Aproximácia hodnotových funkcií
(Strojové učenie II)

M. Mach

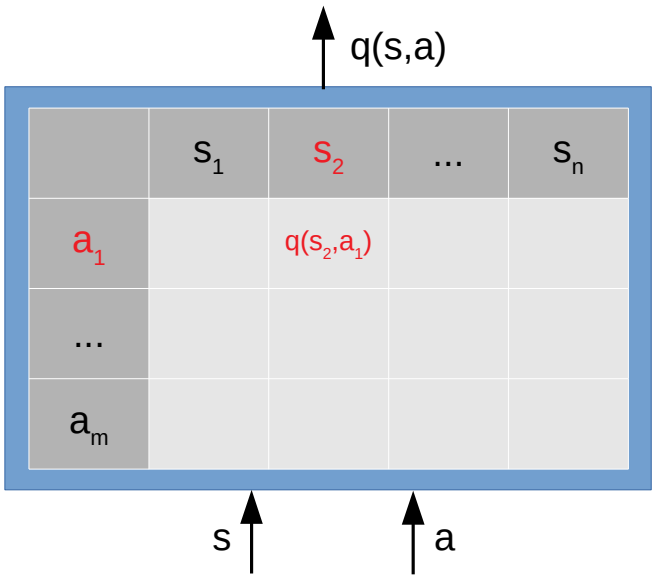
Katedra kybernetiky a umelej inteligencie, FEI, TUKE

február 2021 - marec 2024

Problém škálovania

- Konečný MDP
- Najčastejším problémom škálovania je škálovanie počtu stavov
  - konečný počet stavov
    - Backgammon: 10^{20} stavov
    - Go: 10^{170} stavov
  - nekonečný počet stavov
    - spojité problémy - diskretizácia
      (jemnejšie rozlišovaníe $\rightarrow$ veľký počet stavov)

Presná reprezentácia hodnotovej funkcie



- Stav
  - atomický (tabuľka $1+1$ dimenzií)
  - vektor n príznakov (tabuľka $n+1$ dimenzií)
- Každý stav / pár (stav,akcia) má jedinečnú reprezentáciu
  - s rastúcim počtom stavov či akcií sa tabuľka zväčšuje

- Výhoda
  - rýchle určenie hodnoty = vyhľadanie príslušného údaja v tabuľke
- Problémy
  - príliš veľa miesta v pamäti
  - nutnosť uvažovať osobitne každý stav / pár (stav,akcia)

Aproximátor - parametrický

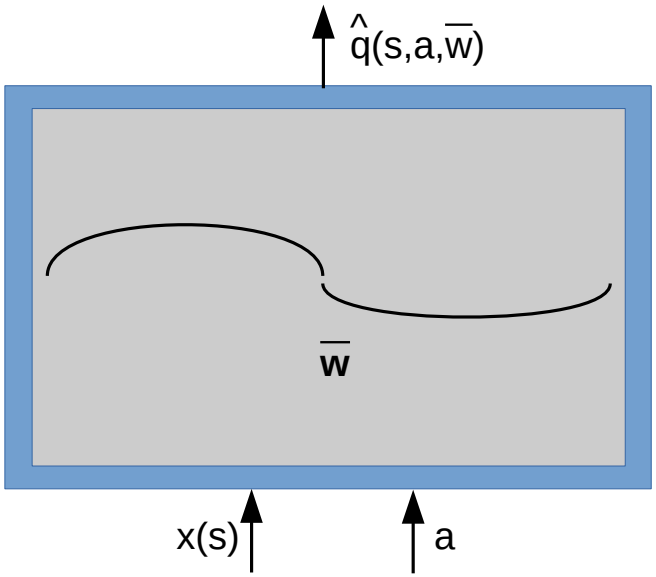
- Parametrický aproximátor $\hat{v}(s, \bar{w})$ má tvar vhodne zvolenej parametrickej funkcie
  - funkcia je definovaná vektorom numerických parametrov
  - počet váh je oveľa menší ako počet stavov
  - lineárny aproximátor
    - lineárna kombinácia príznakov (parametre = váhy)
  - nelineárny aproximátor
    - umelá neurónová sieť (parametre = váhy)
    - rozhodovací strom (parametre = split pointy)
- Aktualizácia hodnotovej funkcie pre nejaký stav
  - realizovaná ako zmena váh aproximačnej funkcie
  - ovplyvnenie viacerých stavov naraz (zmena jednej váhy ovplyvní hodnotu viacerých stavov)
- Po aktualizácii sa dáta zahodia

