

# Matching Matrix를 사용하여 운전자와 승객의 관계를 반영한 강화학습 기반 유동적인 가격 책정 체계

## Dynamic Pricing Based on Reinforcement Learning Reflecting the Relationship between Driver and Passenger Using Matching Matrix

박 준 형\* · 이 찬 재\*\* · 윤 영\*\*\*

\* 주저자 : 홍익대학교 정보컴퓨터공학부 컴퓨터공학전공 학사과정  
 \*\* 공저자 : 홍익대학교 산업융합협동과정 인공지능·빅데이터전공 석사과정  
 \*\*\* 교신저자 : 홍익대학교 컴퓨터공학과 조교수

Jun Hyung Park\* · Chan Jae Lee\*\* · Young Yoon\*\*\*

\* Dept. of Computer Engineering, Univ. of Hongik  
 \*\* Dept. of Artificial Intelligence·Big Data, Univ. of Hongik  
 \*\*\* Dept. of Computer Engineering, Univ. of Hongik  
 † Corresponding author : Young Yoon, young.yoon@hongik.ac.kr

Vol.19 No.6(2020)

December, 2020  
 pp.118~133

pISSN 1738-0774  
 eISSN 2384-1729  
<https://doi.org/10.12815/kits.2020.19.6.118>

Received 11 November 2020  
 Revised 4 December 2020  
 Accepted 17 December 2020

© 2020. The Korea Institute of Intelligent Transport Systems. All rights reserved.

### 요 약

최근 통합교통서비스(Mobility-as-a-Service)의 개념을 도입하여 이용자들의 이동성과 접근성을 향상시키고자 하는 연구가 진행되고 있다. 특히 카셰어링, 택시 등에 대해 수요와 공급에 따라 지역을 구분하여 가격을 책정하는 유동적인 가격 책정 전략을 도입하여 단일 요금제가 가지는 서비스 기피 등의 문제를 해결함과 동시에 기업과 운전자들의 수익성에 긍정적인 영향을 줄 수 있을 것으로 기대되고 있다. 본 연구에서는 승객과 운전자간의 배차거리, 승객의 운행거리, 승객의 목적지에 대한 HDBSCAN 알고리즘을 통해서 정밀하게 인식된 수요 밀집지역, 승객과 운전자가 생각하는 선호가격을 고려하여 승객과 운전자의 입장에서 Matching Matrix를 생성한다. 이를 조합하고 보상에 반영하여, 강화학습이 더욱더 현실적인 유동적인 가격 책정 전략을 도출할 수 있는 새로운 방법론을 제안한다.

핵심어 : 강화학습, 동적 가격 책정, 매칭 매트릭스, 승객과 운전자의 관계, 통합교통서비스

### ABSTRACT

Research interest in the Mobility-as-a-Service (MaaS) concept for enhancing users' mobility experience is increasing. In particular, dynamic pricing techniques based on reinforcement learning have emerged since adjusting prices based on the demand is expected to help mobility services, such as taxi and car-sharing services, to gain more profit. This paper provides a simulation framework that considers more practical factors, such as demand density per location, preferred prices, the distance between users and drivers, and distance to the destination that critically affect the probability of matching between the users and the mobility service providers (e.g., drivers). The aforementioned new practical features are reflected on a data structure referred to as the Matching Matrix. Using an efficient algorithm of computing the probability of matching between the users and drivers and given a set of precisely identified high-demand locations using HDBSCAN, this study developed a better reward function that can gear the reinforcement learning process towards finding more realistic dynamic pricing policies.

Key words : Reinforcement Learning, Dynamic Pricing, Matching Matrix, Relationship of Driver and Passenger, Mobility as a Service

## I. 서 론

2004년 7월부터 시행된 서울 버스와 수도권 전철 간의 통합 환승 할인제로 교통의 수요는 더욱더 폭발적으로 증가하였으나, 기존 대중교통 서비스는 도어투도어 (Door-to-Door) 서비스와는 거리가 있기 때문에, 퍼스트마일 (First Mile) 과 라스트마일 (Last Mile) 을 충족시키기 위한 카셰어링이나 자전거, 전동 킥보드 등의 다양한 공유 모빌리티 서비스들이 등장하였다. 최근에는 공유 서비스를 기존의 대중교통 수단과 연계하여 이용자들의 이동성과 접근성을 향상시키고자 하는 통합모빌리티서비스 (Mobility-as-a-Service) 가 연구되고 있다. 그러나 모빌리티 서비스의 단일 요금제가 가지는 수익성에 대한 문제로 광고 수익 등으로 손실을 메꾸거나 정부의 지원에 기대는 경우가 여전히 많다. 또한 수요가 낮은 지역에서의 서비스 기피로 인한 공급의 불균형이 발생하고 있다. 이로 인한 이용자들의 불편함을 해결하고, 모빌리티 서비스의 수익성을 극대화하기 위해 수요와 공급량 및 다양한 상황에 따라 탄력적으로 가격을 변화시키는 유동적인 가격 책정과 같은 전략이 논의되기 시작하였다. 그러나 유동적인 가격 책정은 고객 행동특성, 공정요금, 시장구조, 제품의 수요 및 공급, 제품에 대한 인식, 계절성 등 다양한 요인에 의해서 구성되며 고객의 신뢰도에 영향을 미쳐 공정성의 문제를 발생시킬 수 있다는 우려가 있다. Haws and Bearden(2006)은 유동적 가격책정 방식의 공정성 문제를 파악하기 위하여 판매자, 소비자 및 시간 관점에 따른 인식을 분석하였고, 소비자 간의 가격차이가 불공정성과 가장 낮은 만족도를 유발하는 것으로 나타났다. 그러나 소비자가 가격 책정 과정에 참여할 경우 공정성과 만족도가 높은 것으로 확인되었으며, 미국의 라이드셰어링 업체인 우버나 리프트는 실제로 승객의 수요와 택시의 공급 정보와 인공지능에 기반하여 유동적인 가격정책을 수립하여 수익성을 향상시키고, 이용자들의 만족도를 향상시키고 있다.

과거 유동적인 가격 책정은 경쟁사 정보 및 수요통계와 같은 시행착오를 통하여 감각적으로 가격을 책정하여 5개 이내의 세부 구간에 따라 가격 차별화를 수행하였지만, 수요와 공급 관련 데이터에 기반한 보다 정교한 가격 차별화 전략들이 실시간 빅데이터 분석에 기반하여 개인 맞춤형 전략으로 진화하고 있다. Wu et al.(2016)은 수요 대응형 승차 요청 서비스와 유사한 동적 상황 시뮬레이션을 구축하여 다양한 방법의 가격책정 대안 결과를 비교 분석하였다. 이때, 강화학습을 통한 유동적인 가격 책정 결과가 플랫폼의 수익, 택시 기사의 수입 측면에서 확연하게 좋은 결과를 나타내었음을 볼 수 있다.

본 연구에서는 유동적 가격 정책을 펼치는 택시 서비스를 가정하고 강화학습 기반 유동적인 가격 책정 시뮬레이션 진행 시 기존에는 고려되지 않은 한계점에 대해 분석한 후 이를 해결하기 위해 지역별 수요 밀도, 운전자와 승객간의 거리, 운행 거리 등 실시간으로 도로 위의 다양한 요소를 고려한 Matching Matrix를 구성하였다. 이를 활용해 보상 함수를 조정하여 보다 현실적인 강화학습 기반 유동적인 가격 책정 시뮬레이션 체계를 구성하고자 한다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 유동적인 가격 책정에 관련된 다양한 연구과 강화학습을 통한 유동적인 가격 책정에 대해 검토하고, 최적의 수익성을 고려한 강화학습 기반 유동적인 가격 책정 시뮬레이션 체계 구성 및 한계점을 분석한다. III장에서는 한계점을 개선하기 위해 운전자와 승객의 관계를 고려한 Matching Matrix를 활용하여 보상 함수를 조정한 강화학습 기반 유동적인 가격 책정 시뮬레이션 모델을 구축한다. IV장에서는 모델의 실험 결과를 확인하고 시각화를 통해 결과를 분석하고자 한다. 마지막 V 장에서는 실험에 따른 결론과 시뮬레이션 체계의 확대 적용 등 정책적 제언을 하고자 한다.

