**Artificial Intelligence in Games**

**Session 8**

1. Dynamic programming
   1. Dynamic programming algorithms can be used to compute optimal policies given a perfect model of the environment (one-step dynamics) when the sets of states and actions are finite
   2. The problem of finding the optimal value functions has optimal substructure: it can be solved by breaking it into sub-problems and then recursively finding the solutions to the sub-problems
2. Policy evaluation
   1. Policy evaluation is an iterative algorithm to compute the state value function V^pi for an arbitrary policy pi
   2. It relies on creating a sequence V0, V1, . . . of estimates of V^pi given by  
        
      V\_k+1(s) = sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]) for all s in S.
   3. The initial value estimate V0 can be arbitrary
   4. The sequence V0(s), V1(s), . . . converges to V^pi(s) for all s in S
3. In-place policy evaluation
   1. Instead of computing the new estimate V\_k+1 using the old estimate V\_k , it is also possible to change a single estimate V in-place using
   2. V\_k+1(s) 🡨 sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]) for all s in S.
   3. The estimate V(s) also converges to V^pi(s) for all s in S after repeated passes over all states
   4. Algorithm: Iterative policy evaluation (in-place)
   5. Input: policy pi, one-step dynamics functions P and R, discount factor lambda, tolerance theta.
   6. Output: Value function V = V^pi when theta 🡪 0.
      1. For each s in S do
         1. V(s) 🡨 0
      2. End for
      3. Repeat
         1. Delta 🡨 0
         2. For each s in S do
            1. V 🡨 V(s)
            2. V\_k+1(s) 🡨 sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]) for all s in S
            3. Delta 🡨 max of (Delta, Modulus of (v – V(s)))
         3. End for
      4. Until delta < theta
4. Convergence of iterative policy evaluation:  
     
   NOTE:  
   I am only writing out the equations here otherwise it would take me years to complete this slide deck.   
   1. Norm:
      1. Consider a vector space Z over a field F. A function || · || : Z → [0, ∞) is a norm if
         1. ||u + v|| ≤ ||u|| + ||v||,
         2. ||av|| = |a|||v||,
         3. ||v|| = 0 =⇒ v = 0,
   2. Euclidean norm:
      1. ||v^2|| = sqrt(sum of(v\_i^2))
   3. Maximum norm:
      1. ||v||\_infinity = max |v\_i|
   4. Convergence of a sequence:
      1. ||v\_n - v|| = 0
   5. Bellman operator:
      1. T^pi(v)\_s = sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_s’])
   6. Convergence of iterative policy evaluation:
      1. Consider the Bellman operator T^pi : R^{|S|} → R^{|S|} for the policy pi. Given an arbitrary v0 in R^|S|, consider also the sequence (vn)n >= 0 where v\_k+1 = T^pi(vk ). Finally, consider the vector v pi in R^|S| such that v^pi\_s = V pi (s), for any s in S. For any n >= 0,
      2. vn → v pi ,
      3. v pi = T^pi (v^pi ),
      4. ||vn − v pi || <= lambda^n ||v0 − v pi ||
   7. Cauchy sequence:
      1. Consider a normed vector space (Z, || · ||). A sequence (vn)n>=0 = v0, v1, . . . of vectors in this space is Cauchy if
      2. As n approaches infinity (supplied m >= n) =|| v\_n – v\_m || = 0
      3. In other words, if a sequence (vn)n >= 0 is Cauchy, then for every epsilon > 0 there is an N such that for every n >= N, we have ||v\_n – v\_m|| < epsilon.
   8. Banach space:
      1. A Banach space is a normed vector space (Z, || · ||) where if (vn)n>=0 is a Cauchy sequence then v\_n → ||·|| v for some vector v.
      2. For any d, both (R^d , || · ||\_2) and (R^d , || · ||\_infty) are Banach spaces, although we omit the corresponding proofs.
   9. L-contraction:
      1. Consider a normed vector space (Z, || · ||) and a function T : Z → Z. The function T is L-Lipschitz if:
      2. ||T(u) − T(v)|| ≤ L||u − v||
      3. For all u, v in Z. If L < 1, then T is also an L-contraction.
   10. Lemma:
       1. Consider a normed vector space (Z, || · ||), an L-Lipschitz function T : Z → Z, and a sequence (vn)\_n = v0, v1, . . . of vectors in this space. If vn → v, then T(v\_n) → T(v).
       2. Proof:
       3. For any n >= 0, by the definition of an L-Lipschitz function,
          1. 0 ≤ ||T(vn) − T(v)|| <= L||vn − v||.
       4. Since vn → v,
          1. lim n approached infty L||vn − v|| = L lim n → infty ||vn − v|| = 0.
       5. By the squeeze theorem,
          1. lim n→ infty ||T(vn) − T(v)|| = 0.
   11. Banach’s fixed point theorem:
       1. If (Z, || · ||) is a Banach space and T : Z → Z is an L-contraction, then T has a unique fixed point v. Furthermore, for any v0 in Z, let v\_n+1 = T(vn). For any n >= 0,
       2. vn → v,
       3. ||vn − v|| <= L^n ||v0 − v||
       4. Proof.
       5. We first show that the sequence (vn)\_{n >= 0} is Cauchy, which guarantees that vn → v for some vector v.   
            
