Algoritmy řízeného náhodného výběru v regresní analýze

Ivan Křivý

1. dubna 1994

1 Úvod

Příspěvek je věnován algoritmům řízeného náhodného výběru (CRS algoritmům) a jejich využití při určování odhadů parametrů v nelineárních regresních modelech. Je dobře známo, že přímý výpočet takových odhadů není možný. Numerické metody používané pro nelineární modely lze zhruba klasifikovat takto (viz [2,5,11]):

- metody založené na výpočtu derivací nějaké vhodné funkce regresních parametrů, zpravidla součtu čtverců odchylek (gradientní metoda, Newtonova-Gaussova metoda a jejich modifikace);
- přímé metody (simplexová metoda, řada dalších metod založených na náhodném výběru).

CRS algoritmy patří zřejmě do kategorie přímých metod, jež pracují pouze s funkčními hodnotami.

2 Priceův algoritmus

Pojem CRS algoritmu zavedl jako první Price [5] k označení svého algoritmu pro stanovení globálního minima dané funkce, algoritmu založeného na efektivním spojení simplexové metody [4] s technikou náhodného výběru do jediného spojitého procesu. Předpokládejme, že chceme v parametrickém prostoru $\Omega = D_1 \times D_2 \times \ldots \times D_n$, kde D_i jsou neprázdné množiny (definiční obory parametrů $p_1, p_2, \ldots, p_n$), určit globální minimum funkce $f(p_1, p_2, \ldots, p_n)$. Výchozí oblast V parametrického prostoru Ω se definuje tím, že se specifikují podmínky kladené na jednotlivé parametry. Pak se v této oblasti náhodně vybere předem zvolený počet N bodů $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N$, $\mathbf{P}_k = [p_{k1}, p_{k2}, \ldots, p_{kn}]$, $k = 1, 2, \ldots, N$, a spočtou se hodnoty funkce f ve všech vybraných bodech. Konfigurace takové N-tice bodů je v paměti počítače popsána N vektory typu $(p_{k1}, p_{k2}, \ldots, p_{kn}; y_k)$, kde $y_k = f(p_{k1}, p_{k2}, \ldots, p_{kn})$. V aktuální konfiguraci se zvolí náhodně $n+1$ různých bodů $\mathbf{P}_{i_1}, \mathbf{P}_{i_2}, \ldots, \mathbf{P}_{i_{n+1}}$ $(n \ll N)$ takových, že tvoří simplex v euklidovském prostoru $\mathbf{E}_n$. Nový zkušební (trial) bod $\mathbf{P}$ se definuje předpisem

$$\mathbf{P} = 2\mathbf{G} - \mathbf{R} \tag{1}$$

v němž $\mathbf{R}$ představuje náhodně zvolený vrchol (pól) simplexu $\mathbf{P}_{i_1}\mathbf{P}_{i_2}\ldots\mathbf{P}_{i_{n+1}}$ a $\mathbf{G}$ těžiště soustavy zbývajících n vrcholů. Z hlediska geometrického je nový bod $\mathbf{P}$ obrazem vrcholu $\mathbf{R}$ ve středové souměrnosti se středem v těžišti $\mathbf{G}$. Pokud nový bod $\mathbf{P}$ leží v oblasti V, spočte se odpovídající funkční hodnota $f(\mathbf{P})$. Je-li splněna podmínka $f(\mathbf{P}) < f(\mathbf{M})$, kde $\mathbf{M}$ značí bod, jemuž přísluší nejvyšší funkční hodnota ze všech N bodů uvažovaných v dané iteraci, nahradí se bod $\mathbf{M}$ v paměti novým bodem $\mathbf{P}$. V případě, kdy platí $f(\mathbf{P}) > f(\mathbf{M})$, stejně jako v případě, že bod $\mathbf{P}$ neleží v oblasti V, je nově generovaný bod zavržen a hledá se jiný bod $\mathbf{P}$ již popsaným postupem. Pro danou konfiguraci N bodů existuje zřejmě $(n+1)\binom{N}{n+1}$ stejně pravděpodobných možností, jak vybrat nový bod $\mathbf{P}$. V každém okamžiku se v paměti uchovává aktuální informace o N v jistém smyslu nejlepších bodech z oblasti V.

