Linear Value Function Approximation and Linear Models

Ronald Parr
Duke University

Joint work with Christopher Painter-Wakefield (Duke) and Michael Littman (Rutgers)
Why Study Linear Methods?

• Simplicity
• Opacity

• Recent trend in machine learning towards using “embellished” linear methods
  – Boosting
  – SVMs
  – Recent work of Mahadevan et al. for RL
Outline

• Introduce terminology

• Various forms of linear value function approximation

• Linear approximate model formulation

• Show equivalence between linear fixed point approximation and linear model approximation
Focus on Value Determination

• Compute expected (discounted) value of a policy
  – Return on an investment strategy
  – Reward for navigating a robot successfully to a goal
  – Cost of an equipment maintenance strategy

• Value determination is (often) a precursor to optimization
Notation and Assumptions

• State space: $S$
• Reward function: $R(s)$
• Transition function: $P(s'|s)$, and matrix $P$
• Discount factor: $0 \leq \gamma < 1$
• Value of a state

$$V(s) = \sum_{i=0}^{\infty} \sum_{s'} \gamma^i P(s_i = s'|s_0 = s)R(s')$$

• Value function

$$V = R + \gamma P V$$
$$= (I - \gamma P)^{-1} R$$
Approximation

• Since $|S|$ is typically large, would like to approximate $V$ more succinctly

• Many ways to approach this

• We consider approximations that, loosely speaking, aim to achieve what linear regression would do given true $V$
Regression Notation

- Given some target vector $x=[x_1 \ldots x_n]$
- Set of features/basis vectors/basis functions $h_1(x) \ldots h_k(x)$
- Find weight vector $w=[w_1 \ldots w_k]$ s.t.

$$\sum_{j=1}^{k} w_j h_j(x_i) \approx x_i$$
More Regression Notation

K basis functions

\[
A = \begin{bmatrix}
h_1(s1) & h_2(s1) & \cdots \\
h_1(s2) & h_2(s2) & \cdots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]

Data points \(x_1 \ldots x_n\)

- \(A\) is a design matrix
- \(Aw\) is our approximation to \(x\)
Still more notation…

• We want: \( Aw \approx x \)

• Regression/orthogonal projection/least squares/max likelihood yield

\[
    w = (A^T A)^{-1} A^T x
\]

• \( w \) = projection weights

• Projection into column space of \( A \)

\[
    A(A^T A)^{-1} A^T
\]
Weighted Projections

- Can introduce a diagonal weight matrix $\rho$
- Weighted projection is a projection in a skewed space; minimizes weighted error

$$A(A^T \rho A)^{-1} A^T \rho$$

- We omit $\rho$ for compactness
  (but remember that we have the option!)
Fixed Points of Linear Approximations

- Approximation solution of: \( V = R + \gamma P V \)

- Substitute linear approximation:
  \[
  Aw = R + \gamma PAw
  \]

- Problem: Solution may not exist because RHS may not be in column space of \( A \)
Approximation w/Projection Step

\[ Aw = A(A^T A)^{-1} A^T (R + \gamma PAw) \]

Projection Matrix

• Leads to several algorithms distinguished by
  – Direct vs. Indirect solution for w
  – Assumptions about P and R
  – (Recall that P and R are too big!)

• Varying convergence, optimality properties
Outline

• Introduce terminology
• Various forms of linear value function approximation
• Linear approximate model formulation
• Show equivalence between linear fixed point approximation and linear model approximation
Indirect Update

$$Aw = A(A^T A)^{-1} A(R + \gamma P A w)$$

$$w^{i+1} = (A^T A)^{-1} A(R + \gamma P A w^i)$$

- Convergence is not guaranteed in general

- Can guarantee convergence w/projection weighted by stationary distribution of $P$ [Tsitsiklis & Van Roy 96]

- Still not practical if done explicitly ($P$ and $R$ too big)
Indirect Update, Factored Model

\[ w^{i+1} = (A^T A)^{-1} A(R + \gamma \mathbf{P} A w^i) \]

- Suppose \( \mathbf{P} \) can be factored (Bayes net)
- Suppose basis functions have limited support
- Can project exponentially many states in polynomial time [Koller & Parr 99]
- Can (optionally) approximate stationary distribution to do weighted projection
Direct Solution

\[ Aw = A(A^T A)^{-1} A(R + \gamma PAw) \]

\[ w = (A^T A - \gamma A^T PA)^{-1} R \]

- Solution may exist even if iterative solution is unstable
  - Solution almost always exists (depending on \( \gamma \))
  - Can use SVD for linearly dependent basis fns.
- Not practical in general (P and R too big)
- Efficient for factored models, bases with small support [Koller & Parr 00]
Direct Solution w/Sampling

\[ w = (A^T A - \gamma A^T PA)^{-1} R \]

- In general, can’t explicitly construct A, P
- Assume a corpus of samples: (s,r,s’)
- Construct A^T A from s in (s,s’) samples
- Construct A^T PA from (s,s’) pairs
- If states are drawn from \( \rho \), converges to \( \rho \) weighted fixed point.

