Reinforcement Learning for Traffic Signal Control

Prashanth L.A.

Postdoctoral Researcher, INRIA Lille – Team SequeL

work done as a PhD student at Department of Computer Science and Automation, Indian Institute of Science

October 2014

On a good day, the traffic is . . .

And on a bad day, it can be . . .

Traffic Light Control (TLC)

Aim: Maximize traffic flow
(long-term performance criterion)

Input:
Coarse congestion estimates

Output:
Policy for switching traffic lights

L1 L2

Low Medium High

Traffic Light Control (TLC)

Aim: Maximize traffic flow
(long-term performance criterion)

Input:
Coarse congestion estimates

Output:
Policy for switching traffic lights

L1 L2

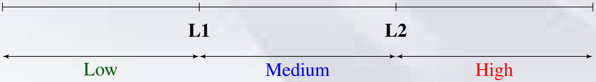
Low Medium High

Traffic Light Control (TLC)

Aim: Maximize traffic flow
(long-term performance criterion)

Input:
Coarse congestion estimates

Output:
Policy for switching traffic lights



Traffic Light Control (TLC)

Aim: Maximize traffic flow
(long-term performance criterion)

Input:
Coarse congestion estimates

Output:
Policy for switching traffic lights

Desirable attributes of TLC algorithm

Dynamic:
Adapts to traffic conditions

Model free:
Do not assume a system model

Scalable:
Easily implementable on large road networks

Solution:
Reinforcement Learning

Desirable attributes of TLC algorithm

Dynamic:
Adapts to traffic conditions

Model free:
Do not assume a system model

Scalable:
Easily implementable on large road networks

Solution:
Reinforcement Learning

Desirable attributes of TLC algorithm

Dynamic:
Adapts to traffic conditions

Model free:
Do not assume a system model

Scalable:
Easily implementable on large road networks

Solution:
Reinforcement Learning

Desirable attributes of TLC algorithm

Dynamic:
Adapts to traffic conditions

Model free:
Do not assume a system model

Scalable:
Easily implementable on large road networks

Solution:
Reinforcement Learning

Desirable attributes of TLC algorithm

Dynamic:
Adapts to traffic conditions

Model free:
Do not assume a system model

Scalable:
Easily implementable on large road networks

Solution:
Reinforcement Learning

Traffic Signal Control MDP

State. $s_n = (q_1, \cdots, q_N, t_1, \cdots, t_N)$

Actions. $a_n = \{\text{feasible sign configurations in state } s_n\}$

Cost.

$$k(s_n, a_n) = r_1 * \left(\sum_{i \in I_p} q_i(n) + t_i(n) \right) + s_1 * \left(\sum_{i \notin I_p} q_i(n) + t_i(n) \right)$$

more weightage to main road traffic

Qlearning based TLC algorithm

Q-learning

$$Q(s_{n+1}, a_{n+1}) = Q(s_n, a_n) + \alpha(n) \left(k(s_n, a_n) + \gamma \min_a Q(s_{n+1}, a) - Q(s_n, a_n) \right).$$

Why function approximation?

- need look-up table to store Q-value for every (s, a)
- **Computationally expensive**
  - two-junction corridor: 10 signalled lanes, 20 vehicles on each lane
  - $|S \times A(S)| \sim 10^{14}$
- Situation aggravated when we consider larger road networks

Q-learning with Function Approximation

Approximation.

$$Q(s,a) \approx \theta^{T} \sigma_{s,a}$$

Parameter $\theta \in \mathbb{R}^d$

Feature $\sigma_{s,a} \in \mathbb{R}^d$

Note: $d << |S \times A(S)|$

Feature-based analog of Q-learning.

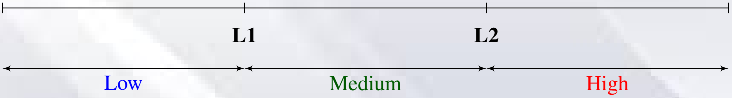
$$\theta_{n+1} = \theta_n + \alpha(n)\sigma_{s_n,a_n}(k(s_n,a_n) + \gamma \min_{v \in A(s_{n+1})} \theta_n^T \sigma_{s_{n+1},v} - \theta_n^T \sigma_{s_n,a_n})$$

Feature Selection

State (s_n)	Action (a_n)	Feature (σ_{s_n,a_n})
$q_i(n) < L1$ and $t_i(n) < T1$	RED GREEN	0 1
$q_i(n) < L1$ and $t_i(n) \geq T1$	RED GREEN	0.2 0.8
$L1 \leq q_i(n) < L2$ and $t_i(n) < T1$	RED GREEN	0.4 0.6
$L1 \leq q_i(n) < L2$ and $t_i(n) \geq T1$	RED GREEN	0.6 0.4
$q_i(n) \geq L2$ and $t_i(n) < T1$	RED GREEN	0.8 0.2
$q_i(n) \geq L2$ and $t_i(n) \geq T1$	RED GREEN	1 0

Threshold tuning using SPSA

- Problem: hard to obtain exact queue lengths in practice
- Solution: Use broad congestion estimates based on thresholds



- How to optimize *Li*'s? Use **Simultaneous Perturbation Stochastic Approximation**
- Combine the optimization procedure with TLC algorithms:
  - Full state Q-learning algorithm with state aggregation
  - Function approximation Q-learning TLC
  - Priority based (naive?) scheme

Feature Adaptation

Recall the approximation.

$$Q(s,a) \approx \theta^T \sigma_{s,a}$$

Parameter $\theta \in \mathbb{R}^d$

Feature $\sigma_{s,a} \in \mathbb{R}^d$

Is to possible to adapt features online to make them optimal?

We propose an *online feature adaptation* algorithm
to find the "optimal" features

Publications I

- **Prashanth L. A.** and S. Bhatnagar,
  Reinforcement learning with function approximation for traffic signal control
  IEEE Transactions on Intelligent Transportation Systems, 2011.
- **Prashanth L. A.** and S. Bhatnagar,
  Threshold Tuning using Stochastic Optimization for Graded Signal Control,
  IEEE Transactions on Vehicular Technology, 2012.
- **Prashanth L.A.** and S.Bhatnagar,
  Reinforcement Learning with Average Cost for Adaptive Control of Traffic Lights at Intersections,
  IEEE Conference on Intelligent Transportation Systems, 2011.
- S. Bhatnagar, V. Borkar and **Prashanth.L.A.**,
  Adaptive Feature Pursuit: Online Adaptation of Features in Reinforcement Learning,
  Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, by F. Lewis and D. Liu (eds.), IEEE Press Computational Intelligence Series.
- S.Bhatnagar, H.L.Prasad and **Prashanth.L.A.**,
  Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods,
  Lecture Notes in Control and Information Sciences Series, Springer (Accepted), 2012.

Publications II

S. Bhatnagar and **Prashanth L. A.**,
Simultaneous Perturbation Newton Algorithms for Simulation Optimization,
Journal of Optimization Theory and Applications, 2013.

Prashanth L. A. and Mohammad Ghavamzadeh,
Actor-Critic Algorithms for Risk-Sensitive MDPs
Advances in Neural Information Processing Systems (NIPS), 2013 (**Full oral presentation**).

The road ahead