V průběhu výpočtu se aktuální konfigurace N bodů uvnitř oblasti V chová tak, že její body mají tendenci vytvářet shluk kolem minima, kterému přísluší nižší hodnota, než je právě platná hodnota $f(\mathbf{M})$. Pravděpodobnost, že průběžně aktualizovaná konfigurace „konverguje" ke globálnímu minimu funkce f v oblasti V, a také rychlost této konvergence závisí na hodnotě N, složitosti funkce f, jejíž globální minimum hledáme, povaze omezujících podmínek kladených na parametry a způsobu výběru nových bodů.

Priceův algoritmus má všechny přednosti přímých metod, tj. funkce f nemusí mít derivace a parametry mohou nabývat diskrétních hodnot. Klade důraz na úplnost prohledávání oblasti V, přitom je mnohem efektivnější než prostý náhodný výběr. Nemá mimořádné nároky na kapacitu operační paměti, a to ani v případě vysoké dimenze řešeného problému (velkého počtu parametrů).

Základní nevýhodou popsaného algoritmu je pomalá konvergence, typická pro přímé postupy. Testování algoritmu (viz [5]) ukázalo, že nalezení globálního extrému vyžaduje, aby bylo provedeno řádově několik tisíc iterací.

Urychlení konvergence Priceova algoritmu je možno dosáhnout v podstatě dvěma způsoby:

- Vrcholy simplexu pro určení nového bodu se nevybírají náhodně z celé aktuální konfigurace N bodů, ale jen z nějaké její podmnožiny, jež zahrnuje body s nejnižšími funkčními hodnotami.

- Při generování nového bodu pomocí vztahu (1) se za vrchol $\mathbf{R}$ simplexu nebere libovolný z jeho vrcholů, ale právě ten vrchol, kterému odpovídá nejvyšší funkční hodnota.

Obě uvedené modifikace Priceova algoritmu vedou sice k urychlení konvergence extremalizačního procesu, ovšem pouze na úkor úplnosti prohledávání oblasti V parametrického prostoru.

3 Heuristická metoda PAROPTMULTI

Metoda PAROPTMULTI (viz [8,9]) byla navržena pro multikriteriální optimalizaci a identifikaci simulačních modelů. Předpokládá se existence dobrého simulačního modelu sledovaného systému a optimum se hledá cestou experimentování s jednotlivými variantami tohoto modelu, jež se liší právě hodnotami svých parametrů. Varianty simulačního modelu se vyvíjejí paralelně v čase. Zatímco nevhodné varianty jsou v průběhu simulace vyřazovány, nové varianty se v parametrickém prostoru generují pomocí heuristických algoritmů založených na řízeném náhodném výběru. Cílem experimentování je nalézt nějaké vhodné varianty ve smyslu předem zvoleného vektorového (vícesložkového) kritéria.

Metoda PAROPTMULTI obohacuje klasické optimalizační postupy o následující prvky a principy:

- další heuristické algoritmy pro generování nových variant modelu (bodů v parametrickém prostoru);
- princip střídání vyhledávacích algoritmů během výpočtu na základě průběžného hodnocení jejich úspěšnosti;
- vektorové kritérium pro posuzování kvality jednotlivých variant modelu z různých rozumných hledisek;
- pomocné skalární kritérium pro stanovení pořadí (uspořádání) variant daného modelu.

3.1 Vyhledávací algoritmy

Předpokládejme, že je nutno optimalizovat celkem n parametrů. Pak každá varianta modelu může být reprezentována bodem z prostoru $[0, 1]^n$. Navržené algoritmy realizují v zásadě zobrazení

$$([0, 1]^n)^L \longrightarrow [0, 1]^n;$$

vycházejí tedy z nějaké L-tice bodů v prostoru $[0, 1]^n$ a generují nový bod v témže prostoru. Jednotlivé vyhledávací algoritmy se liší také hodnotou L (počtem výchozích bodů). Souřadnice výchozích bodů se vybírají střídavě ze tří následujících, průběžně aktualizovaných seznamů:

- úplný seznam všech „živých" variant;
- seznam obsahující údaje o třech nejlepších (ve smyslu pomocného skalárního kritéria) variantách vytvořených prostým náhodným výběrem;
- seznam obsahující údaje o třech absolutně nejlepších variantách bez ohledu na algoritmus jejich generování.