## II. 관련 연구 고찰

### 1. 유동적인 가격 책정 개념

유동적 가격 책정 (Dynamic Pricing) 은 동일한 제품 및 서비스에 대한 가격을 다양한 시장 상황에 따라 탄력적으로 변화시키는 전략이다. 이는 시간에 따라 적정 가격을 변화시켜 공급의 주체에게는 최대의 이익을 제공하면서도 서비스를 이용하는 고객에게는 수용 가능한 가격 정책 선에서 보다 양질의 서비스를 제공 할 수 있도록 한다. 이러한 유동적인 가격 책정은 기간 만료로 소멸되거나 공급량이 한정된 제품 및 서비스들을 취급하는 항공사, 호텔 및 판매자가 재고를 보관할 수 있는 전자 상거래 산업에서 주로 알려져 왔는데, 이는 가격을 쉽게 조정할 수 있는 환경에서 제품 및 서비스 판매를 최적화 할 수 있기 때문이다. 또한, 유동적인 가격 책정은 승차 공유 서비스에서도 요금에 대한 최적화를 통해 형평성 측면과 수익성 측면에서 긍정적인 영향을 제시하고 있다.

### 2. 교통 분야에서 유동적인 가격 책정 접목

교통 분야에서 유동적인 가격 정책을 적용하기 위해 제안된 방법으로 수요공급 변화 확률 기반 모형, 휴리스틱 모형, 기계학습 알고리즘, 강화학습(Q-learning) 등이 있었으나, 본 연구에서 진행한 Matching Matrix를 통한 새로운 보상 함수를 적용하기 위해서는 수요와 공급 이외에 승객과 운전자간의 관계 등의 다른 요소를 고려 할 수 있어야 하는데, 수요공급 변화 확률 기반 모형은 이를 고려하지 못한다. 또한 Bertsimas and Perakis(2006)에서 소개된 휴리스틱 모형, 다양한 차원의 다이내믹 프로그래밍에 기반한 유동적인 가격 정책 들은 마르코프 의사 결정과정에서 전환 함수에 대해 완벽하게 설명할 수 없었다. 따라서 본 연구에서는 Model-Free 접근법을 이용한 강화학습을 통해 유동적인 가격 정책 시뮬레이션을 진행하고자 하였다.

변동 가격의 하한선 및 상한선을 결정하는데 있어서 공정함과 신뢰성에 영향을 줄 수 있는데, Wu et al.(2016)에 따르면, 기계학습 알고리즘 및 강화학습을 이용한 유동적인 가격 정책 시뮬레이션에서 특별한 상한선 제한이 없어도 초기가격에서 3배 이상 넘어가는 경우가 없다고 한다. 다만, 기계학습 알고리즘에 비해 본 연구는 시장가격 뿐 아니라 승객의 매칭 확률이 높을수록 보상이 높게 책정되도록 강화학습의 장점이 활용되었고, 이는 에이전트가 높은 보상을 받기 위해 매칭이 될 확률이 높아지도록 공정한 가격을 책정함에 있어 운전자와 승객들에게 공정성과 신뢰성을 제공할 것으로 보인다. 또한 Wu et al.(2016)의 연구에서는 택시의 수요 · 공급의 시장 경제 속에서 다양한 가격 책정 대안을 구축하여 결과를 비교하였다. 이때, Statistic Pricing, Proportional Pricing, Batch Update 및 강화학습을 통한 유동적인 가격 책정 전략들을 수익성 측면과 승객의 매칭 비율에 대한 결과를 통해 비교 분석하였다. 그 결과 기업들은 실시간 변화에 대응 가능한 방법을 필요로 하여 더 이상 예측하는 것만으로는 효율적이지 않으며, 강화학습 기반의 에이전트가 승객을 고려하는 최적의 비용을 통해 매출 및 이익 극대화와 관련하여 최고의 성능을 나타냄을 확인하였다.

Castillo et al.(2017)은 승차공유서비스의 공차 문제인 WGC (Wild Goose Chase) 를 해결하기 위한 방안으로 서지 가격 책정 전략을 제안하였고 그 효과를 입증하였다. 과부하 상태인 플랫폼에서 멀리 있는 손님을 태우기 위해 공차 상태로 장거리에 차를 보내게 될 경우 공차 상태의 증가로 인해 고객 수가 감소하게 된다. 이는 운전자의 수익감소에도 영향이 생기기 때문에 운전자의 감소로 이어지게 되며 결과로 플랫폼의 수익이 감소하는 등의 문제가 발생한다. Song et al.(2020)은 승차거부 및 수요가 적은 지역이 목적인 승객이 택시 서비스를 이용함에 있어서 발생하는 형평성 문제를 다뤘다. 심야와 교외 지역에서 수요와 공급의 불균형이

심화되는 문제를 해결하기 위해 강화학습 기반 유동적인 가격 책정을 통한 솔루션을 제안하였다. 오후 10시에서 오전 4시 사이에 적절한 승차 공유 서비스를 위한 가격 결정 시뮬레이션을 구성하였는데, 먼저 교통 서비스 측면에서 공간적으로 소외된 지역을 분석하고 지역(구) 간의 이동 정보를 사용하여 내향/외향 중심성이 라는 지역 지표를 개발하고 이를 강화학습 시뮬레이션에 적용하여 실험하였다.

### 3. 서지 히트 맵

위의 연구들에서는 유동적인 가격 책정의 장점과 강화학습을 이용하였을 때 최적의 수익성을 고려할 수 있다는 결과들을 보았다. 특히 Wu et al.(2016)의 연구에서는 다소 공간적인 요소가 부족하였다. 실제 상용화를 위해서는 지도에 지역에 따라 가격을 다르게 책정해야 할 필요가 있었고 Lu et al.(2018)의 연구에서는 우버 운전자의 지역 이동과 유동적인 가격 책정의 데이터를 토대로 수익을 분석하였다. 특히 ‘서지 히트 맵 (Surge Heat Map)’은 육각형의 단위로 주요 10개 도시를 구분하여 각각 가격을 다르게 책정하였다. 우버의 서지 히트 맵 서비스가 중단되었을 때와 평소간의 차이를 비교하였고, 중단 되었을 때 수익이 현저히 낮게 나타났다는데, 이는 운전자가 서지 히트 맵을 보고 가격이 높거나 낮은 지역으로 이동하여 수익을 최대 약 70%까지 극대화시킨 것으로 나타났다. 이를 통해 유동적인 가격 책정이 적용된 서지 히트 맵은 운전자를 수요가 가장 높은 곳으로 이동하게 만들면서 수익을 증가시켜주고, 공간 검색 마찰과 공간 불일치가 감소하여 승객의 대기 시간과 복지가 향상되었다. Guo et al.(2017)은 우버와 같은 주문형 승차 서비스 (Ubiquitous Service)와 기존 택시 서비스와의 차이를 비교하였는데, 수요의 특성과 수요와 공급을 일치시키는 유동적인 가격 책정 메커니즘을 분석하여 승객의 주문 및 결제 정보 데이터를 토대로 수요 및 동적 가격 분석에 중점을 두었다. 수요 분석으로 일반적인 특성, 승객 그룹화 및 수요 클러스터링을 다루었는데, 거리 기반 클러스터링의 대표적인 기법 중 K-means 클러스터링을 활용하여 수요에 대해 클러스터링 하였다. 또한 클러스터 마다 동적 가격 승수와 패턴을 결정하는 것에 대해 다루었다.

대부분의 선행 연구에서는 시간적인 요소만을 고려한 도로 위의 고정된 상황을 가정하여 실험 및 분석하였으며, 공간적인 요소를 고려한 연구에서는 지역마다 가격을 책정할 때 고정된 단위로 지역을 분리하여 가격을 측정하였고, 수요 좌표 데이터들을 거리에 따라 클러스터링 하였다. 실제 도로 위의 상황에서 규격화된 서지 지역으로 구분하여 가격을 책정하는 것은 현실적인 측면에서 고의적으로 이러한 시스템의 단점이 악용될 수 있는 문제가 있다. 본 논문에서는 운전자가 서지 지역까지 다른 거리를 이동해야하는 공간적인 관점을 고려하여 기본 모델을 확장하고, 승객들의 수요 밀도 기반 클러스터링을 통해 서지 지역을 구분하여 가격을 책정하여 고의적으로 시스템의 단점을 악용 할 수 없으면서도 클러스터에 따른 가격 정책을 제안하는 방법으로 ‘Matching Matrix’ 모델을 제안하고자 한다.

