Kognitívna mapa bludiska

Michal Malý

Katedra aplikovanej informatiky FMFI UK
Mlynská dolina, 824 48 Bratislava
maly@fmph.uniba.sk

Abstrakt

V učení posilňovaním sa využíva model čiastočne pozorovateľných Markovovských rozhodovacích procesov. Zvyčajne sa predpokladá, že agent má predpripravený model sveta (jeho stavov), ktorý na základe pozorovaní a zvolenej stratégie ohodnocuje a na jeho základe vykonáva akcie.

V príspevku sa zaoberáme problémom, či je model sveta možné vytvoriť na základe pozorovaní, aby si ho agent mohol odvodiť sám a nemusel byť explicitne zadaný. Zadefinujeme problém vytvorenia Markovovského modelu. Ukazujeme teoretické obmedzenia a demonštrujeme riešenie pomocou gramatickej indukcie pre zjednodušený prípad s aplikáciou pre problém bludiska. Vytvorený model je vhodný najmä pre využitie v učení posilňovaním.

1 Úvod

1.1 Učenie posilňovaním

Učenie posilňovaním je spôsob učenia inšpirovaný behaviorizmom. Spočíva v tom, že učený agent koná na základe vstupov z prostredia, ale spätnú väzbu – číselné ohodnotenie (odmenu alebo trest) – môže dostať až oneskorene. Nikdy mu teda nie je priamo prezentované správne konanie v danej situácii, a aj to, či vykonal správnu akciu, môže iba nepriamo odvodiť až z (vo všeobecnosti ľubovoľne) oneskoreného ohodnotenia. Cieľom je maximalizovať toto ohodnotenie.

"Učenie posilňovaním je učenie čo robiť – ako namapovať akcie k situáciám - aby sa maximalizoval vstupný signál. Učenému nie je povedané, aké akcie má vykonať, ale miesto toho musí skúšaním zistiť, ktoré akcie prinášajú najväčšiu odmenu.

V najzaujímavejších a najnáročnejších prípadoch akcia nemusí ovplyvniť len najbližšiu odmenu, ale aj časovo nasledujúcu situáciu a cez ňu všetky nasledujúce odmeny. Tieto dve charakteristiky -- učenie cez pokus a omyl, a oneskorená odmena, sú dve najdôležitejšie charakteristiky učenia posilňovaním." [1] Hoci sa vo všeobecnosti používajú viaceré metódy, ľubovoľnú metódu, ktorá spĺňa popísanú charakteristiku, môžeme zaradiť do skupiny učenia posilňovaním.

1.2 Definícia problému učenia posilňovaním

Pri učení posilňovaním spolu interagujú *prostredie* a *agent*. V slede časových okamihov *(t=0,1,2,...)* agent dostáva od prostredia informáciu – pozorovanie stavu s_t – a na základe vlastného uváženia (stratégie) volí akciu a_t z dopredu definovanej množiny. V dôsledku zvolenej akcie sa zmení stav prostredia, o ktorom agent znova dostane informáciu prostredníctvom ďalšieho pozorovania s_{t+1} , a navyše tiež dostane od prostredia číselnú odmenu r_{t+1}. Takáto postupnosť krokov sa opakuje. Číselnú odmenu vypočítava prostredie na základe agentovho výkonu, odmena môže byť teda výsledkom dlhodobej interakcie a nie len dôsledkom práve vykonanej agentovej akcie.

Je obvyklé nazerať na prostredie ako na Markovovský proces. Prechody medzi stavmi prostredia môžu byť stochastické. Zodpovedá to skutočnosti, že akcie agenta nemusia byť presné alebo úspešné. Rovnako sa môže stať, že agent má iba čiastočný prístup k stavu prostredia, napríklad ak jeho senzory sú nepresné, alebo je to prirodzená vlastnosť prostredia: ak agent hrá poker, nevidí do balíčka s kartami.

