Úvod do umělé inteligence (NAIL120)

8. cvičení

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Letní semestr 2023/24

Poslední změna 15. dubna 2024

6. domácí úkol: Lokalizace v čase

Zadání (zkráceno)

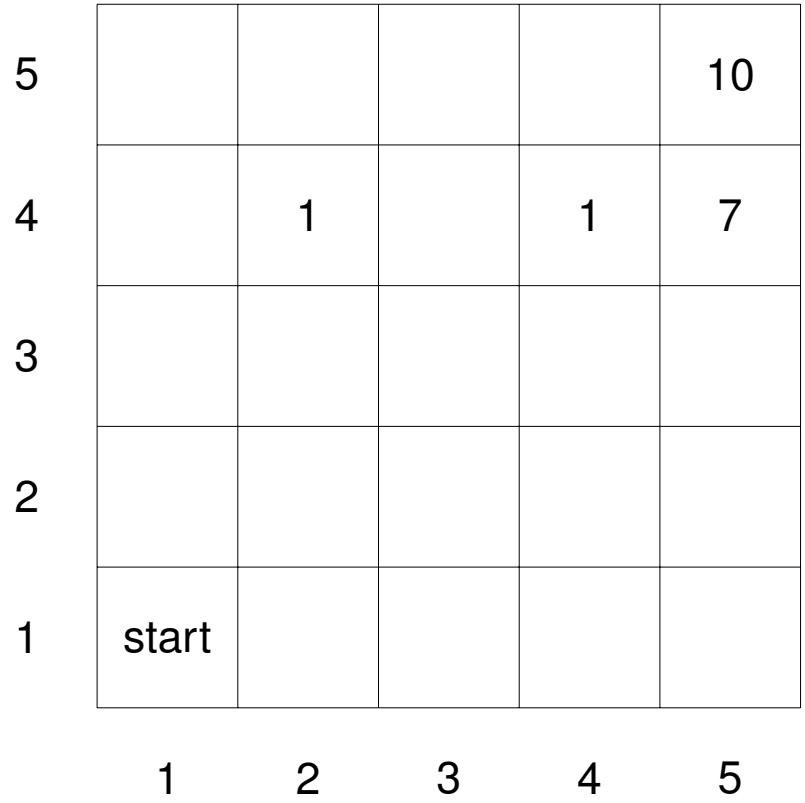
- Na Marsu přistane robot, který se má dostat na základnu
- Přistání není 100 % úspěšné, takže
  - nepřistál přímo na základně, ale musí k ní dojet
  - poškodily se mu navigační systémy a většina senzorů
- Naštěstí fungují motory, takže je schopen se přesně pohybovat ve 4 základních směrech a vždy ví, jaká je jeho relativní pozice vůči místu přistání
- Funguje mu binární čidlo, které náhodně vrací True/False na základě tmavosti/světlosti dané pozice
- Má uloženou mapu, ze které pro každou pozici dokáže určit stupeň šedi
- Cílem robota je dostat se na základnu
  - nesmí vyjet mimo mapu
  - má omezenou kapacity baterie (počet kroků)

Napište program, který v každém kroku dostane binární hodnotu z čidla, a určí směr, ve kterém se má robot posunout.

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)
- Užitková funkce $U(s)$ udává maximální očekávaný užitek ze stavu s do cílem
  - Maximalizujeme přes volby akcí ve všech stavech
  - Očekávaný přes náhodné přechody dány distribucemi $P(s'|s,a)$
- Cílem je vybírat akce maximalizující celkový užitek v cíli
- Předpokládáme Markovův proces, takže užitek $U(s)$ je stacionární
  - Užitek $U(s)$ ze stavu s nezávisí na způsobu, jak jsme se do s dostali
- Bellmanova rovnice z přednášky: $U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