## Ⅲ. 분석방법론

### 1. 분석 개요

본 논문은 유동적인 가격 책정을 실시하는 택시 서비스를 가정하고, 택시 운전자와 승객간의 관계를 반영한 Matching Matrix를 바탕으로 한 새로운 보상 함수를 통해 기존 강화학습 기반 유동적인 가격 책정 시뮬레이션에서 발생한 한계점을 극복하고자 한다. 본 연구에서의 강화학습의 목표는 유동적인 가격 책정을 실시하는

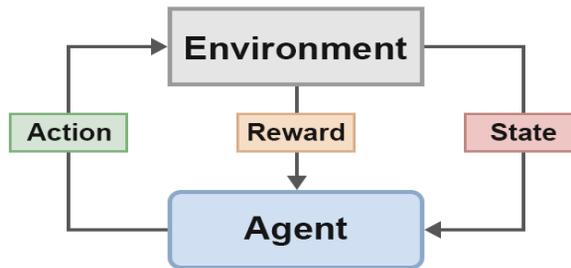
택시 서비스의 수익성 극대화에 초점을 맞추고 있다. 즉, 본 논문의 강화학습 시뮬레이션 에이전트는 가격을 결정하는 택시 서비스 플랫폼이고, 이 에이전트는 시간  $t$ 에서 도시 및 구 단위 지역 내에서 발생한 수요와 공급 간의 다양한 상황과 관계를 고려하여 가격을 결정한다. 이때, 결정된 택시의 시장 가격과 운전자와 승객이 매칭된 수에 비례하여 가격 결정 에이전트에 대한 수익의 일부 금액이 발생하는 것으로 하였다.

보다 현실적인 강화학습 시뮬레이션 진행을 위해 미국 뉴욕시에서 제공하는 실제 택시 운행 데이터에 기반한 분석을 하고자 한다. 뉴욕시 데이터의 경우는 공차 운행 정보가 없고, 승객을 태운 택시의 정보만 있다. 따라서 공차 운행 정보를 새로 모델링하는데, 직전 하차 정보를 중심으로 수요가 밀집된 지역에 대해 추가로 고려할 필요가 있다고 판단하였다. 이후, 현실적으로 정확히 알 수 없는 운전자의 위치를 반영하기 위해 전체 지역에 랜덤으로 운전자의 위치를 낮은 비율로 생성하고자 하였다. 이를 정리하면 다음과 같다.

1. 직전 하차 정보를 이용하여 운전자의 위치를 생성하는 방법 (50%)
2. 수요가 밀집된 지역 주변에 운전자의 위치를 생성하는 방법 (35%)
3. 전체 지역에 랜덤으로 운전자의 위치를 생성하는 방법 (15%)

## 2. 강화학습 기반 유동적인 가격 책정 시뮬레이션 환경 구성

### 1) 강화학습 방법론

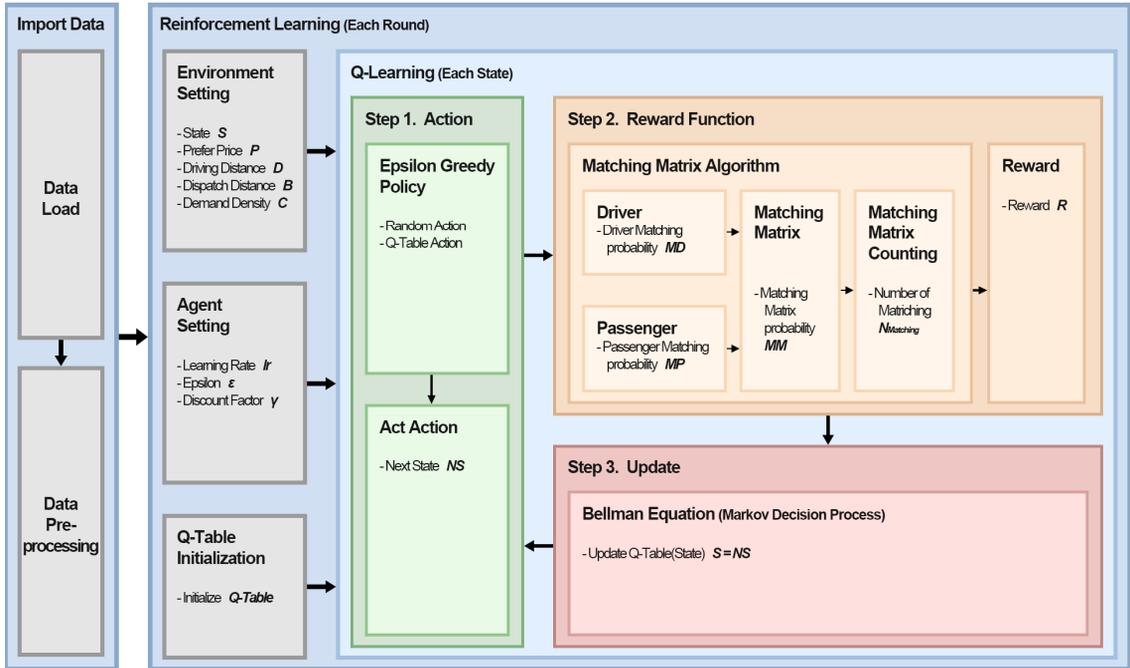


<Fig. 1> Reinforcement Learning Algorithm

강화학습의 목표는 주어진 환경 (Environment) 에서 보상을 최대한 많이 받을 수 있도록 에이전트 (Agent) 를 학습시키는 것인데, 에이전트는 주변 상태 (State) 와 정책 (Policy) 에 맞게 어떠한 행동 (Action) 을 할지 판단을 내리고, 속한 환경에서 결정한 행동에 따라 상태가 바뀌게 되며, 이러한 행동에 따라 보상 (Reward) 을 받을 수 있다. 이러한 강화학습은 다양한 분야에서 사용되지만, 관련 연구 고찰 부분에서 언급하였듯이 유동적인 가격 책정을 시뮬레이션하고 분석한 여러 연구들이 존재한다. 마찬가지로 본 연구에서도 마르코프 의사 결정과정 (Markov Decision Process) 과정으로 유동적인 가격 책정 환경을 구성하였다. 이후 큐 러닝 (Q-Learning)을 반복하여 에이전트를 학습하였다.

이러한 강화학습 알고리즘에 기반하여 본 연구에서 제안하고자 하는 접근 방법의 전반적인 흐름을 표현 하면 <Fig. 2>와 같이 나타낼 수 있으며, 전반적인 흐름을 설명하자면, 데이터를 입력 받아 매 1시간 단위의 라운드마다 총 50,000 단계의 큐 러닝을 반복 수행하여 최적의 가격을 도출하게 되는데, 매 라운드마다 환경과 에이전트에 대한 설정 값을 가져오고, 큐 테이블을 초기화한다. 이후 각 라운드 내에서 각 상태별로 행동, 보상 함수, 업데이트 순으로 3단계를 하나의 업데이트 과정으로 한다. 각 단계 별로 살펴보면, 먼저 행동에서 입실론 탐욕 정책(Epsilon Greedy Policy)를 통해 입실론(Epsilon)에 따라 랜덤한 행동을 취할지, 아니면 큐 테이블에 기반한 행동을 취할지를 정하고, 이에 맞는 행동으로 다음 상태(Next State)인  $NS$ 를 구하고, 이와 함

계 환경과 에이전트, 큐 테이블을 입력으로 사용하여 보상 함수를 진행한다. 이 단계에서 본 논문에서 제안하고자 하는 Matching Matrix를 생성하는데, 운전자와 승객의 입장에서 각각에 해당하는 매칭 확률인  $MD$ 와  $MP$ 를 구한다. 이후 이를 곱하여 Matching Matrix 확률인  $MM$ 을 구하고, Matching Matrix로부터 매칭 수  $N_{Matching}$ 을 구한다. 이로부터 최종적인 해당 상태의 보상  $R$ 을 구하고, 이렇게 구한 값을 벨만 최적 방정식에 따라 큐 테이블에 상태를 업데이트 해준다. 이후 이러한 과정을 반복 연산하여 강화학습을 진행한다.



<Fig. 2> Flowchart (The overall procedure for Reinforcement Learning using Matching Matrix)

## 2) 마르코프 의사 결정과정

마르코프 의사 결정과정은 다음과 같이 구성된다.

