강화학습 기반 Paging 의 이동성 예측

천성진*, 김복근** 추현승***[1]

*성균관대학교 인공지능학과
**성균관대학교 DMC 공학과,
**삼성전자 5G Call S/W R&D
***성균관대학교 소프트웨어대학

{chunsoungjin, choo}@g.skku.edu, bokkeun.kim@samsung.com

Mobility Prediction for Paging with RL

Sungjin Chun*, Bokken Kim**, Hyunseung Choo***[1]
*Dept. of Artificial Intelligence, Sungkyunkwan University
** Dept. of Digital Media Communication Engineering, Sungkyunkwan University,
**5G Call S/W R&D, Samsung Electronics
***College of Software Sungkyunkwan University

요 약

4G 에서 5G 로 기술이 발전하며 무선 통신에 필요한 자원이 급격히 증가하고 있다. 증가된 자원을 효율적으로 관리하는 것은 필수적이며 이를 위해 paging cost 감소 연구들이 진행되고 있다. 순환신경망을 응용한 paging cost 감소 연구에서는 연속 예측으로 인해 예측 정확도 감소 문제가 발생한다. 본 논문에서는 강화학습 기반 이동성 예측 기법을 제안하고 기존 순환신경망 응용 기법에서 발생하는 정확도 감소 문제를 극복한다.

1. 서론

일반적인 무선 통신망에서는 단말이 기지국과 항상 연결되어 있지는 않다. 기지국과 연결이 끊어진 상태에서도 단말은 이동을 할 수 있으며 Core Network 에서는 이 단말과 다시 연결을 하기 위해 paging 탐색을 하여 단말의 위치를 찾는다. 단말의 위치를 예측하고 탐색할 범위를 줄인다면 paging 을 위한 computing 자원을 절약할 수 있을 것이다. 기존에도 paging cost 를 절약하기 위한 연구들이 있다. 본 논문에서는 기존 연구 방법들과 실제 환경과의 차이로 인해 발생할 수 있는 문제와 한계를 설명한다. 또한, 이러한 한계를 극복할 수 있는 방법을 제안하고 실제 handover log data 를 사용하여 가능성을 검증한다.

2. Related Work

무선 통신망에서 paging 절차는 4G 이전부터 존재했다. 무선 망에서 주파수와 같은 무선 자원은 비싸고 한정적인 자원이다. 무선 자원을 효율적으로 관리하기 위해 core network 에서는 어떤 단말이 데이터를 전송하지 않는 상황이라면 단말과 기지국 간에 할당된 자원을 해지한다. 자원을 해지하는 것은 연결을 끊는 것을 의미하고 이를 통신 규격에서는 idle mode (혹은 idle state)라고 한다. 이 idle mode 에서는 단말이 어떤 기지국에도 연결 되어있지 않고 core network 에서는 단말의 위치를 알 수 없다. Core network 에서 idle mode 의 단말에 어떤 데이터를 전송해야 할 때 단말의 위치를 알 필요가 있다. 단말의 위치를 찾는 과정을 paging 이라고 한다. Paging 은 4G 망에서는 Mobility Management Entity (MME)를 통해, 5G 에서는 Access and Mobility Management Function (AMF)를 통해 관리하고 처리하는데, 이 장치들의 computing 자원 중 30 – 35%가 Mobility Management (MM)를 위해 사용된다[1].

MM 에서 사용하는 대부분의 computing 자원은 paging 을 위해 사용된다. computing 자원 소모를 줄이기 위한 연구들이 계속되고 있으며 최근의 연구들로는 machine learning 기법을 사용하여 mobility prediction 을 하고 이를 통해 탐색해야 할 기지국 후보군을 줄여 paging cost 를 절약하는 방법들이 있다[1][2]. 이러한 mobility prediction 의 환경에서는 단말이 idle mode 상태로 경과된 시간에 따라 단말의 이동 반경이 달라질 수 있다. 이전의 RNN 방식을 응용한 연구에서는

[1] 교신저자

이를 극복하기 위해 예측을 연속적으로 하여 이동 반경의 변화를 반영하였다[2]. 그러나 연속 예측을 할 경우, 오차 누적에 의해 예측 정확도가 급격히 감소되는 한계가 있다.

