순차적 의사결정 문제와 해법

Sequential Problem, MDP, RL, and Applications.

서울과학기술대학교
심민규
mksim@seoultech.ac.kr

연사 소개

Current

- 서울과학기술대학교 산업공학과/데이터사이언스학과 부교수 (2019-)
- 대한전기자동차학회 (2025-)

Previous

- 조지아 공과대학 산업공학 박사 (2014)
  - Engineering Economy, Stochastic Processes
  - 금융시장 데이터 확률기반 모델링
- 삼성/KB 자산운용 펀드매니저 (2015-2017)
- 경희대학교 스마트에너지 사업단 연구교수 (2018-2019)

연구관심분야

- 산업분야 확률기반 모델링
- 강화학습을 활용한 의사결정
- 에너지 운영
  - 배터리 열화를 고려한 ESS 최적 충방전 전략
  - 에너지 기술(PV, ESS, V2X) 최적 운영 전략

에너지기술과 전기자동차?

배경 - 인류문명의 변천

- 모든 생명체는 태양에너지로 살아감
  - 농작물, 석탄, 석유
  - 과거에 저장된 에너지로 살아감
- RE100은 실시간 에너지를 활용

RE100 구현의 핵심과제는 에너지의 저장과 인출

- 에너지의 수요-공급에는 시간차이와 공간차이의 존재
- 에너지의 이동을 구현하는 저장 기술
  - temporal shifting: stationary ESS
  - **spatio-temporal shifting: EV**

Today’s talk on sequential decision making problem.

1. Overview
2. MDP & RL
3. Applications

I. Overview

인공지능과 강화학습

인공지능

- 기계가 시연하는 지능
- 지각(Perception), 추론(Reasoning), 학습(Learning), 의사결정(Decision Making)의 과제를 수행
- 데이터를 바탕으로 스스로 추론하고 학습하여, 의사결정을 기계가 수행

강화학습

- 인간 지능의 가장 높은 단계인 의사결정을 목적으로 함
  - 미래에 대한 조망을 바탕으로 한 현명한 의사결정은 지적 능력의 정수
  - 최적의 의사결정은 부가가치가 매우 높은 영역
- 인간이 지능을 사용하여 세상을 배워가는 방식과 닮아있다.

직립보행(Homo erectus)으로 보는 순차적 의사결정

Sequential decision making problem

- 특성 1: 시간적 의존성 (Temporal dependency)
  - 다음 상태는 직전 결과와 행동에 의존한다.
  - 걷는 행위는 자세를 계속적으로 바꾼다.
  - “한 번 어떻게 움직이는 것”을 정하는 것이 아니라 순차적인 의사결정 문제에 해당한다.
- 특성 2: 불확실성 (Uncertainty)
  - 내부적인 제어와 외부 충격으로 부터 불확실성이 존재한다.
  - 고정된 궤적이 아닌 모든 상황(contingency)에 대응하는 행동 규칙이 필요하다.
- 해답: 해답의 형태는 **해(solution)**가 아니라 **정책(policy)**여야 한다.

	해(solution)	정책 (policy)
형태	시간에 따른 제어 $\{a_0, ..., a_T\}$	상태 $\rightarrow$ 행동의 mapping
특징	특정 상황에서의 제어치 산출	모든 상황에서 즉각적인 의사결정치 제공

직립보행(Homo erectus)으로 보는 순차적 의사결정

Sequential Decision Making 관점에서의 해석

- Components
  - 자세, 균형, 발 위치, 속도 등에 대한 인지 (State, $\mathcal{S}$)
  - 한 발을 내딛기, 무릎을 굽히기, 팔은 흔들기 등을 선택 (Action, $\mathcal{A}$)
  - 현재 상태에서 취한 행동에 대한 다음 상태로의 전이 (Transition, $\mathcal{T}$)
  - 전진하는 것과 넘어지는 것에 대한 내외부 피드백 (Reward, $\mathcal{R}$)
- Policy
  - **S를 input으로 하고 A를 output으로 하며 R을 maximize하는 함수 (Policy, π)가 있어야 함**
  - R을 maximize하려면 주어진 S에 대해서 최적의 A를 찾을 수 있어야 한다.
  - $\pi : S \rightarrow A$를 정책함수라고 하며, 최적 정책함수 π^*를 찾으면 가장 잘 걸어다닐 수 있다.

