I007 Osnove umjetne inteligencije

Tema: Učenje s podrškom

28. 4. 2021.

predavač: Darija Marković

asistent: Darija Marković

1 Učenje s podrškom

2 Pasivno

3 Aktivno

4 Aproksimativno Q-učenje

5 Pretraživanje strategija

Učenje s podrškom

Osnovna ideja

- agent dobiva povratnu informaciju u obliku nagrade (ili kazne)
- agentova korisnost se definira pomoću funkcije nagrade
- treba naučiti kako se ponašati u svrhu maksimizacije nagrade
- učenje se bazira na uočenim (prikupljenim) primjerima (ishodima primjera)

- Kod učenja s podrškom pretpostavljamo da se radi o MPO, pri čemu nam je poznato:
  - skup stanja $s \in S$
  - skup akcija $a \in A$
- ne znamo funkciju prijelaza T, niti funkciju nagrade R
- i dalje želimo odrediti strategiju
- kako bi to mogli naučiti potrebno je isprobavati različite akcije u različitim stanjima

Učenje s podrškom

Osnovna ideja

- agent dobiva povratnu informaciju u obliku nagrade (ili kazne)
- agentova korisnost se definira pomoću funkcije nagrade
- treba naučiti kako se ponašati u svrhu maksimizacije nagrade
- učenje se bazira na uočenim (prikupljenim) primjerima (ishodima primjera)

- Kod učenja s podrškom pretpostavljamo da se radi o MPO, pri čemu nam je poznato:
  - skup stanja $s \in S$
  - skup akcija $a \in A$
- ne znamo funkciju prijelaza T, niti funkciju nagrade R
- i dalje želimo odrediti strategiju
- kako bi to mogli naučiti potrebno je isprobavati različite akcije u različitim stanjima

1 Učenje s podrškom

2 Pasivno

3 Aktivno

4 Aproksimativno Q-učenje

5 Pretraživanje strategija

Direktna procjena

Pojednostavljeni zadatak: procjena strategije

- zadana je fiksirana strategija
- cilj je naučiti vrijednosti stanja

- Ideja: usrednjenje svih uočenih vrijednosti primjera
  - ponaša se u skladu sa zadanom strategijom
  - bilježe se svi ishodi za svako stanje pri svakom izvođenju, te se izračuna nagrada
  - sve postignute nagrade se usrednje
- prednosti direktne procjene:
  - jednostavna, ne zahtjeva poznavanje T i R, s vremenom dobije dobre srednje vrijednosti
- nedostaci direktne procjene:
  - ne koristi informacije o povezanosti stanja, svako stanje se uči posebno i u pravilu zahtjeva puno vremena

Direktna procjena

Pojednostavljeni zadatak: procjena strategije

- zadana je fiksirana strategija
- cilj je naučiti vrijednosti stanja

- Ideja: usrednjenje svih uočenih vrijednosti primjera
  - ponaša se u skladu sa zadanom strategijom
  - bilježe se svi ishodi za svako stanje pri svakom izvođenju, te se izračuna nagrada
  - sve postignute nagrade se usrednje
- prednosti direktne procjene:
  - jednostavna, ne zahtjeva poznavanje T i R, s vremenom dobije dobre srednje vrijednosti
- nedostaci direktne procjene:
  - ne koristi informacije o povezanosti stanja, svako stanje se uči posebno i u pravilu zahtjeva puno vremena

Direktna procjena

Pojednostavljeni zadatak: procjena strategije

- zadana je fiksirana strategija
- cilj je naučiti vrijednosti stanja

- Ideja: usrednjenje svih uočenih vrijednosti primjera
  - ponaša se u skladu sa zadanom strategijom
  - bilježe se svi ishodi za svako stanje pri svakom izvođenju, te se izračuna nagrada
  - sve postignute nagrade se usrednje
- prednosti direktne procjene:
  - jednostavna, ne zahtjeva poznavanje T i R, s vremenom dobije dobre srednje vrijednosti
- nedostaci direktne procjene:
  - ne koristi informacije o povezanosti stanja, svako stanje se uči posebno i u pravilu zahtjeva puno vremena