- Known as LSTD [Bradtke & Barto 96]
- Generalized to \( \lambda \)-case [Boyan 99]
- Generalized for control (LSPI) [Lagoudakis & Parr 03]
Linear TD(0)

- Recall indirect update:
  \[ w^{i+1} = (A^T A)^{-1} A(R + \gamma P A w^i) \]

- States, next states, rewards are sampled
- Given \((s,r,s')\), stochastic approximation:
  \[ w^{i+1} = w^i + \alpha [A w^i(s) - \gamma A w^i(s') - r] h(s) \]

- Stable if states are sampled from \(P\)
  [Tsitsiklis & Van Roy 96]
# Linear VFA Summary

<table>
<thead>
<tr>
<th></th>
<th>(In)Direct</th>
<th>Stable</th>
<th>Sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear TD</td>
<td>Indirect</td>
<td>From Trajectories</td>
<td>Yes</td>
</tr>
<tr>
<td>LSTD</td>
<td>Direct</td>
<td>Almost always</td>
<td>Yes</td>
</tr>
<tr>
<td>Factored MDP</td>
<td>Both</td>
<td>Yes*</td>
<td>No</td>
</tr>
</tbody>
</table>

All Solve for same fixed point: \[ Aw = A(A^T A)^{-1} A^T (R + \gamma P A w) \]
Outline

• Introduce terminology

• Various forms of linear value function approximation

• Linear approximate model formulation

• Show equivalence between linear fixed point approximation and linear model approximation
Suppose we start with a linear model…

- Suppose we have:
  - k features \((h_1 \ldots h_k)\)
  - Deterministic feature-to-feature \((k \times k)\) linear model \(Q\)
  - \(x' = Q^T x\), \(AQ\) = matrix of next feature values
  - Deterministic reward \(x^T w_r\), \(Aw_r\) = vector of rewards

- Simple generalizations
  - White noise
  - Noise = convex combination of possible \(Q\) matrices
Value Function for our Model

• Normally:

\[ V = R + \gamma PV \]
\[ = (I - \gamma P)^{-1} R \]

• For our model:

\[ V(x) = x^T w_r + \gamma W(Q^T x) \]

• Matrix form, assuming V is linear:

\[ Aw = Aw_r + \gamma AQw \]
\[ n \times k \text{ next feature matrix} \]

• We never leave the column space of A
Solving for \( w \)

- From the last slide: \( A w = A w_r + \gamma A Q w \)

- Indirect: \( w_{i+1} = w_r + \gamma Q w_i \)

- Direct: \( w = (I - \gamma Q)^{-1} w_r \)

- \( Q \) behaves like \( P \), but
  - \( k \times k \), not \( n \times n \)
  - Not necessarily stable

\[
V = R + \gamma PV = (I - \gamma P)^{-1} R
\]

Standard MDP
Producing Linear Models

• Approximate reward:

\[ A w_r = A (A^T A)^{-1} A^T R \]

Projection

• Find Q minimizing squared error in next features:

\[ A Q = A (A^T A)^{-1} A^T P A \]

Expected next feature vector

Project

Each column of PA
Outline

• Introduce terminology

• Various forms of linear value function approximation

• Linear approximate model formulation

• Show equivalence between linear fixed point approximation and linear model approximation
Putting it all Together

- Linear fixed point solution:

\[ Aw = A(A^T A)^{-1} A^T (R + \gamma PAw) \]
\[ = A(A^T A)^{-1} A^T R + \gamma A(A^T A)^{-1} A^T PAw \]

- Linear model w/approximation:

\[ Aw = Aw_r + \gamma AQw \]

\[ w_r = (A^T A)^{-1} A^T R \quad Q = (A^T A)^{-1} A^T PA \]
Concluding comments

• Linear value function approximation = deterministic linear model approximation

• Questions:
  – Is this unsatisfying?
    • Weren’t we doing stochastic processes?
    • Does it seem crude when viewed this way?
  – Should we address model approximation head-on?
  – How does this inform feature selection?