

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

A. Kalenchuk-Porkhanova, L. Vakal

CONSTRUCTING THE BEST UNIFORM APPROXIMATIONS FOR MANY-VARIABLES FUNCTIONS

An algorithm for the best uniform approximation of a many-variables functions by a generalized polynomial is proposed. The algorithm is based on leading the approximation problem to the maximum problem of linear programming. Some approximation results are given.

Запропоновано алгоритм найкращого рівномірного наближення функції багатьох змінних узагальненим поліномом. В алгоритмі використано зведення задачі наближення до максимум-задачі лінійного програмування. Наведені приклади наближень.

© А.О. Каленчук-Порханова,
Л.П. Вакал, 2007

УДК 004.421.2:519.651.2

А.О. КАЛЕНЧУК-ПОРХАНОВА, Л.П. ВАКАЛ

ПОБУДОВА НАЙКРАЩИХ РІВНОМІРНИХ НАБЛИЖЕНЬ ФУНКЦІЙ БАГАТЬОХ ЗМІННИХ

Вступ. На практиці часто виникає необхідність дослідження складних процесів, що характеризуються багатопараметричними залежностями, дискретне представлення яких виражається багатовимірними масивами даних. Задача заміни таких масивів даних аналітичними виразами зводиться до задачі побудови наближень функцій багатьох змінних.

У роботі для розв'язання цієї задачі пропонується спосіб найкращого рівномірного наближення і наводиться алгоритм побудови такого наближення функції багатьох змінних узагальненим поліномом. Цей спосіб має суттєві переваги за точністю порівняно з іншими способами наближення [1].

Постановка задачі. Задача найкращого рівномірного наближення функції $f(X)$ k змінних $X = (x_1, \dots, x_k)$ узагальненим поліномом за системою n лінійно незалежних базисних функцій $\varphi_j(X)$

$$F_n(X; Z) = \sum_{j=1}^n z_j \varphi_j(X), \quad (1)$$

де $Z = (z_1, \dots, z_n)$, на множині N точок $\{X^{(1)}, \dots, X^{(N)}\}$ ($N > n$) полягає у знаходженні такого набору коефіцієнтів $Z^* = (z_1^*, \dots, z_n^*)$ апроксиманта $F_n(X; Z)$, який задовольняє умову чебишовського мінімаксу:

$$L(Z^*) \equiv \rho = \min_Z L(Z),$$

$$L(Z) = \max_{1 \leq i \leq N} \left| f(X^{(i)}) - \sum_{j=1}^n z_j \varphi_j(X^{(i)}) \right|. \quad (2)$$

Найбільш поширений спосіб знаходження узагальненого полінома найкращого рівномірного наближення полягає у зведенні задачі (1) – (2) до задачі лінійного програмування [2–7]. Якщо ввести функції

$$\Phi_i(Z) = \sum_{j=1}^n z_j \varphi_j(X^{(i)}) - f(X^{(i)}), \quad \Phi_{N+i}(Z) = -\Phi_i(Z) \quad (i = \overline{1, N}), \quad (3)$$

то задачу найкращого рівномірного наближення (1)–(2) можна представити у вигляді задачі лінійного програмування з $n + 1$ невідомими і $2N$ обмеженнями:

$$\Lambda = \min, \quad (4)$$

$$\xi_i = \Lambda - \Phi_i(Z) \geq 0 \quad (i = \overline{1, 2N}). \quad (5)$$

Нехай для базисних функцій $\varphi_j(X)$, $j = \overline{1, n}$, на множині N точок виконується мала детермінантна умова, тобто хоча б один з детермінантів n -го порядку не дорівнює нулю. Тоді в (5) серед системи $2N$ функцій ξ_i є не менше, ніж $n + 1$ лінійно незалежних. Якщо їх вибрати за нові незалежні змінні $\xi_{i_1}, \dots, \xi_{i_{n+1}}$ (як це можна зробити практично – описується далі в алгоритмі), то задача (4) – (5) запишеться у більш звичному вигляді задачі лінійного програмування з невід’ємними коефіцієнтами

$$\Lambda = \sum_{v=1}^{n+1} C_v \xi_{i_v} + \tau = \min; \quad (6)$$

$$\xi_{i_{n+s}} = \sum_{v=1}^{n+1} k_{v, n+s} \xi_{i_v} + k_{0, n+s}, \quad (s = \overline{2, 2N-n}); \quad \xi_i \geq 0 \quad (i = \overline{1, 2N}). \quad (7)$$

Для знаходження оптимального розв’язку задачі (6) – (7) можна застосувати загальні методи лінійного програмування, наприклад симплекс-метод. Але урахування специфіки задачі наближення дозволяє розробляти більш ефективні алгоритми найкращого рівномірного наближення функцій багатьох змінних.

