Odhad regresních parametrů metodou řízeného náhodného výběru

Josef Tvrdík

1. dubna 1994

Úvod

Simplexovou metodu minimalizace funkce obohatil Price [6], který algoritmus rozšířil o řízený náhodný výběr (controlled random search – CRS) vrcholů simplexu z vektoru dosud zkoumaných a zapamatovaných možných vrcholů. Algoritmus je popsán i v tomto sborníku [1]. Ve stručnosti uvedeme důležité kroky:

Algoritmus vyhledává bod $B = (b_1, b_2, \ldots, b_n)$ v prostoru parametrů Ω, ve kterém funkce $f(b_1, b_2, \ldots, b_n)$ nabývá minima. Prostor parametrů $\Omega = D_1 \times D_2 \times \ldots \times D_n$, D_i je obor hodnot parametru b_i , například interval $D_i = \langle bmin_i, bmax_i \rangle$, $bmin_i$ je nejmenší a $bmax_i$ největší možná hodnota parametru b_i, $i = 1, 2, \ldots, n$.

V parametrickém prostoru Ω se náhodně vybere N bodů, $P_1, P_2, \ldots, P_N$ a v každém tomto bodě se vypočte hodnota funkce z_i

$$z_i = f(p_{i1}, p_{i2}, \ldots, p_{in}), \qquad i = 1,\ 2,\ \ldots,\ N$$

Tato N-tice vektorů $(p_{i1}, p_{i2}, \ldots, p_{in}, z_i)$ se uspořádá vzestupně podle hodnot z_i a udržuje se v paměti během výpočtu s tím, že N-tá složka tohoto pole vektorů (t.j. složka s maximální hodnotou minimalizované funkce) se postupně nahrazuje nějakým novým vhodně zjištěným vektorem $(p_1^*, p_2^*, \ldots, p_n^*, z^*)$ takovým, pro který platí $z^* < z_N$. Vektor $(p_1^*, p_2^*, \ldots, p_n^*, z^*)$ se do N-tice vektorů zařazuje tak, aby zůstalo zachováno vzestupné uspořádání podle hodnot z_i .

Nové hodnoty parametrů se $(p_1^*, p_2^*, \ldots, p_n^*)$ se získávají simplexovou metodou, $n+1$ vrcholů simplexu se z N-tice dosud zapamatovaných vektorů $(p_{i1}, p_{i2}, \ldots, p_{in})$ vybírá náhodně. Algoritmus můžeme zapsat přehledněji (i když ne co do podrobností zcela přesně) v pseudokódu:

```
čti N
for i:= 1 to N do
    vygeneruj P_i ∈ Ω
    vypočti z_i
endfor
seřaď (p_i1, p_i2, ..., p_in, z_i) vzestupně podle z_i
repeat
    repeat
        vyber n + 1 bodů
```

```
        najdi zkušební bod P* = (p1*, p2*, ..., pn*)
    until P* ∈ Ω
    vyhodnoť z*
    if z* < zN then
        zařaď (p1*, p2*, ..., pn*, z*) do N-tice zapamatovaných bodů
        ITERACE := ITERACE+1
    endif
until splněno kritérium pro ukončení výpočtu
```

Úspěšnost algoritmu při hledání globálního minima v prostoru parametrů a rychlost, s jakou k tomuto minimu bude konvergovat, závisí na řadě dalších „detailů", které prozatím v zápisu algoritmu nejsou obsaženy. Pokud tvar funkce $f(p_1, p_2, \ldots, p_n)$ je dán a parametrický prostor Ω vymezen, pak mezi tyto „detaily" patří zejména

- hodnota N (počet zapamatovaných bodů pro náhodný výběr vrcholů simplexu),
- způsob výběru nového zkušebního bodu P^*,
- definice podmínky pro ukončení výpočtu.

Pro implementaci algoritmu jsou tyto faktory podstatné a je těžké předem rozhodnout o jejich obecně přijatelné volbě.

S rostoucí hodnotou N roste pravděpodobnost, že opravdu bude nalezeno globální minimum, ale současně se zvětšují paměťové a zejména časové nároky programu. Price doporučuje volit hodnotu N přibližně 25 krát vyšší než je počet argumentů minimalizované funkce, t.j. $N \sim 25n$.