1. 일련의 환경 및 에이전트의  $State S$
2. 에이전트가 할 수 있는 일련의  $Action A$
3. 어떠한 상태에서 에이전트의 행동에 대한  $Reward Function R(s, a)$
4. 각 상태에 대한 각 행동의 영향을 설명하는  $Transpose Function T(s, a, s')$

Van and Wiering(2012)에서 제안된 마르코프 의사 결정과정을 우리의 시뮬레이션에 적용하기 위해서 에이전트를 가격 결정 플랫폼 (시스템) 으로 설정하였고, 환경에서 택시의 시장 가격을 상태로, 에이전트가 취할 수 있는 행동을 택시의 시장 가격을 조정하는 것으로 설정하였다. 보상 함수 (Reward Function) 은 조정된 택시의 시장 가격과 운전자와 승객이 매칭 된 수에 비례하여 가격 결정 에이전트에 대한 수익의 일부 금액이 발생하므로 두 가지 변수를 곱한 값으로 설정하여, 보다 현실적으로 플랫폼의 입장에서 수익성을 극대화하기 위한 모델을 제안하고자 한다.

### 3) 큐 러닝

강화학습을 통해 마르코프 의사 결정과정을 구성할 때, 전환 함수 (Transpose Function)  $T$ 에 대해 완벽하게 설명할 수 없다는 한계점이 존재한다. 따라서 정확히 전환 함수를 모르더라도 접근 가능한 방법으로 Model-Free 접근법을 활용하여 시뮬레이션을 구성한다. 큐 러닝 알고리즘은 초기에  $Q$ 를 0으로 초기화시킨 후, 시간  $t$ 에서 에이전트에게 주어진 상태에서 선택한 행동에 따라 보상과 함께  $Q$ 를 학습한다. 즉, 주어진 상태에서 주어진 행동에 따라 기대 효용성을 제공하는 행동-가치 함수 ( $Q$  함수)를 학습하도록 하며, 이를 Sutton and Barto(2018)에서 제안된 벨만 최적 방정식으로 표현할 수 있다. 벨만 최적 방정식은 다음 식(1)은 다음과 같다.

$$Q(s_t, a_t) = (1 - \alpha_t) Q(s_t, a_t) + \alpha_t [R(s_t, a_t, s_{t+1}) + \gamma \max_a Q(s_{t+1}, a)] \dots\dots\dots (1)$$

여기서,  $Q(s_t, a_t)$  = 현재 가치  
 $\alpha$  = 학습률 (Learning Rate)  
 $\gamma$  = 감가율 (Discount Factor)

현재 가치는 State  $s$ 에서 Action  $a$ 을 취할 때의 값이 되고, 학습률은 0 ~ 1 사이의 값으로 즉각적인 보상 또는 미래에 받을 보상의 기댓값에 대해  $Q$  값을 업데이트하는 정도를 나타낸다. 감가율은 미래의 보상은 현재의 보상보다 가치가 낮기 때문에, 보상에 미래 시점의 시간 단계만큼 곱해주는 0.0 ~ 1.0 사이의 실수 값이다. 감가율이 높아질수록 에이전트는 현재 보상보다 미래의 보상에 대해 보다 중요하게 판단하게 된다. 본 연구에서는 실험적으로 학습률과 감가율을 설정하였는데, 학습률을  $\frac{500}{(500 + Time Step)}$  으로 설정하여 각 라운드에서 시간 단계가 0 ~ 50,000 진행 될 때, 학습률은 1에서 약 0.01로 순차적으로 낮아져  $Q$ 에 대해 영향력을 낮춘다. 또한 이때의 감가율은 0.9로 설정하였다.

### 4) 입실론 탐욕 정책

강화학습에서 에이전트는 보상을 최대로 받기 위해 주어진 상태에 따라 최적의 행동을 선택한다. 이 때, 에이전트는 탐험 (Exploration) 과 이용 (Exploitation) 이라는 두 가지 선택지로부터 하나의 행동을 결정하여 행동하는데, 먼저 탐험은 상태를 고려하지 않고 랜덤한 행동을 취하여 행동에 대한 보상들을 큐 테이블에 업데이트하는 것이다. 이용은 큐 테이블을 활용하여 주어진 상태에 대해서 최적의 행동을 선택하는 것으로, 탐욕적으로 행동을 선택 할 수 있다. 이러한 탐험과 이용을 통해서 큐 테이블을 업데이트하고, 업데이트된 큐 테이블을 이용 할 수 있게 된다.

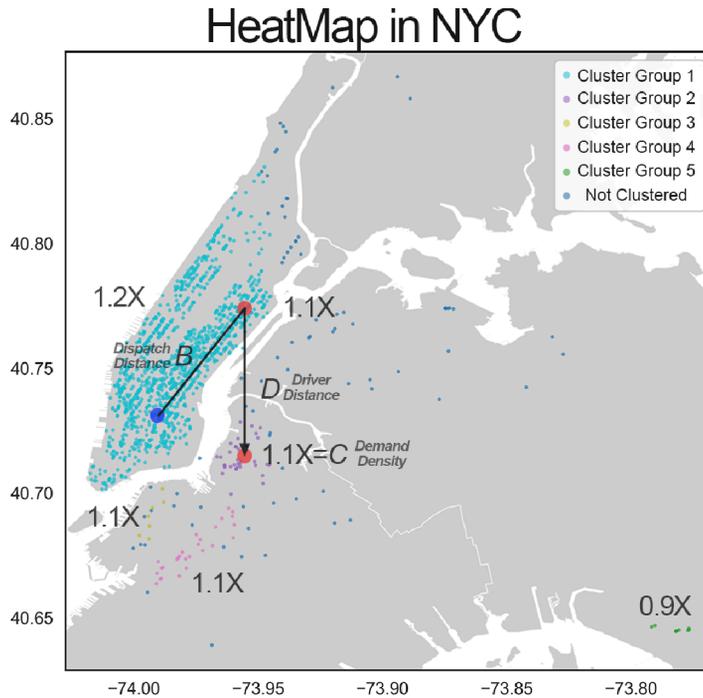
입실론 탐욕 정책의 입실론  $\epsilon$ 의 값에 따라 난수  $p$ 를 비교하여 에이전트가 탐험과 이용 중 하나의 행동을 할 것으로 결정하게 되는데, 이 때 난수  $p$ 가 입실론  $\epsilon$ 보다 작은 경우에는 탐험을 하게 되고, 크거나 같은 경우에는 큐 테이블을 이용하게 된다. 본 연구에서는 매 라운드 마다 입실론  $\epsilon$ 를 1.0으로 초기화하였고, 이후 단계 마다 입실론 Decay 값 0.9999를 곱해주어 입실론  $\epsilon$ 값을 천천히 낮춰준다. 이 때, 입실론 최솟값을 0.01로 설정하여 입실론  $\epsilon$ 가 그보다 낮아질 수 없게 하였다. 약 45,000 단계 이후의 입실론  $\epsilon$ 값은 최솟값에 수렴하게 되고, 이후에는 약 99% 확률로 큐 함수(Q Function)를 이용하여 최적의 행동을 선택하게 된다. 또한 1%의 확률로 탐험적인 행동을 하며, 총 보상 값이 더 높은 방향으로  $Q$ 를 업데이트 할 수 있도록 하였다. 이러한 연산 방식을 통해서 처음에는 랜덤한 값을 행동으로 사용하는 탐험을 통해 비어 있는 큐 테이블을 업데이트 하고, 천천히 큐 테이블을 채워가면서 충분히 학습이 되면, 학습된 내용에 기반하여 행동을 취하게 되는 구조로 작동하게 된다. 또한 충분히 학습을 하였다고 하여도, 1%의 확률로부터 보다 좋은 결과를 찾을 수 있도록 다른 선택도 해보게 된다.

## 5) 보상 함수 (Matching Matrix 알고리즘)

### (1) Matching Matrix 모델링

Matching Matrix를 구성하기 위해 고려한 4가지 요소에 대한 정의는 다음과 같으며, <Fig. 3>는 수요에 대한 Matching Matrix로, 수요밀도와 배차거리, 운행거리를 같이 표현한 것이다.