인간의 직립보행

인간은 어떻게 π^*를 찾는가?

1. 한 번 걸어보는 세션(Episode)을 시작함
2. 물리적 환경(Environment)내에서 상태(S)가 주어짐
3. 의사결정 메커니즘(π)을 통해 어떤 행동(A)를 선택함
4. 보상(R)을 피드백으로 받고 다음 상태(S)로 이동
5. 다시 행동(A)을 선택하고 다음상태(S)로 이동하고 보상(R)을 받음
6. 넘어져서 세션이 종료할 때까지 위의 과정을 반복
7. 현재의 의사결정 메커니즘(π)을 반성하고 개선함
8. 위의 1.-7.을 반복하다보면 π가 π^*로 수렴함

몇 가지 용어들

- Reinforcement 과정
  - Experience replay (승리한 대국의 복기는 이기는 습관을, 패배한 대국의 복기는 이기는 준비를 만들어준다 - 조훈현 국수)
  - 위의 7번 과정의 피드백을 통한 positive reinforcement vs negative reinforcement.
- Exploration vs Exploitation
  - 초기에는 이것저것 시도하여 경험을 쌓고(Exploration) 점점 지식내의 최선의 행위를 선택(Exploitation)한다.
- Return은 Reward의 누적합
  - Return = $\sum_{\forall t}$ Reward
- State value function $V^{\pi}(s)$
  - 현재 상태 s에서 부터 π의 의사결정 방식을 취한다면 몇 걸음을 더 걸을 수 있는가?
- State-action value function $Q^{\pi}(s, a)$
  - 현재 상태 s에서 행동 a를 취하고 그 이후에 π의 의사결정 방식을 취한다면 몇걸음을 더 걸을 수 있는가?

로봇의 직립보행

로봇의 직립보행 학습 전략

	인간의 직립보행	로봇의 직립보행
상태(S)의 인식 보행 환경	본능적으로 상태인식 물리적 중력공간이 제공됨	1. 상태변수의 정의가 필요 2. 시뮬레이션 공간정의 필요 $P : S \times A \to S$
행동	본능적인 신체의 움직임	3. 관절의 움직임 정의 필요
정책함수	뇌신경계에 탑재된 초기 π 함수	4. 정책함수 초기화 필요
보상	주위의 피드백 및 본능적 만족감	5. 보상 수치의 정의 필요
Replay	비주기적으로 시행	6. 주기와 방식을 명시해주어야 함

- Markov Decision Process로 문제를 정의하고 Reinforcement Learning Algorithm으로 최적정책 π^*를 학습한다.
- MDP로 문제 정의
  - 1. 상태 변수 정의 / 2. 전이 확률 정의 / 3. 행동 집합 정의 / 5. 보상 함수 정의
- RL로 최적 정책함수 도출
  - 4. 정책함수 초기화 / 6. 학습 주기, 알고리즘 등 선택

II. MDP & RL

MDP (Markov Decision Process)

- 현재 상태와 행동으로 다음 상태·보상이 결정되는 확률적 의사결정 문제를 수학적으로 정의한 프레임워크

특성/표기/목적

- 특성
  - Temporal dependency: 현재의 상태와 행동이 다음 상태에 영향을 미친다.
  - Sequential decision: 매 순간 상태가 주어지고 이에 기반해 행동을 선택해야 한다.
  - Uncertainty: 상태의 전이와 보상의 결정에 있어서 uncertainty가 존재할 수 있다.
- 표기: 5-tuple notation $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$
  - $\mathcal{S}$: 상태집합, 에이전트가 관찰할 수 있는 모든 상황
  - $\mathcal{A}$: 행동집합, 각 상태에서 취할 수 있는 선택지
  - P: 전이확률, $P(s'|s,a)$ 상태 s에서 a를 선택하면 s'으로 갈 확률
  - $R(s,a)$: 보상함수, 전이에 대해서 받는 즉시 보상
  - γ: 할인율, 미래 보상의 현재 가치로의 조정
- 목적
  - 누적기대보상 $G_0 = \sum_{t=0}^{\infty} \gamma^t R_t$를 최대화 시키는
  - 최적 정책함수 $\pi^* : S \to A$를 찾는다.

