

군중과 로봇의 상호작용

유승태, 정재형

Crowd-Robot Interaction

Seung Tai Yoo, Jaehyoung Jeong

요 약

Crowd-Robot Interaction 관련 논문인 CrowdNav[1]를 활용하여 다양한 환경에서 실험을 진행해 본다. 그리고 각 환경에서의 모델 성능을 분석한다. 실험 결과물을 분석하여 문제점을 찾고 발견된 문제점에 대해 개선을 시도한다

Abstract

In this report, Experiment in various environments using CrowdNav[1], a paper related to Crowd-Robot Interaction. And analyze the model performance in each environment. Analyze the experimental results to find problems and try to improve the problems found.

Key words

deep learning, crowd-robot interaction, reinforcement learning

I. 서 론

Crowd-robot interaction은 에서 다루는 것은 군중 사이를 통과해서, 안전하고 사회 규범을 지키는 경로를 찾아 목적지를 찾아가는 문제로, 공장, 식당, 병원 등 다양한 장소에서 로봇이 사용됨에 따라 그 중요도가 날로 더해지고 있다. Crowd-robot interaction은 단순히 장애물 피하기가 아니라 사람과 사람, 사람과 로봇 사이의 상호작용을 이해하고 예측할 수 있어야 하므로 그 난이도가 더 어렵게 된다. 본 프로젝트에서는 기존에 작성된 논문 가운데 CrowdNav[1]의 특징을 분석하고, 발견한 문제점에 대해 개선을 시도하는 것을 목표로 한다.

II. 기존 연구

CrowdNav[1]이전에 작성되었던 crowd-robot interaction 관련 논문을 우선 살펴보고자 한다. 가장 시초가 되는 모델로는 SocialForce[4]가 있다. 이는 시뮬레이션과 실제 환경 모두에서 성공적으로 적용된 적 있다. Interacting Gaussian Process는 각 에이전트의 궤적을 개별 Gaussian Process로 모델링한다. RVO[5], ORCA[6]는 모든 에이전트에 동일한 정책이 적용되는 다중 에이전트 설정에서 공동 장애물 회피 속도를 추구한다. CADRL[7]은 최악의 경우에 대해 최선의 조건을 선택하는 연산을 통해 2-agent에서 multi-agent case로 확장하였다. LSTM-RL[8]은

LSTM을 이용해서 LSTM 모델을 사용하여 로봇까지의 거리의 역순으로 각 이웃의 상태를 순차적으로 처리한다.

이러한 기존 모델들은 모든 상호작용을 나타내기 어렵고, 로봇에게 영향을 줄 수 있는 crowd간의 상호작용을 무시한다는 공통적인 특징이 있다. 따라서 CrowdNav는 crowd간의 상호작용을 고려하고자 한다.

III. 사용 딥러닝 모델

III.1 CrowdNav[1]

CrowdNav[1]은 crowd와 로봇의 상호작용을 고려하는 attention 메커니즘을 사용한다. 이를 통해 보다 효율적으로 crowd와의 상호작용에 반응할 수 있다.

그리고 crowd와 다른 crowd 사이의 상호작용이 미치는 영향에 대해서도 고려하는데, crowd끼리 주고받는 영향이 간접적으로 로봇에게 영향을 주기도 한다.

CrowdNav[1]는 다음과 같은 세 가지 모듈로 구성된다. Figure 1[1]을 통해 전체 모델 구조를 확인할 수 있다.

- Interaction module: crowd-로봇 상호 작용을 명시적으로 모델링하고 거친 로컬 맵을 통해 crowd간의 상호 작용을 인코딩한다.

- Pooling module: crowd 간의 상호작용을 self-attention 메커니즘에 의해 고정 길이 임베딩 벡터로 합산한다.

- Planning module: social navigation을 위해 로봇과 crowd의 합상태의 값을 추정한다.