1. 수요밀도 (Demand Density) : 승객의 목적지 위치의 수요 밀도 클러스터에 따른 비례 계수  $C$
2. 배차거리 (Dispatch Distance) : 승객과 운전자의 위치에 따른 거리  $B$
3. 운행거리 (Driving Distance) : 승객의 출발 위치에서 목적지까지의 거리  $D$
4. 선호가격 (Prefer Price) : 승객과 운전자의 선호 가격을 가우시안 분포를 통해 결정  $P$



<Fig. 3> Elements of the Matching Matrix for Demand (Demand Density, Dispatch Distance, Driving Distance)

### (2) 수요 밀도

비지도 학습의 대표적인 클러스터링 알고리즘 중에서 밀도 기반 클러스터링 (Density based Clustering) 기법을 사용하여 수요에 대한 좌표에 대한 수요밀도를 계산한다. 특히 밀도 기반 클러스터링의 대표적인 알고리즘인 Ester et al(1996)에서 제안된 DBSCAN 기법이 아니라 지역 밀도에 대한 정보와 계층적 구조를 반영한 Campello et al.(2015)에서 제안된 HDBSCAN 기법을 사용하였다. DBSCAN을 사용하게 되면, 매 라운드마다 수요 밀도를 계산하기 위해서 두 가지의 하이퍼파라미터를 조정 해주어야만 한다. Karypis et al.(1999)에서 DBSCAN의 하이퍼파라미터 두 가지인 클러스터에 이웃을 포함할 수 있는 최대 반지름 거리 (Eps,  $\epsilon$ ) 와 클러스터를 형성하기 위해 필요한 최소 이웃의 수 (MinPts) 에 따라 클러스터링 결과가 상당히 다른 결과를 보여지게 되기 때문이다. 그런데 매 라운드마다 일관적인 하이퍼파라미터를 사용 하면, 어떠한 라운드에서는

최적의 클러스터링이 이루어질 수 있지만, 다른 라운드에서는 제대로 클러스터를 형성 할 수 없게 될 수 있다. 이는 데이터 셋이 조금만 변하게 되더라도 이전에 결정된 최대 반지름 거리와 최소 이웃의 수가 더 이상 최적의 값이 아니게 되기 때문이다. 따라서 본 연구에서는 이런 부분을 일부 개선하여  $\epsilon$ 를 더 이상 하이퍼파라미터로 사용하지 않으며 최소 이웃의 수만 고려하는 HDBSCAN을 사용하고자 한다. HDBSCAN의 경우에 데이터 셋을 밀도/희소 의미를 포함한 값으로 변환하여 포인트간의 상호 도달가능 거리 (Mutual Reachability Distance)값을 기반으로 미니멈 스패닝 트리 (Minimum Spanning Tree)를 만든다. 이후 연결된 컴포넌트들의 클러스터 계층을 구축하고 앞에서 설정한 최소 이웃의 수를 기반으로 클러스터 계층을 축약한다. 축약된 클러스터 계층에서 견고한 클러스터만 선택하는 방식으로 진행되기에 알고리즘에 필요한 하이퍼파라미터는 최소 이웃의 수뿐이다. 또한 매 라운드 마다 수요 밀도를 계산하기 위해 수요 데이터를 클러스터링 할 경우 4~15개의 클러스터로 한정될 수 있도록 최소 이웃의 수를 오토 튜닝하여 사용하였다. 직접 구현한 오토 튜닝 방법으로는 최소 이웃의 수가 클러스터 개수를 결정하는 것에 핵심적으로 영향을 미치기에 클러스터 개수가 15이상으로 커질 경우 최소 이웃의 수를 높여 클러스터의 개수를 줄이고 4개 이하로 줄어들 경우 최소 이웃의 수의 값을 줄여 4개 이상의 클러스터가 생성되도록 하였다.

또한, 데이터에 노이즈가 다수 존재하는 경우 적절한 클러스터를 찾는 데에 있어서 방해가 될 수 있기 때문에 아웃라이어의 영향을 최소화 할 수 있도록 평균과 분산 대신에 중앙값(median)과 사분위값(IQR)을 사용한 로버스트 스케일링(Robust Scaling)을 거친 후 클러스터링을 진행하였다.

이후 수요(승차) 좌표 데이터에 위치 정보(위도와 경도)에 따라 새로운 Y변수를 추가한다. 승객의 목적지 좌표에 대한 수요 밀도가 높을수록 택시는 다음 승객을 다음 목적지에서 보다 쉽게 태울 수 있기 때문에, 승객의 목적지 좌표의 수요 밀도에 따라 운전자의 매칭 확률이 영향을 받는다. 승객의 목적지는 하차 좌표 데이터로 나타낼 수 있고 Samworth, R. J. (2012)에서 제안된 Distance Weighted K-Nearest Neighbor 알고리즘을 이용하여 승객의 목적지 위치 정보(위도와 경도)에 대해 Y변수를 계산한다. 이는 모든 승객에 대해서 승차, 하차 좌표 데이터가 존재하기 때문에 각 승객에 대해 목적지 클러스터 번호를 생성하였다. 이후, 클러스터 번호에 따라 비례 계수를 산정한다.

### (3) Matching Matrix 생성

승객과 운전자 각각의 시각에서 Matching Matrix를 생성한다. 승객과 운전자와의 관계를 고려한 4가지 요소 중에서 승객의 입장에서 식(2)와 같이 배차 거리, 선호 가격을 고려하고, 운전자의 입장에서 식(5)과 같이 수요밀도, 배차거리, 운행거리, 선호가격 4가지 모두 고려한다. 승객의 입장에서 배차거리를 고려하는 이유는 운전자와의 배차거리에 따라 서비스 호출부터 탑승하기까지 시간에 비례하기 때문이다. 이를 활용하여 다음 식(2) ~ 식(8)과 같은 Matching Matrix 알고리즘을 제안하고자 한다.

$$MP = B * (PP - NS) \dots\dots\dots (2)$$

- 여기서,  $MP$  = 운전자에 대한 모든 승객의 매칭 확률 벡터
- $B$  = 승객과 운전자의 위치에 따른 거리 (배차거리)
- $PP$  = 승객의 선호 가격 (가우시안 분포를 통해 형성)
- $NS$  = 주어진 상태에서 행동에 따른 시장 가격

이 때, 가우시안 분포의 식은  $\bar{x} = \int_{-\infty}^{\infty} \frac{x}{\sigma \sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$  와 같다.

$$MP_{Normal} = (MP - MP_{Min}) / (MP_{Max} - MP_{Min}) \dots\dots\dots (3)$$

여기서,  $MP_{Min}$  = 운전자에 대한 모든 승객의 매칭 확률 벡터의 최솟값  
 $MP_{Max}$  = 운전자에 대한 모든 승객의 매칭 확률 벡터의 최댓값  
 $MP_{Normal}$  = 정규화 된 운전자에 대한 모든 승객의 매칭 확률 벡터

$$MP' = \begin{cases} 0 & MP_{Normal} < MP_{Median} \\ MP_{Normal} & MP_{Normal} \geq MP_{Median} \end{cases} \dots\dots\dots (4)$$

여기서,  $MP_{Median}$  = 정규화 된 운전자에 대한 모든 승객의 매칭 확률 벡터의 중위 값 (Scalar)  
 $MP'$  = 승객의 매칭 확률 벡터

$$MD = B \times D \times C \times (NS - PD) \dots\dots\dots (5)$$

여기서,  $MD$  = 승객에 대한 모든 운전자의 매칭 확률 벡터  
 $D$  = 승객의 출발 위치에서 목적지까지의 거리 (운행거리)  
 $C$  = 승객의 목적지 위치의 수요밀도 클러스터에 따른 비례 계수 값 (수요밀도)  
 $PD$  = 운전자의 선호 가격 (가우시안 분포를 통해 형성)

$$MD_{Normal} = (MD - MD_{Min}) / (MD_{Max} - MD_{Min}) \dots\dots\dots (6)$$

여기서,  $MD_{Min}$  = 승객에 대한 모든 운전자의 매칭 확률 벡터의 최솟값  
 $MD_{Max}$  = 승객에 대한 모든 운전자의 매칭 확률 벡터의 최댓값  
 $MD_{Normal}$  = 정규화 된 승객에 대한 모든 운전자의 매칭 확률 벡터

$$MD' = \begin{cases} 0 & MD_{Normal} < MD_{Median} \\ MD_{Normal} & MD_{Normal} \geq MD_{Median} \end{cases} \dots\dots\dots (7)$$

여기서,  $MD_{Median}$  = 정규화 된 승객에 대한 모든 운전자의 매칭 확률 벡터의 중위 값 (Scalar)  
 $MD'$  = 운전자의 매칭 확률 벡터

$$MM = MP' \times MD' \dots\dots\dots (8)$$

여기서,  $MM$  = 승객과 운전자의 Matching Matrix의 확률 벡터의 곱

식(2)는 운전자에 대한 모든 승객들의 매칭 확률 벡터를 구하기 위해 승객과 운전자의 위치에 따른 거리인 배차거리와 승객의 선호 가격 (시물레이션이므로 실제의 가격이 아닌 가우시안 분포를 통해 랜덤하게 형성하였음) 에 주어진 상태에서 행동에 따른 시장 가격을 뺀 값을 곱한다. 이를 통해 시장에서 승객의 선호 가격에 대해 시스템에서 제안하고자 하는 임의의 값을 입력으로 하였을 때의 승객의 매칭 확률을 나타내고자 하였다. 이후 식(8)에서와 같이 운전자의 매칭 확률과 곱하게 되므로, 식(3)에서와 같이 정규화 해준다. 다음으로 식(4)에서와 같이 승객의 정규화 된 매칭 확률의 중위 값을 기준으로 중위 값보다 작은 값은 0으로 치환해준다.

