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

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

Presented By - Vaibhav Pandey

Outline

- Linear Fixed-Point Solution = Linear-Model Solution
- Analysis of Error
- Feature Selection

Framework and Notation

- Main results and experiments consider the uncontrolled or policy evaluation case.
  - Also applicable to the controlled case
- Uncontrolled case : Markov Rewards Process

$$M = (S, P, R, \gamma)$$

- Need to find solution to the Bellman equation

$$V[s_i] = R_i + \gamma \sum_j P_{ij} V[s_j]$$

Linear Value Functions

- Common to use some form of parametric value-function approximation, such as linear combination of features or basis functions

$$\hat{V} = \sum_{i}^{k} w_i \phi_i,$$

  Where $\Phi = \{\phi_1, \ldots, \phi_k\}$ is a set of linearly independent basis functions of the state, $\phi_i(s)$ is defined as value of feature i in state s.
- We can think of $\boldsymbol{\Phi}$ as a design matrix where $\boldsymbol{\Phi}$[i, j] = $\boldsymbol{\phi}_j(s_i)$
- If weights **w** are expressed as a column vector, we have

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

Linear Value Functions

- Different methods can be used for finding **w**
  - Linear TD, LSTD and LSPE
- They all solve to the fixed point

$$\hat{V} = \Phi \mathbf{w}_\Phi = \Pi_\sigma (R + \gamma P \Phi \mathbf{w}_\Phi),$$

- $\mathbf{\Pi}_\sigma$ is the σ-weighted L_2 projection onto the column space of $\mathbf{\Phi}$
- If $\mathbf{\Sigma} = diag(\sigma)$,

$$\Pi_\sigma = \Phi(\Phi^T \Sigma \Phi)^{-1} \Phi^T \Sigma$$

- If we solve for $\mathbf{w}_\Phi$, we get

$$\begin{aligned} \mathbf{w}_\Phi &= (I - \gamma(\Phi^T \Phi)^{-1} \Phi^T P \Phi)^{-1} (\Phi^T \Phi)^{-1} \Phi^T R \\ &= (\Phi^T \Phi - \gamma \Phi^T P \Phi)^{-1} \Phi^T R. \end{aligned}$$

Linear Value Functions

- Assumption
  - P is known
  - Φ can be constructed exactly

Linear Models

- While value approximation uses features to predict values, we try to predict next features.
- We define $\Phi(s`|s)$ as random vector of next features

$$\Phi(s'|s) \overset{s' \sim P(s'|s)}{=} [\phi_1(s'), \ldots, \phi_k(s')]^T,$$

- Our goal is to calculate a *k x k* matrix P_Φ which predicts expected next feature vectors and minimizes the expected feature prediction error

$$P_\Phi{}^T \Phi(s) \approx E_{s' \sim P(s'|s)}\{\Phi(s'|s)\}$$

$$P_\Phi = \arg\min_{P_k} \sum \|P_k^T \Phi(s) - E\{\Phi(s'|s)\}\|_2^2.$$

Linear Models

- One way to solve the minimization problem in the previous slide is to compute the expected next feature explicitly as the *nxk* matrix $P\Phi$.
- We can then solve the system of equations $\Phi P_\Phi \approx P\Phi$, since ith row of ΦP_Φ is P_Φ's prediction of next feature values and the ith row of $P\Phi$ is the expected value of these features
- The least squares solution is

$$P_\Phi = (\Phi^T \Phi)^{-1} \Phi^T P\Phi,$$

- We could do a standard least squares projection to compute an approximate rewards predictor

$$r_\Phi = (\Phi^T \Phi)^{-1} \Phi^T R,$$

Outline

- Linear Fixed-Point Solution = Linear-Model Solution
- Analysis of Error
- Feature Selection

