Plánovanie a učenie

(Strojové učenie II)

M. Mach

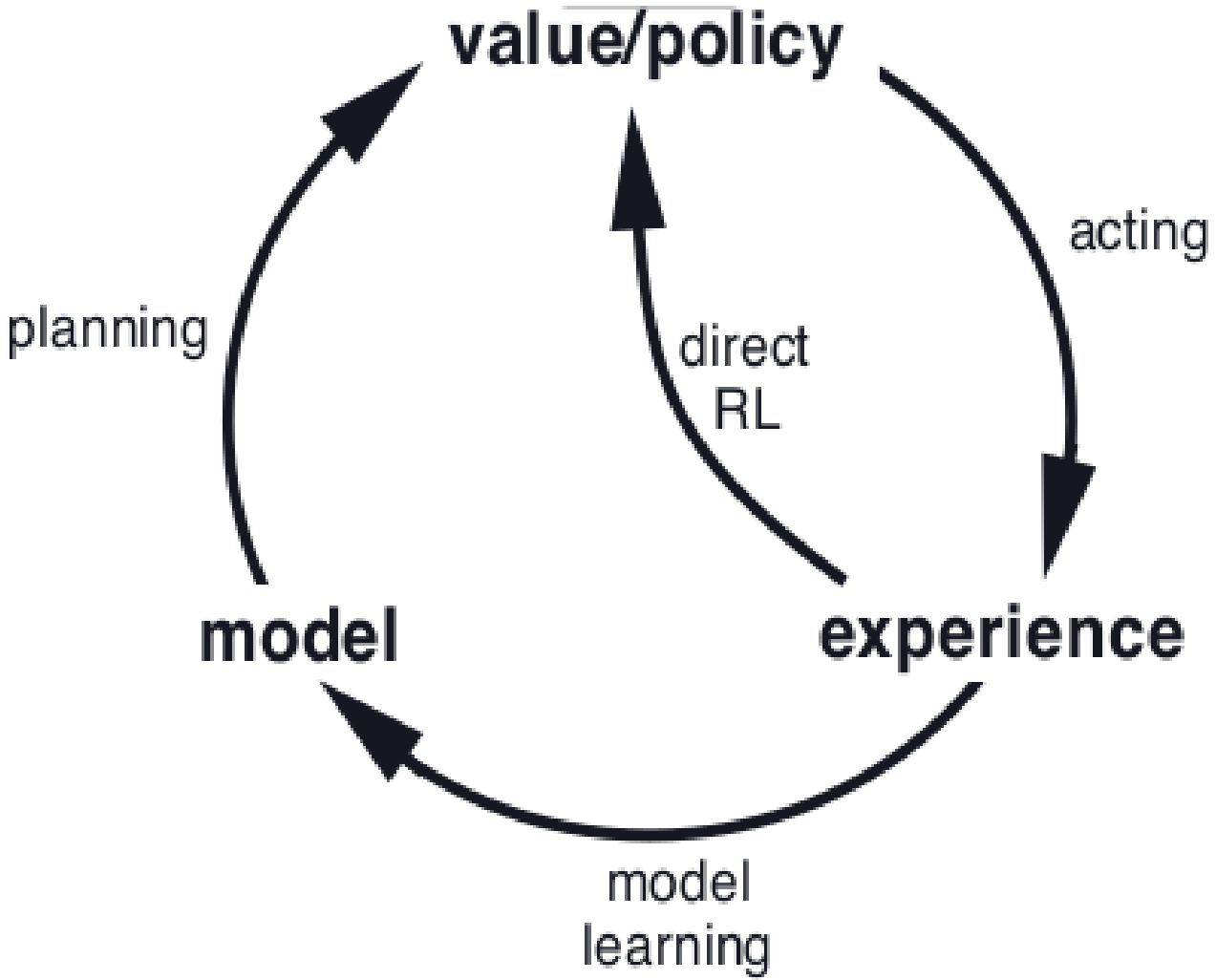
Katedra kybernetiky a umelej inteligencie, FEI, TUKE

marec 2023

Model prostredia

- Model umožňuje predikovať reakciu okolia na zvolenú akciu A_t v kontexte aktuálneho stavu S_t
  - predikuje odmenu R_{t+1} a nasledujúci stav S_{t+1}
- Typy modelov
  - distribučný model
    - poskytuje množinu alternatív a distribúciu ich pravdepodobností: $p(r, s'|S_t, A_t)$, $r \in \{\ldots\}$ a $s \in \{\ldots\}$
    - takýto model je potrebný pre DP
  - vzorkovací model
    - poskytuje priamo jednu z možných alternatív (alternatívy vzorkuje, vyberá jednu vzorku): R_{t+1}, S_{t+1}
    - postačuje pre MC a TD
- Model umožňuje simulovať skúsenosť získateľnú z prostredia - simulovaná skúsenosť
  - plánovanie - používanie skúsenosti simulovanej modelom
  - učenie - používanie (reálnej) skúsenosti s prostredím