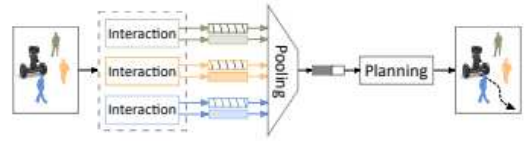


Fig. 1: Interaction, Pooling, and Planning 세 가지 모듈로 구성된 CrowdNav[1] socially attentive navigation 방법의 개요이다. 로봇과 각 crowd 간의 상호 작용은 Interaction 모듈에서 추출된 후 Pooling 모듈에서 집계된다. Planning 모듈은 Crowd 탐색을 위한 로봇과 crowd의 합동 상태 값을 추정한다.

III.2 Relational Graph Learning for Crowd Navigation[2]

CrowdNav[1]는 crowd와 로봇 사이의 attention 만 반영하였지만 Relational Graph Learning for Crowd Navigation[2]은 Graphral neural network를 이용하여 crowd 간의 attention 또한 반영하였다

Human node feature들을 이용해 미래 시점의 crowd 위치를 예상하고 Robot node feature를 이용해 상태 가치를 계산한다.

CrowdNav[1]에 비해 성능 향상이 이루어졌으나 그 개선의 폭이 크지 않고 오히려 CrowdNav[1]의 반복에 가까운 점이 한계점으로 작용한다.

Figure 2[2]를 통해 Relational Graph learning에 대한 내용을 확인할 수 있다.

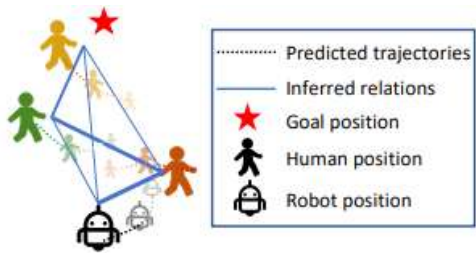


Fig. 2: 논문[2]의 relational graph learning 방식을 표현한 그림이다. (파란색 관계 그래프를 통해 확인 가능하다.) 로봇과 crowd 사이의 상호작용을 추론하고 궤적을 예측한다. (선 두께는 상호작용의 강도를 나타낸다. 큰/중간/작은 그림은 현재 위치, 다음 단계의 예측 위치, 두 단계 뒤의 예측 위치를 나타낸다.) 로봇과 빨간색 crowd, 초록색 crowd와 빨간색 crowd 사이의 강한 관계를 추론하고 예측된 crowd 궤적을 조건으로 계획함으로써 목표를 향한 안전한 경로를 찾을 수 있다.

III.3 Social-NCE[3]

Social-NCE[3]는 self-supervision을 통해 충돌과 같은 부정적인 예를 명시적으로 모델링한다.

모델링 방법은 다음과 같다.

- 합성된 negative 사건으로부터 실제 positive 사건을 식별함으로써 추출된 모션 표현을 정규화하는 social contrastive loss를 도입하였다.
- 드물지만 위험한 상황에 대한 사전 지식을 기반으로 유익한 negative 샘플을 구성하였다.

최근 궤적의 충돌 비율을 크게 줄임으로서 예측, 행동 복제 및 강화 학습 알고리즘 등 여러 벤치마크에서 state-of-art method를 능가하였다.

Social-NCE[3]는 Socially-aware motion 표현을 학습하기 위한 대조적 방법을 제시하였고 Social-NCE loss[3]와 Social-NCE[3]의 sampling 전략과 결합되어 recent human trajectory forecasting 및 crowd navigation 알고리즘의 성능을 크게 향상시켰다.

negative data augmentations을 통한 반대로부터의 학습을 통해 기존 interactive data 수집에 대한 유망한 대안이 될 수 있다.

하지만 Social-NCE[3]는 CrowdNav[1]와 마찬가지로

로 train 상황과 test 상황이 다를 경우 성능하락이 발생한다는 문제점이 있으며 CrowdNav[1]에서 사용 가능한 시나리오가 Social-NCE[3]에서는 사용 불가능한 경우가 있는 문제점이 있다.