Linear Fixed-Point Solution = Linear-Model Solution

- **The uncontrolled case**
- In the approximation model
  - If **x** is a feature vector for a state, then $r_{\Phi}{}^{\mathsf{T}}\mathbf{x}$ is the reward for this state and $P_{\Phi}{}^{\mathsf{T}}\mathbf{x}$ is the next state vector.
  - Thus, the Bellman equation for state **x** is

$$V[\mathbf{x}] = r_{\Phi}{}^{T}\mathbf{x} + \gamma V[P_{\Phi}{}^{T}\mathbf{x}] = \sum_{i=0}^{\infty} \gamma^{i} r_{\Phi}{}^{T} (P_{\Phi}{}^{i})^{T}\mathbf{x}.$$

  - With respect to the original state space, the value function becomes

$$V = \Phi \sum_{i=0}^{\infty} \gamma^{i} P_{\Phi}{}^{i} r_{\Phi}$$

Linear Fixed-Point Solution = Linear-Model Solution

- Since V = Φw for some w, the fixed point equation becomes

$$\begin{aligned} V &= \hat{R} + \gamma \widehat{P\Phi}\mathbf{w} \\ \Phi\mathbf{w} &= \Phi r_\Phi + \gamma \Phi P_\Phi \mathbf{w} \\ \mathbf{w} &= (I - \gamma P_\Phi)^{-1} r_\Phi \end{aligned}$$

- This is called the linear model solution.
- A solution exists if P_Φ has a spectral radius less than $1/\gamma$.
- If the spectral radius exceeds $1/\gamma$ then the value function defined by P_Φ and r_Φ assigns unbounded value to some states.

Linear Fixed-Point Solution = Linear-Model Solution

- **Theorem**: *For any MRP M and set of features Φ the linear model solution and linear fixed-point solution are identical.*
  - This can be shown by using the linear model solution from the previous slide, and substituting the values of P_Φ and r_Φ.

$$\begin{aligned} \mathbf{w} &= (I - \gamma P_\Phi)^{-1} r_\Phi \\ &= (I - \gamma(\Phi^T\Phi)^{-1}\Phi^T P\Phi)^{-1}(\Phi^T\Phi)^{-1}\Phi^T R \\ &= \mathbf{w}_\Phi. \blacksquare \end{aligned}$$

  - The model based view gives a new perspective on error analysis and feature selection

Linear Fixed-Point Solution = Linear-Model Solution

- The controlled case : LSPI
  - Policy, $\pi : S \rightarrow A$.
- The fixed point solution to the Bellman Equation is

$$Q^{\pi}[s_i, a] = R_i^a + \gamma \sum_j P_{ij}^a Q^{\pi}[s_j, \pi(s_j)].$$

- As in the previous case, Q^{π} can be approximated as $\hat{Q} = \sum_{i=1}^{k} w_i \phi_i$
- In the policy evaluation step, LSTDQ algorithm, which is the Q version of LSTD algorithm, is used to compute Q_i.

Linear Fixed-Point Solution = Linear-Model Solution

- An MDP controlled by a fixed policy, is equivalent to an MRP with state space *SxA*.
- LSTDQ can be viewed as LSTD running over this induced MRP
- Thus the results of the uncontrolled case are valid for controlled case as well
- Therefore, the intermediate value functions, Q_i, found by LSPI are exact value functions of the respective approximate linear models with the smallest L_2 error.

Outline

- Linear Fixed-Point Solution = Linear-Model Solution
- Analysis of Error
- Feature Selection

Analysis of Error

- Uncontrolled case
  - Bellman Error $BE(\hat{V}) = R + \gamma P\hat{V} - \hat{V}$
- In context of linear value functions and linear models, Bellman Error for a set Φ of features is

$$BE(\Phi) = BE(\Phi \mathbf{w}_\Phi) = R + \gamma P \Phi \mathbf{w}_\Phi - \Phi \mathbf{w}_\Phi$$

- We define two components of the model error
  - The reward error $\Delta_R = R - \hat{R}$
  - The per-feature error $\Delta_\Phi = P\Phi - \widehat{P\Phi}$
- The per-feature error is the error in the prediction of the expected next feature values, both terms can be viewed as residual error of the linear model.

