Value Function Approximation

CSCI 2951-F
Ron Parr
Brown University

Value function approximation

- Markov assumption, "curse of dimensionality" -> big state spaces
- Often impractical to run value iteration/policy iteration
- Classical approach:
  - Use an over-simplified model, designed by hand
  - Gives correct answer to the "wrong" question.
- Increasingly popular approach (though has classical roots)
  - Use function approximation to represent value function
  - Not obviously/theoretically better but has had some practical success

Living with imperfect value functions

$$\|V - TV\|_\infty \leq \epsilon \rightarrow \|V - V^*\|_\infty \leq \frac{\epsilon}{1-\gamma}$$

T is the Bellman operator

- How reassuring is this?
- Does this worst case hold in practice?

Fitted value iteration (model-based)

- Assume:
  - Very large state space - can't represent the value function as a vector
  - Generic machine learning "fit" operator that fits a continuous function based upon a set of training points
- Fitted VI algorithm:
  - Randomly initialize approximate value function V_0
  - i=0
  - Repeat until done*
    - Sample states $S=s^1 \ldots s^m$
    - Fit V_{i+1} on $TV_i(s^1) \ldots TV_i(s^m)$. ← T is the Bellman operator
    - i=i+1
- Shorthand: $V_{i+1}=\text{fit}(TV_i)$
- How do we define "done"?

How to compute TV(s) in approximate VI

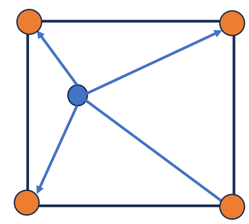
- Challenges:
  - V is not a vector, but some other representation
  - TV involves an expectation over next states, next states which may not be in original sample set S, i.e. off-sample extrapolation is likely *required*
- If number of next states is large and/or no model is available
  - Sample next states too
  - Evaluate expected next state value by Monte Carlo
    - Generate many next states for each state
    - Possible if model/simulator can be easily reset

Properties of Fitted VI (FVI) – part I

- Properties of FVI depend upon properties of Fit function
- Recall that Bellman operator "T" is a contraction in max norm, i.e., $||V_1 - V_2||_\infty < \epsilon \rightarrow ||TV_1 - TV_2||_\infty < \gamma\epsilon, 0 \leq \gamma < 1$
- If two operators, F and G are contractions (i.e. for any value function FV and GV are contractions) then F(GV) is a contraction
- Non-expansion: If H is a non-expansion in max norm, then:
  $$||V_1 - V_2||_\infty < \epsilon \rightarrow ||HV_1 - HV_2||_\infty \leq \gamma\epsilon$$
- If one of F or G is a non-expansion in max norm, and the other is a contraction, the F(GV) is a contraction

Properties of Fitted VI (FVI) – part II

- Follows from previous slide that if Fit is a non-expansion in max norm, then fitted VI is a contraction in max norm
- What choices of Fit are non-expansions?
- Most common examples are averagers, e.g., interpolation



- Fitted VI with interpolation:
  - Pick $S=s^1 \ldots s^m$ to be a grid of points
  - Implementing Fit:
    - For points in S, store TV(s) exactly
    - For points outside of S, use a distance-weighted average of nearest neighbors

Properties of Fitted VI with averagers

- It converges!
- But to what?
- Suppose ε = largest approximation error introduced at any iteration
- Total error is bounded by $\varepsilon/(1-\gamma)$

Is this good news?

- Good news:
  - Convergence yay! ☺
  - In some cases it may be possible to estimate ε
- Bad news:
  - Averagers do not scale well
  - Keeping ε small requires dense S
  - Achieving dense S is exponentially expensive in dimension of space

Beyond Averagers

- Moving beyond averagers requires more powerful function approximation
- Linear approximation is more powerful than averagers because it can extrapolate beyond points in $S=s^1 \ldots s^m$
  (For averagers, any point not in $s^1 \ldots s^m$ has value > $\min(V(s^1) \ldots V(s^m))$ and < $\max(V(s^1) \ldots V(s^m))$)
- Non-linear approximation (e.g. neural networks) is even more powerful than linear approximation

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.

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

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

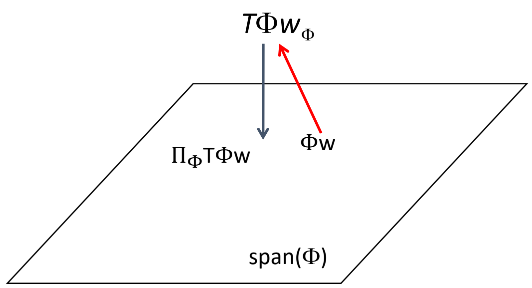
W is a kx1 column vector
Φ is an mxk matrix
(m is number of states sampled)

Why is linear regression so important?