RL

최적 정책함수 π^*를 찾는 알고리즘

- 고수준 학습절차
  1. 경험수집 - 환경으로 부터 (s, a, r, s') history를 수집
  2. 개선과정 - 크게 아래의 셋 중에 하나의 접근을 선택
     - Value-based - 가치함수를 Bellman 오차를 이용해서 개선
     - Policy-based - 정책함수를 정책 그래디언트로 개선
     - Actor-Critic - 위의 Value-based와 Policy-based의 하이브리드
  3. 위의 1번과 2번을 반복하여 $\pi \rightarrow \pi^*$

Remaining questions

- Value-based
  1. 가치함수란 무엇인가
  2. Bellman 방정식이란 무엇인가
  3. Bellman 오차를 이용해서 가치함수를 개선한다는 것은 무엇인가
- Policy-based
  1. 정책함수란 무엇인가
  2. 정책 그래디언트란 무엇인가
  3. 정책 그래디언트를 이용해서 개선한다는 것은 무엇인가
- Actor-Critic
  1. Actor는 무엇이고 Critic은 무엇인가
  2. 어떤 방식으로 하이브리드 방식을 구현하는가
- Fundamental issues
  1. Exploration vs Exploitation
  2. On-policy vs Off-policy

한 문장 답변 (1/2)

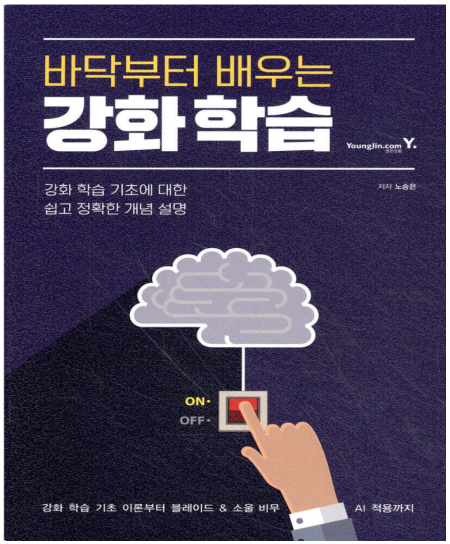
- Value-based
  1. 가치함수란 무엇인가
     - 특정 정책 하에서 한 상태(또는 상태-행동)에서 얻을 장기 누적보상의 기댓값
  2. Bellman 방정식이란 무엇인가
     - 가치함수가 “즉시 보상 + 할인된 다음 상태 가치의 기댓값”과 같다는 자기 일관성 식
  3. Bellman 오차를 이용해서 가치함수를 개선한다는 것은 무엇인가
     - Bellman 식의 양변 차이를 최소화하도록 파라미터를 조정하는 numerical approach
- Policy-based
  1. 정책함수란 무엇인가
     - 주어진 상태에서 각 행동을 선택할 확률(또는 결정적 행동)을 매핑하는 함수
  2. 정책 그래디언트란 무엇인가
     - 정책 파라미터에 대한 기대 누적보상의 기울기로, 이를 따라가면 보상이 증가
  3. 정책 그래디언트를 이용해서 개선한다는 것은 무엇인가
     - 수집된 샘플로 그 기울기를 추정해 파라미터를 업데이트함으로써 정책을 점진적으로 최적화하는 과정

한 문장 답변 (2/2)

- Actor-Critic
  1. Actor는 무엇이고 Critic은 무엇인가
     - 행동을 생성하는 정책 네트워크이고, Critic은 Actor의 가치를 추정해 업데이트 신호(Advantage)를 제공하는 가치 네트워크
  2. 어떤 방식으로 하이브리드 방식을 구현하는가
     - Critic이 계산한 Advantage를 이용해 Actor를 정책 그래디언트로, Critic 자체는 TD 오차로 학습시키며 두 네트워크를 동시 업데이트
- Fundamental issues
  1. Exploration vs Exploitation
     - 탐험(미지의 행동을 시도해 정보를 얻음)과 활용(검증된 행동으로 즉시 보상을 얻음)사이의 trade-off
  2. On-policy vs Off-policy
     - On-policy: 데이터 수집과 학습에 같은 정책을 사용
     - Off-policy: 다른 정책으로 모은 경험을 목표 정책 학습에 활용