Analysis of Error

- **Theorem**: *For any MRP M and features* $\boldsymbol{\Phi}$

$$BE(\Phi) = \Delta_R + \gamma \Delta_\Phi \mathbf{w}_\Phi$$

  - Using the definitions of *BE(*$\boldsymbol{\Phi}$*)*, $\boldsymbol{\Delta}_R$ and $\boldsymbol{\Delta}_{\boldsymbol{\Phi}}$

$$\begin{aligned} BE(\Phi) &= R + \gamma P \Phi \mathbf{w}_\Phi - \Phi \mathbf{w}_\Phi \\ &= (\Delta_R + \hat{R}) + \gamma(\Delta_\Phi + \widehat{P\Phi})\mathbf{w}_\Phi - \Phi \mathbf{w}_\Phi \\ &= (\Delta_R + \gamma \Delta_\Phi \mathbf{w}_\Phi) + \hat{R} + (\gamma \Phi P_\Phi - \Phi)\mathbf{w}_\Phi \\ &= (\Delta_R + \gamma \Delta_\Phi \mathbf{w}_\Phi) + \hat{R} - \Phi(I - \gamma P_\Phi)\mathbf{w}_\Phi \\ &= (\Delta_R + \gamma \Delta_\Phi \mathbf{w}_\Phi) + \hat{R} - \Phi r_\Phi \\ &= \Delta_R + \gamma \Delta_\Phi \mathbf{w}_\Phi. \end{aligned}$$

Analysis of Error

- The decomposition of error lets us think of Bellman Error as composed of two separate sources: reward error and per-feature error.
- This view can give insight into feature selection
  - We should be cautious as there can be interactions between $\mathbf{\Delta}_R$ and $\mathbf{\Delta}_\Phi$.
- A similar result may be possible for the controlled case but there are some subtleties
  - Not explored in this paper

Outline

- Linear Fixed-Point Solution = Linear-Model Solution
- Analysis of Error
- Feature Selection

Feature Selection

- $\mathbf{\Delta}_R = \mathbf{\Delta}_\Phi = 0$ is sufficient to achieve zero Bellman Error.
- This means that features of the approximate model should be able to capture the structure of the reward function, and are sufficient to predict the expected next features.
- Features that cannot represent immediate reward are problematic
  - This increases the Bellam Error (as the first term)
- For $\mathbf{\Delta}_\Phi = 0$, Bellman error is completely determined by $\mathbf{\Delta}_R$ with no dependence on γ.

Feature Selection

- Incremental Feature Generation
  - Krylov basis
  - Bellman Error Basis Function (BEBF)
  - Proposed : Model Error Basis Function (MEBF)
  - All three methods are equivalent given same initial conditions
- Krylov Basis
  - Defined in terms of powers of transition matrix multiplied by **R.**
  - Krylov basis with k basis functions, starting from **X**
    - $Krylov_k(\mathbf{X}) = \{P^{i-1}\mathbf{X} : 1 \leq i \leq k\}$
  - For an MRP, typically **X** = **R.**

Feature Selection

- BEBFs
  - Based upon the residual error in the current feature set
  - Formally
    - If $\mathbf{\Phi}_k \mathbf{w}_{\mathbf{\Phi}k}$ is the current value function
    - Then $\boldsymbol{\phi}_{k+1} = BE(\mathbf{\Phi}_k)$ is the next basis function
  - The basis resulting from *k-1* iterations of BEBF, starting from **X,** $BEBF_k(\mathbf{X})$.

Feature Selection

- **Theorem**: For any $k \geq 1$, $span(Krylov_k(R)) = span(BEBF_k(R))$
  - Proof by induction

$$Krylov_1(R) = BEBF_1(R) = R$$

  - We assume equality up to *k*, so for both methods, the value function can be written as

$$\Phi_k \mathbf{w}_{\Phi_k} = \sum_{i=1}^{k} w_i P^{i-1} R$$

  - Bellman error is next basis function for BEBF

$$BE(\Phi_k) = R + \gamma P(\sum_{i=1}^{k} w_i P^{i-1} R) - \sum_{i=1}^{k} w_i P^{i-1} R$$

  - The only part of the above expression not already in the Krylov basis is the contribution from $\boldsymbol{P^{k+1}R}$, which is what we add in $Krylov_{k+1}(\mathbf{R})$.

