

# GQ( $\lambda$ ) Quick Reference Guide

Adam White and Richard S. Sutton

August 9, 2010

This document should serve as a quick reference for the linear GQ( $\lambda$ ) off-policy learning algorithm. We refer the reader to Maei and Sutton (2010) for a more detailed explanation of the intuition behind the algorithm and convergence proofs. If you have questions or concerns about the content in this document or the attached java code please email [awhite@cs.ualberta.ca](mailto:awhite@cs.ualberta.ca).

## 1 Notation

For each use of GQ( $\lambda$ ) you need to provide the following four *question functions*. (In the following  $S$  and  $A$  denote the sets of states and actions.)

- $\pi : S \times A \rightarrow [0, 1]$ ; target policy to be learned. If  $\pi$  is chosen as the greedy policy with respect to the learned value function, the algorithm will implement a generalization of The Greedy-GQ algorithm as described in the recent ICML-10 paper (Maei, Szepesvari, Bhatnagar & Sutton, 2010).
- $\gamma : S \rightarrow [0, 1]$ ; termination function ( $\gamma(s) = 1 - \beta(s)$  in GQ paper)
- $r : S \times A \times S \rightarrow \mathfrak{R}$ ; transient reward function
- $z : S \rightarrow \mathfrak{R}$ ; terminal reward function

The nature of the approximation you will get will depend upon the following four *answers functions* (these also must be provided):

- $b : S \times A \rightarrow [0, 1]$ ; behaviour policy
- $I : S \times A \rightarrow [0, 1]$ ; interest function (can set to 1 for all state-action pairs or indicate selected state-action pairs to best approximate)
- $\phi : S \times A \rightarrow \mathfrak{R}^n$ ; feature-vector function
- $\lambda : S \rightarrow [0, 1]$ ; eligibility-trace decay-rate function

The following data structures are internal to GQ:

- $\boldsymbol{\theta} \in \mathfrak{R}^n$ ; the learned weights of linear approximation ( $\mathbf{Q}^\pi = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s, a)$ )
- $\mathbf{w} \in \mathfrak{R}^n$ ; secondary set of learning weights
- $\mathbf{e} \in \mathfrak{R}^n$ ; eligibility trace vector

Parameters internal to GQ:

- $\alpha$ ; step-size parameter for learning  $\boldsymbol{\theta}$
- $\eta \in [0, 1]$ ; relative step-size parameter for learning  $\mathbf{w}$  ( $\alpha\eta$ )

## 2 Equations

We can now specify GQ( $\lambda$ ). Let  $\mathbf{w}$  and  $\mathbf{e}$  be initialized to zero and  $\boldsymbol{\theta}$  be initialized arbitrarily. Let the subscript  $t$  denote the current time step. Let  $\rho_t$  denote the importance sampling correction:

$$\rho_t = \frac{\pi(s_t, a_t)}{b(s_t, a_t)} \quad (1)$$

and  $\bar{\boldsymbol{\phi}}_t$  denote the expected next feature vector:

$$\bar{\boldsymbol{\phi}}_t = \sum_a \pi(s_t, a) \boldsymbol{\phi}(s_t, a) \quad (2)$$

The following equations fully specify GQ( $\lambda$ ):

$$\delta_t = r_{t+1} + (1 - \gamma_{t+1})z_{t+1} + \gamma_{t+1}\boldsymbol{\theta}_t^\top \bar{\boldsymbol{\phi}}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t \quad (3)$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha[\delta_t \mathbf{e}_t - \gamma_{t+1}(1 - \lambda_{t+1})(\mathbf{w}_t^\top \mathbf{e}_t)\bar{\boldsymbol{\phi}}_{t+1}] \quad (4)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\eta[\delta_t \mathbf{e}_t - (\mathbf{w}_t^\top \boldsymbol{\phi}_t)\boldsymbol{\phi}_t] \quad (5)$$

$$\mathbf{e}_t = I_t \boldsymbol{\phi}_t + \gamma_t \lambda_t \rho_t \mathbf{e}_{t-1} \quad (6)$$

## 3 Algorithm

The following provides a complete algorithm for GQ( $\lambda$ ).

```

GQLearn( $\phi, \bar{\phi}, \lambda, \gamma, z, r, \rho, I$ )
 $\delta \leftarrow r + (1 - \gamma)z + \gamma \boldsymbol{\theta}^\top \bar{\phi} - \boldsymbol{\theta}^\top \phi$ 
 $\mathbf{e} \leftarrow \rho \mathbf{e} + I \phi$ 
 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha (\delta \mathbf{e} - \gamma (1 - \lambda) (\mathbf{w}^\top \mathbf{e}) \bar{\phi})$ 
 $\mathbf{w} \leftarrow \mathbf{w} + \alpha \eta (\delta \mathbf{e} - (\mathbf{w}^\top \phi) \phi)$ 
 $\mathbf{e} \leftarrow \gamma \lambda \mathbf{e}$ 

```

```

Initialize  $\boldsymbol{\theta}$  arbitrarily and  $\mathbf{w} = \mathbf{0}$ 
Repeat (for each episode):
  Initialize  $\mathbf{e} = \mathbf{0}$ 
   $s \leftarrow$  initial state of episode
  Repeat (for each step of episode):
     $a \leftarrow$  action selected by policy  $b$  in state  $s$ 
    Take action  $a$ , observe next state,  $s'$ 
     $\bar{\phi} \leftarrow \mathbf{0}$ 
    For all  $a \in A(s)$ :
       $\bar{\phi} \leftarrow \bar{\phi} + \sum_{a'} \pi(s', a') \phi_{s', a'}$ 
       $\rho = \frac{\pi(s, a)}{b(s, a)}$ 
      GQLearn( $\phi_{s, a}, \bar{\phi}, \lambda(s'), \gamma(s'), z(s'), r(s, a, s'), \rho, I(s, a)$ )
     $s \leftarrow s'$ 
  until  $s'$  is terminal

```

## 4 Code

The file GQLambda.java contains an implementation of the GQLearn function described above. We have deliberately excluded optimizations (e.g., binary features or efficient trace implementation) to ensure the code is simple and easy to understand. We leave it to the reader to provide environment code for interfacing to  $GQ(\lambda)$  (e.g., using RL-Glue).

## 5 References

Maei, H. R., Szepesvari, Cs., Bhatnagar, S., Sutton, R. S. (2010). Toward Off-Policy Learning Control with Function Approximation. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel.

Maei, H. R. and Sutton, R. S.  $GQ(\lambda)$ : A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In Baum, E., Hutter, M., and Kitzelmann, E. (eds.), *AGI 2010*, pp. 9196. Atlantis Press, 2010.

Sutton, R. S., Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.