Přírodou inspirované algoritmy 1

Vojtěch Šára

March 15, 2021

Evoluční algoritmy - podle Darwina
jednoduchý příklad:
binární čísla odpovídající číslům z množiny, které vybírám k součtu
snažím se trefit součtem do nějakého čísla.
Genetické algoritmy - typ evolučních - slouží k symbolické regresi

AI - výpočetní (evoluční algo, neuronové sítě, Fuzzy logika) / symbolická (reprezentace logikou, plánování, prohledávání)

0.1 Machine learning

1. s učitelem (klasifikace / regrese) - mám vstupní data spojená s výstupními, chci najít funkci, která to bude splňovat
2. bez učitele (shlukování / klustrování) - mám jen vstupní data, snažím se v nich najít nějaké zákonitosti
3. zpětnovazební učení (agent v prostředí) - má akce, stav, cíl, z toho se chci naučit nějakou strategii chování

0.2 Neuronové sítě

Dopředu spojené neuronové sítě - pokud mohou být i dozadu, tak jde o rekurentní sítě.
Konvoluční sítě

0.3 NEAT, hyperNEAT - neuroevoluce

Evoluční algoritmus nad neuronovou sítí. Možnost dělat jen váhy, nebo jen topologii, nebo váhy + topologii
Neuro Evolution Augmenting Topology - NEAT.

0.4 Rojové algoritmy

Mravenci, ptáci - lze pomocí toho například řešit nejkratší cestu

0.5 Artificial Life

Celulární automaty, Tierra, Creatures

1 Pravděpodobnost a statistika

Množina elementárních jevů - Ω Prostor jevů - $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ + axiomy:

1. $\emptyset \in \mathcal{F} \wedge \Omega \in \mathcal{F}$
2. $A \in \mathcal{F} \implies \Omega - \in$

$P : \mathcal{F} \Rightarrow [0, 1]$ + axiomy:

1. $P(\emptyset) = 0, P(\Omega) = 1$
2.

Pravděpodobnostní prostor je trojice $\Omega, \mathcal{F}, P$

Šance - "1 ku 2" $\frac{P(A)}{P(A^c)}$
Příklady
Diskrétní
Spojitý
Bernoulliho krychle

2 Druhá přednáška pravděpodobnosti

Zřetězené podmiňování:
$P(A \cap B) = P(B)P(A|B)$ Pokud $A_1, \ldots, A_n \in \mathcal{F}$ a $P(A_1 \cap \cdots \cap A_n) > 0$ tak:

$$P(A_1 \cap \cdots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)...P(A_n|\bigcap_{i=1}^{n-1} A_i)$$

Příklad: vytáhneme 3 karty z balíčku 52 karet, jaká je P(žádné srdce).
A_i = i-tá karta není srdce. $P(A_1 \cap A_2 \cap A_3)$ - podle vzorečku = $P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)$ = výsledek.

Věta o úplné pravděpodobnosti

Spočetný systém množin $B_1, B_2 ... \in \mathcal{F}$ je rozklad (partition) Ω pokud:

1. $B_i \cap B_j = \emptyset$
2. $\bigcup_i B_i = \Omega$

Pokud $B_1, B_2...$ je rozklad a $A \in \mathcal{F}$, tak:

$$P(A) = \sum_i P(B_i)P(A|B_i)$$

Sčítance s $P(B_i) = 0$ ignorujeme. Důkaz triviální z manipulace množin.