Plánovanie a učenie - unifikačný pohľad



©*Sutton-Barto: Reinforcement Learning, 2nd ed., 2018*

Plánovanie a učenie

- Proces učenia a plánovania
  - agent na základe politiky interaguje s prostredím, získava skúsenosť
  - skúsenosť môže byť používaná dvojako
    - učenie hodnotových funkcií alebo politiky (priame RL)
    - učenie (alebo vylepšovanie) modelu
  - model je používaný na učenie hodnotových funkcií
  - ak sú hodnotové funkcie, tak z nich je odvodená politika
- Spôsoby použitia
  - učenie: value/policy - experience (MC a TD na základe reálnej skúsenosti)
  - plánovanie: model - value (DP, MC a TD na základe simulovanej skúsenosti)
  - plánovanie a učenie: kombinácia dvoch ciest
    - politika - skúsenosť - hodnotová funkcia (priame RL)
    - politika - skúsenosť - model - hodnotová funkcia (nepriame RL)

Algoritmus Dyna-Q

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:

(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \varepsilon$-greedy(S,Q)
(c) Take action A; observe resultant reward, R, and state, S'
(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Loop repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

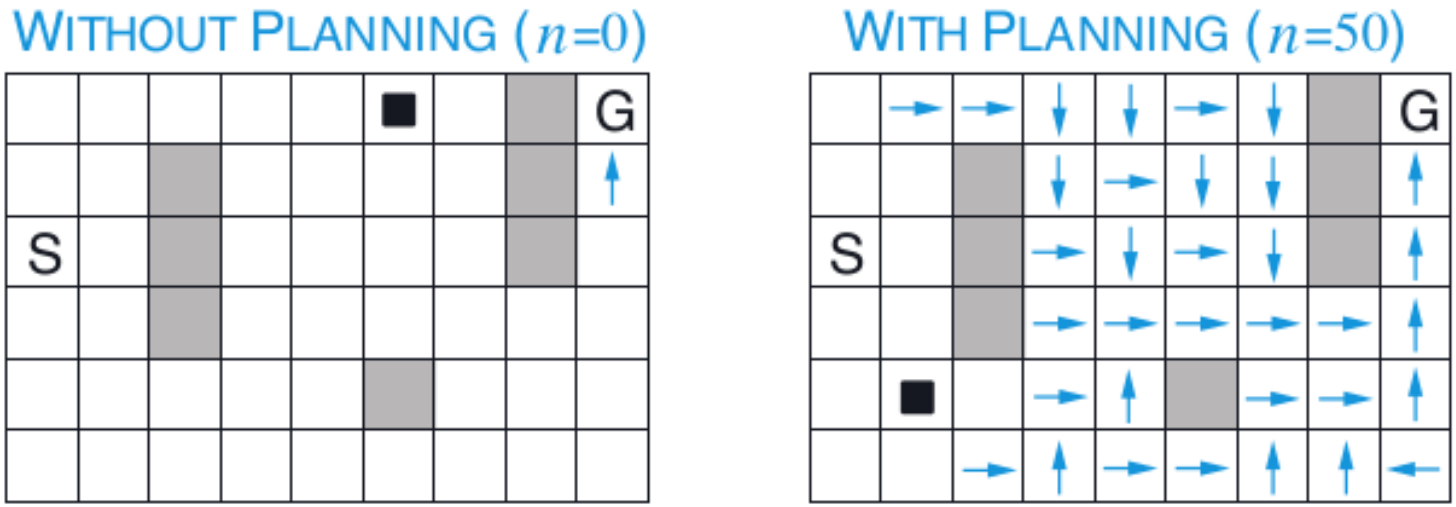
©*Sutton-Barto: Reinforcement Learning, 2nd ed., 2018*

Dyna-Q

- Dátové štruktúry
  - hodnotová funkcia $q(s, a) \in \mathcal{R}$ - tabuľková reprezentácia
  - deterministický model $m(s, a) = (r, s')$
    - na začiatku prázdny
    - učený z reálnej skúsenosti
    - umožňuje simulovanú skúsenosť
- Učenie (krok d)
  - 1-krokový tabuľkový Q-learning z reálnej skúsenosti
- Plánovanie (krok f)
  - 1-krokový tabuľkový Q-learning zo simulovanej skúsenosti (s náhodným vzorkovaním dát vložených do modelu)
- Učenie modelu (krok e)
  - pamätanie reálnej skúsenosti
- Nepriame RL umožní lepšie využitie získanej skúsenosti (opakované prehrávanie)
  - potrebných menej interakcií s prostredím