Agent sa pohybuje v prostredí počas dlhšieho časového obdobia. Úloha môže skončiť po splnení cieľa, alebo po uplynutí obmedzeného času, alebo môže pokračovať neobmedzene dlho. Cieľom agenta je maximalizovať odmenu v dlhodobom meradle. Táto dlhodobá odmena (počínajúc časom *t*) sa zvyčajne formalizuje ako suma

$R_t = \sum_k^T \gamma^k r_{t+k+1}$, kde T je celkový čas úlohy a $0 < \gamma \leq 1$ je "utlmovací" parameter (discount factor), ktorý určuje, ako cenné sú odmeny vzdialenejšie v čase. Ak je T konečné (epizodická úloha), je prípustná hodnota $\gamma = 1$, vtedy sú zahrnuté všetky odmeny rovnako. Inak musí byť nerovnosť ostrá – máme potom zaručené, že suma nenadobudne nekonečnú hodnotu.

1.3 Prístupy a metódy učenia posilňovaním

Riešenia problému učenia posilňovaním sú založené na tom, že agent si interne udržuje (odhaduje) ohodnotenie stavov a akcií a podľa zvolenej stratégie sa rozhoduje, ktoré akcie vykoná.

Ohodnocovacia funkcia stavu $V^{\pi}(s)$ vyjadruje, akú odmenu môže agent očakávať, ak sa nachádza v stave *s* a bude sledovať stratégiu π.

Ohodnocovacia funkcia akcie $Q^{\pi}(s,a)$ vyjadruje, akú odmenu môže agent očakávať, ak sa nachádza v stave *s*, vykoná akciu *a*, a následne bude sledovať stratégiu π. Ohodnocovacia funkcia je užitočná vtedy, ak agent chce vykonať inú akciu, ako by prislúchala podľa jeho stratégie – napríklad nechce vykonať akciu s najlepším ohodnotením, ale chce vykonať neznámu akciu a preskúmať prostredie.

Prístup pomocou **dynamického programovania** je založený na priamom iteratívnom prepočítaní ohodnotení stavov a akcií. To si zvyčajne môžeme dovoliť, ak nás netrápi výpočtová náročnosť, a ak je k dispozícii dokonalý model sveta (je plne známy Markovovský proces vrátane pravdepodobností prechodov).

Monte Carlo metóda sa učí priemerovaním vzoriek z reálnej skúsenosti z prostredia, alebo tiež zo skúsenosti simulovanej. Nepotrebuje zadaný Markovovský model a ani si ho nevytvára.

Temporal difference learning (učenie na základe chyby v odhade odmeny) vylepšuje Monte Carlo metódu tak, že zatiaľ čo Monte Carlo čaká s užitočným updatom pokiaľ nie je známy výsledok akcie, TD-učenie použije jeho predbežný odhad, podobne ako sa to robí v dynamickom programovaní. Rovnako si nevytvára model prostredia.

Stratégia pre Monte Carlo alebo TD-učenie je obvykle voľba najužitočnejšej akcie, prípadne výber náhodnej akcie s malou pravdepodobnosťou.

Pokročilejšie metódy majú oddelenú voľbu akcie od ohodnocovacej funkcie od voľby akcie (**actor-critic**). Pri voľbe náhodnej akcie sa totiž odchyľujeme od zvolenej stratégie, čo môže viesť k inému výsledku, aký by sme dosiahli, ak by sme sa riadili stratégiou. Výsledok by sa teda nemal zaratávať do ohodnotenia daného stavu. Miesto toho sa určí, či sa očakávaná odmena zlepšila alebo zhoršila, a podľa toho sa posilní alebo oslabí vykonanie zvolenej akcie v tomto stave.
Na odhad odhodnotení sa tu s úspechom používajú napr. neurónové siete, ktoré poskytnú zovšeobecnenie skúseností. Môžu byť tiež využité v spojitých prípadoch.