Funkci jednotlivých vyhledávacích algoritmů (názvy podle autora) lze stručně charakterizovat takto.

- RANDOM provádí prostý náhodný výběr v prostoru $[0, 1]^n$. Souřad- nice nového bodu jsou realizací náhodného výběru z rovnoměrného rozdělení na intervalu $[0, 1]$.
- AMI představuje modifikaci Priceova algoritmu řízeného náhodného výběru.
- CENTRING pracuje tak, že se jedna (náhodně vybraná) souřadnice nového bodu určí jako realizace náhodného výběru z rovnoměrného rozdělení na intervalu $[0, 1]$ a ostatní souřadnice tohoto bodu jako prostý aritmetický průměr příslušných souřadnic ve všech variantách zařazených do právě zpracovávaného seznamu.
- FLOP vytváří zrcadlový obraz bodu reprezentujícího poslední (tedy nejhorší) variantu zpracovávaného seznamu podle těžiště bodů odpovídajících všem variantám zařazeným do tohoto seznamu.

- GEN vychází z absolutně nejlepší varianty. Hodnoty všech souřadnic nového bodu výjma jediné (náhodně vybrané) se rovnají příslušným souřadnicím nejlepší varianty. Zbývající souřadnici se přiřadí hodnota této souřadnice pro první položku seznamu nejlepších variant vytvořených prostým náhodným výběrem.

- HAZARD pracuje podobně jako algoritmus GEN , jen s tím rozdílem, že se hodnota zbývající (náhodně vybrané) souřadnice určí jako realizace náhodného výběru z rovnoměrného rozdělení na intervalu [0, 1].

- BROWN pracuje tak, že každé souřadnici nově vytvářeného bodu přiřadí hodnotu téže souřadnice ve variantě náhodně vybrané (pro každou souřadnici zvlášť) ze zpracovávaného seznamu variant.

- INTERVAL přiřazuje všem souřadnicím nového bodu (s výjimkou jediné, náhodně vybrané) hodnoty odpovídajících souřadnic pro první (nejlepší z hlediska skalárního kritéria) variantu zpracovávaného seznamu. Hodnota zbývající souřadnice se pak určí jako pseudonáhodné číslo z rovnovnoměrného rozdělení na intervalu $[a^*, b^*]$, kde

$$a^* = \max(0, (a+b)/2 - 5 \mid a - b \mid),$$

$$b^* = \min((a+b)/2 + 5 \mid a - b \mid, 1),$$

  a (b) je hodnota této souřadnice pro první (poslední) variantu právě zpracovávaného seznamu.

Každý z CRS algoritmů obsahuje navíc test, zda právě nevytvořil absolutně nejlepší variantu ve smyslu pomocného skalárního kritéria. Jestliže je tomu tak, pak je nový bod generován ve směru vektoru určeného předcházející a právě generovanou rekordní variantou. V případě vytvoření další rekordní varianty se taková extrapolace opakuje.

3.2 Střídání vyhledávacích algoritmů

Popsané vyhledávací algoritmy se v průběhu optimalizace střídají. Kritériem pro opakované zařazení určitého algoritmu do výpočtu je jeho úspěšnost při generování nových variant. Mechanismus rozhodování o nasazování jednotlivých algoritmů do optimalizačního procesu je zřejmý z následujícího fragmentu zdrojového textu programu (v jazyku SIMULA):

```
CYCLE: AMI; while AGAIN do AMI;
       BROWN; while AGAIN do BROWN;
       CENTRING; while AGAIN do CENTRING;
       FLOP; while AGAIN do FLOP;
       GEN; while AGAIN do GEN;
       HAZARD; while AGAIN do HAZARD;
       INTERVAL; while AGAIN do INTERVAL;
       RANDOM; goto CYCLE;
```

V tomto schematickém zápisu značí AGAIN volání booleovské procedury rozhodující o tom, který vyhledávací algoritmus se uplatní v následujícím kroku. Pokud má tato procedura hodnotu true, opakuje se algoritmus z předchozího kroku; jinak se zařadí do výpočtu algoritmus uvedený na bezprostředně následujícím řádku.

