Heuristické optimalizačné procesy

Algoritmy založené na teórii hier, strategizácia

prednáška 7
Ing. Ján Magyar, PhD.
ak. rok. 2025/2026 ZS

Teória hier

matematická odvetvie skúmajúce štruktúrované rozhodovanie

skutočné správanie vs. optimálne správanie

interaktívne výpočty

Charakteristiky hier

štruktúrovaná interakcia v kolách

počet hráčov

nulový/nenulový súčet

kooperatívne/antagonistické hry

úplná/neúplná informácia

Nashova rovnováha

pre nekooperatívne hry

berie do úvahy stratégiu ostatných hráčov

žiaden hráč nevie zmeniť svoju stratégiu tak, aby dosiahol lepší výnos

pre stratégiu hráča i s_i:

$u_i(s_i{}^*, s_{-i}{}^*) \geq u_i(s_i, s_{-i}{}^*)$ pre všetky $s_i \in S_i$

Väzňova dilema

hra s nenulovým súčtom

rovnováha vedie k suboptimálnemu riešeniu

viac iterácií môže viesť k optimálnej stratégii

Ľudské rozhodovanie

1. analýza
2. stratégia
3. taktika

Strojové prevedenie

na základe pravidiel

lačné rozhodovanie

preskúmanie všetkých možností

prezeranie (čo najviac) dopredu

Bodovacia funkcia

číselné ohodnotenie výsledných aj medzistavov

popísaná niekoľkými príznakmi alebo vlastnosťami:
$g(f_1, f_2, ..., f_n)$

zvyčajne lineárna polynomiálna funkcia: $c_1 f_1 + c_2 f_2 + ... + c_n f_n$

v pokročilejších stavoch počet príznakov klesá

Minimax algoritmus

dvaja hráči v adverzariálnej hre

hráč 1 sa snaží maximalizovať svoje výnosy, hráč 2 chce minimalizovať výnosy protivníka

algoritmus určuje optimálnu stratégiu oboch hráčov

Pseudokód Minimax

```
function minimax(node, maximizingPlayer):
    if uzol je terminálny then
        return bodovacia hodnota uzla
    if maximizingPlayer then
        value := −∞
        foreach potomok uzla
            value := max(value, minimax(child, !maximizingPlayer))
        return value
    else
        value := +∞
        foreach potomok uzla
            value := min(value, minimax(child, !maximizingPlayer))
        return value
```

Alfa-beta orezávanie

umožní orezať vetvy, ak pred tým bolo dokázané, že existuje lepšie riešenie (pre maximalizujúceho aj minimalizujúceho hráča)

alfa = minimálna hodnota, ktorú vie dosiahnuť maximalizujúci hráč

beta = maximálna hodnota, ktorú vie dosiahnuť minimalizujúci hráč

zložitosť algoritmu $O(b^d) \rightarrow O(\sqrt{b^d})$

Pseudokód alfa-beta orezávania

```
function alphabeta(node, α, β, maximizingPlayer):
    if uzol je terminálny then
        return bodovacia hodnota uzla
    if maximizingPlayer then
        value := −∞
        foreach potomok uzla
            value := max(value, alphabeta(child, α, β, !maximizingPlayer))
            if value > β then
                break
            α := max(α, value)
        return value
    else
        value := +∞
        foreach potomok uzla
            value := min(value, alphabeta(child, α, β, !maximizingPlayer))
            if value < α then
                break
            β := min(β, value)
        return value
```

Učenie posilňovaním

cieľom je natrénovať agenta, ktorý reaguje na prostredie tak, aby dosiahol určený cieľ