Direktna procjena

Pojednostavljeni zadatak: procjena strategije

- zadana je fiksirana strategija
- cilj je naučiti vrijednosti stanja

- Ideja: usrednjenje svih uočenih vrijednosti primjera
  - ponaša se u skladu sa zadanom strategijom
  - bilježe se svi ishodi za svako stanje pri svakom izvođenju, te se izračuna nagrada
  - sve postignute nagrade se usrednje
- prednosti direktne procjene:
  - jednostavna, ne zahtjeva poznavanje T i R, s vremenom dobije dobre srednje vrijednosti
- nedostaci direktne procjene:
  - ne koristi informacije o povezanosti stanja, svako stanje se uči posebno i u pravilu zahtjeva puno vremena

Učenje modela

Osnovna ideja

- naučiti aproksimacijski model s obzirom na iskustvo (primjere)
- odrediti vrijednost kao da je naučeni model točan

- Korak 1.: naučiti empirijski model za MPO:
  - izbrojati ishode s' za svaki par (s, a)
  - normalizirati kako bi se dobila procjena za $\hat{T}(s, a, s')$
  - otkriti svaki $\hat{R}(s, a, s')$ iz danih primjera
- Korak 2.: Riješiti naučeni MPO
  - npr. koristeći procjenu strategije:

$$V^{\pi}_{k+1}(s) \leftarrow \sum_{s'} \hat{T}(s, \pi(s), s')[\hat{R}(s, \pi(s), s') + \gamma V^{\pi}_{k}(s')]$$

Učenje vremenskih razlika (Temporal Difference Learning - TDL)

- želimo naučiti iz svakog primjera, te ažuriramo vrijednost $V(s)$ nakon svakog novog primjera (s, a, s', r)
- vjerojatniji ishodi s' će češće doprinijeti ažuriranju
- uvodimo faktor učenja α

- TDL učenje vrijednosti (uz fiksiranu strategiju π):
  - novi primjer za $V(s)$: $R(s, \pi(s), s') + \gamma V^{\pi}(s')$
  - ažuriramo staru vrijednost $V^{\pi}(s)$

$$V^{\pi}(s) \leftarrow (1 - \alpha)V^{\pi}(s) + \alpha[R(s, \pi(s), s') + \gamma V^{\pi}(s')]$$

Učenje vremenskih razlika (Temporal Difference Learning - TDL)

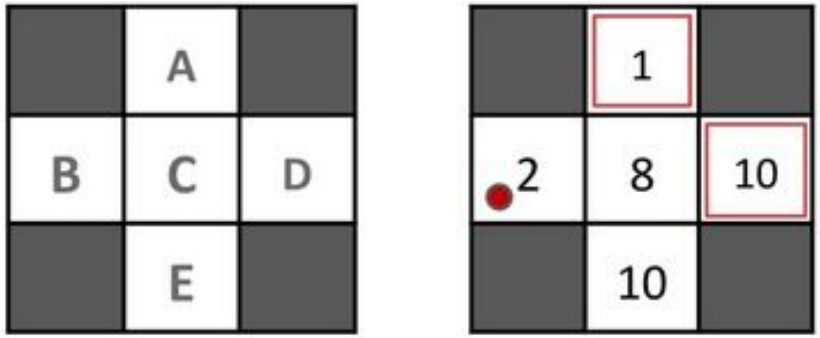
- Eksponencijalno pomični prosjek
  - ažuriranje je oblika $\bar{x}_n = (1-\alpha) \cdot \bar{x}_{n-1} + \alpha \cdot x_n$
  - na ovaj način novi primjeri imaju veću važnost:

$$\bar{x}_n = \frac{x_n + (1-\alpha) \cdot x_{n-1} + (1-\alpha)^2 x_{n-1} + \cdots}{1 + (1-\alpha) + (1-\alpha)^2 + \cdots}$$

  - zaboravljamo prošlost (udaljene stare vrijednosti su i onako bile krive)
- uz opadajući faktor učenja može se postići konvergencija prosjeka

Primjer 1. Učenje vremenskih razlika (TDL)

Zadana je sljedeća mreža s procijenjenim vrijednostima:



Nakon toga dobijemo sljedeći primjer: $B, \rightarrow, C, -2$.
Uz $\gamma = 1$ i faktor učenja $\alpha = 0.5$ odredite nove vrijednosti za procjenu strategije.