Figure 3[3]를 통해 social contrastive learning에 대한 내용을 확인할 수 있다.

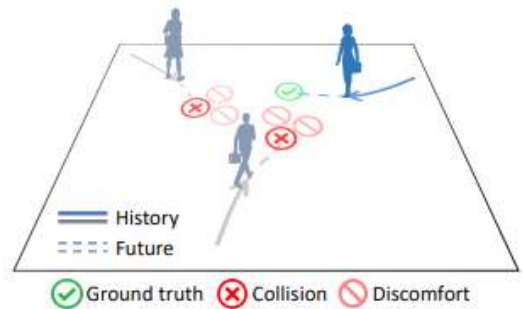


Fig. 3: social contrastive learning을 표현한 그림이다. Social-NCE[3]의 방법은 multi-agent context에서의 negative data의 augmentation과 결합된 contrastive representation learning을 통해 신경 운동 모델의 견고성을 촉진한다.

IV. 실험 결과

실험 환경은 CrowdNav[1]에 준하게 설정되었다.

- crowd는 ORCA[4]에 의해 움직인다.
- robot은 OM-SARL[1]에 의해 움직인다.
- robot은 invisible하다고 가정한다.
- 실험환경은 Google Colab Pro
- Python 버전은 3.6.8
- GPU를 사용해서 학습한다.(논문에서는 CPU를 사용한다)
- 모방학습
 - ORCA[4] 에피소드 3000개를 이용
 - learning rate 0.01
 - 50 epoch

● 강화학습

- 10,000 에피소드
- 0 ~ 4000 에피소드: 엡실론 0.5 에서부터 순차적으로 감소
- 4000 에피소드:엡실론 0.1

Method	Success	Collision	Time	Reward
논문	1.00	0.00	10.46	0.342
5명 case	1.00	0.00	10.60	0.3367
10명 case	0.91	0.09	12.56	0.2237
20명 case	0.46	0.54	14.32	-0.0469

표 1: Crowd-Nav[1]을 활용하여 사람이 5명일 때, 10명일 때, 20명일 때 세 가지 경우로 실험을 진행한 결과이다.

사람이 5명일 때, 10명일 때, 20명일 때 세 가지 경우로 실험해보았다.

5명일 경우 논문의 설정과 동일했고, 그 성능도 논문과 거의 동일함을 확인할 수 있었다.

10명으로 실험한 결과 5명일 때보다 성공확률, 충돌률, 소모시간, 보상 모든 부분에서 나빠졌다. 또한 사람과 로봇이 너무 가까이 위치하는 경우 또한 증가하였다.

20명으로 실험한 결과 아예 성공확률보다 충돌률이 더 커지게 되었다. 보상 수치 또한 음수가 된다.

사람이 늘어날수록 모델 성능 하락이 일어나는 것을 볼 수 있다.

Figure 4.1, 4.2를 통해 사람 10명에 대해 실험했을 때의 로봇과 crowd의 진행 경로를 확인할 수 있다.

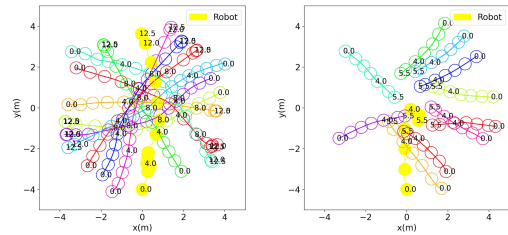


Fig. 4.1, 4.2: 사람 10명으로 실험을 진행했을 때 사람과 로봇의 진행경로를 표시한 그림이다.

4.1 성공한 경우, 노란색으로 표시된 로봇의 경로가 가운데를 가로질러 목적지까지 도달하는 것을 볼 수 있다.