3.3 Vektorové kritérium

Podstatným přínosem metody PAROPTMULTI je zavedení vektorového kritéria optimalizace. To znamená, že na rozdíl od Priceova algoritmu nejde pouze o nalezení globálního extrému jediné funkce, ale o stanovení takových hodnot parametrů, které odpovídají optimálním hodnotám všech složek vektorového kritéria. Vektorové kritérium kvality hraje významnou roli při vyřazování variant z optimalizačního procesu. Postačující podmínkou pro okamžité vyřazení nějaké varianty M ze skupiny sledovaných variant je existence takové varianty $\tilde{M}$, která je ve všech složkách vektorového kritéria lepší než vyřazovaná varianta M.

3.4 Pomocné skalární kritérium

Toto kritérium kvality dovoluje zavést na množině všech „živých" variant modelu úplné uspořádání. Jeho definice je přirozená. Jednotlivým složkám c_i ($i = 1, 2, \ldots, m$) vektorového kritéria c se přiřadí subjektivní statistické váhy $w_i \geq 0$. Celkové skalární ohodnocení varianty M je pak dáno výrazem

$$\sum_{i=1}^{m} w_i d_i,$$

kde d_i značí počet těch živých variant modelu, které jsou v i-té složce vektorového kritéria c dominovány uvažovanou variantou M. Skalární kritérium se uplatní při redukci počtu paralelně zpracovávaných živých variant modelu v tom případě, kdy jejich počet překročí zadanou horní mez. Použití skalárního kritéria nezaručuje vyřazení nejhorší varianty či nejhorších variant. Nicméně pravděpodobnost toho, že takto dojde k eliminaci opravdu kvalitní varianty (z hlediska některé složky vektorového kritéria), je zřejmě malá.

Aplikace metody PAROPTMULTI vyžaduje, aby uživatel zajistil:

- doprogramování všeho, co souvisí s řešením jeho konkrétního problému,
- řízení průběhu optimalizace, zejména rozhodování o jejím ukončení.

V uživatelském programu se definují:

- struktura experimentálních dat,
- vlastní model včetně předpisu pro výpočet hodnot a vah jednotlivých složek vektorového kritéria kvality,
- zobrazení z prostoru $[0, 1]^n$ do skutečného parametrického prostoru Ω,
- omezení kladená na hodnoty parametrů modelu.

Řídící údaje pro optimalizaci zahrnují počet parametrů modelu, počet složek vektorového kritéria kvality, násadu pro generátor pseudonáhodných čísel, počet průběžně sledovaných živých variant modelu, horizont simulace, způsob diskretizace simulovaného období a režim výpočtu (automatický nebo interaktivní). Uživatel může zafixovat hodnoty některých parametrů modelu, popř. vkládat expertní varianty (nulté aproximace hodnot parametrů modelu). Celé simulované období se v závislosti na řídících datech rozloží na dílčí časové intervaly. Na konci každého z nich má uživatel k dispozici aktualizovaný seznam všech živých variant uspořádaný podle hodnot pomocného skalárního kritéria. V těchto okamžicích

se rozhoduje o počtu nově vytvořených a analyzovaných variant v následujícím časovém intervalu. Po dosažení simulačního horizontu může uživatel restartovat výpočet s tím, že nejlepší, resp. nejzajímavější varianty modelu jsou do opakované simulační studie vloženy znovu jako expertní varianty.

Ve srovnání s Priceovým algoritmem má metoda PAROPTMULTI dvě významné přednosti:

- Střídání vyhledávacích CRS algoritmů garantuje rozumný kompromis mezi úplností hledání v parametrickém prostoru a rychlostí konvergence optimalizačního procesu.
- Zavedení vektorového kritéria umožňuje daleko víc než odhadnout polohu globálního extrému jediné kriteriální funkce. Uživatel může nalézt nejen optimální řešení svého problému, ale i posoudit celou řadu dalších zajímavých řešení z pohledu různých kritérií. Při vyřazování variant (na základě dominance lepší variantou či na základě hodnocení pomocí skalárního kritéria) nemůže dojít k zavržení varianty, jež je v nějakém ohledu nejlepší.

