

ZÁKLADNÉ PRINCÍPY KOGNITÍVNYCH ALGORITMOV

(BASIC PRINCIPLES OF THE COGNITIVE ALGORITHMS)

Kristína MACHOVÁ

Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky Technickej univerzity v Košiciach,
Letná 9, 042 00 Košice, tel. 055/602 4152, E-mail: Kristina.Machova@tuke.sk

SUMMARY

This paper focuses on basic principles of learning algorithm design. It presents various principles employed in known learning algorithms dealing with different representations of concept definitions. It discusses pros and cons of application of the presented basic principles in relation with classification task. The focus is also on systematization of algorithm election or design in several steps. First, a problem / learning task is analyzed. Next, those of basic principles are selected, which match the given learning task requirements. Lastly, a general algorithm is designed in the form, which can be transformed into a selected programming language without difficulties.

Keywords: learning algorithm, basic principles, learning task, machine learning, metalearning

1. ÚVOD

V praxi sa často stretávame s postupom, v ktorom sa k riešeniu kognitívnej úlohy pristupuje tak, že sa vyskúša väčšia skupina algoritmov (viacmenej intuitívne). Následne sa vyberie jeden, dva algoritmy, ktoré pre danú úlohu dávajú najlepšie výsledky. Kognitívne algoritmy môžu byť rôzneho druhu, môžu realizovať učenie s učiteľom aj bez učiteľa, so zameraním na riešenie kognitívnych úloh rôzneho typu, a napokon môžu pracovať s rôznou reprezentáciou naučenej znalosti. Niektoré základné princípy majú však spoločné. Úlohou príspevku je abstrahovať z veľkej množiny kognitívnych algoritmov takzvané základné princípy. Daným základným princípom bude taktiež venovaný tento článok. Zameriame sa aj na súvislosti medzi základnými princípmi učiacich algoritmov a charakteristikami učiacich úloh. Tieto súvislosti môžu pomôcť pri systematizácii výberu jedného zo známych alebo návrhu nového kognitívneho algoritmu.

Budeme predpokladať, že na vstupe učiaceho algoritmu máme množinu pozorovaní teda množinu tréningových príkladov. Na výstupe je pojem, resp. trieda, ktorej popis alebo definíciu chceme získať z danej množiny príkladov. Na konkrétnej reprezentácii pojmu nám teraz nezáleží. Môže ísť o ktorúkoľvek z najčastejšie používaných reprezentácií: logickú konjunkciu, produkčné pravidlá, rozhodovacie stromy, rozhodovacie zoznamy, prahové pojmy, pravdepodobnostný popis, etalóny, neurónové siete a pod. Pojem, resp. trieda bude charakterizovaná skupinou atribútov a k nim vzťahovaných podmienok, ktorých splnenie postačuje na zaradenie nového pozorovania k danému pojmu (triede).

2. ZÁKLADNÉ PRINCÍPY

Základné princípy kognitívnych algoritmov môžeme rozdeliť do dvoch skupín na charakte-

ristické a dodatkové. K charakteristickým princípom patria: usporiadanie priestoru pojmov, horolezecký princíp, delenie priestoru príkladov na podpriestory, riadenie výnimkami a súťaživý princíp. K dodatkovým princípom patria: skórovacia funkcia a redukcia počtu kandidátov. Charakteristické princípy charakterizujú podstatu algoritmu, ktorý ich používa. Dodatkové princípy sú spravidla kombinované s niektorým z charakteristických princíпов. Charakteristické princípy sa medzi sebou tiež môžu kombinovať. V ďalšom budú rozobraté jednotlivé základné princípy a budú uvedené aj príklady algoritmov, v ktorých sa jednotlivé princípy využívajú. Podrobnejšie popisy týchto algoritmov je možné nájsť v [3], [4] a [6].

