

Pravdepodobnostné riadenie mobilného robota tréňované evolučným algoritmom

Pavel Petrovič

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky
Mlynská dolina, 842 48 Bratislava
ppetrovic@acm.org

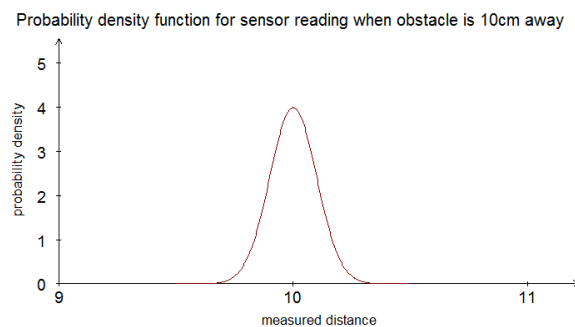
Abstrakt

Mobilné roboty získavajú informácie z prostredia prevažne pomocou senzorov, ktoré sú nepresné, zašumené a nie sú jednoznačné. Väčšina súčasných prístupov k lokalizácii, navigácii, mapovaniu i k iným riadiacim úlohám pre mobilné roboty využíva pravdepodobnostné modelovanie prostredia [1]. Parametre týchto distribúcií sú buď stanovované odhadom alebo metódou učenia s učiteľom (supervised learning) [2] a teda nie sú použiteľné vo vopred neznámom prostredí. Táto práca navrhuje spôsob ako určiť parametre distribúcií pomocou evolučného algoritmu v simulácii a s reálnym robotom.

1 Úvod

Roboty získavajú informácie o prostredí prostredníctvom senzorov. Táto informácia spravidla nemá symbolický charakter a už vôbec nepodlieha zákonom predikátového počtu a logiky. Napríklad, ak ultrazvukový senzor na meranie vzdialenosti oznámi, že prekážka je v danom smere vo vzdialenosti 3 metre, v skutočnosti je celkom možné, že prekážka je vo vzdialenosti 1m, ale sonar zachytil zosilnené vzdialené echo susedného sonaru, ktoré sa odrazilo od pevnejšej prekážky, alebo, že povrch prekážky je z nevhodného materiálu, obsahuje hrany, alebo nastal nejaký iný nežiadany, ale neodstrániteľný jav. Napriek tomu je táto informácia použiteľná, pretože vo väčšine prípadov je správna (napokon, iná ani k dispozícii nie je) – nemožno sa na ňu ale 100% spoľahnúť. Nemožno sa na ňu 100% spoľahnúť ani po dvoch, či troch meraniach, pravdepodobnosť pravdivosti údajov sa ale zvyšuje. Podobne, keď infračervený senzor na meranie vzdialenosti oznámi, že prekážka je vo vzdialenosti 10cm, môže to rovnako znamenať, že prekážka sa nachádza vo vzdialenosti 9.8cm alebo 10.2cm alebo, že prekážka sa nachádza vo vzdialenosti 9.5cm alebo 10.45 cm, ale už s menšou pravdepodobnosťou. Nameraný údaj 10cm preto treba chápať v tomto kontexte – v kontexte apriori znalostí o vlastnostiach použitého senzora. Tieto apriori znalosti sú k dispozícii jednak v rámci špecifikácie od výrobcu –

ktorý parametricky alebo graficky popíše správanie senzora a jeho presnosť a jednak sa dá (a väčšinou je to vhodné) určiť empirickým meraním, na základe ktorého možno určiť pravdepodobnostnú distribúciu náhodnej premennej – t.j. napríklad s akou pravdepodobnosťou nameriame príslušnú hodnotu x' , ak skutočná vzdialenosť od prekážky je x . Tá môže mať napríklad tvar ako na obrázku 1.



Obr. 1. Pravdepodobnostné správanie senzorov.

