Reinforcement Learning-Based Optimal Public Transportation Route Search Considering Time and Cost

SeoYeong Kim
Applied Artificial Intelligence
Sungkyunkwan University
Seoul, Korea
carsk0607@g.skku.edu

***Abstract*—As public transportation is considered useful to commute, research of searching an optimal route to destination has been actively conducted. As traffic increases, road conditions change unexpectedly by sudden accident, traffic congestions or weather and it causes delay of lead time of commutes dynamically. Therefore, searching an optimal route to destination reflecting road conditions in real-time is needed. Traditional methods to search optimal route such as A* algorithm and Dijkstra algorithm cannot reflect non-linear factor and dynamically changing conditions. Also, they cannot generalize road conditions so that they have to search again from the beginning. So, I suggest Reinforcement Learning based methods to search optimal public transportation route reflecting road conditions in real-time. In this paper, I set up a task of arriving at Sungkyunkwan University starting from Seoul Station and implement environment of grid world. Obstacle states which cause delay of commute time are randomly set up. And an action space consists of public transportation such as bus, metro, and walk. To consider time and cost simultaneously, reward consists of two components: time reward and cost reward. And total reward considers both components at once. In this paper, Q-learning, SARSA, and Expected SARSA are applied. As a result, Expected SARSA performs best at overall. In detail, Expected SARSA performs best in terms of time and SARSA performs best in terms of cost. Future work of importing data in real-time using IoT technology and learning by setting up more specific environment is needed.**

Keywords—reinforcement learning, optimal route search, Q-learning, SARSA, Expected SARSA

I. INTRODUCTION

Most people use public transportation for their daily commutes. But, there are some factors that can unexpectedly affect on public transportation such as traffic congestion, weather conditions, or traffic control [1]. In this kind of situation, people become aware of disruptions of public transportation on their way, and it causes significant inconvenience in commutes. Therefore, a method to search optimal public transportation route reflecting real-time changes in road conditions is needed.

Traditionally, researches of finding optimal route are mainly based on A* algorithm or Dijkstra algorithm. But these methods cannot consider non-linear factor like road conditions such as traffic congestion, weather conditions, or traffic control. Also, these methods cannot reflect dynamic environment which changes by time step. Moreover, these algorithms cannot generalize the environment, so they cannot be applied when the situation changes. It makes search again from the beginning and it takes much time. So, I apply Reinforcement Learning method to solve this problem. I apply Q-learning, SARSA, Expected SARSA to this problem and compare the results. Also, optimal route must be not only fast but also cheap. To find fastest and cheapest way to arrive at the destination, I consider reward in two aspects: time reward and cost reward. Total reward must consider both two components. Background is presented in section II, and method is described in section III. Finally, results are given in section IV.

II. BACKGROUND

A. Search optimal route

A* algorithm and Dijkstra algorithm are well known method to search optimal route. A* algorithm finds the straight-line distance value to the target from starting point by using a heuristic method, which assigns priority [2]. Dijkstra algorithm calculates the shortest distance from the starting point to all points on the graph. It is valid only when the weights of all edges are positive [3]. These algorithms cannot consider non-linear factor like road conditions such as traffic congestion, weather conditions, or traffic control. Also, when road conditions change while searching, they should reset the topology and calculate again [3]. Moreover, they cannot generalize the environment, so they only can be applied when prior knowledge of the road conditions are known. So these methods are not appropriate to find optimal routes when the environment changes in real-time.

B. Reinforcement Learning

Reinforcement Learning is a method which imitates the learning of people and animals. It is a method of learning when agent receives feedback in the form or rewards for its action by interacting with a given environment. It aims to maximize the rewards. And it can find optimal policy by interacting with the environment.

C. Q-learning

Q-learning is one of the off-policy methods. It learns which action to take in the current state can maximize the rewards by using Q-function. Q-learning can respond to dynamic changes and figure out the route when the connections of nodes are not known yet. Also, it can update rewards in real-time [2].

D. SARSA

SARSA is one of the on-policy methods. Policy evaluation and policy improvement occur simultaneously. It learns reward of certain state and certain action by using current state, action, reward and next state, action. It only consider the next action which it will take.

E. *Expected SARSA*

SARSA is also one of the on-policy methods but in terms of using different policy from the target policy to generate behavior, it can be considered one of the off-policy methods. Unlike SARSA, Expected SARSA consider all of the next action which it can take.

III. METHOD DESIGNED

In this paper, a virtual environment of a 16*16 grid world has been implemented. And it learn a route from Seoul Station to Sungkyunkwan University.

