Spodbujevano učenje pri igranju namiznih iger
(angl. *Reinforcement learning in board games*)

Tim Kalan

Mentor: izr. prof. dr. Marjetka Knez

Fakulteta za matematiko in fiziko

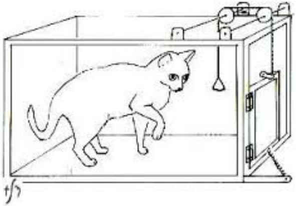
30. marec 2021

Napovednik

- Motivacija,
- problem spodbujevalnega učenja,
- algoritmi,
- namizne igre.

Motivacija: Instrumentalno pogojevanje

- Psihološko motivirana podlaga.
- **Nagrade in kazni**.




Okvir

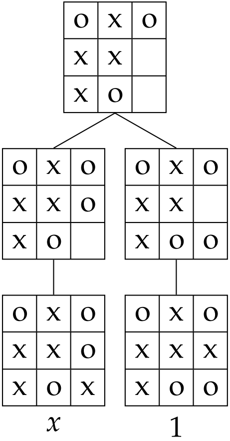

Okolje
Situacija
(senzorika)
nagrajevanje
kaznovanje
Dejanje
(aktiranje)

Primer 1: robot se uči hoje

- **Situacija/Stanje**: položaj v sobi in stanje nog,
- **Nagrada**: 1 za doseg vrat, 2 za ključ, -0.5 za časovni korak,
- **Okolje**: soba in senzorji, ki govorijo o položaju,
- **Akcija**: Premik noge.

Primer 2: križci in krožci

- **Situacija/Stanje**: stanje na plošči,
- **Nagrada**: 1 za zmago, -1 za poraz, x za izenačenje/korak,
- **Okolje**: nasprotnik, plošča, sodnik, nagrajevalec,
- **Akcija**: postavitev X oz. O na ploščo.



Ideja

- Agent »pade« v okolje.
- S poskušanjem se nauči pravilnih akcij.
- Svoje znanje izkoristi za maksimizacijo nagrade.

Ideja

- Agent »pade« v okolje.
- S poskušanjem se nauči pravilnih akcij.
- Svoje znanje izkoristi za maksimizacijo nagrade.

Hipoteza 1 (Hipoteza o nagradi).
Vse cilje je mogoče opisati kot maksimizacijo neke kumulativne numerične nagrade.

Formalizacija: Markovski proces odločanja 1

Definicija 2 (Markovska veriga).
Slučajni proces $(S_t)_{t=0}^T$ *na končnem verjetnostnem prostoru* $(\Omega, \mathcal{F}, P)$ *je* ***Markovska veriga****, če velja Markovska lastnost*

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, ..., S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

Formalizacija: Markovski proces odločanja 1

Definicija 2 (Markovska veriga).
Slučajni proces $(S_t)_{t=0}^T$ *na končnem verjetnostnem prostoru* $(\Omega, \mathcal{F}, P)$ *je* ***Markovska veriga****, če velja Markovska lastnost*

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, ..., S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

- Prihodnost je neodvisna od preteklosti, če poznamo sedanjost

Formalizacija: Markovski proces odločanja 1

Definicija 2 (Markovska veriga).
*Slučajni proces $(S_t)_{t=0}^T$ na končnem verjetnostnem prostoru $(\Omega, \mathcal{F}, P)$ je **Markovska veriga**, če velja Markovska lastnost*

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, ..., S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

- Prihodnost je neodvisna od preteklosti, če poznamo sedanjost
- $p_{ss'} := P(S_{t+1} = s' \mid S_t = s) \to \mathcal{P} := [p_{ss'}]_{s,s' \in \mathcal{S}}$, $\mathcal{S}$ je množica stanj
- Markovska veriga je torej dvojica $(\mathcal{S}, \mathcal{P})$

Formalizacija: Markovski proces odločanja 2

Definicija 3 (Markovski proces nagrajevanja).
Markovski proces odločanja *je nabor* $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$*, kjer je*

- $\mathcal{S}$ *je (končna) množica stanj*
- $\mathcal{P}$ *je prehodna matrika, kjer* $p_{ss'} = P(S_{t+1} = s' \mid S_t = s)$
- $\mathcal{R}$ *je nagradna funkcija* $\mathcal{R}_s = E[R_{t+1} \mid S_t = s]$
- $\gamma \in [0, 1]$ *je diskontni faktor*

Formalizacija: Markovski proces odločanja 3

Definicija 4 (Markovski proces odločanja).
Markovski proces odločanja *je nabor* $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$*, kjer je*

- $\mathcal{S}$ *je (končna) množica stanj*
- $\mathcal{A}$ *je (končna) množica akcij oz. dejanj*
- $\mathcal{P}$ *je prehodna matrika, kjer* $p^a_{ss'} = P(S_{t+1} = s' \mid S_t = s, \mathbf{A_t} = \mathbf{a})$
- $\mathcal{R}$ *je nagradna funkcija* $\mathcal{R}^a_s = E[R_{t+1} \mid S_t = s, \mathbf{A_t} = \mathbf{a}]$
- $\gamma \in [0, 1]$ *je diskontni faktor*

Primer: MDP

Agent 1

- Strategija (angl. *Policy*)
- Vrednostna funkcija (angl. *Value function*)
- (Model)

Agent 2: strategija

Definicija 5.