Vo všeobecnosti je informácia zo senzorov ovplyvnená rôznym šumom a obsahuje určitý stupeň neurčitosti. Podobne, akcie robotov nie sú presné, ale podliehajú nepresnostiam, nerovnostiam robota i prostredia, tepelnému a mechanickému namáhaniu materiálu, a ďalším fyzikálnym vplyvom. Preto podobne ako v prípade senzorov, ak robot dostane príkaz na posun o 1m vpred, v skutočnosti sa môže presunúť o 103 cm, prípadne s menšou pravdepodobnosťou o 107 cm a pod. Opäť máme k dispozícii apriori informáciu (podobnú obrázku 1) o tom, aký bude výsledok akcie robota. Na roboty, ktoré plnia úlohy v skutočnom prostredí, t.j. ak zdieľajú prostredie so živými bytosťami sa kladú obrovské nároky z hľadiska ich bezpečnosti a spoľahlivosti. Preto je nutné, aby roboty maximálne využili čo najviac informácii, ktoré sú dostupné, najmä pokiaľ ide o ich interakciu s prostredím a najmä ak je vopred známe, že informácia zo senzorov je nespoľahlivá a obsahuje chyby. Z toho dôvodu si reálne robotické aplikácie – a pre túto komunitu ani žiadne inteligentné robotické systémy nemôžu dovoliť nespracovať aj apriori poznatky o správaní senzorov a aktuátorov robota, ktoré

*1 Takto sa píšú poznámky pod čiaru.

sú vyjadrené vo forme pravdepodobnostných distribúcií. Využitím tejto informácie budú akcie a model robota automaticky zahŕňať predpoklad, že informácia z prostredia je neúplná a nepresná. Výsledkom je oveľa robustnejšie správanie. Pravdepodobnostný model sa takto sám postará o správne naloženie s jednotlivými chybnými údajmi samotná logika modelu sa nimi ďalej nemusí podstatne zaoberať – model automaticky pracuje so zašumenými dátami. Štúdiom pravdepodobnostných prístupov k riešeniu úloh navigácie, lokalizácie, mapovania prostredia, riadenia robota a pod. Sa zaoberá Pravdepodobnostná robotika [1]. Pravdepodobnostná robotika rieši predovšetkým úlohy modelovania pohybu robota, modelovania senzorických systémov – odometrie, laserových či sonarových smerových snímačov (beam models of range finders), aktualizovania polohy a orientácie robota na mape – či už pomocou metód výskytu v mriežke (grid occupancy), alebo Monte Carlo prístupov (particle filters), vytvárania máp prostredia a súčasnej lokalizácie (SLAM – simultaneous localization and mapping), vrátane pre nás najviac relevantného plánovania a výberu akcie na základe cieľov a pravdepodobnostnej aktualizácie modelu sveta. Unifikujúci framework pre využitie pravdepodobnostnej robotiky poskytuje tzv. Bayesovské robotické programovanie [2].

2 Teória Bayesovského robotického programovania

Bayesovské robotické programovanie pracuje s výroky ($a, b, a \wedge b$), náhodnými premennými – ktoré sú buď diskrétné (t.j. náhodná premenná X a vzájomne vylučujúce sa výroky x_i), alebo spojité, ktoré môžu nadobúdať continuum hodnôt. K diskrétnym náhodným premenným môžu existovať apriori poznatky (väčšinou označované π), konkrétne priradenie pravdepodobnostných hodnôt jednotlivým výrokom náhodnej premennej. Pravdepodobnosť platnosti nejakého výroku a sa potom označuje ako $\mathbf{P}(a)$ a keďže pre každé zmysluplné odvedenie uvažujeme nejaké apriori poznatky, píšeme $\mathbf{P}(a | \pi)$. Ak prijmeme, že pravdepodobnosť pravdivého výroku je $\mathbf{P}(True) = 1$, pravdepodobnosť nepravdivého výroku je $\mathbf{P}(False) = 0$, pravdepodobnosť akéhokoľvek výroku je v rozmedzí 0 a 1, stačí nám pridať dve pravidlá, konkrétne:

1. Pravidlo konjunkcie:

$$\begin{aligned} \mathbf{P}(a \wedge b | \pi) &= \mathbf{P}(a | \pi) \times \mathbf{P}(b | a \wedge \pi) \\ &= \mathbf{P}(b | \pi) \times \mathbf{P}(a | b \wedge \pi) \end{aligned}$$