2 Motivácia

Takmer všetky existujúce prístupy učenia posilňovaním sú založené na implicitnom určení stavového priestoru. Priestor je určený zvyčajne rozsahom prípustných pozorovaní, môže byť diskrétny aj spojitý.
Napríklad v hre piškvorky môže agent pozorovať hraciu plochu a rozmiestnenie značiek na nej. Všetky potenciálne označkovania tvoria stavový priestor. Niektoré stavy (napr. štyri značky X a len jedna O) sa môžu ukázať ako nedosiahnuteľné, prípadne môžu byť vylúčené zo stavového priestoru už pri návrhu agenta.
Čo však v prípade, že pozorovania nedokážu pokryť skutočný priestor možností, ktorý je určený prostredím?
Napríklad, ak uvažujeme problém navigácie v bludisku: Ak sú súradnice *(x,y)* pozorovateľné, poskytujú plný popis polohy agenta v priestore bludiska a umožňujú učiacim metódam vypočítať ohodnotenia stavov a ohodnotenia akcií v jednotlivých stavoch. Ak však môže agent pozorovať len svoje okolie (povedzme, pole na ktorom stojí a 4 polia okolo), a nevie dopredu rozmery bludiska, ako mu zadefinovať stavový priestor? Polí s rovnakým usporiadaním môže byť viacero; agent v nich dostane však rovnaké pozorovanie.
Cieľom je z postupnosti pozorovaní a akcií vytvoriť model, ktorý by zahŕňal možnosť viacerých stavov s rovnakým pozorovaním.
Tento prístup môže byť užitočný aj vtedy, ak síce priestor pozorovaní zodpovedá stavovému priestoru, ale predpokladáme, že vo svete agenta bude možné nájsť užitočné vzťahy medzi jednotlivými stavmi, o ktorých dopredu nevieme, alebo ich nechceme explicitne zadávať kvôli náročnosti.

3 Predošlý výskum

Obmedzenia agentov bez správneho modelu sú známe už dávno. V [3] je ilustrovaný tento problém v nasledujúcej úlohe: "Uvažujme úlohu balenia darčeka, ktorá zahŕňa 4 kroky: otvoriť krabicu, vložiť darček, zatvoriť ju, a zalepiť. Agent, ktorý je vedený len jeho aktuálnym vizuálnym vnemom nedokáže úlohu splniť, pretože ak má pred sebou zatvorenú krabicu, nevie, či darček je už vo vnútri, a teda sa nevie rozhodnúť, či má krabicu zalepiť alebo otvoriť." Autori zároveň analyzujú tri konekcionistické pamäťové architektúry, ktoré extrahujú príznaky z histórie a tak doplňujú stav agenta.
Problém *perceptuálneho aliasingu (perceptual aliasing),* čiže skutočnosti, že rôzne stavy môžu mať rovnaký perceptuálny vnem, bol pozorovaný už v [4], kde bol navrhnutý algoritmus *Lion* (lev). Algoritmus sa snaží udržovať si interný stav, ktorý je konzistentný s pozorovaniami. Keďže v dôsledku neúplného pozorovania dôjde k zámene odhadovaného a skutočného

stavu, Lion rieši túto neočakávanú situáciu tak, že identifikuje akciu, ktorá viedla k nekonzistentému stavu, a dočasne jej ohodnotenie zresetuje na nulovú hodnotu. Tým zabráni jej vykonaniu a umožní agentovi rozhodnúť sa pre iné, vhodnejšie akcie.
Kaelbling a Chapman [5] navrhli G-algoritmus, ktorý rekurzívne delí svet na stále jemnejšie kúsky, vytvárajúc stromovú štruktúru pre ohodnocovaciu funkciu akcií $Q^{\pi}(s,a)$.
Ďalej boli navrhnuté prístupy založené na agregácii [6] a delení stavov [7].

4 Problém vytvorenia Markovovského modelu

4.1 Markovovský rozhodovací process

Markovovský rozhodovací proces je matematická štruktúra vhodná pre zachytenie takého prostredia, kde vykonanie akcie v nejakom stave spôsobí prechod do iného stavu alebo stavov, pričom prechod do konkrétneho stavu je stochastický – výsledný stav záleží od pravdepodobnosti, určenej prechodovou funkciou. Agent okrem toho dostane príslušnú odmenu.
V Markovovskom modeli nezáleží na histórii – správanie prostredia je (až na náhodu) plne určené aktuálnym stavom.
Formálne, ak S je množina stavov, A množina akcií, prechodová funkcia *P(a,s,s')* určuje pravdepodobnosť prechodu zo stavu *s* do stavu *s',* a *R(a,s,s')* je odmena, ktorú agent získa.
V prípade, že agent môže stav s pozorovať iba čiastočne, vstupuje do hry ešte pravdepodobnosť, že z množiny *O* pozorovaní dostane agent pri prechode do stavu *s'* informáciu (pozorovanie) *o*. Táto pravdepodobnosť je určená $\Omega(o,s',a)$.