Pridaná hodnota plánovania

- Znázornenie vytvorenej politiky v polovici druhej epizódy
- Odmena: 1 pre cieľové pole, inak 0



©*Sutton-Barto: Reinforcement Learning, 2nd ed., 2018*

- V oboch prípadoch na konci 1.epizódy sa naučí iba jeden stav
- Bez plánovania počas druhej epizódy k učeniu ešte nedošlo
- Počas začiatku druhej epizódy sa plánovaním hodnota z predposledného stavu spätne šíri na predchádzajúce stavy (agent sa zatiaľ túla v blízkosti štartu)

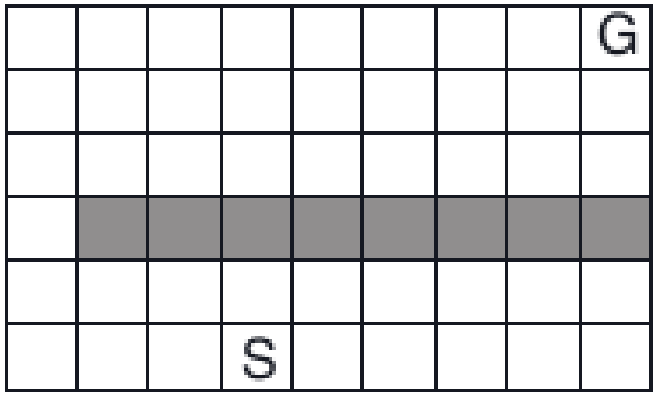
Korektnosť modelu

- Ak je model nekorektný, plánovanie pravdepodobne bude mať za následok suboptimálnu politiku
- Model sa môže stať nekorektným
  - prostredie je stochastické a skúsenosť s ním poskytla iba limitované množstvo vzoriek
  - model bol naučený použitím aproximácie hodnotových funkcií s nedokonalým zovšeobecnením
  - prostredie sa zmenilo a zmena ešte nebola pozorovaná skúsenosťou s prostredím
- Explorácia vs exploatácia (v kontexte plánovania)
  - explorácia - skúšanie akcií, ktoré vylepšia model
  - exploatácia - vyberanie akcií spôsobom, ktorý je optimálnym podľa modelu
  - keďže chceme čo najmenej používať reálnu skúsenosť a čo najviac simulovanú, explorácia v interakcii s prostredím má menej miesta

Explorácia v kontexte plánovania

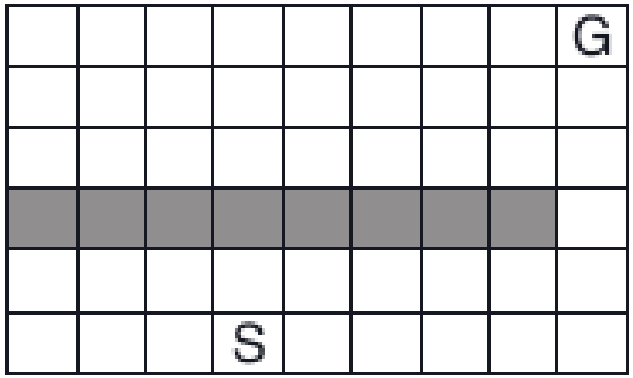
- Explorácia pri plánovaní môže byť podporovaná pomocou heuristík
- Algoritmus Dyna-Q+
  - čím dlhšie sa nejaká akcia nepoužila v reálnom prostredí, tým je väčšia šanca že údaj v modeli už nie je korektný
    - ak akcia nebola skúšaná τ krokov, tak odmena za jej použitie bude $r + \kappa\sqrt{\tau}$ pre malé κ
    - zvyšovanie odmeny v simulovanej skúsenosti pre pár (s, a) vedie k nadhodnoteniu $q(s, a)$
    - nadhodnotenie podnecuje agenta aby (s, a) zvolil a opätovne vyskúšal v reálnom prostredí
  - uvažované sú v modeli aj akcie, ktoré v nejakom stave neboli nikdy vyskúšané v prostredí
    - takéto akcie sú modelované tak, že vedú späť na daný stav s nulovou odmenou
    - ak v stave s bola reálne použitá iba akcia a, tak všetky ostatné nepoužité akcie b získajú $q(s, b) = \gamma q(s, a)$