2. Pravidlo normalizácie:

$$\mathbf{P}(a | \pi) + \mathbf{P}(\neg a | \pi) = 1$$

Z týchto pravidiel potom možno odvodiť užitočné pravidlá pravdepodobnostnej inferencie, konkrétne

1. Konjunkcia náhodných premenných (teda $X \diamond Y = \{x_i \wedge y_i; x_i \in X \wedge y_i \in Y\}$):

$$\begin{aligned} \mathbf{P}(X \diamond Y | \pi) &= \mathbf{P}(X | \pi) \times \mathbf{P}(Y | X \wedge \pi) \\ &= \mathbf{P}(Y | \pi) \times \mathbf{P}(X | Y \wedge \pi) \end{aligned}$$

2. Normalizácia náhodných premenných:

$$\sum_x \mathbf{P}(X | \pi) = 1$$

3. Marginalizácia premenných:

$$\sum_x \mathbf{P}(X \diamond Y | \pi) = \mathbf{P}(Y | \pi)$$

Základom prakticky všetkých algoritmov Bayesovskej a pravdepodobnostnej robotiky je Baysovské pravidlo podmienenej pravdepodobnosti:

$$\mathbf{P}(X | Y) = \mathbf{P}(Y | X) \mathbf{P}(X) / \mathbf{P}(Y)$$

Jeho využitie možno demonštrovať na nasledujúcom príklade. Uvažujme robot, ktorý pomocou senzora meria výšku schodu, na ktorý má vykročiť, pričom podstatné sú dva stavy – privysoký, resp. vhodný. Senzor ale vracia iba 10 rôznych hodnôt 1-10. Z predchádzajúcich meraní máme k dispozícii pravdepodobnostnú distribúciu popisujúcu opačnú závislosť: aká je pravdepodobnosť, že senzor vráti nejakú hodnotu, ak schod je (resp. nie je) privysoký, t.j. $\mathbf{P}(hodnota_senzora | privysoký)$. Bayesovské pravidlo nám umožňuje obrátiť podmienenosť a určiť aká je pravdepodobnosť, že robot sa nachádza pred privysokým schodom, ak nameral konkrétnu hodnotu:

$$\begin{aligned} \mathbf{P}(privysoký | hodnota_senzora) &= \\ \mathbf{P}(hodnota_senzora | privysoký) \mathbf{P}(privysoký) / & \\ \mathbf{P}(hodnota_senzora) & \end{aligned}$$

kde $\mathbf{P}(privysoký)$ je náš pôvodný odhad, že schod je privysoký (napr. 0.5) a $\mathbf{P}(hodnota_senzora)$ je hodnota väčšinou nezávislá od konkrétnej hodnoty merania – je vždy rovnaká a preto sa označuje ako normalizačná konštanta.

Bayesovské programovanie robotov sa dá analogicky prirovnať k logickému programovaniu: program pozostáva v prvej časti z nejakého popisu sveta: v logickom programovaní ide o bázu faktov a pravidiel, v Bayesovskom programovaní ide o identifikáciu náhodných premenných, ich parametrických foriem, a vzťahov, ktoré medzi nimi platia; a v druhej časti z

otázky – v logickom programovaní nejaký výrok, ktorý logický inferenčný mechanizmus splní, alebo vyvráti a v Bayesovskom programovaní sa pýtame na to, aká je pravdepodobnosť platnosti nejakého výroku. Dôležitým prvkom je teda získanie empirických dát, na základe ktorých sa určí pravdepodobnostná distribúcia apriori poznatkov.