4.2 Akcie, pozorovania a model

Predpokladajme, že agent začal vykonávať akcie, pozoroval prostredie a dostával nejaké odmeny.
Získame postupnosť $o_1, a_1, o_2, r_2, a_2, o_3, r_3, a_3, \ldots, a_{n-1}, o_n, r_n$. Je možné odvodiť taký model sveta, ktorý by čo najlepšie zodpovedal tejto postupnosti pozorovaní, akcií a odmien? Predpokladajme, že máme nejaký model k dispozícii. Je možné určiť jeho vierohodnosť.

Definícia. *Vierohodnosť modelu.* Nech je daná postupnosť pozorovaní, akcií a odmien $P= o_1, a_1, o_2, r_2, a_2, o_3, r_3, a_3, \ldots, a_{n-1}, o_n, r_n$. kde o_i je pozorovanie, r_i je odmena a a_i je akcia v časovom okamihu *i*. Nech $M = (S, A, O, T, \Omega, R)$ je čiastočne pozorovateľný Markovovský rozhodovací proces. Potom vierohodnosť modelu *B(M)* je

$$B(M) = \max_{s \in S^n} \prod_{t=1}^{n-1} P(a_t, s_t, s_{t+1}) \cdot \Omega(o_{t+1}, s_{t+1}, a_t)$$

Spomedzi všetkých možných modelov by sme potom teoreticky mohli vybrať najvierohodnejší (ak by sme nebrali ohľad na výpočtové obmedzenia). Avšak takýto model by mohol byť príliš komplikovaný. V súlade s princípom Occamovej britvy by bolo vhodnejšie vybrať síce menej vierohodný, ale jednoduchší model. Bolo by tu nutné určiť, ako vyvážiť jednoduchosť a vierohodnosť.
V praxi je takáto formulácia problému vzhľadom na výpočtové obmedzenia príliš široká. V ďalšom texte sa obmedzíme na zjednodušený prípad, keď akcie a pozorovania budú deterministické.

4.3 Problém bludiska

Predpokladajme, že máme bludisko, ktoré agent dopredu nepozná. Ako takéto bludisko prehľadať, ako vytvoriť jeho mapu?
Samozrejme, existujú štandardné prístupy k riešeniu tohto problému za predpokladu, že je možné si do prostredia značiť, že sme daným políčkom už išli (Trémauxov algorimus, Tarryho prieskum a pod.)
V našom prípade takéto značkovanie nie je možné. Tiež nechceme dať agentovi explicitnú informáciu/algoritmus závisiaci na tom, či je v bludisku. Vyžadujeme, aby danú úlohu splnil bez tejto znalosti, čo bude demonštrovať jeho schopnosť riešiť aj všeobecnejšie problémy. Agent má k dispozícii len množinu akcií *A* a môže prostredie spoznávať cez pozorovania z *O,* čím dostane informáciu o svojom najbližšom okolí. Akcie ani pozorovania nenesú ďalšiu informáciu (súradnice, smer, a pod.). V princípe si možno predstaviť, že agenta vložíme bludiska ľubovoľných rozmerov (2D, 3D), za predpokladu, že upravíme veľkosti množín *A* a *O*. Môžeme tiež do bludiska umiestniť napr. jednosmerné dvere, ktorými možno prejsť len jedným smerom, alebo "teleporty", čiže políčka, ktoré agenta prenesú na iné, vzdialenejšie miesto v bludisku.

Veta. *Nemožnosť preskúmania bludiska vo všeobecnosti.* Nie je možné preskúmať ľubovoľné bludisko, ak agent nemôže do prostredia umiestňovať značky. [2]
Dôkaz. Nemožnosť vyplýva z toho, že dva regulárne grafy s rovnakým počtom vrcholov nie sú z pohľadu agenta rozlíšiteľné (Napr. trojuholník a štvorec, čiže tri, resp. štyri vrcholy spojené do cyklu. Každý vrchol má dve hrany, ak vrcholy nie sú odlíšené, nemožno určiť rozdiel.) □

Napriek tomuto výsledku má zmysel zaoberať sa riešením problému bludiska z pohľadu učenia posilňovaním. V dôkaze sme totiž využili, že všetky vrcholy sú rovnaké. Takéto prostredie je relatívne neštandardné. V tom prípade je asi rozumné predpokladať najjednoduchší model a riadiť sa ním. Limitácia je, že môžu existovať stavy, o ktorých nevieme, a vďaka zvolenej stratégii sa k nim nikdy ani nedostaneme.