Příklad: Máme tři mince: P+O, P+P, O+O. Náhodně jednu vyberu a hodím s ní. Jaká je pravděpobodobnost, že padne orel?
Druhý příklad: Máme a korun, protihráč má b korun, hrajeme opakovaně spravedlivou hru o 1 Kč, dokud někdo nepřijde o vše. Jaká je pravděpobodobnost, že vyhrajeme?
To jsem nestihnul pobrat

Bayesova věta

Pokud $B_1, B_2...$ je rozklad, $A \in \mathcal{F}, P(A) > 0, P(B_j) > 0$ tak:

$$P(B_j|A) = \frac{P(B_j P(A|B_j))}{P(A)}$$

Příklad: test na COVID, $P(N|T) = 0.99, P(T|N) = 0.8$ podle Bayesovy věty:

$$P(N|T) = \frac{P(N)P(T|N)}{P(N)P(T|N) + P(N^c)P(T|N^c)}$$

Nezávislost jevů

Definice: dva jevy jsou nezávislé, pokud $P(A \cap B) = P(A)P(B)$. Pak také $P(A|B) = P(A)$ (pokud $P(B) > 0$).
Nezávislost více jevů - jevy $\{A_i : i \in I\}$ jsou nezávislé, pokud pro každou konečnou množinu $J \subseteq I$:

$$P(\bigcap_{i \in J} A_i) = \prod_{i \in J} P(A_i)$$

Pokud podmínka platí jen pro dvouprvkové množiny J, nazýváme jevy $\{A_i\}$ po dvou nezávislé.

Spojitost pravděpodobnosti

Nechť pro množiny z prostoru jevů platí:

$$A_1 \subseteq A_2...$$

a

$$A = \bigcup_{i=1}^{\infty} A_i$$

pak platí

$$P(A) = \lim_{i \Rightarrow \infty} P(A_i)$$

Náhodná veličina

Často nás zajímá číslo dané výsledkem náhodného pokusu. Funkci $X : \Omega \to \mathbb{R}$ nazveme diskrétní náhodnou veličinou, pokud obor hodnot X je spočetná množina a pokud pro všechna reálná x platí

$$\{\omega \in \Omega : X(\omega) = x\} \in \mathcal{F}$$

Pravděpodobnostní funkce (probability mass function, pmf) diskrétní náhodné veličiny je funkce $p_X : \mathbb{R} \to [0, 1]$ taková, že:

$$p_X(x) = P(X = x) = P(\{\omega \in \Omega : X(\omega) = x\})$$

3 Třetí přednáška z pravděpobodobnosti

4 Druhá přednáška Přírodou insp. algo

Zpětnovazební učení

Máme agenta a prostředí. Agent pozoruje prostředí a jedná v něm akcemi, dostává za to zpětnou vazbu - reward function.

Příklad - autíčko v ďolíčku.
Stav: x - poloha, v - rychlost
Akce: { jet doleva, jet doprava, zůstat v klidu } Reward function: -1 pokud x < 4 (auto není v cíli) jinak 0 - tím motivuji, aby se auto snažilo dostat do cíle co nejrychleji a zkoumalo nové stavy
Agent maximalizuje celkový reward, který za celý běh dostane.

Prostředí a akce

Prostředí může být diskrétní nebo spojité. U autíčka je spojité prostředí, ale akce jsou diskrétní. Kdyby autíčko mohlo otáčet volantem o libovolný reálný úhel, tak by i jeho akce byly spojité.
Dále řešíme zda prostředí je deterministické / nedeterministické. Stejně tak plně / částečně pozorovatelné. Jednoagentní / multiagentní.

Markovské rozhodovací procesy (MDP)

Je čtveřice (S, A, P, R), kde

1. S - množina stavů ve kterých může prostředí být
2. A - množina akcí (A_S akce které lzou udělat ve stavu $s \in S$)
3. P_a - binární funkce na stavech - přechodová funkce - pravděpodobnost, že když ve stavu s udělám akci a, tak prostředí přejde do stavu s'
4. R_a - binární funkce na stavech - reward function

Chování agenta - strategie (policy) $\pi : S \times A \to [0, 1], \pi(s, a)$ pravděpodobnost provedení akce a ve stavu s.
Cíl učení je najít π, která maximalizuje sumu odměn: $R = \sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$, $a_k = \pi(s_t)$ (akce provedená agentem v t). γ - diskontní faktor - aby suma nebyla nekonečná.
Takhle máme ve vzorečku nedeterminističnost schovanou v pravděpobodobnosti funkci π, lze vyřešit zabalením do středních hodnot.

