Announcements

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

- Assignment 5 is posted online. Due Nov 9.
- Midterms to be handed back Nov 7
- Project title, 1 paragraph description, and list of 3 team members due Nov 9.

Today

- Optimality equations for infinite horizon MDPs
- Value iteration

Reading Roadmap

Use the lectures as a guide for important material to cover in Chapter 5 of Puterman.

Following are important sections of Chapter 6:

- Section 6.1: Policy evaluation
- Section 6.2: Optimality equations
  - Skip 6.2.5
- Section 6.3: Value iteration
  - Skip 6.3.1
- Section 6.4: Policy Iteration
  - Skip 6.4.3, 6.4.4
- Section 6.5.1: Modified Policy Iteration
- Section 6.9: Linear Programming
- Section 6.11: Optimality of Structured Policies

Optimal Policy

For a max problem, an optimal policy π^* satisfies

$$u_\lambda^{\pi^*}(s) \geq u_\lambda^{\pi}(s), \forall s \in S \; and \; \forall \, \pi \in \Pi$$

and can be expressed as

$$u_\lambda^{\pi^*}(s) = \max_{\pi \in \Pi^{MD}} u_\lambda^{\pi}(s)$$

Optimality Equations

The optimality equations for a stationary **finite horizon** MDP are written as

$$v_t(s) = \max_{a \in A} \{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_{t+1}(j) \}, t = 1, \ldots, T-1$$

Boundary condition: $v_T(s) = r(s)$.

Passing to the limit as $t \to \infty$ we have

$$v(s) = \max_{a \in A} \{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \}$$

Optimality Equations

The optimality equations for a stationary **infinite horizon** MDP are:

$$v(s) = \max_{a \in A} \{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j) \}, \forall s$$

These can be expressed compactly as $v = Lv$ where

$$\mathrm{L}v = \max_{d \in D} \{ r_d + \lambda P_d v \}$$

L is an operator denoting the $\max_{d \in D} \{\cdot\}$ operation over decision rules

Some Useful Properties

Definition: Define the **norm** of vector, v, as

$$||v|| = \max_{s \in S} |v(s)|$$

Definition: The **norm of a matrix**, Q , is

$$||Q|| = \max\{||Qv|| : ||v|| \leq 1\}$$

When Q is a transition probability matrix $||Q||$ = 1.

Definition: A linear transformation is **bounded** if there exists some K such that

$$||Qv|| \leq K||v||$$

Solution to Optimality Equations

Theorem (≈6.2.2 Puterman): If there exists a v such that $v = Lv$, then $v = u_\lambda^{\pi^*}$.

Basic Idea: Proof is in two parts:

- Part (a): prove if $v \geq Lv$ then $v \geq u_\lambda^{\pi^*}$
- Part (b): prove if $v \leq Lv$ then $v \leq u_\lambda^{\pi^*}$
- Follows from (a) and (b) if $v = Lv$ then v $= u_\lambda^{\pi^*}$

See supplemental material on canvas for complete proof

Contraction Mapping Property

Definition: Operator L is a **contraction mapping** if there exists a $0 < \lambda < 1$ such that

$$||Lv - Lu|| \leq \lambda ||v - u||$$

for all u and v.

Solution to Optimality Equations

Theorem (≈6.2.3 Puterman): Suppose L is a contraction mapping, then for arbitrary v^0 the sequence $\{\mathrm{v}^0, \ldots, \mathrm{v}^\mathrm{n}\}$ defined by $\mathrm{v}^{\mathrm{n}+1} = \mathrm{L}\mathrm{v}^\mathrm{n}$ converges to v^* as $\mathrm{n} \to \infty$, where $v^* = Lv^*$.

Proof: See supplemental materials on canvas

Value Iteration

Value iteration exploits the contraction mapping property to solve infinite horizon MDPs

Value Iteration:

1. Select some intial vector v^0 *and set* $n = 0$.

2. *Apply* $v^{n+1} = Lv^n$

3. *If* $||v^{n+1} - v^n|| < \epsilon(1 - \lambda)/2\lambda$ *go to step 4. Otherwise n=n+1 and return to step 2.*

4. $d_\epsilon(s) \in argmax_{a \in A} \{\mathrm{r(s,a)} + \lambda \sum_{j \in S} p(j|s,a)v^n(j)\}, for\ all\ s \in S.$

Solution to Optimality Equations

Proposition (≈6.2.4 Puterman): If $0 < \lambda < 1$ then L is a contraction mapping.