učenie je umožnené cez interakciu s prostredím

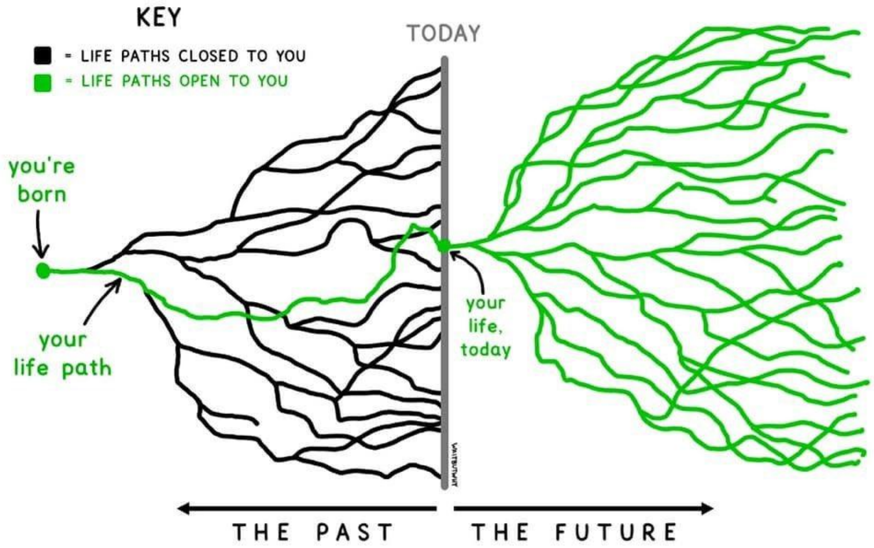
agent dostáva spätnú väzbu

Interakcia s prostredím

agent získa skúsenosti súčasne s učením

agent riadi interakciu štýlom pokus-omyl - potrebuje úspechy aj neúspechy

akcie môžu ovplyvniť budúce možnosti agenta



Spätná väzba

zriedkavá/po každej akcii

oneskorená/okamžitá

často určená pre postupnosť akcií

ťažko odhadnúť (ne)správnosť akcií agenta

pre agenta je skôr relevantná kumulatívna odmena

Terminológia

stav prostredia

akcia agenta

prechod prostredia

politika agenta

odmena

model prostredia

Stav

typy stavu

- stav prostredia s_t^e
- agentov stav prostredia s_t^a
- pozorovanie prostredia o_t

plná pozorovateľnosť

- $s_t^e = s_t^a = o_t$

čiastočná pozorovateľnosť

- $s_t^e \neq s_t^a$
- agent aktualizuje s_t^a na základe predošlých pozorovaní
- $s_t^a = (P[s_t = s_1], P[s_t = s_2], \dots, P[s_t = s_n])$

Akcie

množina akcií, ktoré sú agentovi k dispozícii: $a \in A$

často diskrétne akcie, ale priestor akcií môže byť aj spojitý

v každom stave môžu byť dostupné všetky akcie, alebo môžu byť aj limitované

množina akcií je vždy daná

Prechody

ak agent vyberie niektorú akciu, prostredie na ňu zareaguje a aktualizuje svoj stav

aktualizácia stavu je popísaná prechodom $T: S \times A \to S$

deterministické/nedeterministické

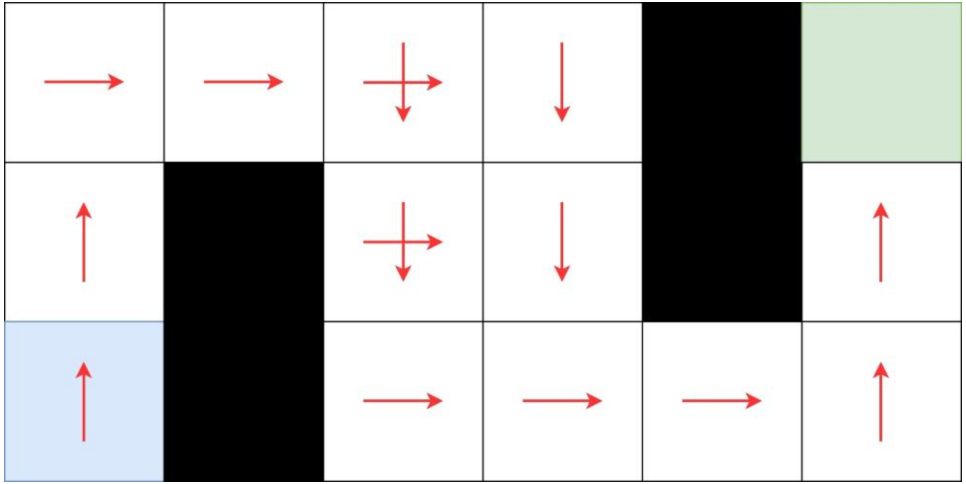
v niektorých problémoch môžu byť časovo závislé