Zavedeme dvě další funkce:
$V^\pi(s) = \mathbb{E}[R] = \mathbb{E}[\sum_0^\infty \gamma^t v_t | s_i = s]$ - hodnota stavu - očekávaná odměna pokud používám strategii π a začal jsem ve stavu s.
$Q^\pi(s, a)$ ... hodnota akce a ve stavu s = celková očekávaná odměna, pokud ve stavu s provedu akci a, potom pokračuji podle π.
Jsem ve stavu s, co udělám? Vyberu akci $\arg\max_{a \in A} Q(s, a)$.

Pokud máme ale jen funkci V, tak volím akci takto:

$$\arg\max_{a \in A} \sum_{s_t} P_a(s, s_t)[R_a(s, s_t) + \gamma V(s_t)]$$

Cíl je najít $\pi^\star$ t.ž.: $V^{\pi^\star}(s) = \max_\pi V^\pi(s)$.
Potřebujeme agenta motivovat, aby se pokoušel o nové strategie. Nebýt tedy greedy, ale ϵ-greedy - ta s pravděpobodobností ϵ vybírá akci (jen během učení). Tím eliminuji zaseknutí se v lokálním extrému.

Monte Carlo metoda pro odhad $Q^\pi(s, a)$

Simuluji chování agenta podle π, čímž "sampluji" funkci Q. Místo zcela náhodné strategie mohu volit π podle už zjištěných hodnot funkce Q (s nějakým ϵ, abych se nezaseknul v lokálním extrému). Velká nevýhoda - musím pořád spouštět simulaci - nevyužívám závislosti odměny za stav a za následující stav, tuto závislost modelují Bellmanovy rovnice. Co tyto závislosti započítává jsou tzv. temporal difference metody. Díky tomu se mi pak i propagují odměny časově odzadu dopředu.

Základní temporal difference metoda se jmenuje Q-učení.

Q-učení

$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a_t))$ Dělám v každém kroku. α je nějaký learning rate.

5 Evoluční algoritmy

Třída optimalizačních algoritmů. Typicky jsou maximalizační - což není omezení, minimalizaci můžeme vždy transformovat na maximalizaci. Na druhou stranu to není vždy úplně snadné a minimalizace je v matematice častější.
Jedinci - kandidáti na řešení - mohou to být vektory, čísla, i neuronové sítě.
Fitness funkce - jedinci $\to \mathbb{R}$ udává kvalitu řešení.

Inicializace - náhodní jedinci. Následuje ohodnocení - zvolím si jednoduché: maximalizuji počet jedniček v daném jedinci (vektoru jedniček a nul). Následuje selekce - vybírá jedince podle jejich fitness, ti co mají větší fitness mají i větší šanci přežít. Následuje křížení - kombinuje selektované jedince - v našem případě rozdělím vektory v náhodném bodě a prohodím části vektorů. Nakonec je mutace - prochází jedincem a náhodně ho změní - v našem případě flipne náhodné bity.

Ruletová selekce :

$$p_i = \frac{f_i}{\sum f_j}$$

Vlastnost této selekce je, že mohu odečítat / přičítat konstanty k fitnessům abych zdůraznil / zmírnil výběr nejsilnějších jedinců.
SUS - stochastic universal sampling.
Turnajová selekce - záleží pouze na uspořádání podle fitness, ne konkrétních hodnotách fitness.

Genetické operátory - křížení a mutace

Buď jedno nebo dvoubodové - jestli rozdělím a spojím dva jedince na jednom nebo dvou místech. Uniformní - na každé pozici buď prohodím nebo neprohodím.
Mutace - dokud mám vektory jedniček a nul velmi jednoduché - prohodím každý bit s malou pravděpodobností.