3 Experimenty s Bayesovským programovaním robotov

V práci [2] natrénovali Oliver Lebeltel a kol. mobilného robota Khepera vybaveného 8 infračervenými senzormi na meranie vzdialenosti pomocou Bayesovského programovania na plnenie niekoľkých rozličných úloh. Model pozostával z troch náhodných premenných – vzdialenosť k najbližšej prekážke (*Prox*), smerový uhol k najbližšej prekážke (*Dir*) a uhlová rýchlosť (*Vrot*), pod ktorou sa má robot otáčať za súčasného pohybu vpred. Trénovacie dáta, na základe ktorých bola určená pravdepodobnosť $P(Vrot | Dir \diamond Prox \diamond \delta \diamond \pi)$, pričom δ zodpovedá trénovacím dátam, ktoré boli získané priamym riadením robota joystickom po dobu približne 30 sekúnd, čo bolo dostatočné na to, aby sa robot naučil príslušné správanie. Pri použití toho istého modelu sa robot buď naučil obchádzať prekážky, alebo sledovať stenu, alebo tlačiť objekt. Alternatívne prístupy na trénovanie správania robotov – napríklad pomocou neurónových sietí alebo pomocou evolučných algoritmov si spravidla vyžadujú výpočtovo oveľa náročnejšie procesy a rádovo dlhšie trénovacie doby. Je to najmä preto, že Bayesovské programovanie využíva všetky získané dáta v kontexte apriori modelu robota a jeho prostredia, zatiaľ čo stochastické prístupy inherentne musia „zahodiť“ a znovuobjaviť mnoho hypotéz, kým sa náhodou objavia cestu v prehľadávacom priestore. Silná stránka Bayesovského programovania je ale najmä vo využití pravdepodobností, t.j. automatického zaradenia neurčitosti sveta do modelu. Bayesovské programovanie ďalej v [2] využili na určovanie pozície svetelného zdroja na základe dát z viacerých senzorov (sensor fusion) a na určenie výsledného správania robota ako kombinácie viacerých vopred natrénovaných správání – v najkomplexnejšej úlohe bol robot natrénovaný na lokalizáciu ohňa, uhasenie, obchádzanie prekážok a návrat na nabíjačku – domovskú lokalitu.

4 Bayesovské evolučné programovanie robotov

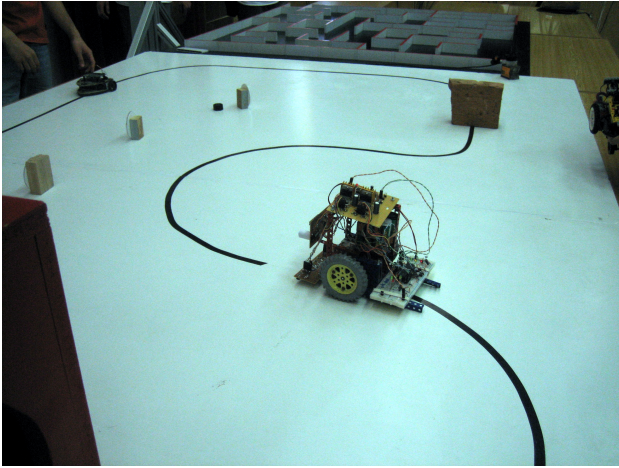
Bayesovské programovanie robotov má veľký potenciál a výhody súvisiace, nevýhodou uvedeného prístupu je, že na určenie experimentálnych dát je potrebný riadený

experiment, t.j. ide o typ učenia s učiteľom. Ak sa majú roboty úspešne pohybovať a plniť úlohy v neznámom prostredí, riadenie joystickom, alebo iný spôsob priameho zadania správnych dát často nie je možný. Naším príspevkom je návrh evolučného Bayesovského robotického programovania, t.j. nahradenie procesu zbierania dát zaradením modelu do evolučného algoritmu a použitím priamej empirickej skúsenosti – ohodnotením správania sa robota pomocou účelovej funkcie (objective function).

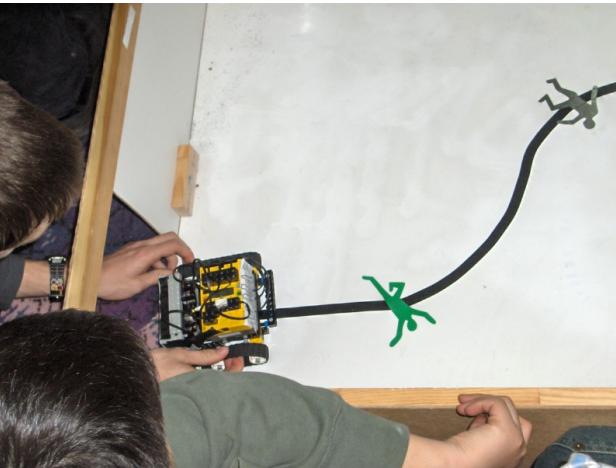
Princíp: Vychádzajúc z paradigmy Bayesovského programovania je pre nás jedným možným riešením v prehľadávacom priestore všetkých riešení konkrétne nastavenie parametrov pravdepodobnostných distribúcií, tak ako to bolo urobené na základe natrénovaných dát joystickom. V našom prípade však tieto dáta nie sú k dispozícii a preto evolučný algoritmus v stochastickom generačnom procese „uhádne“ správne nastavenie parametrov. Samotné vykonávanie programu (beh programu robota) je potom identické s prístupom Bayesovského programovania.