Feature Selection

- MEBFs
  - Add features which capture the residual error in the model
  - Adds $\mathbf{\Delta}_R$ and $\mathbf{\Delta}_{\mathbf{\Phi}}$ to the basis at each iteration to create $\mathbf{\Phi}_{k+1}$.
  - Basis resulting from *k-1* iterations of MEBF, starting from **X,** $MEBF_k(\boldsymbol{X})$
- **Theorem**:

$$span(BEBF_2(\Phi)) \subseteq span(MEBF_2(\Phi))$$

  - This follows from the previous theorem about decomposition of error.

Feature Selection

- **Theorem:** For $k \geq 1$,

$$span(Krylov_k(R)) = span(MEBF_k(R))$$

  - Proof by induction,

$$Krylov_1(R) = MEBF_1(R) = R$$

  - We assume equality up to k, and consider the behaviour of MEBF.
  - For $k \geq 1$, $\mathbf{\Delta}_R = 0$, since R is the first basis function added.
  - Since $\mathbf{\Phi}_k$, is a collection of basis functions of form $\boldsymbol{\phi}_i = P^{i-1}R$ for $1 \leq i \leq k$.
  - Therefore $P\mathbf{\Phi}_i$ is already in the basis for $1 \leq i < k$.
  - Thus the column in $\mathbf{\Delta}_{\mathbf{\Phi}}$ corresponding to $\boldsymbol{\phi}_k$ will be $P^kR - P_{\boldsymbol{\Phi}}P^{k-1}R$.
  - $P_{\boldsymbol{\Phi}}P^{k-1}R$ is necessarily in the span.
  - Thus only contribution to the basis made by MEBF is from P^kR which is what we add in $Krylov_{k+1}(\mathbf{R})$.

Feature Selection

- Invariant Subspaces of P
  - Features for which $\mathbf{\Delta_\Phi} = 0$, reduce the feature selection problem to predicting the immediate reward using these functions.
  - This means that features are prefect linear predictors of their own next state, ie they are subspace invariant with respect to **P**.
    ie $P\Phi = \Phi\Lambda$
  - There are many ways to describe an invariant subspace of **P**,
  - For example, Schur decomposition of a matrix **P** provides a set of nested invariant subspaces of **P**.

Feature Selection

- **Theorem**: For any MRP M and subspace invariant feature set $\mathbf{\Phi}$, $\mathbf{\Delta_\Phi} = 0$.
  - $\mathbf{P_\Phi}$ has a simple form due to subspace invariance

$$P_\Phi = (\Phi^T\Phi)^{-1}\Phi^T P\Phi = (\Phi^T\Phi)^{-1}\Phi^T\Phi\Lambda = \Lambda.$$

  - Substituting into the definition of $\mathbf{\Delta_\Phi}$

$$\Delta_\Phi = P\Phi - \widehat{P\Phi} = P\Phi - \Phi P_\Phi = \Phi\Lambda - \Phi\Lambda = 0$$

Feature Selection

- **Theorem**: For any MRP M and subspace invariant feature set $\mathbf{\Phi}$, $\mathbf{w}_{\mathbf{\Phi}}$ always exists and $\mathbf{\Phi w_\Phi} = (I - \gamma P)^{-1}\hat{R}$.
  - Starting with form of w_{Φ} from previous slides and the fact that $\mathbf{\Delta}_{\mathbf{\Phi}}$=0.

$$\begin{aligned} \mathbf{\Phi w_\Phi} &= \hat{R} + \gamma \widehat{P\Phi} \mathbf{w_\Phi} \\ &= \hat{R} + \gamma P \mathbf{\Phi w_\Phi} \\ \mathbf{\Phi w_\Phi} - \gamma P \mathbf{\Phi w_\Phi} &= \hat{R} \\ \mathbf{\Phi w_\Phi} &= (I - \gamma P)^{-1}\hat{R}. \end{aligned}$$