3. Asynchronous Advantage Actor-Critic

본 논문에서는 기존 연구들의 한계를 강화학습의 방법을 접목하여 극복하고자 한다. 적용하는 강화학습은 Asynchronous Advantage Actor-Critic(A3C)이다. A3C 는 여러 Agent 들이 탐색을 하며 비동기적으로 policy update 를 하는 강화학습법이다[3]. A3C 는 기존 강화학습에서 loss 에 advantage 개념을 추가하며 에피소드의 결과를 판단할 때 필요한 추가적인 정보를 agent 에 제공할 수 있도록 하였다. Reward 를 R, 예측한 가치를 V 라 한다면 Advantage 는 수식 (1)과 같이 표현된다.

$$\text{Advantage} = R - V \quad (1)$$

Advantage 는 가치를 과소추정 할 경우 양의 값을, 과대추정 할 경우 음의 값을 갖는다. 이렇게 정의된 Advantage 를 통해 두 가지 loss 를 정의한다:

$$\text{Value loss} = \text{Advantage}^2 \quad (2)$$

Value loss 는 Advantage 의 제곱으로 정의한다. 이는 실제 받는 보상을 과대, 과소평가하지 않게 조절하는 역할을 한다.

Policy loss 는 Softmax Cross Entropy 를 이용하여 정의한다. Softmax 는 어떤 입력 x 에 대해 수식 (3)과 같이 정의한다.

$$\text{Softmax}(x) = \exp(x) / \sum(\exp(x)) \quad (3)$$

어떤 입력 x 에 대한 분포를 q(x), 목표하고자 하는 출력의 분포를 p(x)라 하면 Cross Entropy 는 수식 (4)와 같이 표현된다.

$$\text{Cross Entropy}(x) = -\sum(p(x)*\log(q(x))) \quad (4)$$

Softmax Cross Entropy 는 Cross Entropy 의 q(x)를 Softmax(x)로 대체하여 사용한다. 즉, 어떤 입력의 분포 q(x)를 어떤 입력이 갖는 상대적 분포로 변환하여 입력으로 해석한다. Softmax Cross Entropy 를 S_CE 로 간략히 줄이면, 수식 (5)와 같이 표현된다.

$$\text{S\_CE}(p(x), x) = -\sum(p(x)*\log(\text{Softmax}(x))) \quad (5)$$

정의된 S_CE 를 따라 policy loss 를 정의한다. Policy loss 는 agent 의 action 을 label 로, Policy 의 output 을 logits 으로 사용한다. Cross Entropy 의 p(x)는 action 에, q(x)는 policy 에 대응한다. 수식 (1)과 (5)에 따라 policy loss 를 수식 (6)과 같이 정의한다.

$$\text{Policy loss} = \text{S\_CE}(\text{action}, \text{policy}) * \text{advantage} \quad (6)$$

Policy loss 는 정책이 출력되는 확률이 확신을 갖고 학습하되, 학습에서 잘못된 경험이 발생한 것은 advantage 에 따라 역방향 학습할 수 있도록 조절한다. A3C 의 final loss 는 정의된 value loss (2)와 policy loss (6)에 따라 수식(7)로 표현한다.

$$\text{Final loss} = \text{policy loss} + 0.5 * \text{value loss} \quad (7)$$

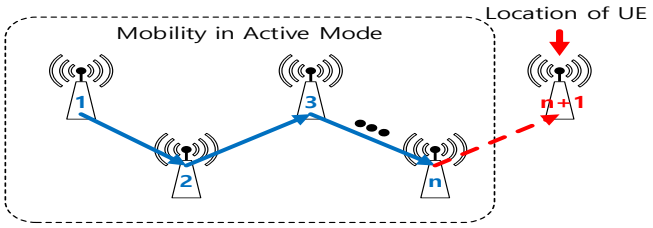
Algorithm S2 Asynchronous advantage actor-critic - pseudocode for each actor-learner thread.

// Assume global shared parameter vectors θ and θ_v and global shared counter $T = 0$
// Assume thread-specific parameter vectors θ' and θ'_v
Initialize thread step counter $t \leftarrow 1$
repeat
  Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
  Synchronize thread-specific parameters $\theta' = \theta$ and $\theta'_v = \theta_v$
  $t_{start} = t$
  Get state s_t
  repeat
    Perform a_t according to policy $\pi(a_t|s_t;\theta')$
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t+1$
    $T \leftarrow T+1$
  until terminal s_t **or** $t - t_{start} == t_{max}$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \text{// Bootstrap from last state} \end{cases}$
  for $i \in \{t-1, \ldots, t_{start}\}$ **do**
    $R \leftarrow r_i + \gamma R$
    Accumulate gradients wrt θ': $d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi(a_i|s_i;\theta')(R - V(s_i;\theta'_v))$
    Accumulate gradients wrt θ'_v: $d\theta_v \leftarrow d\theta_v + \partial(R - V(s_i;\theta'_v))^2/\partial\theta'_v$
  end for
  Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
until $T > T_{max}$