Problemi s učenjem vremenskih razlika

- TDL je bez modelski način procjene strategije koje oponaša Bellmanovo ažuriranje
- no nemoguće je iz dobivenih vrijednosti naučiti (novu) strategiju:

$$\pi(s) = \arg\max_a Q(s,a)$$

$$Q(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V(s')]$$

- ideja: naučiti q-vrijednosti, a ne vrijednosti stanja

Problemi s učenjem vremenskih razlika

- TDL je bez modelski način procjene strategije koje oponaša Bellmanovo ažuriranje
- no nemoguće je iz dobivenih vrijednosti naučiti (novu) strategiju:

$$\pi(s) = \arg\max_a Q(s,a)$$

$$Q(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V(s')]$$

- ideja: naučiti q-vrijednosti, a ne vrijednosti stanja

Problemi s učenjem vremenskih razlika

- TDL je bez modelski način procjene strategije koje oponaša Bellmanovo ažuriranje
- no nemoguće je iz dobivenih vrijednosti naučiti (novu) strategiju:

$$\pi(s) = \arg\max_a Q(s,a)$$

$$Q(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V(s')]$$

- ideja: naučiti q-vrijednosti, a ne vrijednosti stanja

1 Učenje s podrškom

2 Pasivno

3 Aktivno

4 Aproksimativno Q-učenje

5 Pretraživanje strategija

Aktivno učenje s podrškom

- potpuno učenje s podrškom: optimalna strategija
  - ne znamo funkciju prijelaza T
  - ne znamo funkciju nagrade R
  - sami odabiremo akcije
  - cilj: naučiti optimalnu strategiju
- u ovom slučaju
  - agent radi izbor
  - temeljni kompromis: istraživanje nasuprot iskorištavanja
  - ovdje se ne radi o planiranju, agent izvršava akciju u okolišu, proučava novo stanje s' i nagradu r

Q-učenje (Q-Learning)

- Q-učenje: iteracija q-vrijednosti bazirana na primjerima

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

- TDL za učenje q-vrijednosti
  - dobijemo novi primjer (s,a,s',r)
  - uzmemo u obzir staru procjenu $Q(s,a)$
  - uzmemo u obzir procjenu za novi primjer: $R(s,a,s') + \gamma \max_{a'} Q(s',a')$
  - ažuriramo staru vrijednost $Q(s,a)$

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$$

Q-učenje (Q-Learning)

- Q-učenje: iteracija q-vrijednosti bazirana na primjerima

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

- TDL za učenje q-vrijednosti
  - dobijemo novi primjer (s,a,s',r)
  - uzmemo u obzir staru procjenu $Q(s,a)$
  - uzmemo u obzir procjenu za novi primjer: $R(s,a,s') + \gamma \max_{a'} Q(s',a')$
  - ažuriramo staru vrijednost $Q(s,a)$

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$$

Q-učenje (Q-Learning)

- Q-učenje konvergira optimalnoj strategiji - čak i ako se djeluje suboptimalno (uz pretpostavke da je sustav deterministički MPO, nagrade ograničene te da se svaki par stanje-akcija ponovi dovoljan broj puta)
- učenje bez strategije
- ograničenja
  - mora se dovoljno istraživati
  - s vremenom je potrebno faktor učenja smanjiti, no ne prebrzo
  - u limesu nije bitno na koji način su birane akcije

Istraživanje nasuprot iskorištavanja: Kako istraživati?