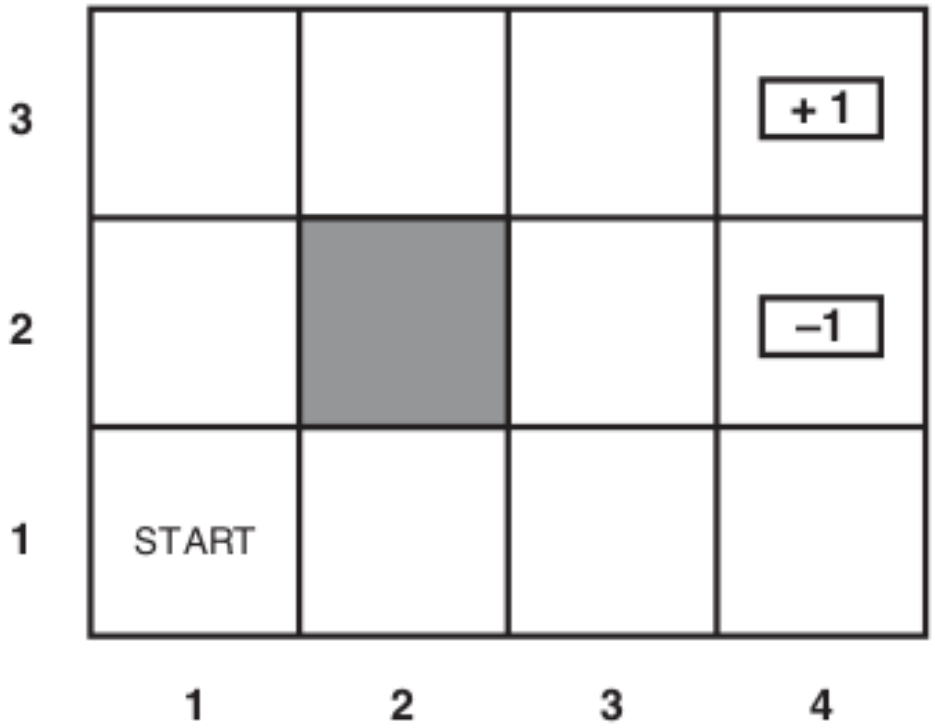
Robot z přednášky

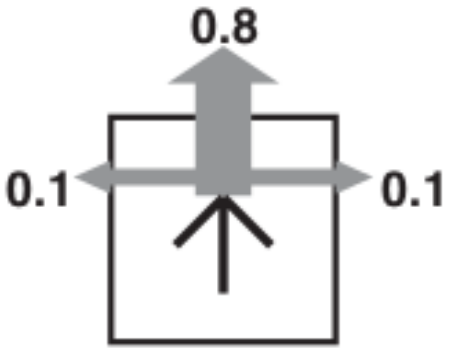


Zjednodušená varianta

- Robot začíná na pozici start a končí na vyznačených pozicích
- Vstupem koncová pole dostane uvedený počet bodů, jinde -0.1
- Robot se může vydat jen nahoru nebo doprava, ale zvolený přesun provede s pravděpodobností 0.8 a druhý s pravděpodobností 0.2
- Jak určit optimální strategii a získaný počet bodů?

Robot z přednášky





Obecná varianta

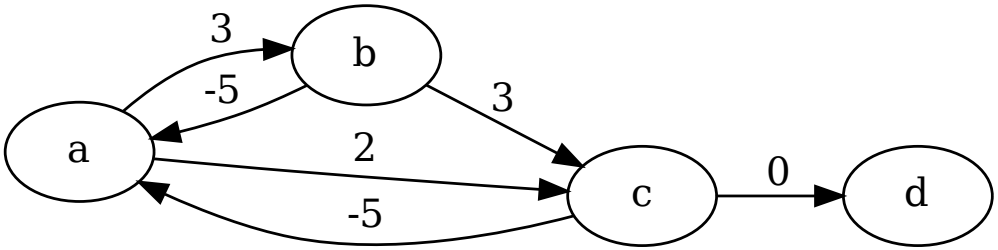
- Jak se postup změní, pokud se robot může pohybovat ve všech 4 směrech?
- Dokážete obecně popsat, kdy stačí použít zjednodušený postup?

Algoritmus iterování užitkové funkce

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

Jednoduchý příklad obecného postupu



Popis úlohy

- Robot začíná na pozici *a* a končí na pozici *d*
- Robot se přesune na zadaný vrchol s pravděpodobností 0.8 a jinak přejde po druhé hraně
- Odměna za přechod po hraně je dána v grafu
- Jak určit, kterou máme zadat u každého vrcholu, aby součet odměn v cíli byl maximální?