Aproximátor - neparametrický

- Pamäťový (neparametrický) aproximátor $\hat{v}(s, \mathcal{M})$ má tvar množiny príkladov
  - $\hat{v}(s, \mathcal{M}) = \sum_{s' \in \mathcal{M}} k(s, s') v(s')$
    - metóda najbližšieho suseda (kernel 0 alebo 1)
    - metóda váženého priemeru (kernel váha z $< 0, 1 >$)

    kernelová funkcia udáva relevanciu s' voči s na základe vzdialenosti (či podobnosti)
- Dáta sa nezahadzujú, ukladajú sa v pamäti
  - presnosť sa zvyšuje so zväčšovaním množstva dát
  - množstvo dát ovplyvňuje rýchlosť vyhľadávania v pamäti (hľadanie susedných/blízkych stavov)
  - stratégie updatu pri naplnení pamäti
- Učenie je odložené až do doby, keď má byť aproximátor použitý
- Robí lokálnu aproximáciu

Parametrická aproximácia funkcie



$$\hat{v}(s, \bar{w}) \approx v_\pi(s)$$
$$\hat{q}(s, a, \bar{w}) \approx q_\pi(s, a)$$

- Namiesto enumeračnej tabuľky použitý aproximátor
- Stav je reprezentovaný príznakmi
- Namiesto hodnôt v/q sa učí súbor váh $\bar{w}$
  - váh je (logaritmicky) menej ako počet stavov
  - hodnoty v/q sa odvodzujú výpočtom

- Zovšeobecnenie skúmaných stavov na neskúmané
  - nie je nutné skúmať každý pár (stav,akcia) osobitne
- Poskytuje iba približné odhady a nie presné hodnoty

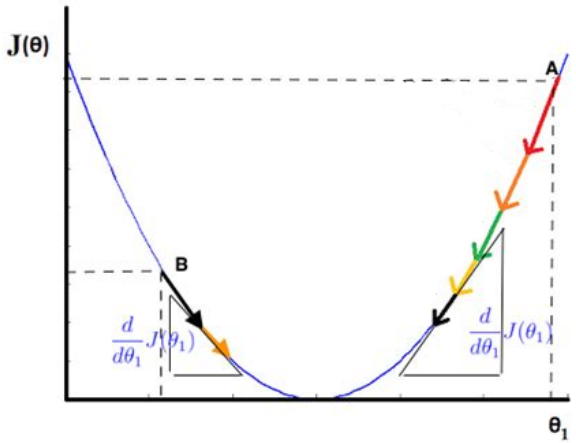
Aproximačný rámec (1)

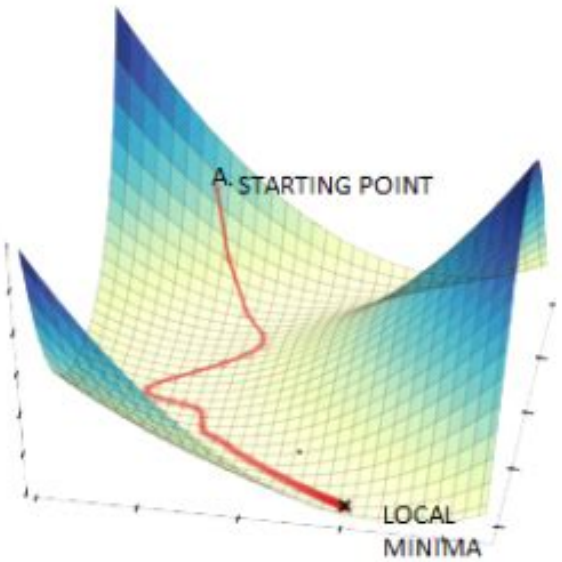
- Cieľová hodnota: $v(S_t) \longmapsto U_t$
  - Dynamické programovanie:
    $$v(S_t) \longmapsto E_\pi[R_{t+1} + \gamma \hat{v}(S_{t+1}, \bar{w}_t) | S_t = s]$$
  - Monte Carlo: $v(S_t) \longmapsto G_t$
  - Temporal-difference: $v(S_t) \longmapsto R_{t+1} + \gamma \hat{v}(S_{t+1}, \bar{w}_t)$
- Individuálna aktualizácia - posun aktuálnej hodnoty k cieľovej hodnote ($\alpha \in (0, 1>$)
  $$\begin{aligned} v_{t+1}(S_t) &= v_t(S_t) + \alpha(U_t - v_t(S_t)) \\ &= (1 - \alpha) v_t(S_t) + \alpha U_t \end{aligned}$$
- Nie je možné robiť individuálnu aktualizáciu ako zmenu jednej hodnoty v tabuľke
- Aktualizácia hodnoty stavu musí byť urobená ako aktualizácia aproximátora
  - zmenou váhového vektora
  - aktualizácia zasiahne súčasne aj iné stavy (generalizácia)