Základní nedostatky metody PAROPTMULTI spatřujeme v tom, že

- optimalizační proces konverguje pomaleji než při použití metod založených na výpočtu derivací;
- metoda neposkytuje žádné informace o přesnosti získaných odhadů parametrů (varianční matici odhadů).

4 Výsledky experimentů s programovým produktem PAROPTMULTI

Programový prostředek pro optimalizaci metodou PAROPTMULTI dodává firma TIMING [10] ve formě samostatně kompilované třídy PARMULTI. (Zdrojový text je v jazyku SIMULA.) Zmíněná třída se v uživatelském programu deklaruje jako externí.

Uživatelský program je nutno zapsat v jazyku SIMULA. Vzhledem k tomu, že jde o jazyk objektově orientovaný, doporučujeme formulovat vlastní regresní model i strukturu dat obecně ve formě třídy (obecné znalosti). Jednotlivé regresní modely se pak liší jen tvarem regresní funkce.

Při testování produktu PAROPTMULTI jsme použili vektorového kritéria o těchto čtyřech složkách:

- reziduální součet čtverců (RSS),
- medián čtvercových odchylek (LMS),
- součet absolutních odchylek (SAD),
- maximální absolutní odchylka.

V úvahu přicházejí i další (méně běžná) kritéria jako např. S-odhad [7] nebo „oříznutý" součet čtverců [6].

Produkt PAROPTMULTI jsme ověřovali zhruba na 30 souborech dat, jednak experimentálních (vlastních), jednak modelových, převzatých z literatury [1-3]. Pokud jde o modelová data, byla vybrána vesměs taková, jejichž zpracování pomocí klasických postupů založených na výpočtu derivací (gradientní metoda, Newtonova-Gaussova metoda) působilo značné problémy.

Výsledky experimentů ukazují, že k nalezení solidních odhadů postačí vygenerovat a vyhodnotit v průměru 1000 – 2000 variant daného modelu, což při práci na personálním počítači kompatibilním s IBM PC/AT486 představuje 2-4 s času CPU. Závěrečné upřesnění odhadů jsme prováděli pomocí programu Nonlinear Regression Analysis (Marquardtův algoritmus) v rámci statistického programového systému SOLO. Toto upřesnění bylo nutné mimi jiné i proto, že produkt PAROPTMULTI poskytuje hodnoty odhadů s přesností jen na dvě platné číslice.

Posuzujeme-li kvalitu získaných odhadů podle finální hodnoty RSS, pak můžeme konstatovat, že naše výsledky jsou srovnatelné s těmi, které se považují v literatuře za nejlepší.

Pro ilustraci možností PAROPTMULTI uvádíme aspoň tři příklady.

4.1 Příklad 1

Data (převzata z práce [1])

$$y = 2 + 2x \qquad \text{pro } x = 1, 2, 3, \ldots, 10$$

Model:

$$\exp(\beta_1 x) + \exp(\beta_2 x)$$

Počáteční aproximace:

$$\mathbf{b}^0 = (0,3;\ 0,4), \qquad RSS(\mathbf{b}^0) \approx 5,2 \cdot 10^2$$

Výsledek optimalizace (1000 variant):

$$\mathbf{b} = (0,26;\ 0,26), \qquad RSS(\mathbf{b}) = 1,2 \cdot 10^2$$

Výsledek upřesnění (3 iterace):

$$\mathbf{b}^* = (0,25781;\ 0,25781), \qquad RSS(\mathbf{b}^*) = 124,36$$

4.2 Příklad 2

Data v tab. 1 (převzata z práce [2])