          As a first step, we show that ||v\_{n+k} − vn|| <= L^n ||v\_k − v0|| for any n, k >= 0. The case n = 0 is trivial. Suppose that the inductive hypothesis is true for some n, and consider the case n + 1:
          1. ||v\_{n+k} + 1 − vn+1|| = ||T(v\_n+k ) − T(vn)|| (definition of the sequence)
          2. <= L||v\_n+k − vn|| (definition of L-contraction)
          3. <= L\_n+1||vk − v0||, (inductive hypothesis)
       6. For k >= 1, ||vk − v0|| = ||vk + (−vk−1 + vk−1) + . . . + (−v1 + v1) − v0||.
       7. Therefore, ||vk − v0|| = sum of( vi – v\_i−1) (reorganizing terms)
          1. <= sum of(||vi − vi−1||) (triangle inequality)
          2. <= (||v1 − v0||) / (1 – L) (earlier result).
       8. We are now close to showing that (vn)n >= 0 is Cauchy. For any n, k >= 0, combining the previous two results,
          * 1. 0 <= ||v\_n+k − vn|| <= L^n ||vk − v0|| <= L^n (||v1 − v0||) / (1 – L)
       9. Therefore, for any fixed n >= 0,
          1. 0 <= (supplied that k >= 0)||v\_n+k − vn|| <= (supplied that k >= 0) L^n ||vk − v0|| <= L^n (||v1 − v0||) / (1 – L)
       10. Because 0 ≤ L < 1, and the squeeze theorem
           1. Limit of n as it approaches infty (supplied k >= 0) ||vn+k − vn|| = 0
       11. which completes the proof that (vn)\_{n >= 0} is Cauchy. Let v denote the vector such that vn → v
       12. Our next step is to show that v is a fixed point of T. For any n,
           1. 0 <= ||T(v) − v|| = ||T(v) + (−T(vn) + T(vn)) − v|| (introducing zeros)
           2. <= ||T(v) − T(vn)|| + ||T(vn) − v|| (triangle inequality)
           3. <= L||v − vn|| + ||vn+1 − v|| (L-contraction)
       13. Because vn → |v
           1. L||v − vn|| + ||vn+1 − v|| = 0. (limit of n → infty)
       14. Therefore, by the squeeze theorem
           1. ||T(v) − v|| = lim n →infty ||T(v) − v|| = 0.
       15. By the definition of a norm, T(v) − v = 0, which implies T(v) = v, completing the proof.
       16. Our next step is to show that the fixed point of T is unique. Suppose that T(u) = u and T(v) = v for some vectors u and v. In that case,
           1. ||u − v|| = ||T(u) − T(v)|| <= L||u − v||.
       17. If we suppose that ||u − v|| > 0, dividing the inequation by ||u − v|| leads to the conclusion that L >= 1. However, T is an L-contraction, contradicting our supposition. Therefore, ||u − v|| <= 0, which implies that u = v.
       18. Our last step is to show that ||vn − v|| <= L^n ||v0 − v||, for any n. The case n = 0 is trivial. Suppose that the inductive hypothesis is true for some n, and consider the case n + 1:
           1. ||vn+1 − v|| = ||T(vn) − T(v)|| (definition of fixed point)
           2. <= L||vn − v|| ≤ L n+1||v0 − v||, (inductive hypothesis) as we wanted to show.
   12. Theorem (Convergence of iterative policy evaluation):
       1. Consider the Bellman operator T^pi : R^|S| → R^|S| for the policy pi. Given an arbitrary v0 in R^|S|, consider also the sequence (vn)\_{n>=0} where v\_k+1 = T^pi (vk ). Finally, consider the vector v^pi in R^|S| such that v^pi\_s = V^pi (s), for any s in S. For any n >= 0,
          1. vn → v^pi ,
          2. v^pi = T^pi (v^pi ),
          3. ||vn – v^pi || <= gamma^n ||v0 – v^pi ||.
       2. As a first step, we show that vπ is a fixed point of T^pi. For any s in S, by the definition of T^pi and V^pi,
          1. T^pi(v)\_s = sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_s’])