4.4 Hľadanie modelu bludiska

Keďže predpokladáme zjednodušený prípad, že prostredie je deterministické, je možné modely odlíšiť podľa toho, či zodpovedajú postupnosti pozorovaní a akcií, alebo nie (vyššie definovaná vierohodnosť modelu *B(M)=1* resp. *0*).
Spomedzi tých modelov, ktoré túto podmienku spĺňajú, by sme radi vybrali najjednoduchší model (s najmenším počtom stavov). Týmto volíme induktívny bias [8], potrebný na prekonanie neurčitosti spomínanej v predošlej podsekcii. Formalizmus minimálnej popisnej dĺžky zodpovedá filozofii Occamovej britvy.
Nazdávame sa, že najvhodnejší prístup k riešeniu tohto problému je *gramatická indukcia,* prípadne inferencia zvoleného druhu automatu od konečného automatu, cez zásobníkový automat, až po Turingov stroj. V prípade konečného automatu (regulárnej gramatiky) je vzťah s Markovovským modelom zrejmý: stavy automatu (neterminály) zodpovedajú stavom Markovovského procesu, akcie sú vstupné symboly (terminály) pre automat (gramatiku). Bližší postup predstavíme v ďalšej sekcii.
Pre silnejšie formalizmy ako konečné automaty, treba nazerať na výsledný automat (povedzme Turingov stroj) ako na orákulum, ktoré odpovedá na otázky týkajúce sa implicitne reprezentovaného Markovovského modelu, t.j. do akého stavu vedie akcia, aké pozorovanie a odmenu dostane agent v danom stave. Meraná zložitosť je zložitosť automatu (povedzme popisná dĺžka Turingovho stroja), nie Markovoského modelu (ten netreba explicitne reprezentovať). Dôvody a možné výhody pre silnejšie formalizmy uvedieme v diskusii.

5 Algoritmus

Cieľom je nájsť taký (čo najmenší) konečný automat (Markovovský proces), ktorý by vyhovoval postupnosti pozorovaní, akcií a odmien $P= o_1, a_1, o_2, r_2, a_2, o_3, r_3, a_3, \ldots, a_{n-1}, o_n, r_n$.

5.1 Neúspešné zovšeobecnenie cez Myhill-Nerodovu ekvivalenciu

Pokúsili sme sa na problém použiť metódu gramatickej indukcie založenú na Myhil-Nerodovej ekvivalencii (bližší popis a zdrojový kód v [9]), a to tak, že by tie prefixové podpostupnosti akcií $a_1, \ldots, a_k$, kde $k \leq n$), ktoré vedú k dopredu zvolenému pozorovaniu o (t.j. platí $o=o_k$), boli považované za slová jazyka L_o, ku ktorému sa má nájsť regulárna gramatika (konečný automat). Malou modifikáciou metódy by potom bolo možné vytvoriť automaty pre každý z jazykov L_o, pričom tieto automaty by boli až na pozície akceptačných stavov totožné. Stavy a hrany automatov by teda zodpovedali stavom a prechodom v Markovovskom modeli, pozície akceptačných stavov v automate akceptujúcom jazyk L_o by zodpovedali tým stavom v Markovovskom modeli, ktoré dávajú agentovi pozorovanie o.
Žiaľ, táto metóda sa ukázala ako nevyhovujúca. Metóda totiž produkuje gramatiku/minimálny automat akceptujúci len zadanú množinu slov, ktorý má príliš veľa stavov (neakceptovanie iných slov znižuje schopnosť zovšeobecňovať).

5.2 Riešenie cez SAT solver

Rozhodli sme sa preto hľadať model tak, že sformulujeme obmedzenia preň v tvare logických formúl, a tieto formuly necháme vyriešiť SAT solver-u. SAT solver je program, ktorý určí, či zadané logické formuly sú splniteľné, a ak áno, aké priradenie hodnôt premenným treba zvoliť.