Aproximačný rámec (2)

- Úloha riešená kontrolovaným učením
- Aktualizačná metóda musí podporovať
  - inkrementálne učenie
    - skúsenosť je získavaná postupne interakciou s prostredím
    - snaha učiť počas získavania (TD)
  - nestacionárne príklady (pri on-policy)
    - pri vyhodnocovaní sa politika nemení, avšak pri učení politiky sa mení permanentne
    - učenie hodnotovej funkcie za neustálej zmeny politiky (zmena politiky mení hodnoty hodnotovej funkcie)
- Požadovaný posun sa vyjadrí ako príklad žiadaného vstupno-výstupného chovania aproximátora

Gradientová minimalizácia





- $J(\bar{w})$ je diferencovateľná funkcia parametra $\bar{w}$
- Gradient
$$\nabla J(\bar{w}) = \left(\frac{\partial J(\bar{w})}{\partial w_1}, \ldots, \frac{\partial J(\bar{w})}{\partial w_n} \right)^T$$
- Iteračné hľadanie lokálneho minima funkcie $J(\bar{w})$
- Aktualizácia parametra $\bar{w}$ v (opačnom) smere gradientu
$$\Delta \bar{w} = -\alpha \nabla J(\bar{w})$$
- Metóda GD (Gradient descend)

Aktualizácia aproximátora

- Predpokladajme, že chceme aktualizáciu v zmysle $\hat{v}(S_t, \bar{w}_t) \longmapsto U_t$
- Chyba aproximátora pre stav S_t je $\frac{1}{2}(U_t - \hat{v}(S_t, \bar{w}_t))^2$
- Aktualizácia váhového vektora

$$\begin{aligned} \bar{w}_{t+1} &= \bar{w}_t - \alpha \nabla \left(\frac{1}{2}(U_t - \hat{v}(S_t, \bar{w}_t))^2 \right) \\ &= \bar{w}_t + \alpha \left(U_t - \hat{v}(S_t, \bar{w}_t)\right) \nabla \hat{v}(S_t, \bar{w}_t) \end{aligned}$$

- Použitie metódy SGD (Stochastic GD)
  - aktualizácia robená iba podľa jedného (často náhodne vybraného) príkladu, nie viacerých naraz
  - príklad získavaný z interakcie s prostredím (použitím metód MC alebo TD)

Algoritmus TD odhadu v_π

Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^d \to \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$
Algorithm parameter: step size $\alpha > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose $A \sim \pi(\cdot|S)$
    Take action A, observe R, S'
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})\big]\nabla\hat{v}(S,\mathbf{w})$
    $S \leftarrow S'$
  until S is terminal

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

Konvergencia aproximátora

- Neminimalizujeme chybovú funkciu
  - iba malý posun smerom k minimu chybovej funkcie
    - minimalizácia chyby pre nejaký stav by znamenala zväčšenie chyby pre iné stavy
  - potrebné pre vybalansovanie chýb pre rôzne stavy
- Cieľová hodnota U_t pre stav S_t nie je správna hodnota $v_\pi(S_T)$ ale iba jej náhodný odhad
  - $E[U_t|S_t = s] = v_\pi(S_t)$ (nevychýlený odhad - prípad MC)
    - tak $\bar{w}_t$ konverguje k lokálnemu optimu ak α sa postupne zmenšuje tak, že platia vzťahy $\sum_{n=1}^{\infty} \alpha_n = \infty$ a $\sum_{n=1}^{\infty} \alpha_n^2 < \infty$
  - ak U_t závisí na $\bar{w}$ (lebo závisí na odhade $\hat{v}(S_t, \bar{w})$ čo je prípad DP, TD), tak konvergencia nie je garantovaná
    - neuvažuje sa vplyv zmeny váhového vektora na U_t (časť gradientu sa neuvažuje - pri výpočte gradientu bolo U_t považované za konštantu)
    - semi-gradientové metódy - menej robustná konvergencia