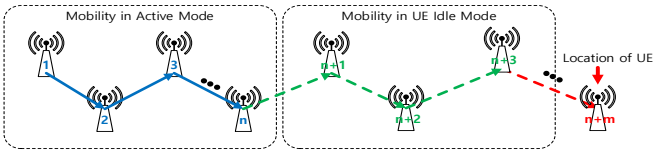
(그림 1) A3C Pseudo Code[3]

4. Proposed Method

기존의 mobility prediction 은 단말이 idle mode 가 된 후 다음 기지국 위치가 어디인지를 예측한다. 순서가 있는 데이터이므로 Long Short-Term Memory (LSTM)과 같은 시계열 패턴 분석 모델을 사용한 연구들도 있다. [3] 이러한 연구들이 가정하는 실험 환경은 그림 1 과 같다. 모델에 입력으로 사용되는 기지국들의 정보는 그림 1 의 Mobility in Active Mode 영역의 정보들이다. LSTM 모델을 사용하여 이러한 이동 패턴을 단말 별로 학습하고 다음 단말의 위치를 예측하는 형태로 모델을 평가한다. 그림 2 는 실제 무선 네트워크 환경에서 얻어지는 기지국 정보를 idle mode 의 단말이 이동한 경로에 맞춰 구성한 그림이다.



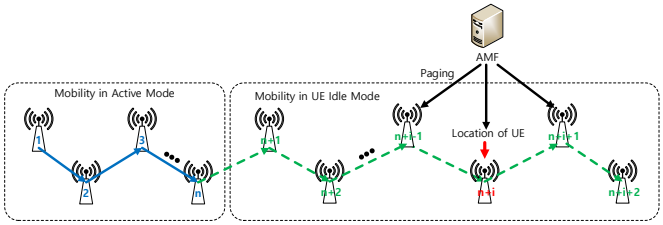
(그림 1) Existing Prediction Method



(그림 2) Prediction Method Considering Mobility in Idle Mode

이전 연구들이 가정하는 단말의 위치와는 다르게 실제 단말은 idle mode 에서도 움직이고 있을 가능성이 높다. 그림 2 에서는 Mobility in UE Idle Mode 영역이 존재한다. 이 영역은 idle mode 의 단말이 이동한 경로를 나타낸다. Idle mode 에서도 이동하는 단말은 n 에서 idle mode 진입 후, n+1 부터 n+m-1 까지 이동 경로를 알 수 없다. 기존 연구 방법은 이러한 실제 환경을 고려한다면 연구들의 예측은 n+1 기지국을 예측하는 것이 한계이며 n+1 예측은 실제 단말의 위치와는 거리가 멀다. 또한, 실제 상황에서는 n+m-1 까지의

이동 경로를 알 수 없으며 이 m 은 크기가 가변이다. 이전 연구 모델의 예측 결과를 입력으로 다시 사용하여 연속적인 예측을 할 경우, 예측 오차가 누적되며 예측이 반복될수록 정확도가 떨어질 수밖에 없다. 즉, 기존 연구는 실제에 적용할 수 없다는 한계를 분명히 보인다.

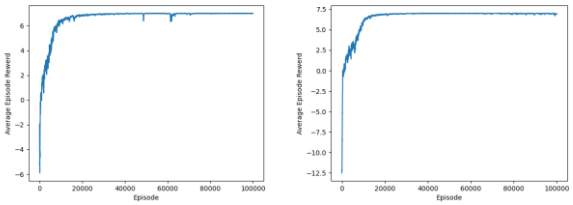


(그림 3) Proposed Prediction Method

이러한 한계를 극복하기 위해 본 논문에서 제안하는 환경은 그림 3 과 같다. 예측하는 target 기지국이 그림 3 과 같이 달라진다. AMF 가 다음 기지국의 위치를 예측하는 것은 기존 연구와 동일하다. 그러나 제안하는 환경에서 강화학습 agent 는 AMF 에서 복수의 기지국을 예측 결과로 도출하도록 구성한다. Agent 가 예측해야 할 기지국의 수 또한 하나의 episode 에서 복수로 존재하며 예측한 기지국의 정밀도에 따라 환경에서는 차등적인 보상을 계산하여 agent 에 할당한다. 예를 들어, 그림 3 과 같이 3 개의 기지국을 예측 목표로 구성한다고 가정한다. 환경은 agent 의 예측 결과가 3 개의 목표 기지국에 속한다면 agent 에 보상을 주고 다음 step 을 생성하여 관측 값으로 전달한다. Agent 의 예측 결과가 목표 기지국 외의 값이라면 현재 episode 는 종료된다. 이러한 환경에서 agent 는 목표하는 3 개의 기지국을 예측하고 높은 누적 보상을 달성하는 방법을 학습할 것이며, 최종적으로 학습된 agent 는 어떠한 관측이 입력되었을 때 복수의 기지국이 갖는 가치를 추정하고 예측할 수 있다. 또한 이러한 예측을 반복하면 단말이 이동한 기지국의 대략적인 경로를 생성할 수 있다.