Pic. 1. Grid world implemented in this paper

A. *State*

Starting point is defined to Seoul Station and destination to Sungkyunkwan University. And obstacle region which can cause delay is randomly selected.

B. *Action*

Action space consists of bus, metro, and walk. When agent takes action of bus or metro, position of agent changes to next station of the bus or metro, reflecting reality. When agent takes action of walk, position can change by 1 cell in grid world. The action space defined in this paper is followed. When agent arrives at certain state, it verifies whether the state is included in each bus or metro station. When it is verified to be included in each station, the agent can take the bus or metro action which includes that state as a station. To reflect the real characteristic of public transportation that only can be used in specific locations, distinct actions for each state are configured in this paper. Walk is considered as default action.

TABLE I. ACTION SPACE

Index	Action	Index	Action
0	Walk (up)	9	Bus 301
1	Walk (down)	10	Bus 704
2	Walk (left)	11	Bus 1102
3	Walk (right)	12	Bus 7022
4	Bus 103	13	Metro line 1
5	Bus 104	14	Metro line 2
6	Bus 151	15	Metro line 3
7	Bus 173	16	Metro line 4
8	Bus 201	17	Metro line 5

Fig. 1. Action Space

C. *reward*

To find not only shortest or cheapest but also satisfy both elements, reward is divided into 2 components: time reward and cost reward. Reward defined in this paper is followed.

TABLE II. REWARD

	Time			Cost
	goal	***obstacle***	***normal***	
Bus	-10	-150	-10	-1500
Metro	-15	-150	-15	-1400
Walk	-150	-150	-150	0

Fig. 2. Reward

Total reward formula defined in this paper is (1). In learning, total reward is used. Time reward and cost reward is only used to compare at the end of all episodes.

$$total\ reward = \frac{time\ reward}{2} + \frac{cost\ reward}{30} \quad (1)$$

D. *Algorithm*

In this paper, above three algorithms are applied. And in each algorithm, 30 episodes are executed.

1) Q-learning

Parameters: step size α, small ε

Loop for each episode:
  Initialize S
  Initialize time R sum, cost R sum, R sum
  Loop for each step of episode:
    Choose A from S using policy derived from Q
    Take action A, observe time R, cost R, total R, S'
    time R sum = time R sum + time R
    cost R sum = cost R sum + cost R
    total R sum = total R sum + total R
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
    S ← S'
  until S is terminal

2) SARSA

Parameters: step size α, small ε

Loop for each episode:
  Initialize S
  Initialize time R sum, cost R sum, R sum
  Choose A from S using policy derived from Q
  Loop for each step of episode:
    Take action A, observe time R, cost R, total R, S'
    time R sum = time R sum + time R
    cost R sum = cost R sum + cost R
    total R sum = total R sum + total R
    Choose A' from S' using policy derived from Q
    time R sum = time R sum + time R
    cost R sum = cost R sum + cost R
    total R sum = total R sum + total R
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
    S ← S'
    $A \leftarrow A'$
  until S is terminal

3) Expected SARSA

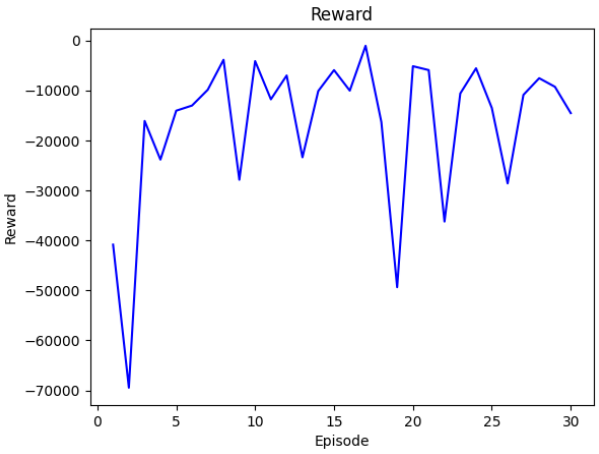
Parameters: step size α, small ε

Loop for each episode:
  Initialize S
  Initialize time R sum, cost R sum, R sum
  Loop for each step of episode:
    Choose A from S using policy derived from Q
    Take action A, observe time R, cost R, total R, S'
    time R sum = time R sum + time R
    cost R sum = cost R sum + cost R
    total R sum = total R sum + total R
    $Q(S,A) \leftarrow Q(S_t,A_t) + \alpha[R_{t+1} + \gamma \sum \pi(a|S_{t+1})Q(S_{t+1},a) - Q(S_t,A_t)]$
    S ← S'
  until S is terminal

IV. Results

I represent a graph using matplotlib in python. In each episode, time R sum, cost R sum, and total R sum is calculated. After each episode ends, these numbers are stored. And after all episodes end, draw a line graph.