Lineárny aproximátor

- Aproximátor má tvar lineárnej kombinácie príznakov reprezentujúcich stav

$$\hat{v}(s, \bar{w}) = \bar{w}^T \bar{x}(s) = \sum_{i=1}^{n} w_i x_i(s)$$

  kde $\bar{x}(s)$ je príznakový vektor stavu s
- Gradient s ohľadom na $\bar{w}$ má jednoduchý tvar

$$\nabla \hat{v}(s, \bar{w}) = \bar{x}(s)$$

- Lineárna funkcia nemá lokálne extrémy
  - ak konverguje tak ku globálnemu optimu
- Aktualizácia parametrov podľa

$$\bar{w}_{t+1} = \bar{w}_t + \alpha \left(U_t - \bar{w}^T \bar{x}(S_t) \right) \bar{x}(S_t)$$

TD konvergencia lineárneho aproximátora

- Konvergencia TD

$$\begin{aligned}
\bar{w}_{t+1} &= \bar{w}_t + \alpha \left(R_{t+1} + \gamma \bar{w}_t^T \bar{x}(S_{t+1}) - \bar{w}_t^T \bar{x}(S_t) \right) \bar{x}(S_t) \\
&= \bar{w}_t + \alpha \left(R_{t+1} \bar{x}(S_t) - \bar{x}(S_t)(\bar{x}(S_t) - \gamma \bar{x}(S_{t+1}))^T \bar{w}_t \right) \\
E[\bar{w}_{t+1} | \bar{w}_t] &= \bar{w}_t + \alpha (E[R_{t+1} \bar{x}(S_t)] - \\
&\quad E[\bar{x}(S_t)(\bar{x}(S_t) - \gamma \bar{x}(S_{t+1}))^T] \bar{w}_t)
\end{aligned}$$

V ustálenom stave sa váhový vektor už nemení:

$$\begin{aligned}
E[\bar{w}_{t+1} | \bar{w}_t] &= \bar{w}_t \\
0 &= E[R_{t+1} \bar{x}(S_t)] - E[\bar{x}(S_t)(\bar{x}(S_t) - \gamma \bar{x}(S_{t+1}))^T] \bar{w}_{TD} \\
\bar{w}_{TD} &= \left(E[\bar{x}(S_t)(\bar{x}(S_t) - \gamma \bar{x}(S_{t+1}))^T] \right)^{-1} E[R_{t+1} \bar{x}(S_t)]
\end{aligned}$$

kde $\bar{w}_{TD}$ je fixed point

Epizodické učenie politiky

- Aproximovaná funkcia $\hat{q}(S_t, A_t, \bar{w})$
  - aproximátor $x(s, a) \rightarrow \hat{q}(s, a, \bar{w})$ na základe $\bar{w}$
- Aktualizácia v zmysle $\hat{q}(S_t, A_t, \bar{w}) \longmapsto U_t$
  ($U_t = G_t$ pre MC, $U_t = R_{t+1} + \gamma \hat{v}(S_t, \bar{w})$ pre TD)
- Chyba aproximátora pre dvojicu S_t, A_t je
  $\frac{1}{2}(U_t - \hat{q}(S_t, A_t, \bar{w}_t))^2$
- Aktualizácia váhového vektora
  $$\bar{w}_{t+1} = \bar{w}_t + \alpha \left(U_t - \hat{q}(S_t, A_t, \bar{w}_t)\right) \nabla \hat{q}(S_t, A_t, \bar{w}_t)$$
- Aktualizácia váhového vektora pre Sarsu
  $$\begin{aligned} \bar{w}_{t+1} \quad = \quad & \bar{w}_t + \alpha(R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}, \bar{w}_t) \\ & - \hat{q}(S_t, A_t, \bar{w}_t)) \nabla \hat{q}(S_t, A_t, \bar{w}_t) \end{aligned}$$
- Zlepšenie politiky $A_t^* = \arg\max_a \hat{q}(S_t, a, \bar{w}_{t-1})$

Algoritmus semi-gradient Sarsa

Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  $S, A \leftarrow$ initial state and action of episode (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    If S' is terminal:
      $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
      Go to next episode
    Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

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

Kontinuálne učenie politiky

- Pri tabuľkovej reprezentácii priemerná odmena pre každý pár (s,a) bola počítaná osobitne
- Použitie aproximácie hodnotovej funkcie spôsobuje
  - zlepšenie pre nejaký stav môže mať za následok zhoršenie pre iný stav
  - teorém zlepšovania politiky neplatí
  - zmena politiky zlepšujúca diskontovanú hodnotu nejakého stavu negarantuje zlepšenie politiky ako celku $\rightarrow$ problém s diskontovaním
- Diskontný prístup (založený na γ) sa nahrádza prístupom založeným na priemernej odmene
  - bezprostredná odmena je považovaná za rovnako dôležitú ako vzdialená odmena
  - zotriedenie politík je rovnaké podľa priemernej diskontovanej odmeny aj podľa priemernej odmeny