5.2.1 Formalizácia

Predpokladajme, že model má mať j stavov očíslovaných *0, 1, …, j-1* a na začiatku sa bez ujmy na všeobecnosti nachádza v stave číslo *0*. Nech počet akcií $|A|=k$, počet možných pozorovaní $|O|=l$. Akcie a pozorovania očíslujeme od *0* po k resp. l, podobne, ako sme očíslovali stavy. Odmeny r sme sa rozhodli pre lepšiu názornosť zanedbať.
Skutočnosť, že v stave s dostáva agent pozorovanie o, zaznačíme predikátom *obs(o,s)*. Skutočnosť, že na akciu a sa zo stavu s dostaneme do stavu s', zaznačíme predikátom *tr(s,a,s')*. Skutočnosť, že sme v čase t boli v stave s zaznačíme ako *pos(t,s)*.

Pre predikát *obs* platia obmedzenia

$$\forall s \in S : \forall o, o' \in O, o \neq o' : \neg obs(o,s) \vee \neg obs(o',s)$$

čiže v jednom stave máme najviac jedno pozorovanie, a tiež

$\forall s \in S \bigvee_{o \in O} obs(o, s)$, čiže máme aspoň jedno pozorovanie.

Pre predikáty *tr* platia podobné obmedzenia

$\forall s, s', s'' \in S, a \in A, s' \neq s'':$

$\neg tr(s, a, s') \vee \neg tr(s, a, s'')$

$\forall s \in S, a \in A: \bigvee_{s' \in S} tr(s, a, s')$, čiže akcia vedie do práve jedného stavu.

Obdobne pre predikát *pos:*

$\forall 0 \leq t \leq n, s, s' \in S: \neg pos(t, s) \vee \neg pos(t, s')$

$\forall 0 \leq t \leq n: \bigvee_{s' \in S} pos(t, s)$

Skutočnosť, že model vyhovuje postupnosti $P = o_1, a_1, o_2, r_2, a_2, o_3, r_3, a_3, \ldots, a_{n-1}, o_n, r_n$ možno zapísať ako

$\forall 0 \leq t \leq n, s \in S: obs(s, o_t) \vee \neg pos(t, s)$

$\forall 0 \leq t \leq n-1, s, s' \in S:$

$tr(s, a_t, s') \vee \neg pos(t, s) \vee \neg pos(t+1, s')$

Predpoklad, že model sa nachádza na počiatku v stave 0, sa zapíše ako *pos(0,0).*

Ostáva už len každé možné dosadenie do predikátov označiť osobitnou pomocnou premennou x_m a hore uvedené vzťahy premenné prepísať pomocou čísel týchto premenných vo formáte akceptovanom SAT solverom.

5.2.2 Výsledky

Testovali sme vytváranie modelu na bludisku (Fig. 1), pričom agent získaval informácie len o 4-okolí, a to v podobe jedného čísla, ktorého 5 bitov v binárnej podobe zodpoedalo od najmenej významného bitu po najvýznamnejší bit tomu, či sa nachádza stena na poli, na ktorom sme, poli vľavo, vpravo, hore, dole. Agent mohol vykonávať štyri akcie: 0,1,2,3 – vpravo, dole, vľavo, hore.

Po určitom počte krokov sme agenta zastavili. Na Fig. 2 je možné vidieť príklad vytvoreného modelu, zodpovedajúci časti 10 polí vľavo hore v bludisku. Model zodpovedá realite, je tiež zaujímavé a dôležité podotknúť, že metóda dokázala odlíšiť dve trojice stavov, v ktorých má agent rovnaké pozorovania (tri polia označené 8 a tri polia označené 16, v bludisku vľavo hore).

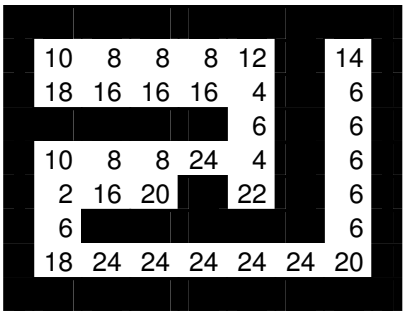

Fig. 1. Bludisko. Čísla vyjadrujú vstup pre agenta (pozorovanie) a zodpovedajú informácii o 4-okolí.