Politika

mapuje stav na akciu agenta $\pi: S \rightarrow A(s)$

deterministická $a = \pi(s)$

stochastická $\pi(a|s) = P[A_t = a|S_t = s]$



Odmena

agent ju obdrží po zásahu do prostredia

číselná hodnota

kladná – odmena, nulová – neutrálna, záporná – trest

deterministická/stochastická

agent by mal maximalizovať kumulatívnu odmenu

-1	-1	-1	-1		10
-1		-1	-1		-1
-1		-1	-1	-1	-1

Model

agentova nepovinná reprezentácia prostredia

predikuje dynamiku a odmenu

nie je perfektný

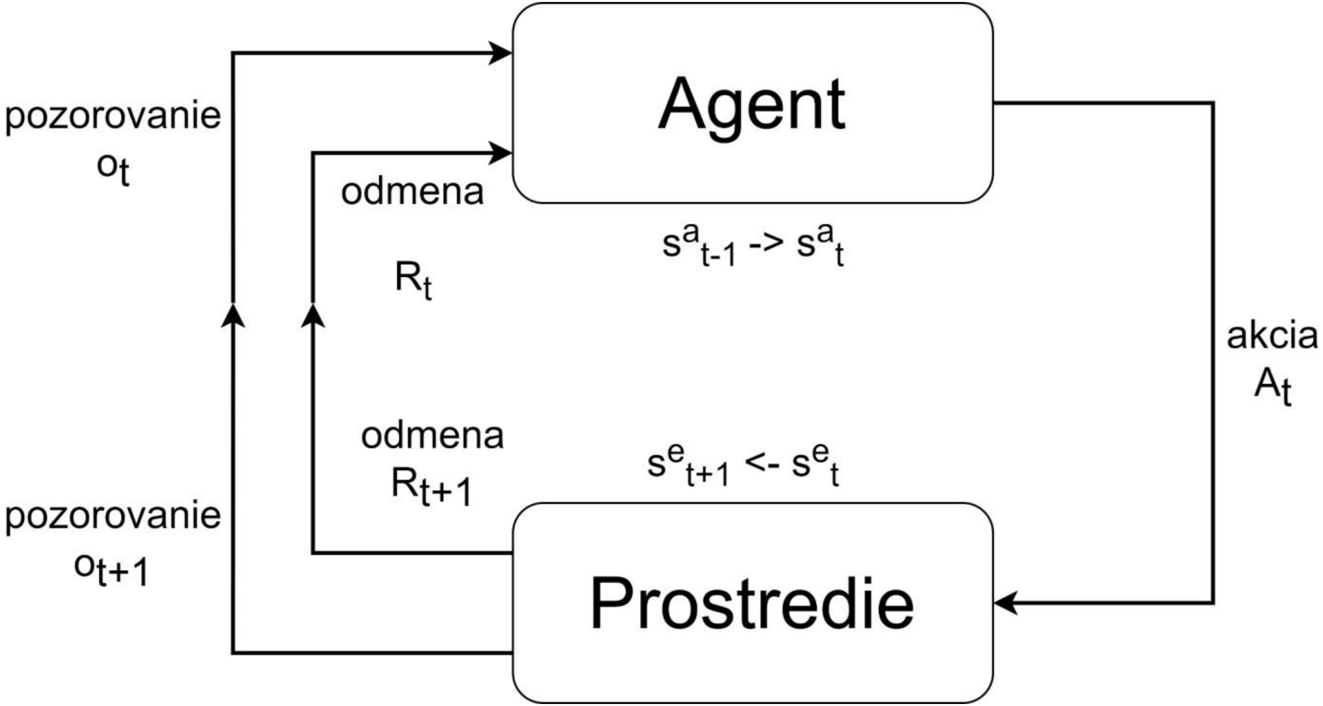
model môže byť poskytnutý,

alebo ho agent sám zostrojí,

alebo ho vôbec nepotrebuje

-1	-1	-1			10
-1		-1	-1		-1
-1			-1	-1	-1

Interakcia agent-prostredie



Kumulatívna odmena

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_T$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ \ldots = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

diskontný faktor $\gamma \in < 0,1 >$

- $\gamma = 0$ – uvažovanie iba okamžitej odmeny
- čím je väčšia γ, tým dlhšiu dobu berie agent do úvahy
- vyhneme sa nekonečne veľkej kumulatívnej odmene