Pro výběr nového zkušebního bodu P^* uvádí Price předpis

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

kde R je náhodně vybraný bod z $(n+1)$ vrcholů simplexu, G je těžiště zbývající vrcholů. Zkušební bod P^* je tedy bod středově symetrický s bodem R kolem těžiště G. Nelder a Mead [5] užívají pro výběr nového zkušebního bodu v simplexové metodě vztah

$$P^* = G - \gamma(R - G) \tag{2}$$

kde γ je nějaká vhodně určená hodnota. Zvára [7] uvádí, že γ má většinou hodnoty z $\{0.5, 1, 2\}$. Vidíme, že rov. (1) je zvláštním případem rov. (2) pro $\gamma = 1$. Je zřejmé, volba hodnoty γ bude mít vliv na rychlost, s jakou se bude vyhledávání blížit bodu, ve kterém je minimum funkce.

Definice kritéria konvergence (podmínky pro ukončení výpočtu) má rovněž vliv na časové nároky programu. Lze očekávat, že k nalezení minima je potřeba mnoha iterací. Kritérium pro ukončení je nutno vyhodnocovat v každém kroku vnějšího cyklu repeat. Proto je vhodné zvolit podmínku pro ukončení tak, aby její vyhodnocení bylo časově nenáročné.

Modifikovaný algoritmus řízeného náhodného výběru.

Po shrnutí poznatků z literatury a uvedených úvah jsme zkoumali několik variant algoritmu řízeného náhodného výběru. Vzhledem k tomu, že cílem bylo užít modifikovaný algoritmus v programu pro odhad regresních parametrů nelineárních regresních modelů, minimalizovanou funkcí je reziduální součet čtverců RSS

$$\mathrm{RSS} = \sum_{i=1}^{n_{obs}} (y_i - \hat{y}_i)^2 \tag{3}$$

kde y_i jsou naměřené hodnoty závisle proměnné, $\hat{y}_i$ jsou odhady vypočtené z odhadu parametrů regresního modelu a n_{obs} je počet pozorování (rozsah výběru). Jako kritérium konvergence je užita relativní diference nejhoršího a nejlepšího zapamatovaného odhadu RSS vzhledem k celkovému součtu čtverců závisle proměnné TSS.

$$\mathrm{TSS} = \sum_{i=1}^{n_{obs}} (y_i - \bar{y}_i)^2 \tag{4}$$

kde $\bar{y}$ je průměr pozorovaných hodnot závisle proměnné, ostatní symboly mají stejný význam jako v předcházející rovnici (3). Výpočet končí, jestliže

$$\frac{\mathrm{RSS}_N - \mathrm{RSS}_1}{\mathrm{TSS}} \leq \epsilon_0 \tag{5}$$

kde ϵ_0 je zadaná vstupní hodnota.

Na různých datech bylo zkoušeno několik variant výběru zkušebního vrcholu simplexu P^*. Nejpodstatnější zkrácení času výpočtu přinesla „znáhodňující“ modifikace rov. (2), kdy místo pevné hodnoty γ byla užito