학습 자료 소개

- 노승은 (2020) 바닥부터 배우는 강화 학습.
  - Good build-up from DTMC, MRP, and MDP. I highly recommend this.
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
  - Complete treatment on reinforcement learning algorithm.
- 심민규 강의 자료
  - 강의노트 / 연습문제 / Youtube 강의 동영상
  - https://github.com/aceMKSim/teaching/tree/master/Stochastic_Processes



Topics

1st semester - Stochastic Processes and Simulation

Part A. Review on quantitative methods

- Math review
- Prob/stat review
- Simulation review

Part B. Newsvendor model

- One-period decision making under uncertainty
- Optimization (always) involves trade-off

Part C. Discrete-time Markov chain (DTMC)

- Multi-period description on stochastic system
- As a foundation of Markov decision process (MDP)

Part D. Bridges between DTMC and MDP

- Extending system description (DTMC) to decision making problem (MDP)
- Dynamic programming (DP) and Markov reward process (MRP) to be covered

Part E. MDP w/ model

- Formulating MDP
- Approaches for finding an optimal policy

2nd semester - Reinforcement Learning

Part F. MDP w/o model

- How to find an optimal policy when model is not present?
- Core components of reinforcement learning

Part G. Functional approximation

- Overcome the curse of dimensionality by functional approximation
- Deep neural network to be reviewed

Part H. Value based agent

Part I. Policy based agent

Part J. Recent advancements in DRL (Deep reinforcement learning)