2.1. Usporiadanie priestoru pojmov

Jeden zo spôsobov, ako získať hľadaný pojem, je vytvoriť priestor všetkých pojmov, ktoré prichádzajú do úvahy ako riešenia a ten prehľadať. Prehľadávanie bude efektívnejšie, ak sa jednotlivé pojmy usporiadajú podľa zvoleného kritéria. Najčastejšie používaným kritériom je všeobecnosť. Priestor pojmov usporiadaný podľa všeobecnosti je možné znázorniť acyklickým grafom. Najvyššia úroveň tohto grafu reprezentuje najvšeobecnejší pojem (tzv. nulovú hypotézu), ktorej atribúty nie sú ohraničené. Žiadny atribút nie je stotožnený s konkrétnou hodnotou. Na najnižšej úrovni sa nachádzajú najmenej všeobecné pojmy, u ktorých je každý atribút konkretizovaný. Tieto pojmy predstavujú tréningové príklady. Smerom od najvyššej úrovne k najnižšej všeobecnosť klesá.

Pri prehľadávaní priestoru pojmov usporiadaného podľa všeobecnosti sa nám ponúkajú tri možné spôsoby prehľadávania: *od všeobecného k špecifickému*, *od špecifického k všeobecnému* a *paralelné prehľadávanie oboma smermi*.

V strojovom učení sa taktiež uplatňujú všetky klasické prehľadávacie metódy umelej inteligencie (do šírky, hĺbky a pod.).

Tento princíp využívajú učiace algoritmy VSS (Version Space Search) [5], EGS (Exhaustive General to Specific) a ESG (Exhaustive Specific to General). Algoritmy VSS sú príkladom prehľadávania obidvoma smermi.

2.2. Horolezecký princíp

Horolezecký princíp (hill-climbing) predstavuje všeobecnú prehľadávaciu stratégiu. Tento princíp je založený na gradientnom hľadaní extrému v lokálnom okolí vyšetřovaného bodu. V každej iterácii preskúma okolie aktuálneho (dosiaľ najlepšieho nájdeného) riešenia. Riešenie sa následne presunie do najslubnejšieho bodu tohto okolia. Nové riešenie bude mať nové okolie a opäť sa presunie do najslubnejšieho bodu tohto nového okolia, čím sa získa ďalšie lepšie riešenie. Slubnosť riešenia sa môže merať nejakou skórovacou funkciou. Nevýhodou týchto algoritmov je, že síce dokážu nájsť lokálny extrém rýchlo, ale potom v ňom uviaznu, keď sa v okolí nenachádza lepšie riešenie. Preto sa používajú na dotiahnutie do extrému, ktorého okolie bolo nájdené inou metódou.

Príkladom algoritmu, využívajúceho tento princíp je algoritmus IWP (Iterative Weight Perturbation) (generuje prahové pojmy), SADE, SOMA [10].

2.3. Delenie priestoru príkladov na podpriestory

Princíp delenia priestoru príkladov na podpriestory je jedným z prvých princípov, ktoré sa začali v rámci strojového učenia uplatňovať. V literatúre ho môžeme často nájsť pod názvom „separuj a panuj“.

Priestor tréningových príkladov sa rekurzívne delí na podpriestory až dovtedy, kým nie je splnená ukončovacia podmienka. Napríklad, kým každý podpriestor neobsahuje iba príklady jednej triedy. Delenie na podpriestory sa uskutočňuje na základe splnenia podmienok. Podmienka sa vyberá pomocou informačnej teórie tak, aby mala najnižšiu entropiu, teda maximálny informačný zisk.

Tento princíp sa využíva v nasledovných algoritmoch: NSC (Nonincremental Separate and Conquer), AQ11 (generujú produkčné pravidlá), ID3, ID5R, C4.5, MDPL (generujú rozhodovacie stromy).

2.4. Riadenie výnimkami

Podstata princípu riadenia výnimkami spočíva v tom, že v každej iterácii učiaceho algoritmu sa všetky tréningové príklady klasifikujú do daných tried. Pre tréningové príklady, ktoré sú chybné klasifikované, teda pre výnimky sa vytvorí nové triedy. Tento proces sa opakuje dovtedy, kým nie sú všetky tréningové príklady správne klasifikované. Ukončovacou podmienkou môže byť vykonanie

daného maximálneho počtu iterácií. Algoritmus môže ukončiť svoju činnosť aj vtedy ak viacero nových iterácií neprineslo lepšie výsledky.