Алгоритм. Алгоритм побудови узагальненого полінома найкращого рівномірного наближення функції багатьох змінних, який описується далі, базується на зведенні задачі наближення (1) – (2) до задачі лінійного програмування, коли головною є двоїста до (6) – (7) максимум-задача:

$$\Lambda' = - \sum_{s=2}^{2N-n} k_{0, n+s} \eta_{i_{n+s}} + \tau = \max; \quad (8)$$

$$\eta_{i_v} = - \sum_{s=2}^{2N-n} k_{v, n+s} \eta_{i_{n+s}} + C_v, \quad (v = \overline{1, n+1}); \quad \eta_i \geq 0 \quad (i = \overline{1, 2N}). \quad (9)$$

Вибір як головної двоїстої задачі (8) – (9) має ряд переваг, зокрема, число обмежень $n + 1$ двоїстої задачі значно менше числа обмежень $2N - n - 1$ прямої задачі (6) – (7), відсутнє виродження при виконанні певних умов та інші [8].

Алгоритм складається з трьох етапів: *перший* – зведення задачі наближення (1) – (2) до максимум-задачі лінійного програмування (8) – (9), *другий* – знаходження оптимального розв’язку максимум-задачі, *третій* – визначення коефіцієнтів узагальненого полінома найкращого наближення на основі отриманого оптимального розв’язку задачі (8) – (9) та оцінка точності наближення.

Для підвищення ефективності алгоритму проведено його оптимізацію за точністю та швидкістю за рахунок таких факторів.

1. В алгоритмі застосовується запропонована в [4] спеціальна процедура зведення задачі (1) – (2) до задачі лінійного програмування (8) – (9), внаслідок якої за рахунок відповідного вибору змінних $\xi_{i_1}, \dots, \xi_{i_{n+1}}$ формується симплекс-таблиця, що вже містить допустимий базисний розв’язок. Це дозволяє розпочати симплекс-процес для двоїстої максимум-задачі (8) – (9) одразу з основної фази.

2. Стандартна симплекс-таблиця замінюється значно меншою, але рівноцінною за поданою інформацією симплекс-таблицею, в якій на основі співвідношення $\xi_i + \xi_{i+N} \equiv 2\Lambda$ [2, с. 486], залишено замість $2N-n-1$ всього $N-n-1$ стовпців (кількість рядків у цих таблицях однакова).

3. В алгоритмі реалізується пряма і двоїста задачі лінійного програмування, причому головною є двоїста максимум-задача, яка розв’язується модифікованим симплекс-методом з урахуванням того, що на практиці кількість рівнянь N значно більше числа невідомих n і таблиця „розширеного базису” розмірністю $(n+2) \times (n+4)$ при модифікованому симплекс-методі суттєво менша, ніж опорна таблиця розмірністю $(n+2) \times N$ при прямому симплекс-методі.

4. Застосовується високонадійний прийом проти можливого зациклювання модифікованого симплекс-методу та проводиться контроль правильності виконаних жорданових перетворень.

Оскільки в алгоритмі при розв’язанні максимум-задачі (8) – (9) модифікованим симплекс-методом відбувається послідовне підвищення величини $L(Z)$, то цей алгоритм є аналогом поліноміального алгоритму підвищувальної дії Є.Я. Ремеза для наближення функції однієї змінної у випадку зведення до задачі лінійного програмування [2].

Вхідними даними для алгоритму побудови узагальненого полінома найкращого рівномірного наближення функції k змінних є:

n – кількість шуканих коефіцієнтів z_j узагальненого полінома вигляду (1);

N – кількість точок $X^{(j)} = (x_1^{(j)}, \dots, x_k^{(j)})$ k -вимірної сітки ($j = \overline{1, N}$);

$f(X^{(j)})$, $j = \overline{1, N}$ – значення функції f у точках сітки або процедура для їх обчислення;

$\varphi_i(X^{(j)})$, $i = \overline{1, n}$; $j = \overline{1, N}$ – значення базисних функцій у точках сітки або процедури для їх обчислення.

Перший етап алгоритму.