- Najjednostavnije: slučajne akcije ($\varepsilon-$greedy)
  - u svakom koraku s (malom) vjerojatnošću ε ponaša se slučajno, a s velikom vjerojatnošću $1-\varepsilon$ ponaša se u skladu s trenutnom strategijom
- Problemi sa slučajnim akcijama?
  - s vremenom se istraži cijeli prostor, no i dalje se (nepotrebno) poduzimaju slučajne akcije
  - jedno rješenje ovog problema je smanjivanje ε s vremenom
  - drugo rješenje: funkcija istraživanja - u obzir uzima procijenjenu vrijednost u i broj posjeta n i vraća optimističnu procjenu, npr. $f(u,n) = u + \frac{k}{n}$

Istraživanje nasuprot iskorištavanja: Kako istraživati?

- Najjednostavnije: slučajne akcije ($\varepsilon-$greedy)
  - u svakom koraku s (malom) vjerojatnošću ε ponaša se slučajno, a s velikom vjerojatnošću $1-\varepsilon$ ponaša se u skladu s trenutnom strategijom
- Problemi sa slučajnim akcijama?
  - s vremenom se istraži cijeli prostor, no i dalje se (nepotrebno) poduzimaju slučajne akcije
  - jedno rješenje ovog problema je smanjivanje ε s vremenom
  - drugo rješenje: funkcija istraživanja - u obzir uzima procijenjenu vrijednost u i broj posjeta n i vraća optimističnu procjenu, npr. $f(u,n) = u + \frac{k}{n}$

Žaljenje/Kajanje

- čak i kada se nauči optimalna strategija, u postupku učenja naprave se pogreške
- žaljenje je mjera ukupne cijene pogrešaka: razlika između očekivanih nagrada (uključujući suboptimalnost) i optimalnih nagrada
- minimiziranje žaljenja je više od učenja optimalnosti: ono zahtjeva optimalnost učenja
- npr. slučajno istraživanje i funkcija istraživanja završavaju s optimalnom strategijom, no slučajno istraživanje ima veće žaljenje

1 Učenje s podrškom

2 Pasivno

3 Aktivno

4 Aproksimativno Q-učenje

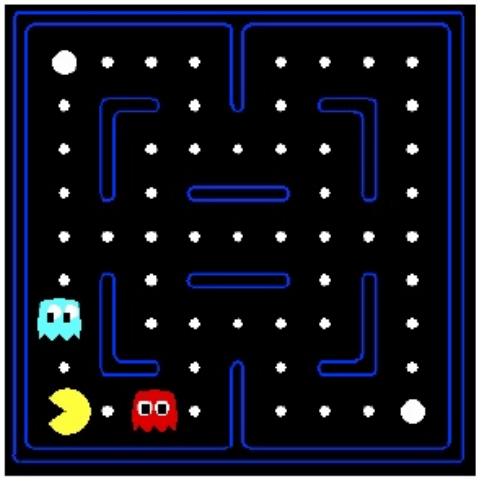
5 Pretraživanje strategija

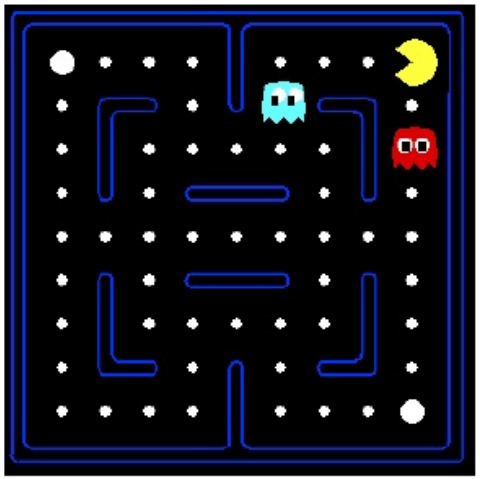
Generaliziranje po stanjima

- osnovno Q-učenje pamti tablicu svih q-vrijednosti
- u realnim situacijama nemoguće je naučiti svako pojedinačno stanje
  - postoji previše stanja koje sva treba posjetiti u treningu
  - postoji prevelik broj stanja da bi sve q-vrijednosti pohranili u memoriji
- umjesto toga želimo generalizirati:
  - naučiti o nekom malom broju stanja iz iskustva (treniranjem)
  - generalizirati iskustvo na nove, slične situacije
  - ovo je jedna od osnovnih ideja strojnog učenja

Primjer 2. Pacman

Pretpostavimo da se kroz iskustvo nauči da je prvo stanje prikazano na slici loše. U naivnom Q-učenju ne bi imali nikakve informacije o preostala dva prikazana stanja.