Algoritmus iterování akcí

function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

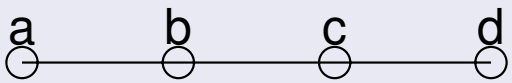
  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\ U[s']$ **then do**
        $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

Nejbezpečnější cesta v grafu

Jak najít očekávanou nejbezpečnější cestu v grafu?

- Známe pravděpodobnost d_{uv}, že se robot na hraně uv rozbije
- Známe pravděpodobnost $t_{u,v,w}$, že robot nacházející se ve vrcholu u mající pokyn jet do vrcholu v pojede do vrcholu w
- Samozřejmě platí $\sum_w p_{u,v,w} = 1$

Nejprve triviální příklad cesty z a do d



Obecný graf

- Sestavte Bellmanovu rovnici
- Popište hledání řešení této rovnice

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Otázka

Je nutné uvažovat $0 < \gamma < 1$? Proč?

Pevný bod

- $x \in M$ je pevným bodem funkce $f : M \to M$, jestliže $f(x) = x$.
- Hledaná užitková funkce je právě pevným bodem Bellmanovi rovnice.

Banachova věta o pevném bodě

- Funkce $f : M \to M$ je kontrakce, jestliže existuje $0 \leq q < 1$ takové, že pro všechna $x, y \in M$ platí $||f(x) - f(y)|| \leq q \cdot ||x - y||$.
- Každá kontrakce na kompaktní podmnožině $\mathbb{R}^n$ má pevný bod.

Brouwerova věta o pevném bodě

Každá spojitá funkce na konvexní kompaktní podmnožině $\mathbb{R}^n$ má pevný bod.

7. domácí úkol: Bellman update

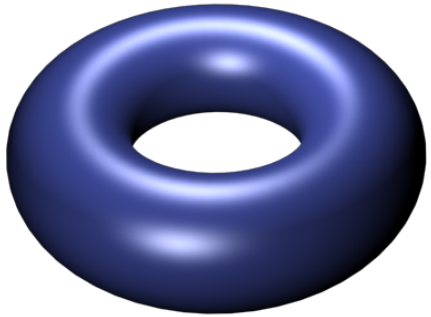
Zadání (zkráceno)

- Na Marsu přistane robot, který se má dostat na základnu
- Přistání není 100 % úspěšné, takže
  - nepřistál přímo na základně, ale musí k ní dojet
  - poškodili se motory a robot často jede jinam než řídící jednotka zadala
- Naštěstí funguje alespoň lokalizace, takže robot vždy ví, kde se na toru
- Při přesunu používá drahocennou energii
- Pro každou pozici je známo, kolik energie je zapotřebí k projetí
- Cílem je najít nejkratší cestu na základnu
  - vzhledem k poškozeným motorům nelze předem spočítat optimální cestu
  - proto pro každou pozici spočítáme nejlepší pokyn, který může řídící jednotka zadat

Postup

- Nejprve si představte, že pozice a přechodové akce tvoří acyklický graf, a vytvořte odpovídající Bellmanovu funkci užitku
- Implementujte jednodušší algoritmus založený jen na „value update“ a zkontrolujte si, že na pomocných testech dává správné řešení
- K implementaci „policy update“ si napište příslušnou soustavu rovnic

7. domácí úkol: Torus



Souřadnicový systém na toru

- Torus je rozdělený do čtvercové mřížky velikosti $n \times m$
- Pozice na toru je dána souřadnicemi (i, j)
  - $0 \leq i < n$ a $0 \leq j < m$
- Z pozice (i, j) se pohybem o jedno políčko na
  - sever dostaneme na pozici $(i - 1 \mod n, j)$
  - jih dostaneme na pozici $(i + 1 \mod n, j)$
  - západ dostaneme na pozici $(i, j - 1 \mod m)$
  - východ dostaneme na pozici $(i, j + 1 \mod m)$

Modulení záporného čísla na reálných počítačích

- Kolik je $-4 \mod 3$ matematicky?
- Kolik je $-4\%3$ v C/C++?
- Kolik je $-4\%3$ v Python?