Phi$ (n×1)
- Error in predicted next features

$$\Delta\Phi = P\Phi - \Phi P_\Phi \quad (n\times k)$$

Expected next feature values — Predicted next feature values

Bellman Error Decomposition

Theorem: $$\underbrace{BE(\Phi w_\Phi)}_{\text{Bellman error of linear fixed point solution}} = \underbrace{\Delta R}_{\text{Reward error}} + \underbrace{\gamma(\Delta\Phi)w_\Phi}_{\text{Per feature error}}$$

Insights into Feature Selection

When ΔR = 0:

- Features the reward
- BE follows from inability of features to predict next features

When ΔΦ = 0:

- Features "predict themselves"
- BE=ΔR, no dependence on γ
- Value function approximation, feature selection reduce to regression problems on R

Achieving Zero Feature Error (ΔΦ = 0)

- When are features sufficient for 0 error in expected next feature values?
  - Rank(Φ)=|S|
  - Φ is composed of eigenvectors of P
  - Φ span an invariant subspace of P
- Invariant subspace: $\exists \Lambda : P\Phi = \Phi\Lambda$

Insight into Adding Features

- Methods for adding features
- Add the Bellman error as a feature (BEBF) [Wu & Givan, 2004; Sanner & Boutilier, 2005; Keller et al., 2006), Parr et al. (2007)]
  - Add the model errors (ΔΦ,ΔR) as features (MEBF)
  - Add P^kR for increasing k (Krylov basis) Petrik (2007)]
- Theorem: BEBF, MEBF, and the Krylov basis are equivalent when initialized with Φ={}

Note: Special thanks to Marek Petrik for demonstrating Krylov=BEBF

Insight into Proto Value Functions

- Proto value functions (PVFs) compute eigenvectors of a modified adjacency graph
- Adjacency graph = approximate [Mahadevan & Maggioni]
- PVFs ~ eigenvectors of P
- ∴ PVFs viewed as approximation to subspace invariant features

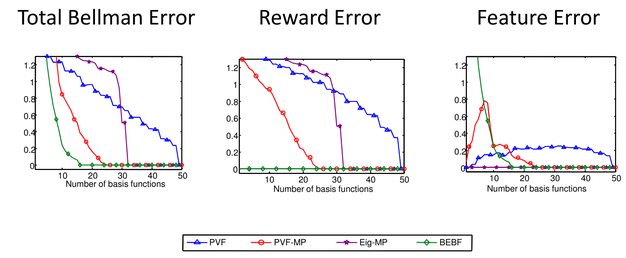
Experiments

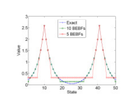
- Four Algorithms
  - PVFs (In order of "smoothness")
  - PVF-MP (Matching pursuits w/PVF dictionary)
  - eig-MP (Matching pursuits w/eigenvectors of P)
  - BEBF (AKA MEBF, Krylov basis)
- Measured (in L_2) as a function of number of basis functions added:
  - Total Bellman error
  - Reward error ΔR
  - Total feature error $\Delta\Phi w_\Phi$

$$BE(\Phi w_\Phi) = \Delta R + \gamma(\Delta\Phi)w_\Phi$$

Chain Results

Total Bellman Error — Reward Error — Feature Error

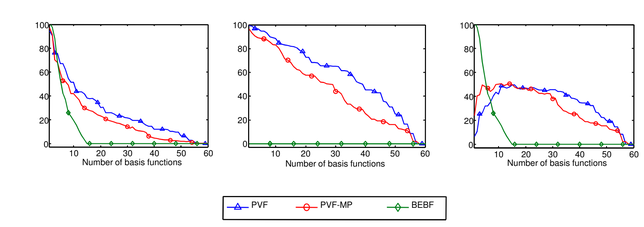


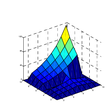


50 state chain from Lagoudakis & Parr

Two Room Results

Total Bellman Error — Reward Error — Feature Error



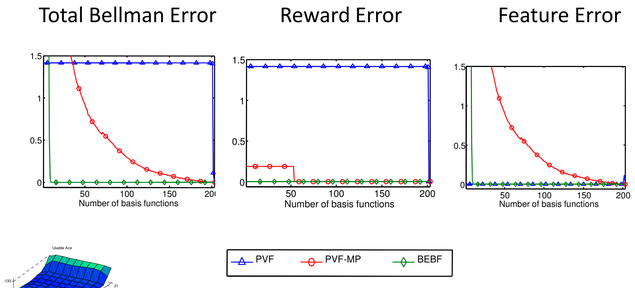


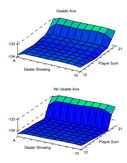
Note: No eig-mp because P is not diagonalizable.

Two Room problem from Mahadevan and Maggioni

Blackjack Results

Total Bellman Error — Reward Error — Feature Error





Note: No eig-mp because P is not diagonalizable.

Blackjack problem from Sutton & Barto

Conclusions From Experiments

- eig-mp will always have ΔΦ=0, but may not be feasible (alternatives, such as Jordan form are also problematic)
- PVFs sometimes approximate subspace invariance
- ΔΦ and ΔR can sometimes cancel (e.g., two-room)
- PVF-MP dominates PVF because PVF ignores R
- BEBF will always have ΔR=0
- BEBF has a more steady/predictable reduction in BE
- Don't ignore R!