4.2 충돌한 경우, 로봇이 사람과 부딪혀 시뮬레이션이 멈춘 것을 확인할 수 있다.

고정 장애물을 이용해본 실험 결과는 표 2를 통해 확인할 수 있다.

고정 장애물 10개만을 이용해 실험한 결과 성공률이 크게 떨어졌다. 이는 충돌은 없으나 시간 초과로 인해 발생한 것으로, 실패한 경우를 정성적으로 평가해본 결과 CrowdNav[1]가 정지한 대상에 대해 미흡한 결과를 보임을 알 수 있었다.

움직이는 사람 5명과 고정 장애물 5개가 섞인 시나리오에서도 0.21의 시간초과 확률을 가지고 있는 것을 통해, 정지한 대상을 마주했을 때 움직이지 못하는 증상이 나타남을 확인할 수 있었다.

CrowdNav[1]가 움직이는 crowd(사람)보다 고정된 crowd(장애물)에 대해 로봇이 행동 수행을 잘 하지 못함을 표의 성능 지표를 통해 확인할 수 있다.

Figure 5.1, 5.2, 5.3를 통해 장애물 5개와 사람 5명에 대해 실험했을 때의 로봇과 crowd의 진행 경로를 확인할 수 있다.

Method	Success	Collision	Time	Reward
논문	1.00	0.00	10.46	0.342
고정 장애물 10개	0.20	0.00	12.35	0.0484
사람 5명 + 고정 장애물 5개	0.78	0.01	12.16	0.2104

표 2: Crowd-Nav[1]을 활용하여 고정 장애물이 10개일 때와 고정 장애물 5개와 사람 5명을 함께 사용했을 때 두 가지 경우로 실험을 진행한 결과이다.

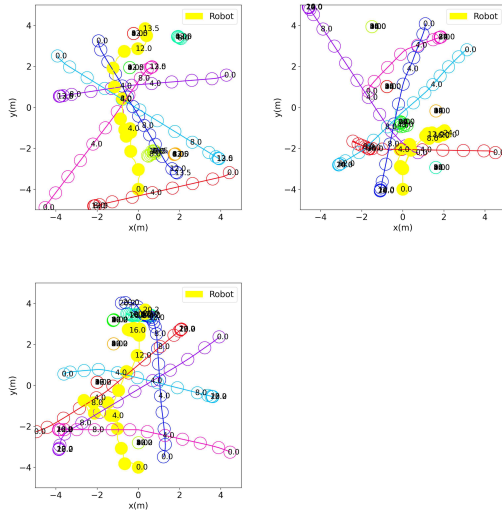


Fig. 5.1, 5.2, 5.3: 장애물 5개와 사람 5명

5.1 성공한 경우, 노란색으로 표시된 로봇의 경로가 가운데를 가로질러 목적지까지 도달하는 것을 볼 수 있다.

5.2 시간초과된 경우, 로봇이 고정된 장애물을 피하지 못하고 시간초과가 된 모습이다.

5.3 충돌한 경우, 고정된 장애물이 목표 위치를 가로막고 있어 충돌하는 것을 볼 수 있다.

V. 문제 해결

CrowdNav[1]로 실험을 진행했을 때 학습한 상황과 다른 상황에서 테스트를 진행할 경우 성능이 크게 떨어지는 문제점이 있었다.

이러한 문제점을 해결하기 위해 사용한 방법은 다음과 같다.

● 더 성능이 좋은 Crowd-Robot Interaction 관련 최신 알고리즘[2], [3]을 활용하여 문제를 해결하고자 하였다.

● test 상황과 같은 환경설정으로 train을 진행하여 문제를 해결하고자 하였다.

V.1 Crowd-Robot Interaction 최신 알고리즘[2], [3] 활용

Relational Graph Learning for Crowd Navigation[2]의 경우 CrowdNav[1]에 비해 성능 향상이 이루어졌으나 그 개선의 폭이 크지 않고 오히려 CrowdNav[1]의 반복에 가까운 점이 한계점으로 작용한다.