  - By design $\hat{R} \in span(\mathbf{\Phi})$, and $(I - \gamma P)^{-1}$ must exists for the actual P and $0 \leq \gamma \leq 1$. Therefore

$$\mathbf{\Phi w_\Phi} = \sum_{i=0}^{\infty} \gamma^i P^i \hat{R}.$$

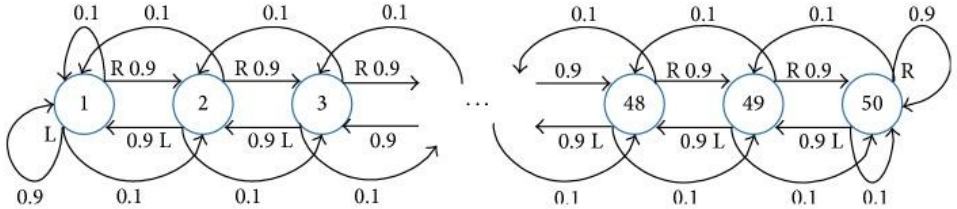
Experimental Results

- Policy evaluation results on three different problems
- Considered four algorithms
    - PVF (Proto Value Function)
        - Eigenvalues of Laplacian derived from an empirically constructed adjacency matrix, in the increasing order of eigenvalue
        - Added adjacency links for all policies, not just the one under evaluation
        - Removal of off-policy links produced worse performance
    - PVF-MP
        - Selects basis functions from the set of PVFs
        - Incrementally based on Bellman Error
        - Basis function *k+1* is the one with highest dot product with with the Bellman Error from previous *k* functions.
    - Eig-MP
        - Similar to PVF MP, but selects from a dictionary of eigenvectors of **P**.
    - BEBF
        - BEBF starting with $\mathbf{\Phi}_0 = \mathbf{R}$.

Experimental Results

- Reported Bellman Error, the reward error and the feature error
- These metrics are presented as a function of number of basis functions
- Problems explored:
    - 50 - state chain problem
    - Two-room problem
    - BlackJack