- ***Deterministična strategija*** *stanju* s *priredi akcijo* a,

$$\pi(s) = a.$$

- ***Stohastična strategija*** *za vsako stanje* s *pove verjetnosti vseh možnih akcij* a,

$$\pi(a|s) = P(A_t = a \mid S_t = s).$$

Agent 3: povračilo

Definicija 6 (Povračilo).

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Agent 4: vrednostna funkcija

Definicija 7 (Vrednostna funkcija).

- ***Vrednostna funkcija stanja*** *je pričakovana vrednost povračila, če se vedemo skladno s strategijo* π

$$v_\pi(s) = \mathrm{E}[G_t \mid S_t = s].$$

- ***Vrednostna funkcija akcije*** *je podobna prejšnji, le da sprosti prvo akcijo*

$$q_\pi(s, a) = \mathrm{E}[G_t \mid S_t = s, A_t = a].$$

Primer - križci in krožci

Algoritmi

- Učenje prek strategije ali **vrednostne funkcije**.
- Celoten problem je **načrtovanje**:
  - Napovedovanje - ugotvaljanje vrednosti.
  - Upravljanje - iskanje optimalne strategije.

Algoritmi: dinamično programiranje

- Poznamo $\mathcal{P}$ in $\mathcal{R}$,
- Bellmanove enačbe,
- vrednostna funkcija - ponovna uporaba rešitev,

Algoritmi: DP - iterativno ocenjevanje strategije

-

Algoritmi: Monte Carlo 1

- Nepoznan epizodični MDP,
- problem napovedovanja,
- empirično povračilo,
- štejemo obiske stanj.

Algoritmi: Monte Carlo 2

- Ob **prvem** obisku stanja s:

$$N(s) \leftarrow N(s) + 1$$
$$S(s) \leftarrow S(s) + G_t$$

- Po koncu učenja:

$$V(s) \leftarrow S(s)/N(s)$$

- Pomni: Računanje povprečja zaporedja $(X_i)_{i\in\mathbb{N}}$

$$\mu_k = \frac{1}{k}\sum_{j=1}^{k} X_j = \mu_{k-1} + \frac{1}{k}(X_k - \mu_{k-1})$$

- Inkrementalni Monte Carlo:

$$V(s) \leftarrow V(s) + \frac{1}{N(t)}(G_t - V(S_t))$$

Algoritmi: Monte Carlo 3

- Inkrementalni Monte Carlo:

$$V(s) \leftarrow V(s) + \alpha(G_t - V(S_t)).$$

- Splošni obrazec:

$$\textit{nova ocena} \leftarrow \textit{stara ocena} + \textit{korak}\,(\textit{tarča} - \textit{stara ocena}).$$

Algoritmi: TD(0)

- Učenje s časovno razliko.
- *Bootstrapping*.
- Ne potrebujejo povračila.
- $G_t \approx R_{t+1} + \gamma V(S_{t+1})$.

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)).$$

Algoritmi: TD(λ) 1

- Povezava med MC in TD(0).
- $G_t^{(n)} = R_{t+1} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+1})$.
- Povprečenje različnih $G_t^{(n)}$: $G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{(n-1)} G_t^{(n)}$.

TD(λ) s pogledom naprej:

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^\lambda - V(S_t)).$$

Algoritmi: TD(λ) 2

- **Sledi upravičenosti** (angl. *eligibility traces*):

$$
\begin{aligned}
E_0(s) &= 0 \\
E_t(s) &= \gamma\lambda E_{t-1}(s) + \mathbb{1}(S_t = s),
\end{aligned}
$$

TD(λ) s pogledom nazaj:

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^{\lambda} - V(S_t)).$$

SLIKE BACKUPOV

Primer - kje se zatakne?

- velike plošče
- pri zgornjih algoritmih hranimo vse v tabeli

Aproksimacija

Motivacija - namizne igre

- Abstraktno mišljenje.
- Model življenja.
- Testiranje algoritmov.

Namizne igre - postanja

Namizne igre - trening

Namizne igre - tradicionalne metode

Namizne igre - združevanje

Nekaj rezultatov

Literatura

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An introduction*. The MIT Press, 2015.

Imran Ghory. *Reinforcement learning in board games*. 2004.