V našom prípade použijeme štandardný genetický algoritmus s genotypom pozostávajúcím z vektora reálnych čísel, s náhodne inicializovanými parametrami distribúcií. Rekombinačný operátor vezme dve distribúcie a vytvorí potomka s parametrami, ktoré sú priemerom rodičovských genotypov. Mutácia spôsobí náhodnú zmenu parametrov distribúcie. Účelová funkcia, podobne ako v experimentoch Evolučnej robotiky [3] otestuje riešenie v simulovanom svete a kvantitatívne ohodnotí vhodnosť výsledného správania robota.

Postup: Naším cieľom je overiť rámec evolučného Bayesovského programovania na experimentoch, ktoré vykonali v [2] a na jednoduchom experimente inšpirovanom kategóriou sledovania čiary v robotickej súťaži Istrobot a kategóriou Rescue v súťaži RoboCup Junior, obrázok 2 a obrázok 3. Robot sa nachádza na bielej ploche, kde jeho dráhu určuje hrubá čiara (vytvorená čiernou izolačnou páskou). Za pomoci svetelných senzorov je úlohou robota dostať sa na koniec dráhy, pričom na dráhe sa môžu vyskytovať prekážky, ktoré treba obísť, čiara môže byť prerušená na niekoľko desiatok centimetrov, môžu na nej byť 90-stupňové zákruty a v súťaži RoboCup Junior aj nalepené fiktívne obeť, ktoré treba označiť. Na overenie rámca najskôr vytvoríme Bayesovský program s trénovaním dát (náhodné premenné v tomto prípade sú: Čiara, Prerušenie_čiar, Obeť, Prudká_zákruta, Vzdialenosť_k_prekážke, Cieľ, Smer_pohybu) a následne sa pokúsime odhadnúť parametre distribúcií pomocou popísaného evolučného prístupu.



Obr. 2. Kategória Stopár v súťaži Istrobot. Foto: David Jablonovský.



Obr. 3. Kategória Rescue v súťaži RoboCup Junior. Foto: Andrej Pančík.

5 Záver

Informácie zo senzorov sú nepresné, zašumené a nespoľahlivé. Pravdepodobnostná a Bayesovská robotika sa zaoberá využitím pravdepodobnostného počtu a inferencie pre aplikácie robotiky. Bayesovské programovanie je možné v kombinácii s tréningovými dátami použiť na učiace sa robotické systémy, ich nedostatkom však je potreba učenia s učiteľom. Táto práca navrhuje modifikáciu tohto postupu kombináciou s prístupmi Evolučnej robotiky, kde pravdepodobnostné distribúcie nastavíme na základe stochastického prehľadania priestoru možných riešení. Doposiaľ sa objavili kombinované prístupy s opačným zameraním, t.j.

Bayesovské inferenčné mechanizmy boli využité na zlepšenie neinformovaného prehľadania priestoru pomocou evolučných algoritmov. Naším cieľom je duálna kombinácia týchto dvoch prístupov – teda využitie evolučných výpočtov v prospech Bayesovských modelov využívaných na riadenie robotických systémov.

Literatúra

- [1] S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics*, The MIT Press, Cambridge, Mass., 2005.
- [2] O. Lebeletel, P. Bessiere, J. Diard, E. Mazer: Bayesian Robot Programming. *Autonomous Robots* (2004), vol. 16, 49-79.
- [3] S. Nolfi, D. Floreano: *Evolutionary Robotics, The Biology, Intelligence, and Technology of Self-Organizing Machines*, The MIT Press, Cambridge, Mass., 2000.