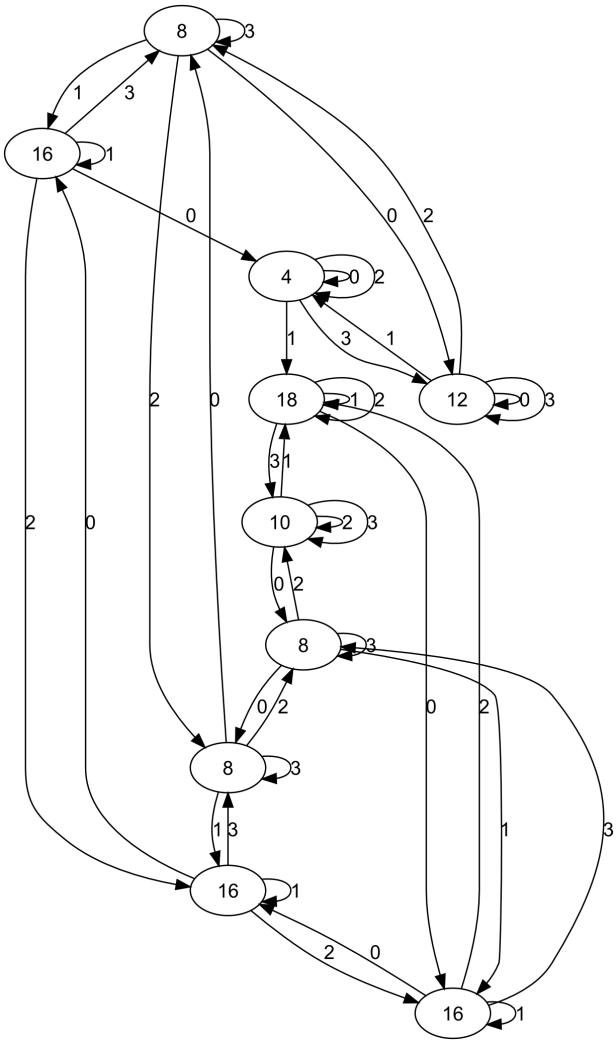

Fig. 2. Model bludiska (kognitívna mapa).

5.2.3 Možné rozšírenia

V uvedených prípadoch sme zanedbali odmeny. Ich pridaním by sme dosiahli ďalšie obmedzenia, spresňujúce model (bolo by nutné uvažovať nad diskretizáciou odmeny v prípade, že táto je pôvodne spojitá – reálne číslo). Ďalej by bolo možné uvažovať o rozdelení pozorovania na jeho príslušné podčasti: v uvedenom prípade sme nedali agentovi možnosť rozpoznať v pozorovaní jeho zložky. Bolo by možné prezentovať sadu pozorovaní (napr. pole v predu – 1 bit, pole vzadu – 1 bit, atď). Tieto by agent mohol lepšie využiť pri zovšeobecnení.

5.3 Možnosti ďalšieho výskumu

Bolo by možné použiť iný spôsob gramatickej indukcie, ako popísaný, napr. pomocou algoritmu ECGI [10], ktorý produkuje regulárnu gramatiku, cez genetické algoritmy [11], alebo cez ďalšie symbolové metódy [12]. Bolo by tiež možné modifikovať náš algoritmus, aby skúšal cez SAT solver hľadať silnejšiu gramatiku (bezkontextovú), prípadne automat s jedným alebo dvoma počítadlami, predpokladáme však, že zložitosť – počet kláuz, ktoré bude treba použiť, bude nesmierne narastať. Treba tiež pamätať na to, že od istej hranice môžeme naraziť na nevypočítateľnosť – problém zastavenia pre Turingove stroje (automat s dvoma počítadlami je pri vhodnom kódovaní ekvivalentný Turingovmu stroju).

6 Diskusia

6.1 Kognitívny význam vytvorenia modelu sveta

Sme názoru, že problém vytvorenia Markovovského modelu matematicky ilustruje niektoré kognitívne aktivity. Pozorujeme svet a vytvárame si jeho model, našu skúsenosť pritom zovšeobecňujeme. Toto nám pomáha lepšie sa rozhodnúť v budúcnosti, keď narazíme na situáciu, ktorá je rovnaká alebo podobná. Taktiež dokážeme robiť úsudky o vlastnostiach prostredia bez toho, aby sme tieto vlastnosti vedeli priamo pozorovať našimi zmyslami.