식(5)는 승객의 매칭 확률과 마찬가지로 승객에 대한 모든 운전자들의 매칭 확률 벡터를 구하기 위해 운전자와 승객의 위치에 따른 거리인 배차거리와 승객의 출발 위치에서 목적지까지의 거리인 운행거리, 승객의 목적지 위치의 수요밀도 클러스터에 따른 비례 계수인 수요 밀도와 주어진 상태에서 행동에 따른 시장 가격에 운전자의 선호 가격 (마찬가지로 시물레이션이므로 실제의 가격이 아닌 가우시안 분포를 통해 랜덤

하게 형성하였음) 을 뺀 값을 곱한다.

이를 통해 시장에서 운전자의 선호 가격에 대해 시스템에서 제안하고자 하는 임의의 값을 입력으로 하였을 때의 운전자의 매칭 확률을 나타내고자 하였다. 마찬가지로 식(8)에서와 같이 승객의 매칭 확률과 곱하게 되므로, 식(6)에서와 같이 정규화 해준다. 다음으로 식(7)에서와 같이 운전자의 정규화 된 매칭 확률의 중위 값을 기준으로 중위 값보다 작은 값은 0으로 치환해준다. 이후 마지막으로 승객의 매칭 확률과 운전자의 매칭 확률을 곱하여 최종적인 Matching Matrix를 생성한다. 이렇게 생성된 Matching Matrix가 곱해지는 값의 범위가 각각 0 에서 1 사이이므로, 곱해도 0 에서 1 사이로 나타나게 된다.

(4) Matching Matrix로부터 매칭 수 계산

<Table 1> Matching Matrix

Matching Matrix	Driver 1	Driver 2	...	Driver 1499	Driver 1500
Passenger 1	0.000	0.000	...	0.596	0.493
Passenger 2	0.283	0.000	...	0.491	0.000
...	...	...	...	...	...
Passenger 1499	0.000	0.000	...	0.000	0.000
Passenger 1500	0.891	0.000	...	0.173	0.267

앞서 구한 Matching Matrix는 각 운전자와 각 승객이 매칭 될 확률을 나타내는데, 매칭 하려는 승객과 운전자 중 한명이라도 서로에게 매칭하고자 하는 확률이 0이었다면, 그 둘의 곱은 0이 되므로 서로 매칭 될 수 없다. 반대로 승객과 운전자가 서로에게 매칭되고자 확률이 존재한다면 해당 승객과 운전자의 칸에 매칭 될 확률을 표시한다. 이후 Matching Matrix에서 매칭 될 확률이 가장 높은 승객과 운전자를 매칭 시키고 이후 매칭에서 제외하는 식으로 반복적으로 매칭하여 매칭 수를 계산 할 수 있다. 그러나 이 때, <Table 1>에서 초록색 상자로 표현한 승객과 같이 모든 운전자에 대해 매칭 될 확률이 전부 0으로 나타나면, 어느 운전자와도 매칭이 될 수 없으므로, 연산에서 제외하면 보다 빠르게 연산을 할 수 있다. 이는 반대로 어느 운전자가 모든 승객에 대해 <Table 1>에서 빨간색 상자로 표현한 운전자와 같이 매칭 확률이 0으로 나타나면, 어느 승객과도 매칭 될 수 없으므로 같은 방법으로 연산에서 제외 할 수 있다.

$$N_{Matching} = \begin{cases} ND & NP - ND \geq 0 \\ NP & NP - ND < 0 \end{cases} \dots\dots\dots (9)$$

여기서,  $N_{Matching}$  = Matching Matrix를 통해 결정된 승객과 운전자의 매칭 수  
 $ND$  = Matching Matrix에서 매칭 가능한 운전자의 수  
 $NP$  = Matching Matrix에서 매칭 가능한 승객의 수

그런데 실제로 이 매칭을 운전자와 승객에 대해 1:1 매칭을 시켜 최종적인 매칭 수를 계산하는 것은 승객과 운전자의 수가 많으면 많을수록 수많은 반복이 이루어져 연산에 큰 부하를 일으키게 된다. 이를 극복하기

위해서 플랫폼 (시스템) 의 입장에서 생각해 보면, 이때의 매칭은 실제로 매칭을 시키는 것이 아닌 매칭 확률로부터 수익을 계산하여 시장에서 택시 요금에 대한 적정 가격 비율을 제시하는 것이므로, 연산 과정에 있어서 모든 승객과 운전자를 반드시 매칭 시킬 필요는 없다. 따라서 식(9)에서와 같이 빠른 연산을 위해 운전자의 수와 승객의 수 중 보다 작은 수를 선택하여 모두 매칭되었다고 놓고 진행하면 보다 빠르게 반복 없이 연산 할 수 있게 되어 부하를 줄일 수 있다. 부하를 줄이기 위해 사용한 알고리즘을 방법 A라고 하고 1:1 전수 매칭을 통해 실질적으로 매칭 된 수를 계산하는 알고리즘을 방법 B라고 하였을 때, 두 방법 간의 매칭 수의 차이 및 소요 시간에 대해 실험한 결과는 다음 <Table 2>와 같다.

<Table 2> Number of Matches and Computational time by Matching Algorithm

Based on 24-hour Data	Matching Algorithm A		Matching Algorithm B	
	Number of Matches	Computation time(s)	Number of Matches	Computation time(s)
Mean	434.54	0.0012	420.62	26.8019

24시간의 데이터를 토대로 매 시간 동일한 Matching Matrix를 생성하여 방법 A와 방법 B에 대해 매칭 수와 연산 시간의 평균 값을 나타낸다. 이때, 매칭 수의 차이는 미미하지만 연산 시간은 매우 큰 차이를 보여 준다. 방법 A의 경우 방법 B보다 매칭 된 수가 약간 더 많은 것으로 나타나는데, 이는 방법 B에서는 1:1 매칭으로 승객과 운전자를 각각 매칭시키기 때문에 한 번 매칭이 된 승객 또는 운전자는 이후 매칭에서 완전히 제외되기 때문이다. 하지만 방법 A에서는 매칭확률이 모두 0인 승객 또는 운전자를 매칭에서 제외시킨 후 나머지는 모두 매칭이 될 수 있다는 가정으로 매칭 수를 계산하기 때문에 방법 B보다 매칭 수가 약간 더 높게 나타났다. 결과로 두 방법 모두 매칭 수에는 큰 차이가 없었지만, 연산 시간에 차이가 크게 두드러졌다. 따라서 방법 A를 사용하여 매칭 수를 계산하였다.