Proof: See supplemental materials on canvas

Convergence of Value Iteration

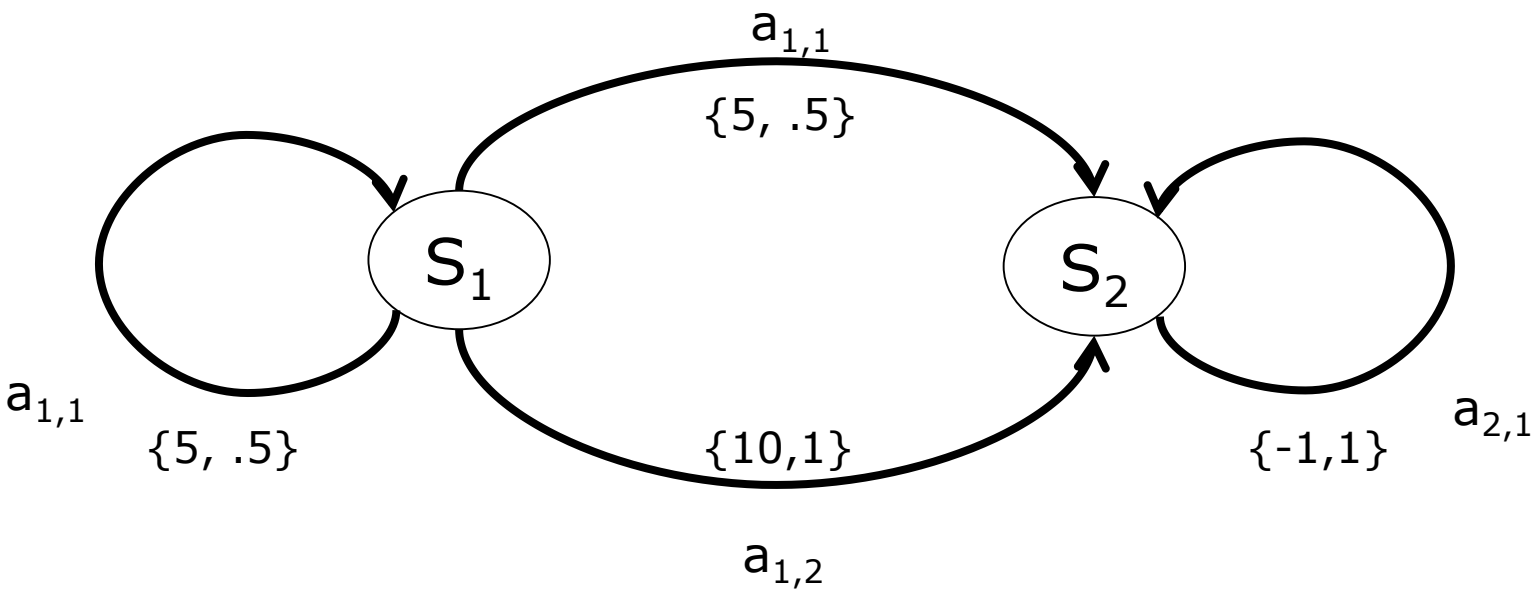
Theorem (≈6.3.1 Part (d) Puterman): Let $\{v^n\}$ satisfy $v^{n+1} = Lv^n$, then:

$$||v^{n+1} - v^*|| \leq \epsilon/2 \text{ whenever} ||v^{n+1} - v^n|| < \frac{\epsilon(1-\lambda)}{2\lambda}$$

Proof: Complete in class

Example Revisited: 2 State MDP

In state S_1 actions $a_{1,1}$ and $a_{1,2}$ are available; in state S_2 only $a_{2,1}$ is available. Rewards and transition probabilities are defined below as {r,p}. At each stage the associated reward is received and then the transition occurs.