6.2 Sila formalizmu ako sila zovšeobecnenia

Ak by sme dokázali použiť silnejšiu metódu, predpokladáme, že by sme dosiahli zaujímavejšie zovšeobecnenie. Uvažujme napríklad prázdnu mriežku, na ktorej sa agent môže voľne pohybovať neobmedzený počet krokov vpravo, vľavo, hore, dole, a môže si na políčko uložiť značku. Zjavne nemôže mať skúsenosť s každým políčkom. Na otázku, kam sa z nového políčka *p* dostane, ak vykoná 3 kroky doprava, 3 hore, 3 doľava, a 3 dole, nemá vo svojej "skúsenosti" odpoveď. Model založený na konečnom automate mu takúto odpoveď tiež neposkytne, keďže každé políčko musí evidovať ako osobitný stav, a teda nezovšeobecní pravidlo, že vykonaním uvedeného pohybu sa vždy dostaneme na pôvodné políčko.
Ak by však agent mohol použiť silnejší automat, napr. automat s dvoma počítadlami, mohol by zjednodušujúci algoritmus odvodiť taký automat, ktorý by na dvoch počítadlách zaznamenával súradnice. Takýto automat bude mať len pár stavov, a teda bude určite jednoduchší ako akýkoľvek konečný automat zaznamenávajúci každé navštívené pole osobitne. Nie je nám známe, či by technika popísaná v [13] pre realtimeové automaty s 2 počítadlami bola schopná odvodiť takýto automat.

7 Záver

Popísali sme princípy a obmedzenia pre vytváranie modelu sveta z pozorovaní. Tiež sme navrhli jednoduchý algoritmus založený na SAT solveri. Otestovali sme metódu na bludisku. Dosiahli sme, že vytvorený model obsahoval stavy zodpovedajúce políčkam bludiska, pričom dokázal odlíšiť aj tie políčka, na ktorých mal agent rovnaký perceptuálny vnem.
Model sveta je použiteľný pre také úlohy učenia posilňovaním, kde sú doterajšie metódy neúspešné, pretože nemajú k dispozícii správny model.

8 Poďakovanie

Tento výskum bol podporený grantom VEGA 1/0439/11.

Literatúra

[1] R. S. Sutton, A. G. Barto. Reinforcement Learning: An Introduction. MITPress, Cambridge, MA, 1998.

[2] G. Dudek, M. Jenkin, E. Milios, D. Wilkes.: Robotic Exploration as Graph Construction. In: *IEEE Transactions on Robotics and Automation 7* (2002): 859–865.

[3] L. J. Lin, T. M Mitchell.: Memory approaches to reinforcement learning in non-Markovian domains. Technical Report CMU-CS-92-138, School of Computer Science, Carnagie Mellon University, Pittsburgh, PA.

[4] S. D. Whitehead, D. H. Ballard: Learning to perceive and act by trial and error. *Machine Learning 7* (1991): 45–83.

[5] D. Chapman, L. P. Kaelbling: Learning from delayed reinforcement in a complex domain. *Twelfth International Joint Conference on Artificial Intelligence*, 1991.

[6] S. Singh, T. Jaakkola, M. I. Jordan: Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, 1995: 361–368.

[7] R. A. McCallum: Overcoming incomplete perception with utile distinction memory. *Proceedings of the Tenth International Conference on Machine Learning*, 1993: 190–196.

[8] T. M. Mitchell.: The need for biases in learning generalizations. CBM-TR 5-110, Rutgers University, New Brunswick, NJ, 1980.

[9] M. Malý: Semi-automatic creation of a stemming dictionary of an inflecting language using grammatical induction. Študentská vedecká konferencia FMFI UK, Bratislava 2010, ISBN 978-80-89186-69-3: 295-300, dostupné online http://www.fmph.uniba.sk/fileadmin/user_upload/editors/studium/svk/2010/AIN/14.pdf

[10] H. Rulot, N. Prieto, E. Vidal: Learning accurate finite-state structural models of words through the ECGI algorithm. *International Conference on Acoustics, Speech, and Signal Processing*, 1989: 643–646.

[11] F. Javed, B. R Bryant, M. Črepinšek, M. Mernik, A. Sprague: Context-free grammar induction using genetic programming. Proceedings of the 42nd annual Southeast regional conference, 2004: 404–405.

[12] René Alquézar Mancho. Symbolic and connectionist learning techniques for grammatical inference, 1997. doctoral thesis.

[13] Amr F. Fahmy, Robert S. Roos: Efficient learning of real time two-counter automata. *Lecture Notes in Computer Science, 1996, Volume 1160*, 113-126.