(5) 보상 함수

기본적으로 보상 함수는 마르코프 의사 결정과정에서 다룬 것과 같이 조정된 택시의 시장 가격과 운전자와 승객이 매칭 된 수에 비례하여 가격 결정 에이전트에 대한 수익의 일부 금액이 발생하므로 두 가지 변수를 곱한 것으로 식(10)과 같이 설정하여 보다 많은 수의 승객과 운전자의 매칭을 할 수 있도록 하는 적정 가격 비율을 강화학습의 보상 함수를 통해 찾아가게 된다.

$$R(s, a) = NS * N_{matching} \tag{10}$$

여기서,  $R(s, a)$  = 주어진 상태에서 선택한 행동에 대한 보상

6) 큐 러닝

시뮬레이션은 라운드의 집합으로 설정하는데, 라운드는 1시간 단위로 강화학습을 통해 결정된 최적의 가격을 도출하도록 하였다. 또한 매 라운드마다 모든 환경, 상태, Q를 초기화한다. 따라서 한 라운드가 다음 라운드에 미치는 영향이 없도록 하였다. 여기서, 시작 상태는 택시의 시장 가격으로 3,800원으로 하였고, 가격 조정 행동인 행동은 [-80, -20, 0, 20, 80] 다섯 가지로 구성하였다. 또한 승객과 운전자 수는 해당 라운드의 수요 데이터의 수, 운전자 데이터의 수로 설정하였다. 각 승객과 운전자들의 선호가격은 초기 상태값인 3,800을 평균으로 하여 가우시안 분포를 이용하여 결정하였다.

## IV. 적용 및 평가

### 1. 사용 데이터

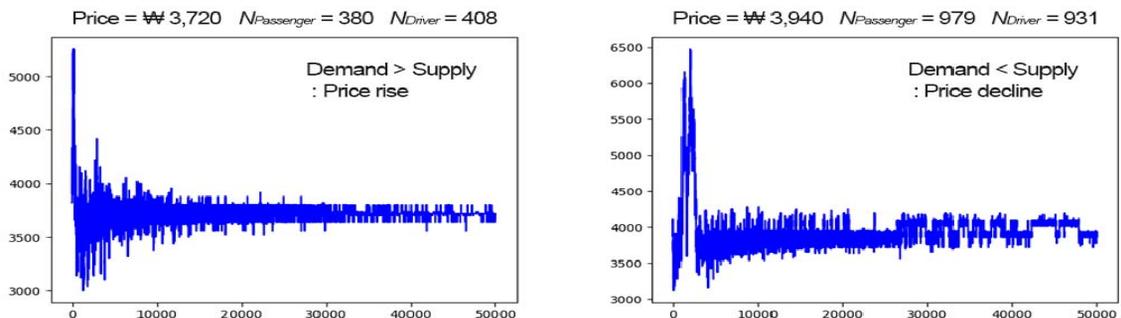
시뮬레이션에서 사용한 데이터는 kaggle(<https://www.kaggle.com/>)의 2014년 뉴욕 택시 데이터이다. 약 1,500만 개의 택시 승차 및 하차에 대한 데이터로, 18개의 속성이 존재하는데, 본 연구에서는 승차 시간과 승차 좌표(경도 및 위도), 하차 시간과 하차 좌표(경도 및 위도)를 사용하였다. 데이터의 시간적 범위는 승차부터 하차까지 2014년 01월 01일 00시 00분 00초 ~ 2014년 03월 01일 00시 55분 00초이다. 또한 공간적 범위는 뉴욕의 맨해튼 지역과 브루클린 지역의 택시 승하차 관련 정보로 구성되어 있다. 일부 하차 좌표 중에 두 지역을 벗어난 경우도 존재한다. 본 연구에서 사용한 데이터의 특성은 다음 <Table 3>과 같다.

<Table 3> Data Attribute

Data	Pickup Datetime	Pickup Longitude	Pickup Latitude	Dropoff Datetime	Dropoff Longitude	Dropoff Latitude
Max	2014-02-28 23:59:00	-73.101985	40.998447	2014-03-01 00:55:00	-73.041458	40.99974
Min	2014-01-01 00:00:00	-74.778437	40.503816	2014-01-01 00:01:00	-74.718568	40.502122
Unit	YYYY-MM-DD hh:mm:ss	Longitude	Latitude	YYYY-MM-DD hh:mm:ss	Longitude	Latitude
Type	Time Series	Float64	Float64	Time Series	Float64	Float64

데이터의 특성에서 Pickup Longitude, Pickup Latitude는 승객의 위치 정보(좌표 데이터)이자 수요 데이터로써 활용하였는데, 이는 현실적으로 택시를 승차하는 위치가 곧 승객이 택시를 호출 또는 기다리는 승객의 위치이기 때문이다. 이에 반해 시뮬레이션을 위한 실제 운전자의 위치는 정확한 공차를 포함한 택시의 위치를 파악 할 수 없으므로, 모수 추정에 어려움이 존재하였다. 따라서 다양한 방법을 통해 가상의 운전자에 대한 데이터를 실제와 비슷하게 구성하여 시뮬레이션에 활용하고자 하였다. 이는 3장의 분석방법론의 분석개요에서 다루었던 3가지 택시 위치 샘플링 방법을 조합하여 시뮬레이션에서 운전자 데이터로써 활용하였다. 이렇게 생성한 승객과 운전자의 데이터는 시뮬레이션에서 1시간 단위의 라운드로 나누어 적용하였다.

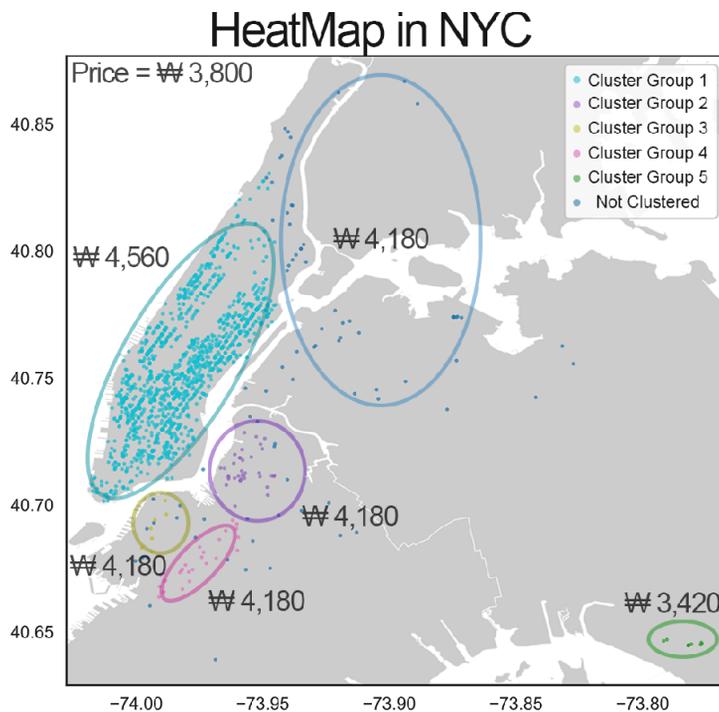
### 2. 강화학습 기반 시뮬레이션 결과



<Fig. 4> Result of Dynamic Pricing considering the relationship between Passengers and Drivers

시간적인 요소를 고려하여 1 라운드 마다 최적의 동적 택시요금을 책정한다. 1 라운드 시뮬레이션에 소요되는 시간은 약 30분이며, 실험에 따른 결과는 다음 <Fig. 4>와 같이 나타난다. <Fig. 4>의 Y축은 택시 가격(상태)을 의미하며, X축은 라운드의 단계를 의미한다. 즉 50,000 단계를 반복하여 큐 러닝을 통해 학습을 진행한 후에 최종적으로 가격이 도출된다. 낮은 단계에서는 입실론 탐욕 정책에 따라 높은 확률로 에이전트는 탐험을 통해 행동을 결정하게 되는데, 낮게는 1,200원부터 높게는 5,500원까지 상태가 자유롭게 변동되며 큐 테이블에 학습되어진 것을 볼 수 있다. 이후 입실론 Decay에 의해서 약 7,000 단계 이후부터 입실론 값은 약 0.50 미만으로 내려가고, 약 15,000 단계 이후부터 입실론 값은 약 0.22로 탐험을 통한 행동 선택보다 Q를 이용(Exploitation)하여 행동을 결정하는 확률이 높아지게 된다. 따라서 이전에 충분히 업데이트된 큐 테이블을 충분히 이용할 수 있게 되고, 점차 가격이 수렴하게 되는 현상이 나타난다. 대개 수요의 수가 공급의 수와 비례하여 수요가 더 많은 경우에는 최종 택시 가격이 기존 시장가격인 3,800원보다 상승하였으며, 반대로 공급이 수요보다 많은 경우에는 최종 택시 가격이 기존 시장가격보다 하락하는 경향을 볼 수 있다. 그러나 본 연구에서 제안하는 모델은 수요와 공급의 수만 고려하는 것이 아닌 좌표 에 기반한 모든 승객과 운전자의 관계 및 다양한 상황을 고려하였기 때문에, 수요가 더 많은 경우에도 가격이 내려간 경우가 있고, 공급이 더 많은 경우에도 가격이 올라가는 경우 또한 존재하였다.