A. *Q-learning*



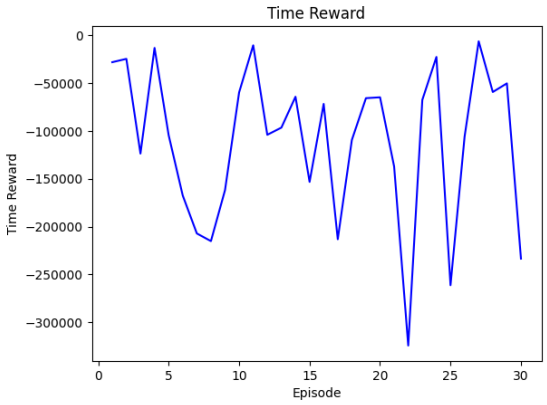
Pic. 2. Total reward graph of Q-learning

Pic. 3. Time reward graph of Q-learning

Pic. 4. Cost reward graph of Q-learning

B. *SARSA*

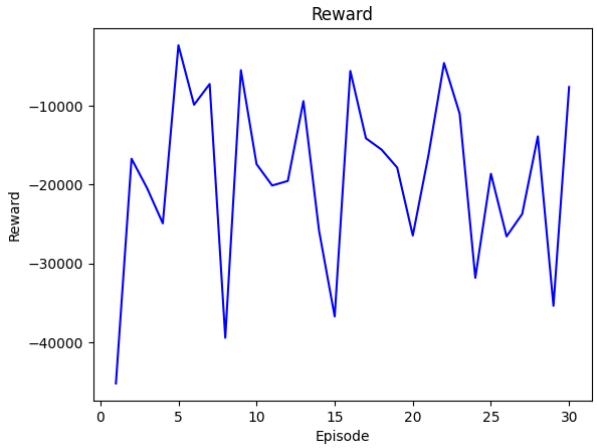
Pic. 5. Total reward graph of SARSA



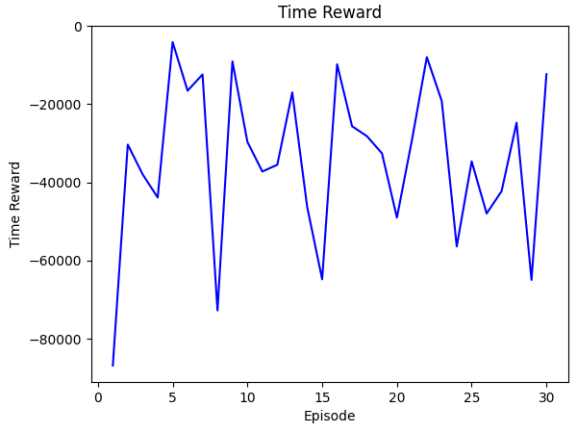
Pic. 6. Time reward graph of SARSA

Pic. 7. Cost reward graph of SARSA

C. *Expected SARSA*



Pic. 8. Total reward graph of Expected SARSA



Pic. 9. Time reward graph of Expected SARSA

Pic. 10. Cost reward graph of Expected SARSA

Overall, Expected SARSA performs best. In terms of time, Expected SARSA shows the highest reward. In terms of cost, SARSA shows the highest reward.

In conclusion, Expected SARSA is appropriate to search optimal route in dynamically changing environment. SARSA can be used when specially cost is considered significantly. In future works, importing data in real-time using IoT technology should be handled. Also, if the environment such as reward based on state and action is specified, performance would be better.

REFERENCES

[1] G. Neelakantam, D. D. Ontnoni, R. K. Sahoo, “Reinforcement Learning Based Passeners Assistance System for Crowded Public Transportation in Fog Enabled Smart City”, Electronics, vol. 9, pp. 1501-1519, September 2020

[2] C. Seung-Hee and Y. Sang-Jo, “Q-learning Based Optimal Escape Route Decision in a Disaster Environment”, The Journal of Korean Institute of Communications and Information Sciences, vol. 46, no. 4, pp.638-650, April 2021

[3] G. Da-Sol and L. Tae-Kyung, “Optimal Path Search using Reinforcement Learning Technique”, Korean Information Processing Society, vol. 21, no. 2, pp.886-889, November 2014