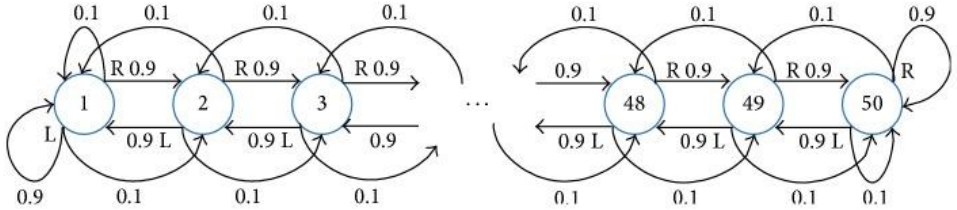
Experimental Results

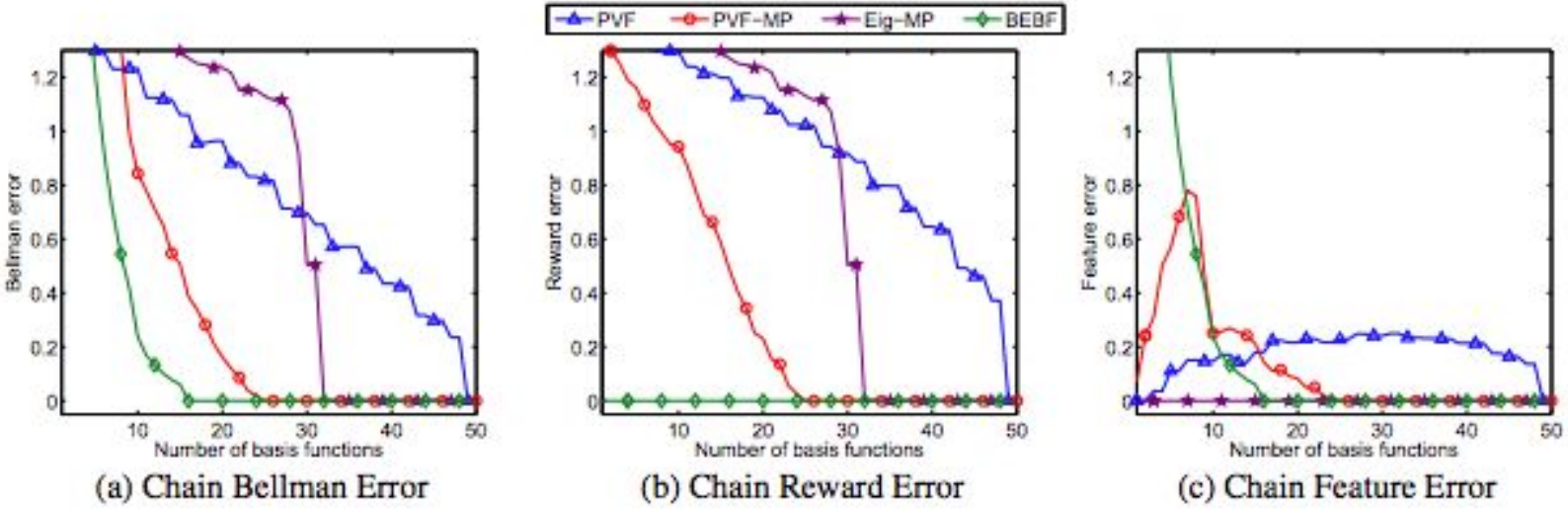


- 50 state chain problem
  - Applied all algorithms to 50 state chain problem
  - Eig-MP has 0 feature error (as predicted)
  - BEBF has 0 rewards error after the first basis function is added
  - PVF appear to be approximately subspace invariant with low feature error
  - Eig-MP and PVF perform poorly as reward is not easily modelled as linear combination of small number of PVFs.
  - PVF-MP does better as it is actively trying to reduce the error