Výnimkami sú riadené také algoritmy ako NCD (Nonincremental Induction of Competitive Disjunctions) a ICD (Incremental Inductions of Competitive Disjunctions) [3], ktoré generujú etalóny a NEX (Nonincremental Induction of Decision List with Exclusions), ktorý generuje rozhodovacie zoznamy.

2.5. Súťaživý princíp

Súťaživý princíp je založený na tom, že rôzne potenciálne pojmy, resp. triedy sa ohodnotia pomocou nejakej hodnotiacej funkcie. Vyberie sa tá trieda, ktorá bude hodnotiacou funkciou najlepšie ohodnotená. Príkladom takejto hodnotiacej funkcie môže byť pravdepodobnosť triedy podmienená hodnotami atribútov daného klasifikovaného príkladu. Ďalšou možnosťou je vzdialenosť klasifikovaného príkladu (napríklad euklidova) od typických reprezentantov jednotlivých tried (etalónov).

Tento princíp je využívaný Bayesovským klasifikátorom a algoritmi, ktoré generujú etalóny: NCD (Nonincremental Induction of Competitive Disjunctions) a ICD (Incremental Inductions of Competitive Disjunctions) [3].

2.6. Skórovacia funkcia

Ak chceme prehľadávaním priestoru pojmov nájsť optimálne riešenie, prehľadáme celý priestor pojmov. Avšak úplné prehľadávanie priestoru pojmov je pomalé a náročné na čas a pamäť. Alternatívou je systém s prehľadávacími preferenciami (search bias), ktorý bude niektoré pojmy uvažovať skôr ako ostatné. Taký systém musí definovať skórovaciu funkciu, ktorá najslubnejší pojem ohodnotí najvyšším číslom. Vo všeobecnosti výsledok tejto funkcie narastá priamo úmerne s P_c počtom pokrytých pozitívnych tréningových príkladov a s N_{nc} počtom nepokrytých negatívnych tréningových príkladov. Dobrá metrika berie do úvahy aj celkové množstvo pozitívnych P a negatívnych N príkladov.

$$Score = \frac{P_c + N_{nc}}{P + N}$$

Uvedená funkcia **Score** môže nadobúdať hodnoty z intervalu $\langle 0,1 \rangle$.

Zložitejšie prístupy používajú štatistické alebo informačné miery, napríklad výpočet entropie. Ako skórovacia funkcia sa môže taktiež použiť signifikancia, ktorá vychádza z definície entropie [2], [4]. Je závislá na takých veličinách ako: počet tréningových príkladov N , počet tried R , počet tréningových príkladov triedy r N_r , počet všetkých tréningových príkladov pokrytých pravidlom K a počet tréningových príkladov triedy r pokrytých

pravidlom K_r . Táto funkcia môže nadobúdať reálne hodnoty.

$$Signif = 2 \sum_{r=1}^R K_r \log_2 \frac{K_r N}{N_r K}$$

Princíp skórovacej funkcie využívajú také algoritmy ako HGS (Heuristic General to Specific), HSG (Heuristic Specific to General) (generujú logické konjunkcie) a HCT (Heuristic Criteria Tables) (generuje tabuľku kritérií). Taktiež ho používajú algoritmy generujúce rozhodovacie stromy ID3 [7], ID5R a C4.5 [8], pričom ako skórovaciu funkciu používajú informačný zisk. Tento princíp využíva aj algoritmus CN2 (generuje rozhodovacie zoznamy aj neusporiadané množiny pravidiel) [2] a ako skórovaciu funkciu využíva vyhodnotenie signifikancie.