5. Experiment & Result

제안하는 방법을 검증하기 위해 판교 center 로부터 수집한 handover log 를 사용한다. Handover log data 에는 단말이 특정 AP 에 연결될 때마다의 AP id 가 기록되어 있다. Agent 의 action 수는 AP 수에 따르나, 본 논문에서는 알고리즘 검증을 위해 다양한 action 수를 검증한다. 환경은 판교 handover log 에서 길이가 32 인 에피소드를 생성한다. A3C agent 는 time step 25 인 LSTM 모델로 구성하였다. 환경은 관측 값으로 AP 경로를 묶어 생성하며, target AP 는 관측의 가장 마지막 AP 로부터 다음 몇 개 step 의 AP 들로 설정한다. 예를 들어, Agent 가 handover 관측을 받아 가장 마지막 AP 에서 다음 3 개 step 내의 AP 중 가장 확률이 높은 AP 를 예측하도록 학습시킬 수 있다. Agent 의 예측이 target AP 3 개 중 하나로 적중된다면, 환경은 reward 1 과 target AP 가 포함된 새로운 관측 값을 agent 에게 전달한다면 복수의 예측 확률을 학습하는 agent 가 된다. 본 논문에서는 큰 환경을 탐색할 때도 동작하는지 검증하기 위해 정확한 예측을 할 때만 reward 1 을 전달한다. 즉, target AP 를 1 개만 설정한다. 예측에 실패할 경우 해당 state 는 반복되고 보상은 -0.5 가 전달된다. State 반복은 에피소드 길이만큼 반복된다.



(그림 4) Average Training Reward
Action 8 (R) Action 12 (L)

그림 4 의 그래프들은 학습을 따라 변화되는 reward 그래프다. Action 의 개수를 8 부터 12 으로 확장하며 탐색해야 할 action space 를 넓혔고, 이에 따른 수렴 지점 차이를 비교하였다. 에피소드의 길이는 32, 한 번의 action 으로 얻는 보상은 최대 1 이다. 예측을 연속적으로 성공한다면 time step 25 일 때 최대 보상은 7 이다. 예측 실패 시 reward 는 -0.5 이며 에피소드 길이 32 를 따라 최대 32 회 state 가 반복된다. 따라서 하나의 에피소드에서 얻을 수 있는 최소 보상은 -16 이다. Action 수가 8 일 때에는 최소 보상이 -6, 12 일 때에는 최소 보상 -12.5 를 달성한다. 이는 탐색해야하는 action space 가 넓어짐에 따라 초기 에피소드들에서 보상을 받지 못하는 경우가 많이 발생한 것을 의미한다.

6. Conclusion & Future Work

판교 center 가 아닌 실제 5G networks 에서는 기지국의 수가 수만 개에서 수십만 개 이상이다. 따라서, 기지국 수 혹은 AP 수가 더 많은 환경을 위해서는 agent 의 action 수가 늘어나야 한다. Action 의 수가 늘어나는 만큼 agent 가 탐색해야 하는 에피소드 종류 또한 많아지며 이는 수렴을 위한 에피소드 학습의 양이 많아짐을 의미한다. Random Network Distillation 과 같이 근본적으로 탐색의 효율을 늘릴 수 있는 기법 등을 적용하여 확장할 필요가 있다[5]. 향후 연구는 확장될 action space 를 대비하여 탐색 효율에 관한 기법을 적용하여 확장하고자 한다.

참고문헌

[1] Sivasankar, Sivanandham, and Rajesh Challa. "Closed Loop Paging Optimization for Efficient Mobility Management." 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2021.

[2] Jeong, Jaeseong, et al. "Mobility Prediction for 5G Core Networks." IEEE Communications Standards Magazine 5.1 (2021): 56-61.

[3] Mnih Volodymyr, et al. "Asynchronous methods for deep reinforcement learning," International conference on

machine learning, PMLR, 2016.
[4] Fattore, Umberto, et al. "AutoMEC: LSTM-based User Mobility Prediction for Service Management in Distributed MEC Resources." Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. 2020.
[5] Burda, Yuri, et al. "Exploration by random network distillation." arXiv preprint arXiv:1810.12894 (2018).

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 Grand ICT 연구센터지원사업(IITP-2021-2015-0-00742), 인공지능대학원지원사업(No.2019-0-00421), ICT 명품인재양성사업(IITP-2021-2020-0-01821)의 연구결과로 수행되었음