       3. Our next step is to show that the Bellman operator T^pi: R^|S| → R^|S| is a gamma-contraction. Because (R^|S| , || · ||\_infty) is a Banach space and v^pi is a fixed point of T^pi , the desired results follow from Banach’s fixed point theorem.
       4. Note that, for any two vectors u, v in R^|S| and every state s in S,
          1. T^pi(u)\_s – T\_pi(v)\_s = {sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda u (s’)])} – {sum of (pi(s,a)) \* sum of(P^a\_ss’ [R^a\_ss’ + lambda v (s’)])}
          2. = double sum of((pi(s,a)) \* P^{a\_ss’} \* gamma u(s’)) - double sum of((pi(s,a)) \* P^{a\_ss’} \* gamma v(s’))
          3. = gamma \* double sum of((pi(s,a)) \* P^{a\_ss’} \* (u(s’) – v(s’)))
       5. By the definition of maximum norm, for any two vectors u, v in R^|S|,
          1. ||T^pi(u) - T^pi(v)||\_infty
          2. gamma \*max of [double sum of((pi(s,a)) \* P^{a\_ss’} \* (u(s’) – v(s’))) (definition of maximum norm)]
          3. <= gamma \* double sum of(max of [ (pi(s,a)) \* P^{a\_ss’} \* (u(s’) – v(s’))) (triangle inequality)
          4. = gamma \* double sum of((pi(s,a)) \* P^{a\_ss’} \* max of [ (u(s’) – v(s’))) ] (multiplicativity)
          5. <= gamma \* double sum of((pi(s,a)) \* P^{a\_ss’} (u(s’) – v(s’))\_infty) ] (definition of maximum norm)
          6. = gamma ||u - v||\_infty max of [sum of (pi(s,a)) \* sum of(P^{a\_ss’})] (distributivity)
          7. = gamma ||u - v||\_infty (unit measure)
5. Deterministic policies :
   1. A deterministic policy pi is one such that, for all s in S, pi(s, a) = 1 for some a in A and pi(s, b) = 0 for all b != a
   2. In this case, we abuse notation and represent a policy by a function pi : S → A from states to actions.
6. Policy improvement
   1. Let pi and pi’ be any pair of deterministic policies such that, for all s in S, Q\_ pi (s, pi 0 ‘(s)) >= V^ pi (s)
   2. The policy improvement theorem guarantees that V^pi’(s) >= V^ pi (s) for all s in S
   3. For all s in S, a policy pi may be improved to a policy pi’ by letting:
      1. Pi’(s) = maximise argument of Q^pi(s,a) = maximise argument of (sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]))
7. Policy iteration
   1. Policy evaluation and policy improvement can be interleaved
   2. This process produces the sequence
      1. pi0, V^pi0 , pi1, V^pi1 , pi2, V^pi2 , . . .
   3. If pi\_t = pi\_t+1, then pi\_t is optimal by the uniqueness of V\*
   4. The initial policy pi0 can be arbitrary
8. Value iteration
   1. A more efficient alternative iteratively improves the estimates for the value of each state under an optimal policy
   2. It relies on creating a sequence V0, V1, . . . of estimates given by:
      1. maximise argument of (sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]))
      2. The initial estimate V0 can be arbitrary
      3. The sequence V0(s), V1(s), . . . converges to V\*(s) for all s in S
      4. In-place value iteration has the same guarantees
      5. Algorithm:
         1. Input: one-step dynamics (P and R), discount factor gamma, and tolerance theta.
         2. Output: optimal deterministic policy pi when theta → 0.
         3. for each s in S do
            1. V(s) ← 0
         4. end for
         5. repeat
            1. Delta 🡨 0
            2. for each s in S do

v 🡨 V(s)

V(s) 🡨 maximise argument of (sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]))

Delta 🡨 max of (Delta, |v – V(s)|)

* + - * 1. End for
      1. Until Delta < theta
      2. For each s in S do
         1. Pi(s) = maximise argument of (sum of(P^a\_ss’ [R^a\_ss’ + lambda V\_k (s’)]))
      3. End for