2.7. Redukcia počtu kandidátov

Spravidla pri úplnom prehľadávaní priestoru pojmov, máme v každej ďalšej iterácii algoritmu viac pojmov ako v predchádzajúcej. (V závislosti od konkrétnej trérovacej množiny môže nastať aj taký prípad, keď počet pojmov od iterácii k iterácii nenarastá.) Keď je v jednej iterácii, resp. na jednej úrovni všeobecnosti v usporiadanom priestore pojmov, príliš veľa pojmov, hovoríme o širokom priestore pojmov. Prehľadávanie takéhoto priestoru by mohlo zapríčiniť zahltenie počítača. Ak sa vzdáme požiadavky nájdania optimálneho riešenia, nemusíme prehľadávať celý priestor pojmov. Môžeme prehľadávanie priestoru pojmov obmedziť v každej iterácii na určitý počet kandidátov pojmov. Ide vlastne o takzvanú metódu lúčového prehľadávania (beam search).

Výhodné je počet kandidátov pojmov určiť tesne pred spustením učiaceho algoritmu používateľom, t.j. zadať ho ako parameter učiaceho algoritmu. Tento parameter sa zvykne označovať ako BS faktor (z anglického Beam Size). Ak už budeme prehľadávať iba niektoré pojmy, mali by to byť tie najslubnejšie. Preto sa kandidáti pojmov v každej iterácii usporiadajú podľa hodnôt skórovacej funkcie a vyberie sa iba BS najslubnejších pojmov.

V súvislosti s princípom redukcie počtu kandidátov rozoznávame takzvanú tvrdú zaujatosť alebo tvrdé preferencie (hard bias) a mäkkú zaujatosť, resp. mäkké preferencie (soft bias). V prvom prípade sú z prehľadávania tvrdo vopred vylúčené niektoré typy pojmov. V druhom prípade nič nie je vopred vylúčené, teda ide o mäkkšiu zaujatosť. Niektorým pojmom (napríklad lepšie ohodnoteným skórovacou funkciou) sa pri prehľadávaní dá prednosť.

Princíp skórovacej funkcie kombinovaný s princípom redukcie počtu kandidátov vytvára takzvaný heuristický prístup k riešeniu učiacej úlohy, ktorý predstavuje kompromis medzi presnosťou a efektívnosťou. Napokon, horolezecký

princíp je možné považovať za špeciálny prípad lúčového prehľadávania pri BS=1.

Princíp redukcie počtu kandidátov využívajú taktiež algoritmy HGS, HSG (generujú logické konjunkcie) a HCT (generuje tabuľku kritérií).

3. NÁVRH ALGORITMU

Návrhu algoritmu spočíva vo výbere základných princípov algoritmov, ktoré by bolo vhodné uplatniť vzhľadom na charakter učiacej úlohy.

Napríklad princíp úplného prehľadávania usporiadaného priestoru pojmov môžeme použiť na riešenie takmer každej učiacej úlohy. Otázne je s akým výsledkom. Môže sa stať, že pre veľký rozsah trérovacích údajov bude trvať riešenie veľmi dlho, prípadne nám nebude stačiť pamäť počítača (priestor pojmov je veľmi široký). Navyše použitie princípu usporiadania priestoru pojmov vyžaduje definíciu operátorov, ktoré umožnia pohyb v priestore pojmov.

Ak máme zložitú úlohu a veľký rozsah trérovacích údajov, môžeme sa rozhodnúť pre prehľadávanie priestoru pojmov, ale s dopĺňujúcim využitím princípu skórovacej funkcie a redukcie počtu kandidátov.

Dodatočné využitie princípu skórovacej funkcie a redukcie počtu kandidátov pri prehľadávaní priestoru pojmov sa hodí aj pri úlohách, kde hrá svoju úlohu čas a riešenie potrebujeme urýchliť, pričom nemusíme trvať na optimálnom riešení.

Ak máme riešiť učiacu úlohu, ktorá je charakteristická zašumenými údajmi, môžeme použiť princíp riadenia výnimkami alebo princíp skórovacej funkcie.