Reprezentacija na osnovu značajki

- rješenje: opisati stanje pomoću vektora značajki (svojstava)
  - značajke su funkcije koje stanju pridružuju realnu vrijednosti (često 0/1) koje opisuju bitna svojstva stanja
  - primjeri značajki (za igru Pacman)
    - udaljenost do najbližeg duha
    - udaljenost do najbliže točke
    - broj duhova
    - je li Pacman u tunelu (0/1)
  - također je moguće opisati q-stanje (s, a) sa značajkama

Linearna kombinacija značajki

- koristeći reprezentaciju pomoću značajki možemo zapisati q funkciju (ili funkciju vrijednosti) za svako stanje koristeći nekoliko težina:

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \cdots + w_n f_n(s)$$

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \cdots + w_n f_n(s,a)$$

- prednost: naše iskustvo je sažeto u nekoliko bitnih brojeva
- nedostatak: stanja mogu dijeliti značajke ali istovremeno imati vrlo različite vrijednosti

Aproksimativno Q-učenje

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \cdots + w_n f_n(s,a)$$

- Q-učenje s linearnom q funkcijom:

$$\text{primjer} = (s, a, s', r)$$

$$\text{razlika} = [r + \gamma \max_{a'} Q(s', a')] - Q(s,a)$$

  - kod Q-učenja: $Q(s,a) \leftarrow Q(s,a) + \alpha[\text{razlika}]$
  - kod aproksimativnog Q-učenja:

$$w_i \leftarrow w_i + \alpha[\text{razlika}] f_i(s,a)$$

- intuitivna interpretacija: podešavamo težine aktivnih značajki
- formalno opravdanje: metoda najmanjih kvadrata

Aproksimativno Q-učenje

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \cdots + w_n f_n(s,a)$$

- Q-učenje s linearnom q funkcijom:

$$\text{primjer} = (s, a, s', r)$$

$$\text{razlika} = [r + \gamma \max_{a'} Q(s', a')] - Q(s,a)$$

  - kod Q-učenja: $Q(s,a) \leftarrow Q(s,a) + \alpha[\text{razlika}]$
  - kod aproksimativnog Q-učenja:

$$w_i \leftarrow w_i + \alpha[\text{razlika}] f_i(s,a)$$

- intuitivna interpretacija: podešavamo težine aktivnih značajki
- formalno opravdanje: metoda najmanjih kvadrata

Aproksimativno Q-učenje

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \cdots + w_n f_n(s,a)$$

- Q-učenje s linearnom q funkcijom:

$$\text{primjer} = (s, a, s', r)$$

$$\text{razlika} = [r + \gamma \max_{a'} Q(s', a')] - Q(s,a)$$

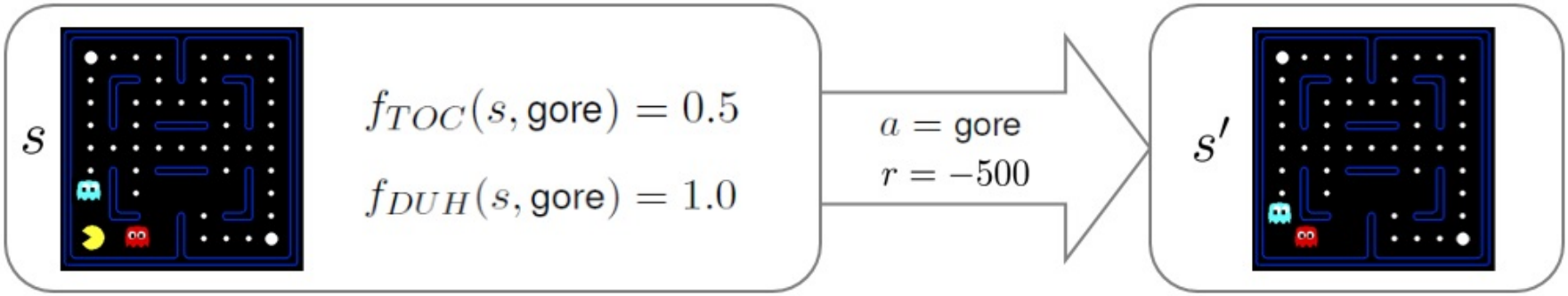
  - kod Q-učenja: $Q(s,a) \leftarrow Q(s,a) + \alpha[\text{razlika}]$
  - kod aproksimativnog Q-učenja:

$$w_i \leftarrow w_i + \alpha[\text{razlika}] f_i(s,a)$$

- intuitivna interpretacija: podešavamo težine aktivnih značajki
- formalno opravdanje: metoda najmanjih kvadrata

Primjer 3. Q-Pacman

$$Q(s,a) = 4.0 f_{TOC}(s,a) - 1.0 f_{DUH}(s,a)$$



$Q(s, \text{gore}) = +1$

$r + \gamma \max_{a'} Q(s', a') = -500 + 0$

$Q(s', \cdot) = 0$

$$\text{razlika} = -501 \implies \begin{aligned} w_{TOC} &\leftarrow 4.0 + \alpha[-501]0.5 \\ w_{DUH} &\leftarrow -1.0 + \alpha[-501]1.0 \end{aligned} \qquad \alpha = \frac{2}{501}$$

$$Q(s,a) = 3.0 f_{TOC}(s,a) - 3.0 f_{DUH}(s,a)$$

Optimizacija: najmanji kvadrati

- model je oblika

$$\hat{y}(x) = \sum_{k=1}^{n} w_k f_k(x)$$

- ukupna greška gleda se kao suma kvadrata svih odstupanja

$$\text{greška} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i \left(y_i - \sum_{k=1}^{n} w_k f_k(x_i) \right)^2$$

Optimizacija: najmanji kvadrati

- uz pretpostavku da smo imali samo jedno mjerenje, tj. samo jednu vrijednost za x imamo

$$\text{greška}(w) = \frac{1}{2}\left(y - \sum_{k=1}^{n} w_k f_k(x)\right)^2$$

$$\frac{\partial \text{greška}(w)}{\partial w_m} = -\left(y - \sum_{k=1}^{n} w_k f_k(x)\right) f_m(x)$$

$$w_m \leftarrow w_m + \alpha\left(y - \sum_{k=1}^{n} w_k f_k(x)\right) f_m(x)$$

- kod aproksimativnog Q-učenja izmjerena vrijednost jednaka je $r + \gamma \max_{a'} Q(s', a')$, a modelom predviđena vrijednost je $Q(s, a) = \sum_{k=1}^{n} w_k f_k(s, a)$

1 Učenje s podrškom

2 Pasivno

3 Aktivno

4 Aproksimativno Q-učenje

5 Pretraživanje strategija

- Problem: strategije bazirane na značajkama koje dobro rade često nisu one koje dobro aproksimiraju V/q vrijednosti
  - prioritet Q-učenja: što točnije odrediti q-vrijednost (modeliranje)
  - prioritet odabira akcije: točno poslagati q-vrijednosti (predviđanje)
- rješenje: naučiti strategiju koja maksimizira nagradu, a ne vrijednosti
- pretraživanje strategija: započeti s nekim prihvatljivim rješenjem (npr. Q-učenje) i onda ga poboljšati koristeći strategiju uspona na vrh po težinama značajki

- Problem: strategije bazirane na značajkama koje dobro rade često nisu one koje dobro aproksimiraju V/q vrijednosti
  - prioritet Q-učenja: što točnije odrediti q-vrijednost (modeliranje)
  - prioritet odabira akcije: točno poslagati q-vrijednosti (predviđanje)
- rješenje: naučiti strategiju koja maksimizira nagradu, a ne vrijednosti
- pretraživanje strategija: započeti s nekim prihvatljivim rješenjem (npr. Q-učenje) i onda ga poboljšati koristeći strategiju uspona na vrh po težinama značajki