- Averagers interpolate (weak, resource hungry approximation)
- Regression extrapolates (potentially more powerful)
- Linear regression = special case of most other methods
  - Neural networks
  - Kernel methods
- If regression fails, not much optimism on other methods

Linear Fixed Point

- $\Pi_\Phi V$=projection of V into span(Φ)



- If we converge, we have: $\Pi_\Phi T\Phi w = \Phi w$

Example: Stability Problem [Tsitsiklis & Van Roy 1996]

Problem: Convergence not guaranteed

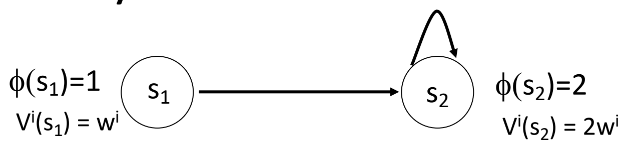
No rewards, γ = 0.9: V* = 0

Consider linear approx. w/ single feature ϕ with weight w.

$$\hat{V}(s) = w \cdot \phi(s)$$

Optimal w = 0
since V*=0

Example: Stability Problem



From iteration i, Belman equation gives

$$T[\hat{V}^i](s_1) = \gamma \hat{V}^i(s_2) = 1.8w^i$$

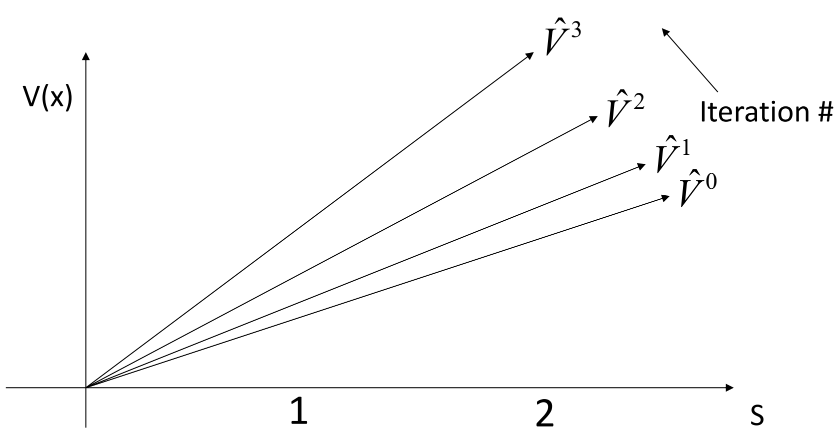
$$T[\hat{V}^i](s_2) = \gamma \hat{V}^i(s_2) = 1.8w^i$$

Can't be represented in our space so find w^{i+1} that gives least-squares approx. to exact backup

After some math linear fit gives us: **$w^{i+1} = 1.2\, w^i$**

What does this mean?

Example: Stability Problem



Each iteration of approximation makes things worse!

Even for this simple problem fitted VI diverges.

Van Roy's Result

- Bellman operator ***fixed policy*** is a contraction in the weighted L_2 norm
- Weights come from the stationary distribution of P
- Linear regression in the **weighted L_2 norm** is non expansive in the weighted L_2 norm
- Understanding this:
  - Weighted norm redefines distance function so that different dimensions in the original space have different importance
  - Equivalent scaling the dimensions of the space
- Combined Regression-Bellman operator is a contraction!

To what does it converge?

$$\left\|V^{\pi} - \widehat{V^{\pi}}\right\|_{2,\rho} \leq \frac{1}{\sqrt{1-\kappa^2}} \left\|V^{\pi} - \Pi V^{\pi}\right\|_{2,\rho}$$

- ρ is the stationary distribution of P_π
- κ is the effective contraction rate ($\leq \gamma$)

Q-iteration: Generalization of Value Iteration

- $\forall s, a: Q(s,a) \leftarrow R(s,a) + \gamma \Sigma_s' P(S'|s,a) V(s')$
- $V(s') = \max_{a'} Q(s', a')$

- Q-iteration has similar convergence properties to value iteration

Application to stopping

- What about optimization?
- How to think about Bellman operator with max
  - Define T^*_Q as the Q-iteration operator
  - T^*_Q is a contraction is Max Norm
- Is T^*_Q a non-expansion in weighted L_2?
- No. ☹
- But… It is non-expansion if max is always done with a constant
- Optimal stopping: Should I continue or stop and receive a payout?

Financial application

- Want to assign a price to an asset with following properties:
  - Can be held by owner for an arbitrary amount of time
  - Can cash out at some future time and receive a state-dependent reward
- Want to compute present value of this asset
- Features:
  - Variables relevant to immediate value of asset
  - Variables relevant to future value of the asset
- Supposedly used by some financial institutions to price assets

Perspective: Is weighted L_2 reasonable?

- In many ways more reasonable than Max norm
  - Worst case over entire state space hard to evaluate
  - Sampling methods can never provide guarantees without additional assumptions
- How do you achieve weighted L_2 in practice?
  (Sample from "real world" states)
- Weighted L_2 gives lower weight to less frequently occurring states
  - Common cases get the most weight
  - Rare events may be wrong but that is forgivable(?)

Q-iteration in general

- What if “Fit” is a neural network?
- Linear value function approximation is a special case of this
- (Lack of) convergence guarantees from linear VFA apply to neural networks, but…
- If approximation error introduced at each step can be bounded by a constant, then overall approximation error is low
  (Note: this is **false** for the Van Roy counterexample.)
- Is this a reasonable assumption? (discuss)

Properties of approximate VI methods

- Convergence not guaranteed, except in special cases
- Success has traditionally required very carefully chosen features and/or dense coverage to achieve low error
- Deep learning, which “automatically” learns feature representations, and uses massive numbers of samples, partially overcomes this