Niekedy sa musíme potýkať s problémom veľkého rozptylu príkladov jednej triedy medzi príklady ostatných tried, čo znamená, že príklady jednotlivých tried nie sú lineárne separabilné. V takom prípade môžeme s úspechom použiť princíp delenia priestoru príkladov na podprieštory resp. „separuj a panuj“, alebo aj princíp riadenia výnimkami.

Ak má náš učiaci problém vlastnosť lineárnej separability, môžeme použiť horolezecký princíp.

Niektoré kombinácie základných princípov sú bežné, ako napríklad kombinácia princípu usporiadania priestoru pojmov s princípom skórovacej funkcie a redukcii počtu kandidátov. Je možné tvoriť aj netradičné kombinácie. Napríklad použitie princípu delenia priestoru príkladov na podprieštory a v rámci jednotlivých podprieštorov aplikovať princíp prehľadávania usporiadaného priestoru pojmov.

Na základe doteraz uvedených skutočností, je možné systematizovať návrh učiaceho algoritmu nasledovným spôsobom:

- Analyzujeme problém resp. učiacu úlohu.
- Na základe výsledkov tejto analýzy, zvolíme tie zo základných princípov, ktoré vyhovujú danej učiacej úlohe.

- Zostavíme všeobecný algoritmus, ktorý by vhodným spôsobom kombinoval vybrané princípy.
- Prepíšeme tento algoritmus do vhodného programovacieho jazyka.

Keď sa už venujeme kombinovaniu rôznych základných princípov v rámci jedného učiaceho algoritmu, bolo by vhodné spomenúť, že jestvuje aj iná možnosť a to kombinovanie rôznych algoritmov založených na rôznych princípoch pri riešení tej istej úlohy. Takým spôsobom boli riešené predikcie špičiek odberu vody a analýza návštevnosti zámok [9]. Tieto úlohy spadali do oblasti dolovania znalostí (data-mining), ktorá využíva pri riešení vlastných problémov aj algoritmy strojového učenia.

4. PRÍKLAD POUŽITIA ZÁKLADNÝCH PRINCÍPOV

Predstavme si, že by sme mali klasifikovať podozrivú bankovú operáciu. Našou množinou príkladov by bolo obrovské množstvo záznamov o bankových operáciách uskutočnených napríklad v priebehu jedného roka. Mohlo by sa stať, že by sme museli zvažovať údaje z niekoľkých bánk. Úplné prehľadávanie priestoru pojmov by neprichádzalo do úvahy. Preto by bolo vhodné použiť princíp prehľadávania priestoru pojmov v kombinácii s heuristickou funkciou a redukciou počtu hypotéz.

Uvažujme problém diagnostiky veľmi zriedkavých diagnóz nejakej špecializovanej oblasti medicíny. Ide o problém klasifikácie do nie veľkého počtu tried. Taktiež tréningová množina nie je príliš početná. Je totiž veľmi malé percento pacientov, trpiacich na dané ochorenia. Tréningové údaje sa však vyznačujú veľkým rozptylom príkladov jednej triedy medzi príklady ostatných tried, pretože uvažované diagnózy majú mnohé z príznakov, vyskytujúcich sa u pacientov, rovnaké alebo podobné. V takom prípade by bolo vhodné priestor príkladov rozdeliť použitím princípu „rozdeľuj a panuj“ (princípu delenia priestoru pojmov na podpriestory) a nad jednotlivými podpriestormi použiť princíp usporiadania priestoru pojmov. Usporiadané podpriestory by sa dali úplne prehľadať. Pre danú učiacu úlohu by mohlo byť zaujímavé použitie kombinácie princípu „separuj a panuj“ s princípom riadenia výnimkami. Princíp riadenia výnimkami totiž formuje pre chybné klasifikované príklady niektorej triedy novú – pseudotriedu. To by mohlo viesť k odhaleniu novej choroby podobného, charakteru, ktorá je trebárs zapríčinená novým zmutovaným druhom vírusu alebo baktérie.