Voľba akcií – politika

politika ja spôsob, ktorým agent volí svoje akcie

formálne je to distribúcia pravdepodobnosti nad akciami v určitom stave:

$$\pi(a|s) = P[A_t = a|S_t = s]$$

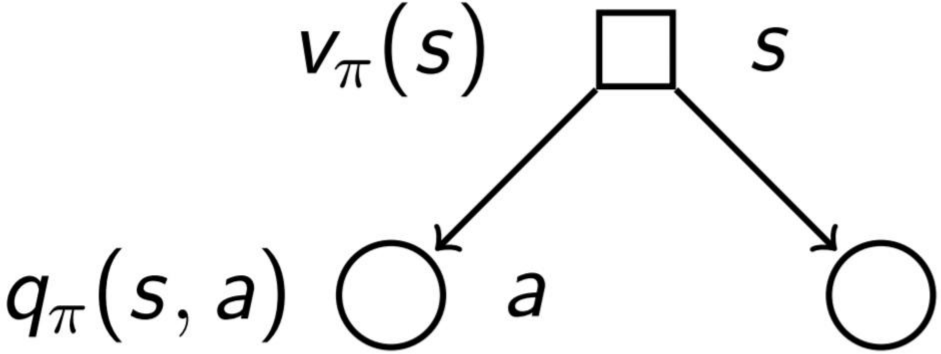
zmyslom učenia posilňovaním je špecifikovať vhodnú politiku agenta na základe skúseností s pôsobením agenta v prostredí

Hodnotová funkcia stavu

$$V_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$$

ako výhodné je byť v danom stave
očakávaná kumulatívna odmena sekvencie začínajúcej v danom stave pri politike π

Bellmanova rovnica očakávania pre v_π



$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s, a)$$

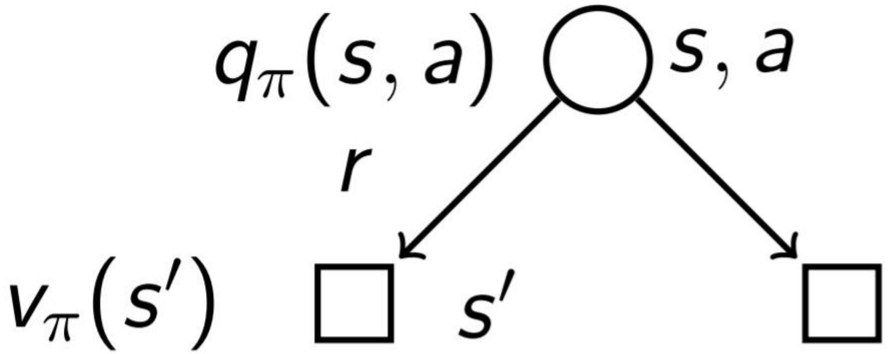
Hodnotová funkcia akcie

$$q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$$

ako výhodné je v danom stave použiť danú akciu pri politike π

očakávaná kumulatívna odmena sekvencie začínajúcej v danom stave danou akciou, ak voľba nasledujúcich akcií je podľa politiky π

Bellmanova rovnica očakávania pre q_π



$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s')$$

Optimálny výber akcií

cieľom je vybrať akcie, ktoré maximalizujú kumulatívnu odmenu

hodnotová funkcia stavu umožňuje parciálne usporiadanie výberových politík

$$\pi \geq \pi' \text{ ak platí, že } v_\pi(s) \geq v_{\pi'}(s) \text{ pre všetky } s \in S$$

optimálna politika π^*

- lepšia alebo rovnako dobrá ako ostatné politiky
- vždy existuje
- môže ich byť viac

Optimálne hodnotové funkcie

$v_*(s)$, $q_*\,(s,a)$ – hodnotové funkcie pri použití optimálnej politiky π^*

- všetky optimálne politiky produkujú rovnaké funkcie $v_*(s)$ a $q_*\,(s,a)$

$v_*(s)$ je maximum hodnotovej funkcie stavu pri uvažovaní všetkých možných politík

$$v_*(s) = \max_\pi v_\pi(s)$$

$q_*\,(s,a)$ je maximum hodnotovej funkcie akcie pri uvažovaní všetkých možných politík

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

Nájdenie optimálnej politiky

ak poznáme $q_* (s, a)$, bezprostredný výber vhodnej akcie:

$$\pi^*(a|s) = \begin{cases} 1, & ak\ a = \operatorname{argmax}_a q(s,a) \\ 0, & inak \end{cases}$$

ak poznáme $v_*(s)$

- prehľadávanie všetkých akcií prípustných v danom stave
- hľadanie do hĺbky 1 (obmedzené iba na jeden krok)
- výber najlepšej možnosti (greedy princíp výberu)

otázky?