Social-NCE[3]는 CrowdNav[1]와 마찬가지로 train 상황과 test 상황이 다를 경우 성능하락이 발생한다는 문제점이 있으며 CrowdNav[1]에서 사용 가능한 시나리오가 Social-NCE[3]에서는 사용 불가능한 경우가 있는 문제점이 있다.

V.2 실험 상황으로 train

Crowd-Nav[1]를 활용하여 test 상황과 같은 환경 설정으로 train을 진행하였다.

표 3을 통해 그 결과를 확인할 수 있다.

test 환경과 train 환경이 같을 때 test 환경과 train 환경이 다를 때에 비해 성능이 상승한 것을 볼 수 있다.

하지만 test 상황과 같은 환경설정으로 train을 진행하여도 논문 성능에 비하면 다소 아쉬움이 있음을 확인할 수 있다.

Method	Success	Collision	Time	Reward
논문	1.00	0.00	10.46	0.342
사람 5명 + 고정 장애물 5개 case 1	0.78	0.01	12.16	0.2104
사람 5명 + 고정 장애물 5개 case 2	0.87	0.02	11.04	0.2746

표 3: Crowd-Nav[1]을 활용하여 고정 장애물 5개와 사람 5명인 상황에서 학습 환경을 다르게 하여 실험을 진행한 결과이다.

사람 5명 + 고정 장애물 5개 case 1의 경우 사람 5명으로 학습을 진행한 모델로 실험을 진행한 결과이다.

사람 5명 + 고정 장애물 5개 case 2의 경우 test 환경과 같은 환경으로 학습된 모델로 실험을 진행한 결과이다.

VI. 결 론

실험에 주로 활용한 CrowdNav[1]를 포함한 Crowd-Robot Interaction 알고리즘들[2],[3]은 훈련 환경과 같은 환경에서 실험했을 때는 높은 성공률을 보인다는 장점이 있다.

하지만 훈련 상황 이외의 다른 상황에서 테스트를 진행할 시 성공률이 크게 떨어지는 단점이 있다.

기존 Crowd-Robot Interaction 알고리즘의 개선을 통해 모델 성능을 향상시키고 훈련 상황 이외의 다른 상황에서 테스트를 진행할 시 성능이 크게 떨어지는 문제점을 해결할 수 있을 것이라 기대한다.

VII. 부 록

실험 관련 참고자료

논문 github 주소

CrowdNav[1] github :

<https://github.com/vita-epfl/CrowdNav>

Relational Graph Learning for Crowd Navigation[2] github :

<https://github.com/ChanganVR/RelationalGraphLearning>

Social-NCE[3] github :

<https://github.com/vita-epfl/social-nce>

팀 github 주소

<https://github.com/yst3147/Crowd-Robot-Interaction>

참 고 문 헌

- [1] Chen, Changan, et al. "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
- [2] Chen, C., Hu, S., Nikdel, P., Mori, G., & Savva, M. (2020). Relational graph learning for crowd navigation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 10007-10013). IEEE.
- [3] Liu, Y., Yan, Q., & Alahi, A. (2021). Social nce: Contrastive learning of socially-aware motion representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 15118-15129).
- [4] D. Helbing and P. Molnr, "Social force model for pedestrian dynamics," Physical Review E, vol. 51,

- no. 5, pp. 4282 - 4286, May 1995.
- [5] J. v. d. Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in 2008 IEEE International Conference on Robotics and Automation, May 2008, pp. 1928 - 1935.
- [6] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in Robotics Research, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer Berlin Heidelberg, 2011, pp. 3 - 19.
- [7] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning," arXiv:1609.07845 [cs], Sep. 2016, arXiv:1609.07845.
- [8] M. Everett, Y. F. Chen, and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," arXiv:1805.01956 [cs], May 2018, arXiv:1805.01956.