III. Applications - “Battery-Supercapacitor Systems in EV”

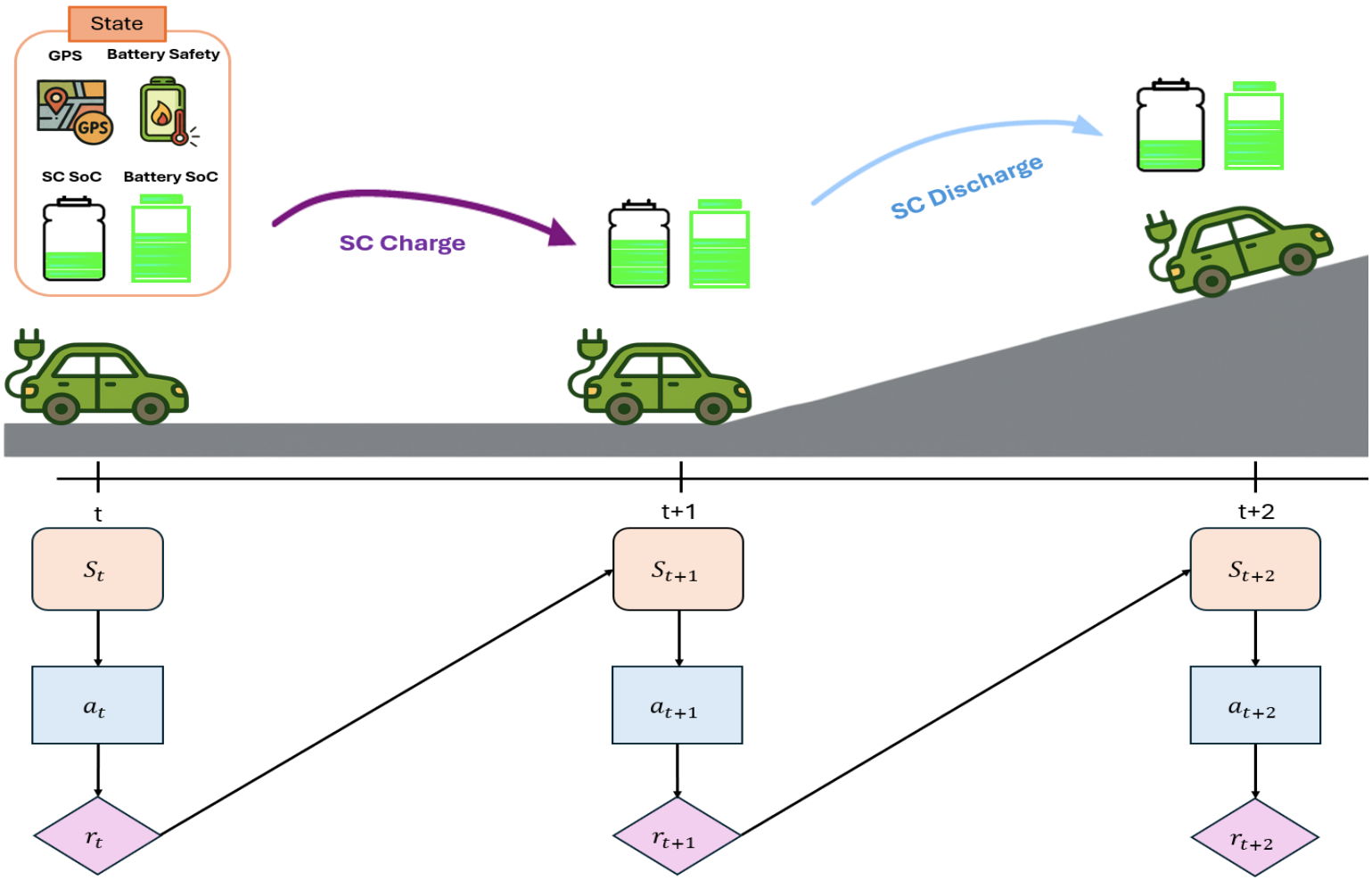
Background

Battery vs Supercapacitor

	리튬이온배터리	슈퍼커패시터
에너지저장방식	화학적저장	물리적저장
반응속도	느림	빠름
에너지밀도	높음	낮음
수명	수천싸이클	수십만싸이클
EV운행시활용	일반적인 상황	순간적 출력

- 저장 vs 출력의 장점을 각각 가지고 있음
- 슈퍼커패시터가 배터리의 열화를 방지하는 기능 제공 (DoD와 온도 제어)

Graphical Abstract



Problem Definition

Goal

- 슈퍼캐퍼시터 활용 여부에 대한 의사결정
  - 고출력이 필요한 상황
  - 회생제동(regenerative braking)이 가능한 상황

MDP formulation (1)

- 의사결정 변수 ($\mathcal{A}$, 행동집합)
  - $a_t = [I_t^{SC}]$
- 상태변수 ($\mathcal{S}$: 상태집합)
  - 의사결정에 필요한 모든 변수를 담고 있어야 함
  - 에이전트가 관찰할 수 있어야 함
  - 배터리 상태: $SoC_t^{bat}, SoC_t^{SC}, T_t^{bat}$
  - 현재 주행 상태: P_t^{load}
  - **미래의 주행 환경**
    - d_t: remaining distance for this trip
    - z_t^{grade}: remaining road condition
    - $z_t^{traffic}$: remaining traffic signal

Novelty

- 특색: **미래의 주행환경을 반영**한 Optimal Control on Hybrid EV
- 동기: Intelligent EV는 GPS trip 정보를 활용하여 차량을 제어해야 한다.
- 해법: World Model Framework

MDP formulation (2)

- 보상함수 ($R(s,a)$)
  - minimize [에너지손실 + 배터리열화비용]
  - with additional constraints…
- 전이확률 (P)
  - $P(s'|s,a)$, 상태 s에서 a를 선택하면 s'으로 갈 확률
  - 운행정보관련 변수: 공개된 주행 데이터셋을 활용해 시뮬레이터 구현
  - 배터리 관련 변수: 전이 다이나믹스 수식을 활용해서 전이 환경 구성

Key Performance Questions

- Q1. (Cost saving) RL agent는 낮은 에너지 소비로 Trip을 완료하는가?
- Q2. (Cost saving) RL agent는 배터리 열화를 최소화하며 Trip을 완료하는가?
- Q3. (Intelligent behavior) SC에 대한 충방전 policy가 intelligent한 것으로 보이는가?
  - 고출력환경(급가속, 오르막길)에서 SC 방전으로 배터리 열화 줄임
  - 가까운 미래의 고출력환경 대비 선제적 SC 충전
  - 회생제동 충전

IV. Applications - Fridge Control

```
"KIEVC 2025 Tutorial"
```