Experimental Results

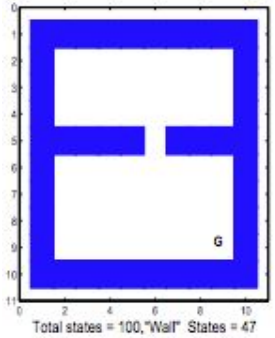
- 50 state chain problem



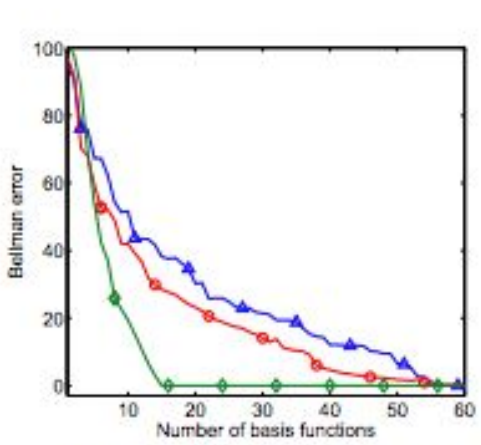


(a) Chain Bellman Error (b) Chain Reward Error (c) Chain Feature Error

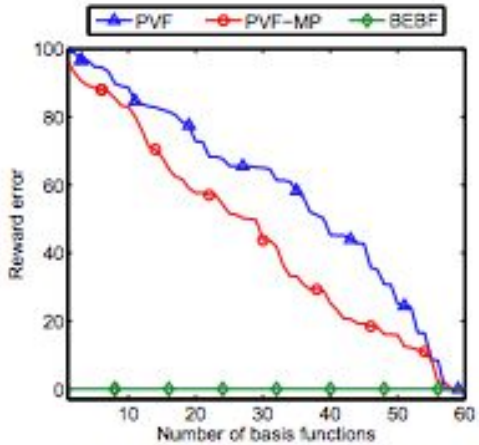
Experimental Results



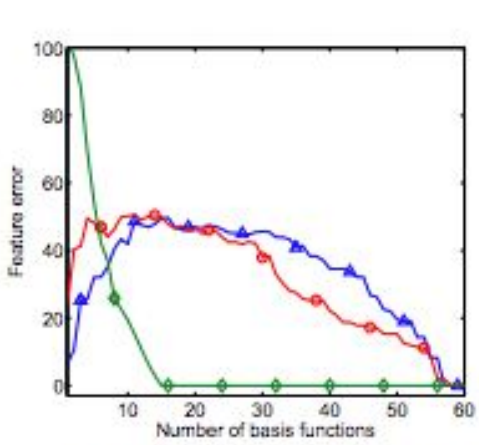
- Two room problem
  - Eig-MP not reported as Adjacency matrix was not diagonalizable.
  - PVF behaves less like an invariant subspace with high reward and feature error.



(d) Two-room Bellman Error



(e) Two-room Reward Error



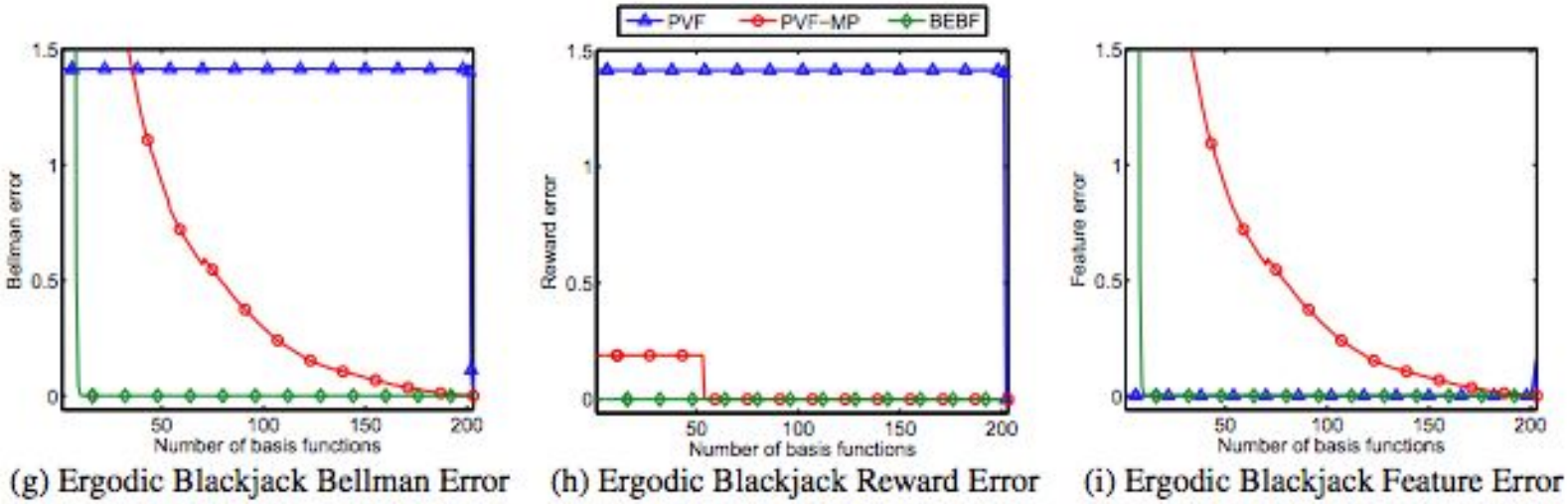
(f) Two-room Feature Error

Experimental Results

- Blackjack
  - Implemented an ergodic version which resets to an initial distribution over hands with value 12 or larger and used a discount of 0.999.
  - BEBF has 0 reward error and lowers the Bellman Error fairly rapidly
  - PVF forms an invariant subspace
  - PVF-MP does not result in subspace invariant feature set but reduces rewards error earlier.

Experimental Results

- Blackjack



Discussion and future work

- A significant contribution is the close relationship between value-function approximation and model based learning.
- While subspace invariant features have highly desirable properties, the features should model the immediate reward as well.
- Eigenvector methods are also limited by the cost of computing eigenvectors of P, or an approximation of P via the Laplacian.
- Future direction : seek a deeper understanding of interaction between feature-selection and policy-improvement algorithms.

Thank You