Q-LEARNING AND BAIRD'S COUNTEREXAMPLE

Jonathan Campbell
COMP-767
March 31, 2017

OVERVIEW

- Summary of Q-learning/Greedy-GQ and Baird's counterexample.
  - Q-learning (tabular)
  - Q-learning (linear function approximation)
  - Baird's counterexample
  - Greedy-GQ
  - Performance on Baird's counterexample

Q-LEARNING

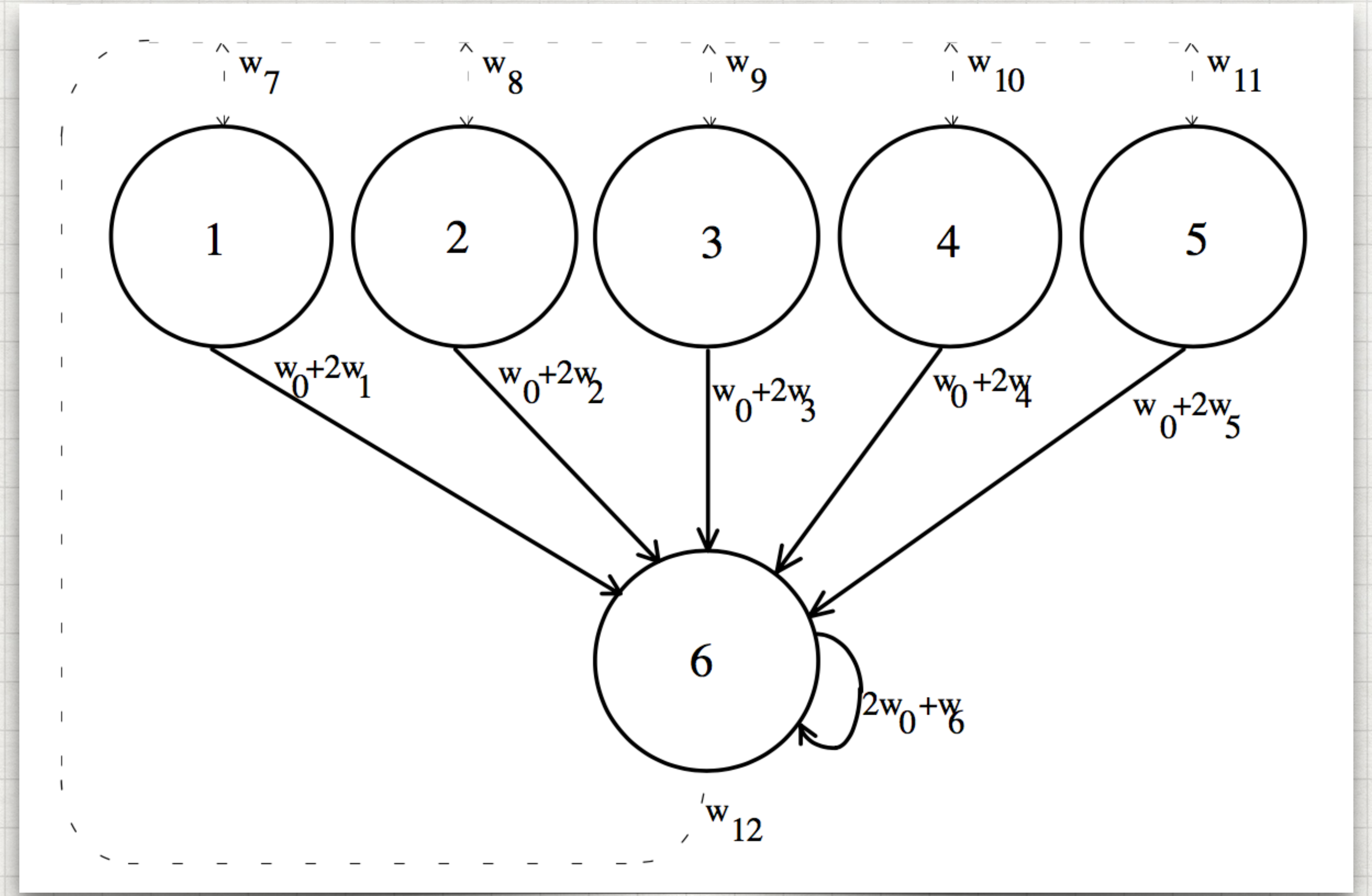
- Initialize $Q(s, a)$ arbitrarily
- Repeat (for each episode):
  - Initialize s
  - Repeat (for each step of episode):
    - Choose a from s using policy derived from Q (e.g. ε-greedy)
    - Take action a, observe r, s'
    - $\delta = r + \gamma \max_{a'} Q(s', a') - Q(s, a)$
    - $Q(s, a) = Q(s, a) + \alpha * \delta$
    - $s = s'$
  - ... until s is terminal