Model:

$$\frac{\beta_1 \beta_3 x_1}{1 + \beta_1 x_1 + \beta_2 x_2}$$

Počáteční aproximace:

$$\mathbf{b}^0 = (10,39;\ 48,83;\ 0,74), \qquad RSS(\mathbf{b}^0) = 0,0365$$

Výsledek optimalizace (1000 variant):

$$\mathbf{b} = (3,7;\ 15;\ 0,69), \qquad RSS(\mathbf{b}) = 5,2 \cdot 10^{-5}$$

Výsledek upřesnění (5 iterací):

$$\mathbf{b}^* = (3,132;\ 15,16;\ 0,780), \qquad RSS(\mathbf{b}^*) = 4,355 \cdot 10^{-5}$$

x_1	x_2	y
1.0	1.0	0.126
2.0	1.0	0.219
1.0	2.0	0.076
2.0	2.0	0.126
0.1	0.0	0.186

Tabulka 1: Data k příkladu 2

x	y
12	7.31
13	7.55
14	7.80
15	8.05
16	8.31
17	8.57
18	8.84
19	9.12
20	9.40
21	9.69
22	9.99
23	10.30

Tabulka 2: Data k příkladu 3

4.3 Příklad 3

Data v tab. 2 (převzata z práce [3])
Model:

$$\beta_1 x^{\beta_3} + \beta_2 x^{\beta_4}$$

Počáteční aproximace:

$$\mathbf{b}^0 = (100;\ 0,1;\ 2;\ 10), \qquad RSS(\mathbf{b}^0) = 2,9 \cdot 10^{25}$$

Výsledek optimalizace (15000 variant):

$$\mathbf{b} = (3,5;\ 0,0045;\ 0,25;\ 2,0), \qquad RSS(\mathbf{b}) = 3,9 \cdot 10^{-3}$$

Výsledek upřesnění (6 iterací):

$$\mathbf{b}^* = (3,7861;\ 0,003947;\ 0,2253;\ 2,0731), \qquad RSS(\mathbf{b}^*) = 2,984 \cdot 10^{-5}$$

5 Závěr

V příspěvku jsou stručně zhodnoceny přednosti a nedostatky CRS algoritmů a posouzena možnost jejich využití ve formě programového prostředku PAROPTMULTI ke stanovení

odhadů parametrů v nelineárních regresních modelech. Ověřování na reálných i modelových datech naznačuje, že uvedený produkt je s to poskytnout odhady regresních parametrů, jež jsou co do kvality (hodnoty RSS) srovnatelné s výsledky speciálních postupů založených na výpočtu derivací (viz např. [1,2]). PAROPTMULTI nabízí rozumný kompromis mezi dvěma základními a přitom protichůdnými požadavky kladenými na přímé metody, tj. mezi úplností prohledávání parametrického prostoru a rychlostí konvergence výpočetního procesu. Jeho základní nedostatky lze prakticky eliminovat tím, že se zajímavé varianty regresního modelu následně upřesní pomocí některého ze standardních programů pro regresi.

Literatura

[1] Jennrich, D.T. – Sampson, P.F.:*Technometrics.* 10, no. 1, 1968, 63–72.

[2] Meyer, R.R. – Roth, P.M. : *J. Inst. Math. Applics.* 9, 1972, 218-233.

[3] Militký, J. – Meloun, M.: *Talanta.* 40, no. 2, 1993, 269–277.

[4] Nelder, J.A. – Mead, R.: *Comput. J.* 7, 1965, 308–313.

[5] Price, W.L.: *Comp. J.* 20, no. 4, 1976, 367–370.

[6] Rousseeuw, P.J.: *J. Am. Statist. Assoc.* 79, 1984, 871–880.

[7] Rousseeuw, P.J. – Yohai, V.: *Robust Regression by Means of S - Estimates.* In: Robust and Nonlinear Time Series Analysis. Editors: J. Franke, W. Hardle, R.D. Martin. Lecture Notes in Statistics. Vol. 26. New York, Springer 1984, 256–272.

[8] Weinberger, J.: *Comp. & Art. Intel.* 6, no. 1, 1987, 71–79.

[9] Weinberger, J. : *Programové prostředky pro optimalizaci a identifikaci simulačních modelů složitých systémů.* Kandidátská dizertace. Praha 1991.

[10] Weinberger, J.: *PAROPTMULTI. Programový systém pro interaktivně řízenou kvaziparalelní, multikriteriální optimalizaci simulačních modelů.* Praha, firma TIMING 1990.

[11] Zvára, K.: *Regresní analýza.* Praha, Academia 1989.