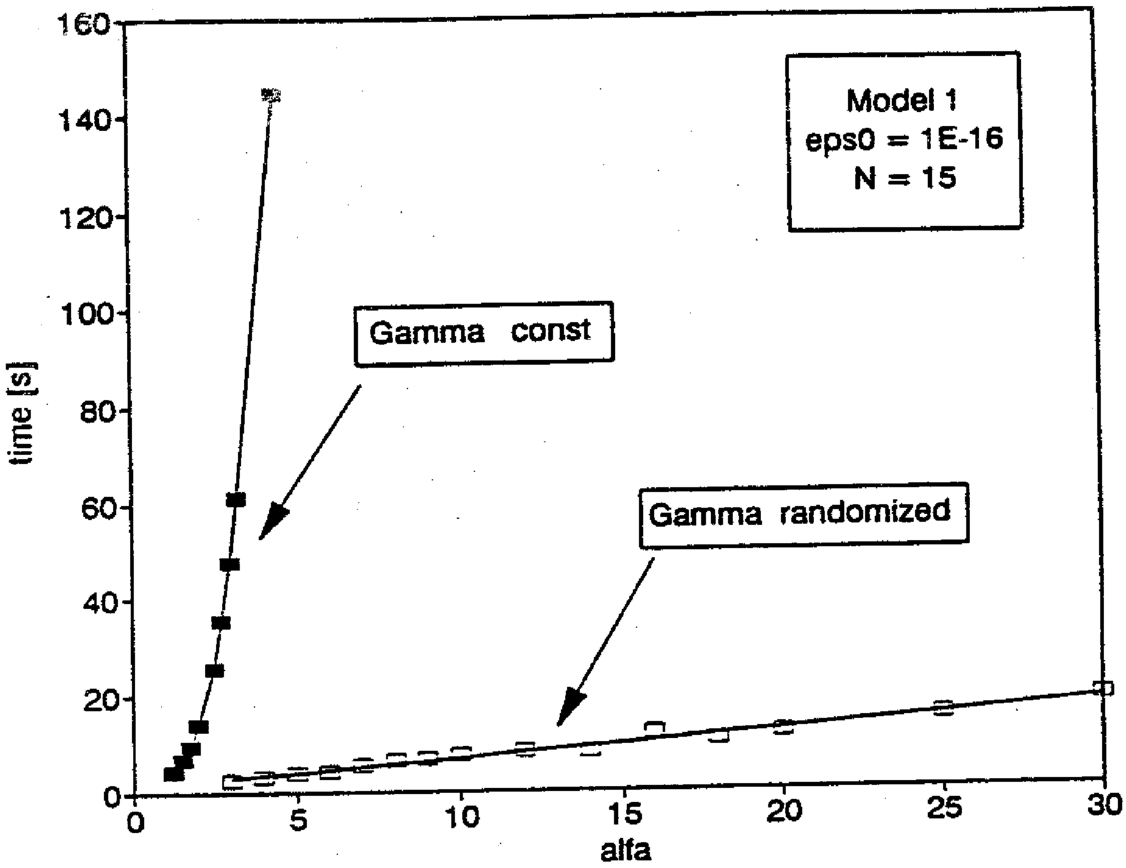
$$\gamma = \alpha X, \tag{6}$$

kde α je vstupní údaj a X je náhodná veličina. Empiricky byla ověřována tato rozdělení:

1. X rovnoměrně rozdělená na intervalu $\langle 0, 1\rangle$
2. $X = Y_1 Y_2$, kde Y_1, Y_2 jsou nezávislé veličiny, rovnoměrně rozdělené na intervalu $\langle 0, 1\rangle$
3. X z přibližně normálního rozdělení N (2, 0.333)

Ukázalo se, že znáhodnění ve většině případů podstatně zrychluje konvergenci k minimu ve srovnání s algoritmem, kdy hodnota γ je v průběhu výpočtu konstantní. Dále bylo zjištěno, že nejkratší časy výpočtu byly dosahovány s variantou 1 - veličina X rovnoměrně rozdělená na intervalu $\langle 0, 1\rangle$. Potěšující je i zjištění, že narozdíl od algoritmu s konstantní hodnotou $\gamma = \alpha$ při využití znáhodnění (6) není čas výpočtu příliš závislý na zadané hodnotě pro poměrně široký interval těchto hodnot. Typickou závislost doby výpočtu na hodnotě vstupního parametru ukazuje obr. 1. Ze zkušenosti s dalšími testovacími příklady lze odvodit doporučení: pokud vstupní zadaná hodnota je z $\langle 4, 8\rangle$, není doba výpočtu příliš vzdálená od minimální hodnoty.

Znáhodnění koeficientu γ rov. (6) dovoluje snížit i hodnotu N, t.j. počet bodů, které je nutno si zapamatovávat), aniž by se neúnosně zvýšilo riziko nenalezení globálního minima. Podle zkušeností s odhadem regresních parametrů několika modelů uváděných v literatuře [3] lze doporučit hodnoty N podstatně nižší, než uvádí Price [6]. Ve většině zkoumaných regresních modelů se osvědčily hodnoty N odpovídající zhruba pětinásobku počtu argumentů minimalizované funkce, t.j. $N \sim 5n$. Tato nižší hodnota N snižuje nejen paměťové nároky algoritmu, ale především podstatně zkracuje čas výpočtu.



Obr. 1: Závislost doby trvání výpočtu na koeficientu α

Program pro odhad regresních parametrů metodou CRS

Na základě uvedených výsledků pokusů s různými variantami algoritmu CRS byl navržen a napsán program MOR0 (MOje Regrese, verse 0). Program je určen k odhadu regresních parametrů v nelineárních modelech. Je to verse pro počítačově zkušenější uživatele, neboť tvar regresního modelu je nutno zadávat jako text ve zdrojovém programovacím jazyku Turbo Pascal a tuto část programu je nutno před výpočtem vždy zkompilovat.

V programu MOR0 je možno si zvolit kromě minimalizace reziduálního součtu čtverců i jiná kritéria, která mají být minimalizována, a to buď součet absolutních odchylek SABSDEV,

$$\mathrm{SABSDEV} = \sum_{i=1}^{n_{obs}} |y_i - \hat{y}_i| \tag{7}$$

nebo maximální absolutní odchylku MABSDEV

$$\mathrm{MABSDEV} = \max_{i=1,2,\ldots,n_{obs}} |y_i - \hat{y}_i|$$

Vstupními daty programu kromě tvaru regresního modelu a volby minimalizovaného kritéria jsou:

n	počet odhadovaných parametrů regresního modelu, dovoleno $1 \leq n \leq 10$
$bmin_i$, $bmax_i$	$i = 1, 2, \ldots, n$, meze parametrického prostoru

Tyto vstupní parametry se zadávají v samostatném textovém souboru s povinnou příponou .PAR.

