SYMBOLIC REGRESSION USING CARTESIAN GENETIC PROGRAMMING

Michaela Šikulová
Master Degree Programme (2), FIT BUT
E-mail: xsikul06@stud.fit.vutbr.cz

Supervised by: Lukáš Sekanina
E-mail: sekanina@fit.vutbr.cz

Abstract: Symbolic regression is the problem of identifying the mathematical description of a hidden system from experimental data. Symbolic regression is closely related to general machine learning. This work deals with symbolic regression and its solution based on the principle of cartesian genetic programming. Cartesian genetic programming is an evolution-based machine learning metod, which automaticaly generates whole programs represented as directed acyclic graph.

Keywords: symbolic regression, cartesian genetic programming

1 ÚVOD

Symbolická regrese je považována za základní úlohu strojového učení, která se zabývá identifikací matematického popisu skryté závislosti experimentálně získaných dat. Trénovací data v úloze symbolické regrese typicky sestávají ze stovek až tisíců bodů. Předpokládá se, že výstupní hodnota závisí na vstupní (závislost měřených hodnot lze vyjádřit funkcí).

Hledání takovéto funkce vyžaduje prohledávání rozsáhlých prohledávacích prostorů, k čemuž je nutné využít heuristické stochastické algoritmy. Mezi takové algoritmy patří genetické programování a jeho různé varianty.

2 KARTÉZSKÉ GENETICKÉ PROGRAMOVÁNÍ

Genetické programování umožňuje automaticky generovat celé programy v určitém programovacím jazyce. John Koza jej navrhl jako modifikaci genetických algoritmů pro tvorbu programů v rámci symbolické regrese. Tyto programy byly reprezentovány syntaktickým stromem. V kartézském genetickém programování (CGP), jak je uvedeno v [1], je kandidátní řešení reprezentováno pomocí orientovaného acyklického grafu, který je zakódován jako lineární pole celých čísel. Vstupy (terminály) a výstupy uzlů jsou postupně očíslovány. Funkce, které mohou vykonávat uzly, jsou také postupně očíslovány. Genotyp je vyjádřen jako seznam propojení uzlů a jejich funkce. Genotyp se následně mapuje na graf, který reprezentuje vykonání programu. Program mapovaný z genotypu se nazývá fenotyp a představuje kandidátní funkci, která je předmětem symbolické regrese. Výstup jednoho uzlu je možné připojit na vstupy více uzlů, což stromová hierarchie neumožňuje.

CGP používá jako primární genetický operátor pro získaní nových kandidátních řešení mutaci. Při mutaci se náhodně vybere gen v chromozomu jedince a nahradí se náhodně vygenerovanou hodnotou, která splňuje kritéria – pro gen na konkrétní pozici existuje množina přípustných hodnot. V důsledku mutace se může změnit funkce uzlu nebo index připojeného vstupu a tím se může změnit i topologie propojení grafu. Genotyp kandidátního řešení má stejnou velikost pro všechny jedince v populaci. Velikost fenotypu se mezi jedinci liší podle toho, kolik aktivních uzlů je na cestě mezi vstupy a výstupy kandidátního programu.

Na počátku algoritmu CGP se náhodně vygeneruje počáteční populace $\lambda + 1$ jedinců. Evoluce pak běží v cyklu, dokud není nalezeno dostatečně kvalitní řešení nebo není dosaženo maximálního počtu generací. V každé iteraci evolučního cyklu se ohodnotí všichni jedinci z populace pomocí fitness funkce. Pak se vybere nejlépe ohodnocený jedinec v populaci. Z nejlepšího jedince se vygeneruje λ jedinců pomocí mutace. Nejlepší jedinec spolu se svými λ potomky tvoří novou populaci.

3 SYMBOLICKÁ REGRESE POMOCÍ CGP

Algoritmus pro symbolickou regresi pomocí kartézského genetického programování jsem implementovala v jazyce C. S ohledem na maximalizaci rychlosti výpočtu jsou všechny potřebné datové struktury definovány staticky (na základě parametrů evoluce), s výjimkou trénovacích dat, která jsou na začátku běhu programu načtena ze souboru.

Graf reprezentující genotyp (kartézský program) je modelován jako pole uzlů umístěných v mřížce o velikosti $n_r \times n_c$ (počet řádků $\times$ počet sloupců). Každý z uzlů má dva vstupní argumenty (pro binární funkce, unární funkce jako e^x potom využívá pouze první vstup uzlu) nad nimiž realizuje jednu z funkcí z množiny dostupných funkcí. Každý uzel má jeden výstup. Dále má kartézský program primární vstupy (v případě funkce jedné proměnné jeden primární vstup), jeden primární výstup jako funkční hodnotu a parametr l-back. Parametr l-back určuje, kolik předchozích sloupců (výstupů uzlů z předchozích sloupců) lze propojit s uzlem v aktuálním sloupci.