5. ZÁVER

Úlohou tohto príspevku bolo z veľkého množstva kognitívnych algoritmov známych dnes v strojovom učení abstrahovať základné princípy

a charakterizovať ich. Uvedené základné princípy vlastne vytvárajú jednotlivé časti akejsi „skladačky“. Z častí tejto skladačky, je možné poskladať ľubovoľný algoritmus podľa objednávky, teda v závislosti od charakteru riešenej úlohy, od platiacich obmedzení, požiadavok na rýchlosť a presnosť spracovania, od požadovanej reprezentácie a od kvality tréningových údajov. Túto takzvanú skladačku je možné využiť tak pri výbere jedného z existujúcich algoritmov, ako aj pri návrhu nového kognitívneho algoritmu v prípade, že žiadny existujúci kognitívny algoritmus nevyhovuje špeciálnym požiadavkám našej kognitívnej úlohy. Táto úloha spadá do oblasti metaučenia (metalearning). Cieľom je získať množinu pravidiel na určenie najlepšieho kognitívneho algoritmu pre danú kognitívnu úlohu. Tieto pravidlá by mohli byť súčasťou znalostného systému so zameraním na metaučenie. V [1] je uvedená metóda, ktorá získava tieto pravidlá kontrolovaným učením (napríklad algoritmom učenia založeného na prípadoch (Instance Based Learning). Na popis úlohy sa používajú značky (landmarks), ktoré môžu byť rôzneho druhu: rozhodovací uzol, ľubovoľne vybraný uzol, najhorší uzol, naivný Bayes, 1NN (prvý najbližší sused), elitný 1NN a lineárny diskriminátor. Hodnota značky sa určuje ako priemerná chyba nad priestorom príkladov. Daná metóda používa aj meta-atribúty založené na informačnej teórii (entropia triedy, priemerná entropia atribútov, mutačná informácia, ekvivalentný počet atribútov a pod.). Tento prístup je však v značnej miere založený na veľkom počte experimentov pri ktorých sa meria chyba klasifikácie pre rôzne kombinácie učiacich algoritmov a databáz.

Ďalší výskum by mohol byť zameraný na všeobecne platné súvislosti kognitívnych algoritmov a kognitívnych úloh.

LITERATÚRA

- [1] Bensusan, H., Giraud-Carrier, C.: Casa Batlo is in Passeig de Gracia or how landmark performances can describe tasks. <http://www.metal-kdd.org/>.
- [2] Clark, P., Niblett, T.: The CN2 Induction Algorithm. *Machine Learning*, Vol.3, No.4, 1989, 261-284.
- [3] Langley, P.: *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1996, 419.
- [4] Machová, K.: *Strojové učenie. Princípy a algoritmy*. Elfa s.r.o., Košice, 2002, 117s.
- [5] Mitchell, T.M.: Version spaces: A candidate elimination approach to rule learning. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA: Morgan Kaufmann, 1977a, 305-310.

- [6] Mitchell, T.M.: Machine Learning. The McGraw-Hill Companies, Inc. New York, 1997b, 414.
- [7] Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 1, 1986, 81-106.
- [8] Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, California, 1993, ISBN 1-55860-238-0.
- [9] Rauber, A., Paralič, J.: Cluster Analysis as a First Step in the Knowledge Discovery Process. Journal of Adv. Comp. Intelligence, Vol.4, No.4, Fuji Technology Press Ltd., Japan, 2000, 158-162.
- [10] Zelinka, I.: Umělá inteligence v problémech globální optimalizace. BEN – technická literatura, Praha, 2002, 189, ISBN 80-7300-069-5.

BIOGRAPHY

Kristína Machová was born 16.02.1961. In 1985 she graduated (MSc.) with honors at the Department of cybernetics of the Faculty of Electrical Engineering at Technical University in Košice. She defended her PhD. in the field of machine learning in 1996; her thesis title was "Knowledge modification with the aid of explanation based learning". Since 1985 she is working as a lecturer at the Department of Cybernetics and AI of the Faculty of Electrical Engineering and Informatics at Technical University. Since 1988 she is working as an assistant professor at the same department. His scientific research is focusing on machine learning and metalearning. In addition to this, he also investigates the questions related to the basic principles of the learning algorithms.