Dalším vstupem programu jsou naměřená vstupní data v textovém souboru čitelném jako matice dat v programovém prostředí ESTAT [8]. Rozsah dat je omezen na maximálně 10 000 naměřených hodnot v datové matici, závisle proměnná se předpokládá v prvním sloupci. Přímo z klávesnice se dále zadávají:

N počet zapamatovávaných bodů prostoru parametrů, dovolena hodnota $n < N \leq 500$, doporučená hodnota $N = 5n$

α koeficient z rov. (6), dovolena libovolná hodnota typu real, doporučeno $4 < \alpha < 8$

ϵ_0 požadovaná hodnota kritéria pro ukončení výpočtu, viz vztah (5). Dovolena libovolná kladná hodnota typu real, doporučená hodnota 1E-16 pro minimalizaci RSS, 1E-8 pro minimalizaci SABSDEV, 1E-6 pro minimalizaci MABSDEV. Kritéria pro ukončení výpočtu pro minimalizaci SABSDEV a MABSDEV jsou definována analogicky se vztahem (5).

Průběh výpočtu (počet iterací, aktuální hodnoty minimalizovaného kritéria a regresních parametrů atd) je zobrazován na displeji po každých dvou sekundách. Výpočet může být ukončen zásahem uživatele (vhodné v situaci, kdy odhady parametrů se delší dobu nemění nebo výpočet zjevně nekonverguje ke globálnímu minimu). Výsledky jsou zobrazeny na displeji a uloženy i do textového souboru pro případné další zpracování.

Program je napsán a odladěn v jazyku Turbo Pascal, verse 6.0. Program sestává z hlavního modulu MOR0.PAS, do kterého se automaticky vkládá textový soubor YHAT.INC, ve kterém je tvar regresního modelu napsán v Pascalu. Hlavní modul MOR0.PAS je nutno vždy před výpočtem zkompilovat. Součástí programu jsou jednotky (units) UMOR0 a UMORTYPY. Dále program MOR0 využívá jednotky UMAJOR a UMTXIO ze systému ESTAT [8]. Všechny tyto jednotky nemusí být k dispozici ve zdrojové formě. K užití programu je nutné mít na počítači (typu PC AT nebo vyšší, matematický koprocesor doporučen) instalován překladač jazyka Turbo Pascal, postačující je jeho dávková verze (program TPC). Případní zájemci o využití programu jej mohou obdržet u autora příspěvku, pokud pošlou disketu s alespoň jedním příkladem (data v textovém souboru + stručný popis úlohy a tvar regresního modelu), na který chtějí program užít. Disketa se jim vrátí se zdrojovým textem programu MOR0, dalšími soubory potřebnými pro užití programu na jejich počítači a výsledky řešení jejich příkladu. Je možné, že program MOR bude již i v uživatelsky přítulnější verzi.

Výsledky testování programu

Program MOR0 byl testován na několika příkladech odhadů regresních parametrů nelineárních modelů (viz tab. 1). Některé z těchto příkladů jsou postrachem i pro renomované softwarové firmy, neboť mnohé metody odhadu parametrů u těchto příkladů selhávají, zvláště v situacích, kdy se „podaří" zadat nevhodné počáteční hodnoty parametrů. Vzhledem k užitému algoritmu program MOR0 počáteční hodnoty parametrů nepotřebuje. Výsledky testování jsou uvedeny v tab. 2.