Q-LEARNING WITH F.A.

- Initialize θ arbitrarily
- Repeat (for each episode):
  - Initialize *s*
  - Repeat (for each step of episode):
    - Choose *a* from *s* using policy derived from **Q** (e.g. **ε**-greedy)
    - Take action *a*, observe *r*, *s'*
    - $\delta = r + \gamma\ \theta^T[\max_{a'} \Phi(s', a')] - \theta^T\Phi(s, a)$
    - $\theta = \theta + \alpha * \delta * \Phi(s, a)$
    - $s = s'$
  - … until s is terminal

Q-value table replaced with parameter vector.

BAIRD'S COUNTEREXAMPLE



6 star problem

BAIRD'S COUNTEREXAMPLE (1)

- Two actions:
    - Solid (goes to terminal state -- state 6)
    - Dotted (goes to any of state [0..5] with uniform probability)
- No rewards.
    - Q-values should converge at 0.
- Behaviour policy: solid action with prob. 1/6, dotted with p. 5/6.

BAIRD'S COUNTEREXAMPLE (2)

- Weight vector size: (num states * num actions) + 1
  - V(s) = linear combination of two weights, as shown in figure.
- Initial parameters:
  - Q-values for solid actions larger than dotted actions
  - Q-value for solid action in terminal state largest.
- Will diverge for Q-learning using linear function approximation.