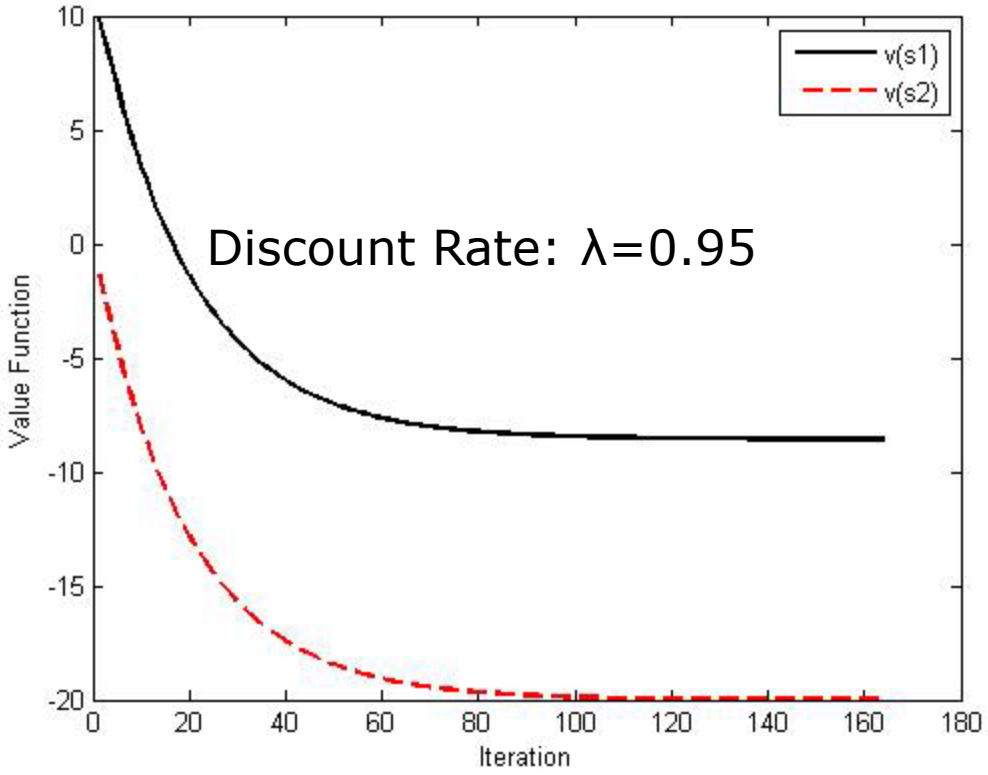
Value Iteration Example (Matlab)

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

```
for t=2:NITERATIONS
   for s=1:S %loop through all states
      for a=1:A %loop through all actions
         q(s,a)=0;
         for j=1:S %sum over states to compute value to go
            if(a==1) q(s,a) = q(s,a) + lambda*P1(s,j)*v(t-1,j);
            else q(s,a) = q(s,a) + lambda*P2(s,j)*v(t-1,j);
            end
         end
         q(s,a) = q(s,a) + r(s,a);

         %save first action evaluated
         if(a==1)
            v(t,s) = q(s,a);
            policy(t,s)=a;
         end

         %if current action better than best save action/value function
         if (q(s,a) > v(t,s))
             v(t,s) = q(s,a);
             policy(t,s)=a;
         end
      end
   end

   %check stopping criteria (exit loop if stopping criteria satisfied)
   norm_v = max(abs(v(t,1)-v(t-1,1)), abs(v(t,2)-v(t-1, 2)));
   if (norm_v < epsilon*(1-lambda)/(2*lambda)) break; end

end
```

Example Revisited: Value Iteration

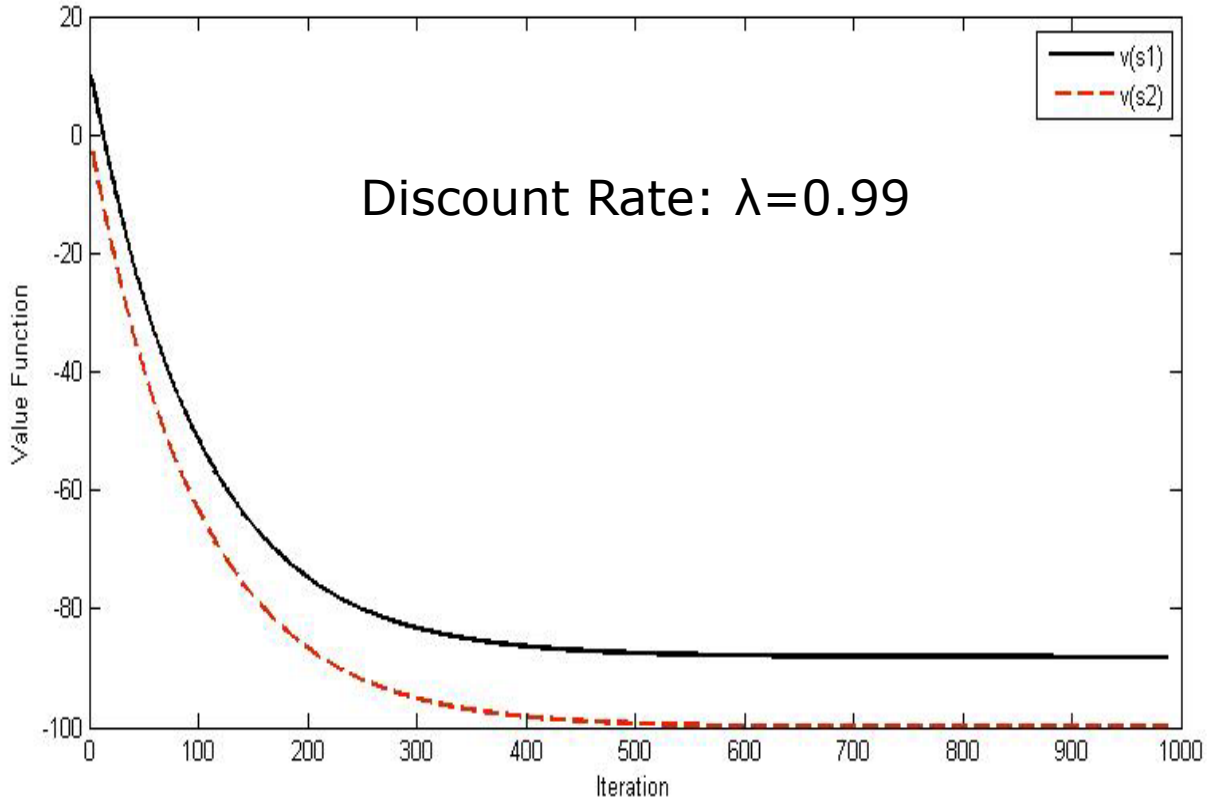
Convergence of value functions with respect to iteration