model	RSS	čas	iterace	i	$bmin$	$bmax$	b
1	5.986E-03	3.8	766	1	0	1000	1.567E+01
				2	0	10	9.994E-01
				3	0	2	2.222E-02
2	1.244E+02	2.0	304	1	0	1	2.578E-01
				2	0	1	2.578E-01
3	8.795E+01	49.4	7729	1	0	10000	5.610E-03
				2	0	50000	6.181E+03
				3	0	10000	3.452E+02
4	3.179E-04	12.9	1156	1	-10000	10000	4.797E+01
				2	-10000	10000	1.020E+02
				3	-1	1	-2.466E-01
				4	-1	1	-4.965E-01
5	1.290E+02	66.2	7688	1	0	1E+08	1.655E+03
				2	0	1E+08	3.404E+07
				3	-2	0	-6.740E-01
				4	-5	0	-1.816E+00
6	2.981E-05	177.0	11878	1	0	1	4.141E-03
				2	1	8	3.802E+00
				3	1	5	2.061E+00
				4	0	1	2.229E-01
t2	8.896E-03	3.0	435	1	-50	50	2.807E-01
				2	-50	50	4.064E-01
t3	9.675E-01	64.9	4283	1	0	100	9.359E+00
				2	0	100	2.029E+00
				3	0	5	1.337E+00
				4	0	1	4.108E-01
				5	0	1	3.551E-01
t5	4.375E-03	10.1	780	1	0	5	5.589E-02
				2	0	5	3.549E+00
				3	0	5	1.482E+00
t6	1.694E-02	380.3	12022	1	-5	5	1.930E+00
				2	-5	5	2.578E+00
				3	0	10	8.017E-01
				4	-5	5	-1.299E+00
				5	0	10	8.991E-01
				6	-5	5	1.915E-02
				7	0	10	3.018E+00
t7	7.147E-05	33.8	6713	1	0	100	2.048E+00
				2	0	100	1.860E+01
				3	0	100	1.802E+00

Tabulka 2: Výsledky testování programu MOR0, počítač PC 486DX, 33 MHz. $N = 5n$, $\alpha = 5$, $\epsilon_0 = 1E - 16$

Označení modelu	Tvar regresní funkce	n_{obs}	Reference
1	$\beta_1 + \beta_2 e^{\beta_3 x}$	10	[2],[3]
2	$e^{\beta_1 x} + e^{\beta_2 x}$	10	[3]
3	$\beta_1 e^{\frac{\beta_2}{\beta_3 + x}}$	16	[2],[3]
4	$\beta_1 e^{-\beta_3 x} + \beta_2 e^{-\beta_4 x}$	10	[3]
5	$\beta_1 e^{-\beta_3 x} + \beta_2 e^{-\beta_4 x}$	16	[3]
6	$\beta_1 x^{\beta_3} + \beta_2 x^{\beta_4}$	12	[3],[1]
t2	$e^{\beta_1 x} + e^{\beta_2 x}$	10	[4]
t3	$\beta_1 + \beta_2 e^{(\beta_3 + \beta_4 x)^{\beta_5}}$	15	[4]
t5	$\beta_1 x^{\beta_2} + \beta_3^{\frac{\beta_2}{x}}$	10	[4]
t6	$\beta_1 + \beta_2 x_1^{\beta_3} + \beta_4 x_2^{\beta_5} + \beta_6 x_3^{\beta_7}$	24	[4]
t7	$\beta_1 \ln(\beta_2 + \beta_3 x)$	12	[4]

Tabulka 1: Tvar regresní funkce a reference na data

Závěr

Modifikace algoritmu CRS znáhodněním koeficientu ve výběru zkušebního bodu pro vrchol simplexu podstatně přispěla ke zrychlení konvergence vyhledávání minima, takže časy potřebné pro vyhledávání globálního minima v prostoru parametrů jsou přijatelné pro většinu aplikací. Všechny dosud řešené příklady se tímto programem podařilo úspěšně vyřešit. Program je vhodné užít zejména v situacích, kdy metody založené na výpočtu derivace minimalizované funkce selhávají. Současná verze programu je však spíše polotovar pro experimentální užití počítačově zkušeným pracovníkem. Pro obecnější užití programu zatím chybí větší uživatelský komfort, odolnost vůči chybnému zadání vstupních dat a podrobnější (i grafické) výstupy.

Literatura

[1] Křivý, I. : *Algoritmy řízeného náhodného výběru v regresní analýze.* sborník ROBUST 94, JČMF, Praha 1994.

[2] Meyer, R. R. – Roth, P. M. : *Modified damped least squares: An algorithm for non-linear estimation.* J.Inst.Math.Applics, 9, 1972, 218–233.

[3] Militký, J. – Kupka, K. : soukromá sdělení, 1994.

[4] Militký, M. – Meloun, M. : *Modus operandi of the least squares algorithm MINOPT.* Talanta, 40(2), 1994, 269–277.

[5] Nelder, J. A. – Mead, R. : *A simplex method for function minimization.* Computer J., 7(1), 1964, 308–313.

[6] Price, W. L. : *A controlled random search procedure for global optimisation.* Computer J., 20(4), 1976, 367–370.

[7] Zvára, K. : *Regresní analýza.* (p.222–224) Academia, Praha, 1989.

[8] Žváček, J. – Řezanková, H. : *ESTAT, statistické programovací prostředí v Turbo Pascalu.* str.192–195, Sborník ROBUST'90, JČMF, 1990.