본 연구에서 수요 데이터에 대한 수요 밀도를 계산하기 위해 밀도 기반 클러스터링(HDBSCAN)으로 클러스터 번호 별로 비례 계수를 계산하였다. 시뮬레이션을 통해 라운드마다 책정된 가격은 각 클러스터 번호의 비례 계수에 곱하여 해당 지역에서의 시장가격으로 결정되도록 하였다. 이러한 시뮬레이션을 통해 도출된 각 라운드의 최종 가격을 지도 위에 나타내면 <Fig. 5>와 같이 표현할 수 있다.



<Fig. 5> Price determination according to Cluster

## V. 결론 및 제언

본 연구에서는 공간적인 요인, 라이드셰어링 승객과 운전자간의 관계를 포함한 다양한 상황을 고려하여 최적의 가격을 도출해내는 강화학습 기반 시뮬레이션을 진행하였다. 실험에서는 뉴욕 지역의 승차 및 하차 데이터로부터 수요 데이터를 추출하고 운전자 데이터를 생성하여 시뮬레이션을 진행하였지만, 시간 정보와 함께 위치 좌표 정보가 있으면 어느 지역이든 같은 방법으로 시뮬레이션이 가능하며, 실제 운전자들의 위치 정보를 알 수 있다면 별도의 운전자의 위치를 생성하기 위한 알고리즘 없이 진행이 가능하다.

또한 본 연구에서는 다양한 선행 연구들에서 제안된 강화학습 기반 시뮬레이션에 기반한 환경을 토대로 하여 구성하였지만, Wu et al.(2016)에서 승객과 운전자의 수를 500씩으로 고정하여 단순히 선호가격과 시장 가격만을 비교하여 진행한 실험보다 구체적이고 현실적인 접근을 하고자 실제 데이터를 사용하였으며, 승객의 목적지 위치의 클러스터에 따른 비례 계수인 수요밀도와 승객과 운전자의 위치에 따른 거리인 배차거리, 승객의 출발 위치에서 목적지까지의 거리인 운행거리, 승객과 운전자의 선호 가격을 고려한 Matching Matrix를 통해 새로운 보상 함수로 보다 현실적인 시뮬레이션이 가능하도록 진행하였다.

그러나 보다 현실적인 시뮬레이션을 구성하고자 하였으나, 강화학습에 기반한 유동적인 가격 책정을 현실에서 진행하기에는 초기 학습 과정에서 다양한 조정을 위해 가격을 변경함에 따른 막대한 피해가 발생 할 수 있으므로, 임의의 확률 분포인 가우시안 분포로 선호 가격을 나타내었는데, 이는 현실의 선호 가격과는 차이가 있을 수 있으므로 연구의 한계점으로 존재한다.

또한 본 연구에서는 각 라운드마다 1시간 단위로 데이터를 나누어 사용하였는데, 앞선 시간의 수요와 공급의 수에 따른 변화가 이후 라운드에 영향을 주지 않는 독립적인 관계이므로, 각 라운드는 시간 순서에 따라 영향을 받지 않는다. 그러나 현실에서는 앞선 시간의 수요와 공급에 따라 현재 시간의 수요와 공급이 달라지므로 이 또한 연구의 한계점으로 남았다.

그러나 이후 연구에서는 앞서 제시한 연구의 한계점 중 하나인 각 라운드가 다음 라운드에 영향을 미칠 수 없었던 부분을 해결하고자 한다. 특히 본 연구에서 진행하였던 큐 러닝 방식은 상태 또는 행동의 차원이 증가할수록 큐 테이블의 크기가 기하급수적으로 증가하여 연산에 과도한 부하를 일으키게 되고, 학습이 불가능한 상황에 이르게 된다. 이에 따라 다양한 조정을 할 수 없었지만, 이 문제를 해결하기 위해 Mnih et al.(2013)와 Mnih et al.(2015)에서 제안된 신경망(Neural Network)에 기반한 DQN (Deep Q-Networks)을 적용하여 각 라운드의 수요와 공급에 대해 다음 라운드에 연속적으로 영향을 미칠 수 있도록 하는 모델에 대한 연구를 진행하고자 한다. 또한, 수익의 극대화에 초점을 맞추어 유동적인 가격 책정을 하였기에 심야시간대에 공유 모빌리티 서비스가 취약한 곳이 발생할 우려가 있다. 이에 Song et al.(2020)에서 제안한 강화학습 기반 유동적인 가격 책정 솔루션을 접목하고자 한다. 국내 데이터를 활용할 때에 시간대에 따른 지역 별 택시 서비스의 이용 분포를 분석하고 출퇴근시간 및 심야시간 등 취약지역에서의 이용자에게도 공정성, 신뢰성 및 형평성을 제공할 수 있도록 연구를 진행하고자 한다.

최근 데이터 3법의 개정에 따라 데이터 관련 규제가 완화되면서 택시 승하차 데이터가 공공데이터 포털을 통해 제공되기 시작하고 있다. 그러나 현재 공개된 데이터는 작은 시간적 범위와 공간적 범위로 인해 분석에 사용하기에는 다소 한계가 있다. 따라서 충분히 분석이 가능 할 정도로 큰 시간적 범위와 공간적 범위에 해당하는 데이터가 개방되면, 해당 데이터를 사용하여 국내에서도 적용 가능한지에 대한 검증을 위해 추가적인 향후 연구를 진행하고자 한다.

## ACKNOWLEDGEMENTS

본 연구는 국토교통부 교통물류연구사업의 연구비지원(20TLRP-B148970-03), 산업통상자원부 재원으로 한국 산업기술진흥원의 지원(P0004602, 친환경 자동차부품 클러스터 조성사업), 과학기술정보통신부의 재원으로 한국연구재단의 지원(2020R1F1A104826411)과 2020학년도 홍익대학교 학술연구진흥비 지원 하에 수행되었습니다.

## REFERENCES

- Bertsimas D. and Perakis G.(2006), Dynamic pricing: A learning approach. In *Mathematical and computational models for congestion charging*, Springer, Boston, MA, pp.45-79.
- Campello R. J., Moulavi D., Zimek A. and Sander J.(2015), "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data(TKDD)*, vol. 10, no. 1, pp.1-51.
- Castillo J. C., Knoepfle D. and Weyl G.(2017), "Surge pricing solves the wild goose chase," In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp.241-242.
- Ester M., Kriegel H. P., Sander J. and Xu X.(1996), "A density-based algorithm for discovering clusters in large spatial databases with noise," *KDD-96 Proceedings*, vol. 96, no. 34, pp.226-231.
- Guo S., Liu Y., Xu K. and Chiu D. M.(2017), "Understanding ride-on-demand service: Demand and dynamic pricing," In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, pp.509-514.
- Haws K. L. and Bearden W. O.(2006), "Dynamic pricing and consumer fairness perceptions," *Journal of Consumer Research*, vol. 33, no. 3, pp.304-311.
- Karypis G., Han E. H. and Kumar V.(1999), "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp.68-75.
- Lu A., Frazier P. and Kislev O.(2018), Surge Pricing Moves Uber's Driver Partners, Available at SSRN 3180246.
- Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D. and Riedmiller M.(2013), *Playing atari with deep reinforcement learning*, arXiv preprint arXiv:1312.5602.
- Mnih V., Kavukcuoglu K., Silver D., Rusu A. A., Veness J., Bellemare M. G. and Petersen S.(2015), "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp.529-533.
- Samworth R. J.(2012), "Optimal weighted nearest neighbour classifiers," *The Annals of Statistics*, vol. 40, no. 5, pp.2733-2763.
- Song J., Cho Y. J., Kang M. H. and Hwang K. Y.(2020), "An Application of Reinforced Learning-Based Dynamic Pricing for Improvement of Ridesharing Platform Service in Seoul," *Electronics*, vol. 9, no. 11, p.1818.
- Sutton R. S. and Barto A. G.(2018), *Reinforcement learning: An introduction*, MIT Press.
- Van Otterlo M. and Wiering M.(2012), "Reinforcement learning and markov decision processes," In *Reinforcement Learning*, Springer, Berlin, Heidelberg, pp.3-42.
- Wu T., Joseph A. D. and Russell S. J.(2016), *Automated pricing agents in the on-demand economy*, University of California.