Note: optimal policy was attained at the first iteration

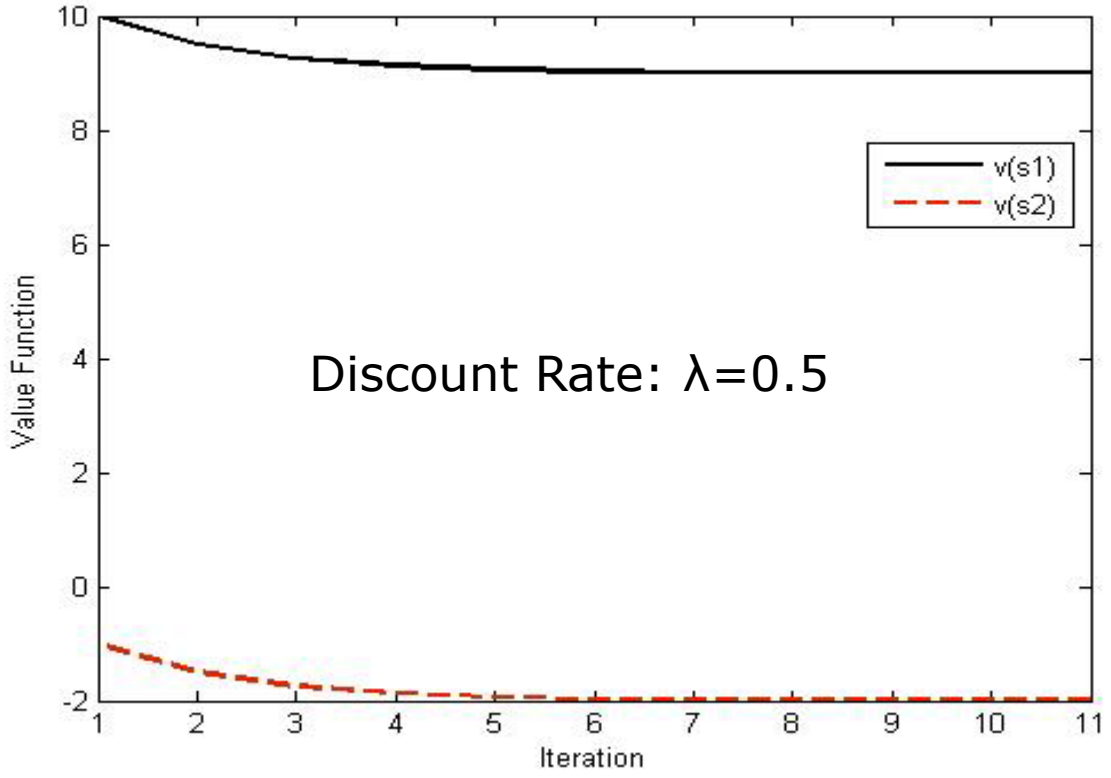
Value Iteration Convergence

Influence of discount rate on value iteration convergence



Value Iteration Convergence

Influence of discount rate on value iteration convergence



Conclusions

- Convergence of the optimal value function is influenced by the discount factor
  - The higher the discount factor the slower the rate of convergence
- In some cases a good policy (sometimes the optimal policy) may be found very quickly
- More advances approaches seek to reduce the number of iterations necessary to find an ϵ-optimal policy