Priemerná odmena

- Priemerná odmena pri použití politiky π

$$
\begin{aligned}
r(\pi) &= \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} E[R_t | S_0, A_{0:t-1} \sim \pi] \\
&= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_r r \sum_{s'} p(s', r|s, a) \\
\mu_\pi(s) &= \lim_{t\to\infty} P[S_s = s | A_{0:t-1} \sim \pi]
\end{aligned}
$$

- Predpokladá sa ergodický MDP
  - z ľubovoľného stavu možno prejsť do ľubovoľného stavu (nemusí bezprostredne v jednom kroku)
  - μ_π je stacionárna (voľba S_0 má iba dočasný efekt)
- Praktické použitie
  - $r(\pi)$ reprezentuje kvalitu politiky π
  - zotriedenie politík podľa dosiahnutej $r(\pi)$
  - za optimálnu politiku bude považovaná tá, ktorá dosahuje maximálne $r(\pi)$

TD pre kontinuálne úlohy

- Diferenčná odmena $G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$
- Diferenčné hodnotové funkcie
  - $v_\pi(s) = E_\pi[G_t|S_t = s], \;\; q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]$
  - pre diferenčné hodnotové funkcie tiež platia Bellmanove rovnice očakávania a optimality
- Diferenčná forma TD chyby

$$\begin{aligned} \delta_t &= \left(R_{t+1} - \overline{R_t} + \hat{v}(S_{t+1}, \bar{w}_t)\right) - \hat{v}(S_t, \bar{w}_t) \\ \delta_t &= \left(R_{t+1} - \overline{R_t} + \hat{q}(S_{t+1}, A_{t+1}, \bar{w}_t)\right) - \hat{q}(S_t, A_t, \bar{w}_t) \end{aligned}$$

  kde $\overline{R_t}$ je odhad $r(\pi)$ v čase t
- Aktualizácia parametrického vektora aproximátora

$$\bar{w}_{t+1} = \bar{w}_t + \alpha\delta_t \nabla \hat{q}(S_t, A_t, \bar{w}_t)$$

Algoritmus Diferenciálna Sarsa

Differential semi-gradient Sarsa for estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step sizes $\alpha, \beta > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)
Initialize average reward estimate $\bar{R} \in \mathbb{R}$ arbitrarily (e.g., $\bar{R} = 0$)

Initialize state S, and action A
Loop for each step:
  Take action A, observe R, S'
  Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
  $\delta \leftarrow R - \bar{R} + \hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})$
  $\bar{R} \leftarrow \bar{R} + \beta\delta$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S, A, \mathbf{w})$
  $S \leftarrow S'$
  $A \leftarrow A'$

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

Triáda nestability

- Zdroje nestability a možnej divergencie
  - Aproximácia hodnotovej funkcie
    - zovšeobecnenie v priestore stavov (neuvažovanie stavov oddelene, ich vzájomné ovlyvňovanie)
    - učenie jedného stavu ovplyvní aj hodnoty iných stavov (zdieľanie príznakov)
  - Bootstrapping
    - aktualizácia hodnôt založená na aktuálnom stave týchto hodnôt
    - prítomné v DP a TD
  - Off-policy učenie
    - rozdielnosť medzi cieľovou a exploračnou politikou
- Kombinácia dvoch je zvládnuteľná (tabuľkový q-learnig), kombinácia všetkých robí problém

Dávkový vs inkrementálny prístup

- Minimalizácia chyby pre jeden stav alebo pár (stav,akcia) - inkrementálny prístup
  - SGD
  - oprava pre jeden stav znamená pokazenie pre iný stav
  - preto pohyb iba malý kúsok v smere opravy
  - skúsenosť sa po jednorazovom použití zahodí
- Minimalizácia chyby pre viac stavov alebo párov (stav,akcia) naraz - dávkový prístup
  - LS (least squares) algoritmy
  - SGD + opakované prehrávanie
    - skúsenosť sa nezahadzuje ale sa pamätá
    - z pamätanej skúsenosti sa vyberá vzorka (dávka)
    - skúsenosť môže byť opakovane vzorkovaná do rôznych dávok