3.1 VÝPOČET FITNESS KANDIDÁTNÍHO ŘEŠENÍ

Jednou z možností výpočtu fitness kandidátních řešení, jak je uvedeno v [2], je spočítání střední absolutní chyby bodů, které jsou v trénovací množině dat, a které generuje kandidátní řešení následovně:

$$\text{fitness}(s) = \frac{1}{n}\sum_{i=1}^{n} |s(x_i) - y_i|, \tag{1}$$

kde s je kandidátní řešení (algebraický výraz), x_i je trénovací vstup funkce a y_i je výstup funkce z trénovací množiny, n je počet vzorků v trénovací množině dat. Čím menší má kandidátní řešení hodnotu fitness, tím je kvalitnější.

Druhým způsobem, jak určit fitness kandidátního řešení, je pomocí přidělování bodů (*score*, uvedeno v [3]) za správný výsledek kandidátního řešení pro daný datový bod. Jestliže výsledek řešení pro daný datový bod odpovídá funkční hodnotě datového bodu s definovanou chybou σ, tak se k fitness hodnotě přičte 1, v opačném případě 0. Čím větší má kandidátní řešení hodnotu fitness, tím je kvalitnější.

$$\text{fitness}(s) = \sum_{i=1}^{n} g(s(x_i)), \text{ kde } g(s(x_i)) = \begin{cases} 0 & \text{pro } |s(x_i) - y_i| \geq \sigma \\ 1 & \text{pro } |s(x_i) - y_i| < \sigma \end{cases} \tag{2}$$

3.2 EXPERIMENTY SE SYMBOLICKOU REGRESÍ POMOCÍ CGP

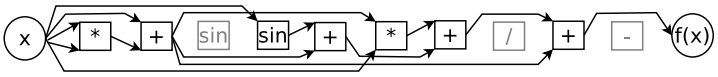
Trénovací data byla vytvořena vzorkováním (dvě stě vzorků pro x z intervalu $\langle -10, 10 \rangle$) funkcí:

$$f_1(x) = x^2, \tag{3}$$

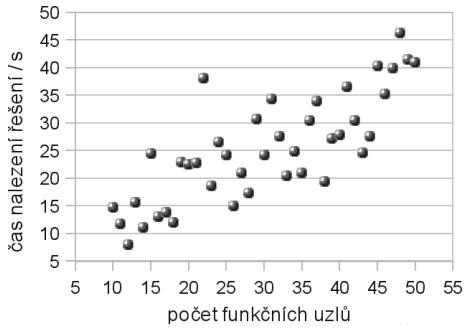
$$f_2(x) = sin(x) + x(x+1) \cdot (x+2), \tag{4}$$

$$f_3(x) = x^2 e^{\sin(x)} + x + \sin\left(\frac{\pi}{4 - x^3}\right). \tag{5}$$

Následně se pro každou funkci hledalo řešení pomocí různých nastavení parametrů evoluce (pro každé nastavení 50 běhů) a vytvářela se statistika. Ukázka nalezeného řešení úlohy f_2 pro 10 uzlů v chromozomu je zobrazena na obrázku 1.



Obrázek 1: Nalezené řešení funkce $f_2(x)$ dané vztahem 4 pro 10 uzlů v chromozomu.



Obrázek 2: Zobrazení závislosti doby, za kterou je nalezeno řešení, na počtu uzlů v chromozomu.

Pro hledání matematického popisu závislosti se ukázala jako vhodná možnost plného propojení kartézského programu, čehož lze docílit nastavením počtu řádků na 1, a parametru l-back na maximum (počet sloupců). Počet funkčních uzlů v grafu je pak roven počtu sloupců n_c.

Jedním ze sledovaných parametrů byla i délka chromozomu. Pro funkce se složitějším průběhem, jako je funkce f_2, byla pravděpodobnost nalezení přesného řešení v průběhu 30 000 generací (8 jedinců v populaci, 3 mutace na vytvářeného potomka) nad 50 % až od deseti uzlů v chromozomu. Avšak se zvyšujícím se počtem uzlů rostla i doba potřebná pro ohodnocení jednoho jedince a tím i celková doba výpočtu (zobrazeno na obrázku 2).

4 ZÁVĚR

Výsledkem tohoto projektu je funkční aplikace, která řeší úlohu symbolické regrese pomocí kartézského genetického programování (CGP). V návaznosti na tuto práci se nyní zabývám symbolickou regresí a jejím řešením založeném na principu genetického programování a koevoluce. Koevoluce fitness prediktorů (jak je uvedeno v [2]) je optimalizační metoda modelování fitness, která snižuje náročnost a frekvenci výpočtu fitness. Mým cílem je srovnání symbolické regrese s užitím koevoluce fitness prediktorů s řešením bez užití koevoluce.

PODĚKOVÁNÍ

Tato práce vznikla v rámci projektu FIT-S-11-1 a MSM 21630528.

REFERENCE

[1] Miller, J. M.; Thomson, P.: Cartesian Genetic Programming. In Proc. of the Third European Conference on Genetic Programming (EuroGP2000), 2000: s. 121-132.

[2] Schmidt, M. D.; Lipson, H.: Coevolution of Fitness Predictors. IEEE Transactions on Evolutionary Computation, ročník 12, č. 6, 2008-12: s. 736-749, ISSN 1089-778X.

[3] Koza, J. R.: Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Dept. of Computer Science, Stanford University, 1990, Technical report STAN-CS-90-1314.