GREEDY-GQ

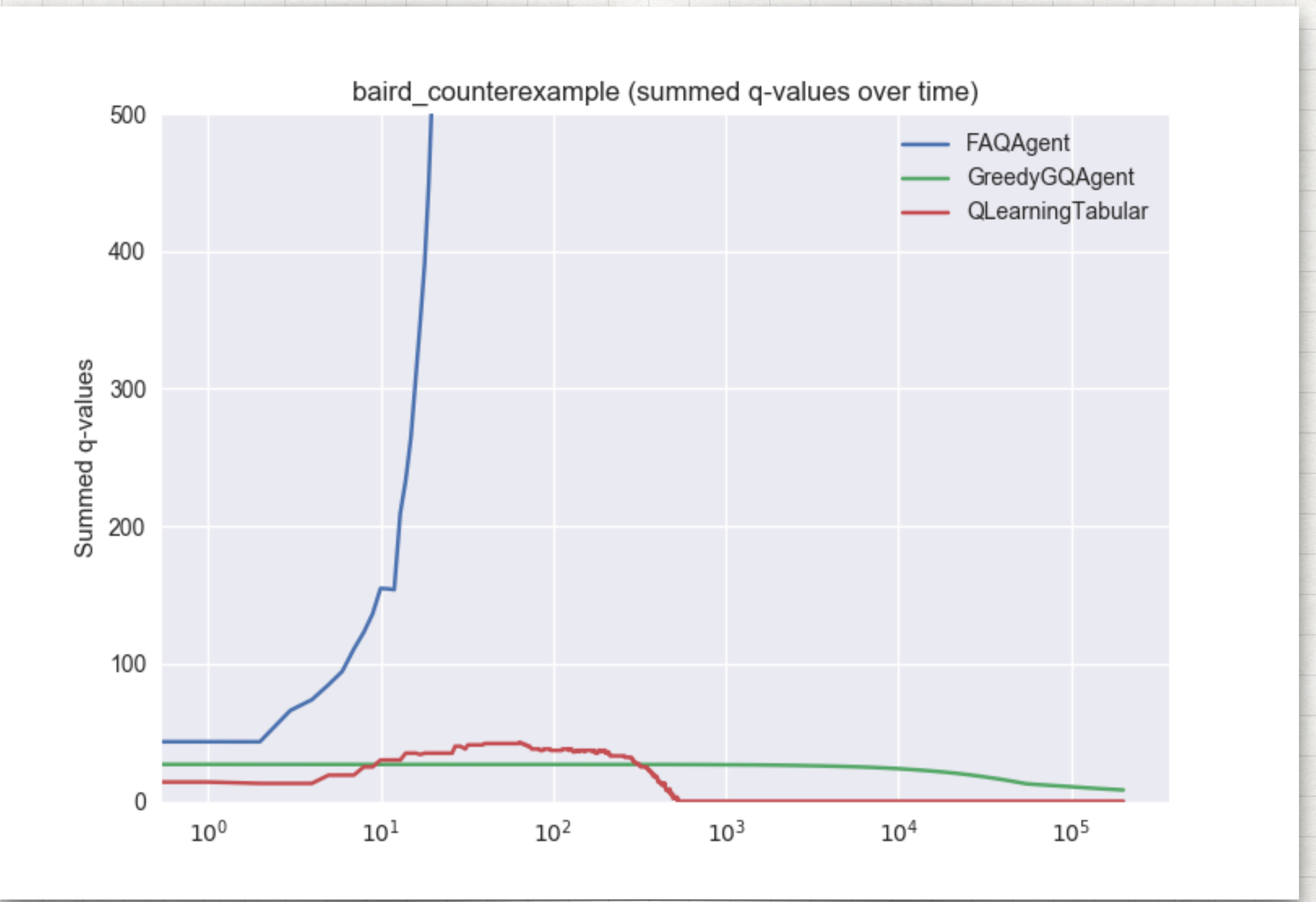
- Action-value learning alg. that is stable for off-policy w/ linear F.A.
  - Extension of gradient-TD methods to control setting.
  - Convergence to equilibrium point of MSPBE.
- Restriction: behaviour policy must be stationary (can't use **ε**-greedy).

GREEDY-GQ

- Initialize θ arbitrarily
- Repeat (for each episode):
  - Initialize *s*
  - Repeat (for each step of episode):
    - Choose *a* from *s* using fixed policy.
    - Take action *a*, observe *r*, *s'*
    - $\delta = r + \gamma\, \theta^T[max_{a'}\, \Phi(s', a')] - \theta^T\Phi(s, a)$ (**δ**: Regular TD-error)
    - $\theta = \theta + \alpha\, [\delta\Phi - \gamma\, (w^T\Phi)\, max_{a'}\, \Phi(s', a')\,]$
    - $w = w + \beta\, [\delta - \Phi^T w]\, \Phi$
    - $s = s'$
  - ... until s is terminal

Two sets of weights, each to store estimate of different expectation (as in GTD).

COMPARISON ON BAIRD'S 6 STAR PROBLEM



REFERENCES

- Greedy-GQ
    - Toward Off-Policy Learning Control with Function Approximation (2010)
      Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, Richard S. Sutton
- GTD(0)
    - A Convergent O(n) Algorithm for Off-policy Temporal-difference Learning with Linear Function Approximation (2009)
      Richard S. Sutton, Csaba Szepesvári, Hamid Reza Maei
- Baird's counterexample
    - Residual Algorithms: Reinforcement Learning with Function Approximation (1995)
      Leemon Baird