Preučenie politiky pri zmene prostredia

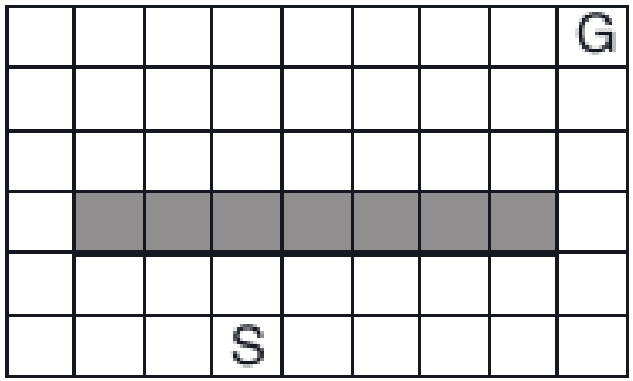


- Iniciálne sa dokáže naučiť najkratšiu cestu
- Po nejakej dobe dôjde k zmene prostredia
- Schopnosť preučiť politiku závisí na type zmeny

Zmena prostredia môže nastať dvojakým spôsobom:



Musí reagovať na zmenu. Dôjde k preučeniu politiky.



Nie je nútený reagovať na zmenu. Menšia šanca na zmenu politiky.

Stratégia vzorkovania modelu

- Náhodné vzorkovanie
  - existuje možnosť vybrať vzorku, ktorá nespôsobí žiadnu zmenu hodnotovej funkcie
- Prioritné deterministické vzorkovanie
  - výber iba tých vzoriek $(S_t, A_t, R_{t+1}, S_{t+1})$ z modelu, ktoré môžu meniť hodnotovú funkciu $q(S_t, A_t)$
    - pre zmenu $q(S_t, A_t)$ muselo dôjsť k zmene $q(S_{t+1}, A_{t+1})$
    - ak sa zmenila hodnota $q(S_{t+1}, A_{t+1})$, tak má zmysel aktualizovať všetky také páry (S_t, A_t), ktoré vedú na S_{t+1}
  - nie všetky zmeny hodnôt funkcie q sú rovnako užitočné
    - čím je zmena väčšia, tým skôr vzorku vybrať
    - má zmysel požadovať minimálnu veľkosť zmeny (obmedzenie počtu čakajúcich zmien)
  - realizácia ako prioritná fronta
    - fronta obsahuje páry (s, a) čakajúce na vzorkovanie z modelu
    - prioritou páru (s, a) je očakávaná veľkosť zmeny $q(s, a)$

Prioritné vzorkovanie modelu

Prioritized sweeping for a deterministic environment

Initialize $Q(s,a)$, $Model(s,a)$, for all s, a, and $PQueue$ to empty
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow policy(S,Q)$
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Model(S,A) \leftarrow R, S'$
- (e) $P \leftarrow |R + \gamma \max_a Q(S',a) - Q(S,A)|$.
- (f) if $P > \theta$, then insert S, A into $PQueue$ with priority P
- (g) Loop repeat n times, while $PQueue$ is not empty:
  - $S, A \leftarrow first(PQueue)$
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
  - Loop for all $\bar{S}, \bar{A}$ predicted to lead to S:
    - $\bar{R} \leftarrow$ predicted reward for $\bar{S}, \bar{A}, S$
    - $P \leftarrow |\bar{R} + \gamma \max_a Q(S,a) - Q(\bar{S},\bar{A})|$.
    - if $P > \theta$ then insert $\bar{S}, \bar{A}$ into $PQueue$ with priority P

©*Sutton-Barto: Reinforcement Learning, 2nd ed., 2018*

Učenie modelu v NAF (spojité S aj A)

- Dve prehrávacie pamäti
  - pre reálne $(S_t, A_t, R_{t+1}, S_{t+1})$ získané zo skúsenosti
  - pre fiktívne $(S_t, A_t, R_{t+1}, S_{t+1})$ generované modelom
  - dávka pre aktualizáciu siete vyberaná kombinovane
- Model $p(\bar{x}_{t+1}|\bar{x}_t, \bar{a}_t) = N(\bar{f}_x\bar{x}_t + \bar{f}_a\bar{a}_t, F)$
  - parametre modelu sa odhadujú vždy po určitom počte prechodov (po zaplnení zásobníka príkladov)
  - do zásobníka sa vkladajú aktuálne príklady zo skúsenosti
  - po aktualizácii sa zásobník príkladov plní nanovo
  - nie globálny (všetky stavy a akcie) ale iba lokálny model
- Plánovanie (periodicky po určenom počte krokov)
  - z reálnych prechodov, na základe ktorých bol naposledy učený model, sa vyberie náhodná vzorka
  - z každej vzorky model vygeneruje sekvenciu dlhú niekoľko krokov (ako alternatívnu skúsenosť)
  - kroky sekvencie sa vložia do fiktívnej prehrávacej pamäti