1. Виконується початкове заповнення матриці T розмірністю $(n+2) \times N$ за

формулами:

$$t_{ij} = \varphi_i(X^{(j)}), t_{n+1,j} = -f(X^{(j)}), t_{n+2,j} = 0 \quad (i = \overline{1, n}; j = \overline{1, N}).$$

2. Покладаються $p_j = t_{n+1,j}$, $d_j = j$ ($j = \overline{1, N}$) та $it = 0$ (it – лічильник числа кроків алгоритму).

3. Для кожного i -го ($i = \overline{1, n}$) рядка матриці T визначається номер v стовпця, в якому знаходиться найбільший за модулем елемент цього рядка $|t_{n+2,v}| = \max_{j=1, N-n} |t_{n+2,j}|$, покладається $d_i = v$ і виконується процедура *Jordan* жорданового перетворення матриці T з елементом t_{iv} як ключовим.

4. Стовпці матриці T переставляються в порядку розташування індексів $d_{n+1}, \dots, d_N, d_1, \dots, d_n$.

5. Обчислюються елементи $(n+2)$ -го рядка матриці T за формулою

$$t_{n+2,j} = p_{d_{n+j}} - \sum_{i=1}^n t_{ij} p_{d_i} \quad (j = \overline{1, N-n}),$$

після чого визначається номер стовпця μ , в якому розташований максимальний за модулем елемент цього рядка $|t_{n+2,\mu}| = \max_{j=1, N-n} |t_{n+2,j}|$. Якщо таких елементів декілька, то вибирається номер μ , для якого додатково $t_{n+2,\mu} > 0$.

6. Якщо $t_{n+2,\mu} > 0$, то перехід до п. 7, інакше – до п. 8.

7. У кожному i -му рядку ($i = \overline{1, n}$) матриці T , в якому елемент $t_{i\mu} > 0$, знаки усіх елементів цього рядка змінюються на протилежні, покладається $d_i = d_i + N$ і здійснюється перехід на п. 10.

8. Змінюється на протилежний знак елемента $t_{n+2,\mu}$ і покладається $d_{n+\mu} = d_{n+\mu} + N$.

9. Якщо в деякому i -му рядку ($i = \overline{1, n}$) матриці T елемент $t_{i\mu} < 0$, то знаки всіх інших елементів цього рядка змінюються на протилежні та покладається $d_i = d_i + N$; якщо ж $t_{i\mu} > 0$, то змінюється на протилежний тільки знак $t_{i\mu}$.

10. Виконуються перетворення елементів матриці T , а саме:

– обчислюються перші $N-n$ елементів $(n+1)$ -го рядка

$$t_{n+1,j} = 1 - \sum_{i=1}^n t_{ij} \quad (j = \overline{1, N-n});$$

– змінюються на протилежні знаки елементів $t_{n+2,j}$ ($j = \overline{1, N-n}$);

– визначаються інші елементи $(n+1)$ -го рядка $t_{n+1,j} = \sum_{i=1}^n t_{ij}$

$(j = \overline{N-n+1, N})$;

– обчислюються останні n елементів $(n+2)$ -го рядка $t_{n+2,j} = -\sum_{i=1}^n t_{ij}q_i$,

$j = \overline{N-n+1, N}$, де $q_i = p_{d_i}$ для $d_i \leq N$ і $q_i = -p_{d_i-N}$ для $d_i > N$;

– змінюються на протилежні знаки елементів на перетині перших n рядків і останніх n стовпців $t_{ij} = -t_{ij}$ ($i = \overline{1, n}$, $j = \overline{N-n+1, N}$).

11. До матриці T застосовується одна операція перетворення Жордана з ключовим елементом $t_{n+1,\mu}$ (з використанням процедури *Jordan*).

12. Обчислюються елементи допоміжних векторів B та R та елементи h_{ij} матриці H розмірністю $(n+2) \times n$, яка буде використовуватися на третьому етапі алгоритму для визначення коефіцієнтів узагальненого полінома:

$$b_i = 2 t_{i\mu}, r_k = k, h_{ij} = t_{i, N-n+j} \quad (i = \overline{1, n+2}, j = \overline{1, n}, k = \overline{1, N}).$$

13. Визначаються елементи матриці C розмірністю $(n+2) \times (n+4)$, в якій формується початкова стиснена таблиця модифікованого симплекс-методу

$$c_{i1} = t_{i\mu}, c_{i, n+4} = 0, c_{ij} = \begin{cases} 1, & j = i+1 \\ 0, & j \neq i+1 \end{cases}, \quad (i = \overline{1, n+2}, j = \overline{2, n+3}).$$

14. Обчислюються елементи матриці T , в якій формується опорна таблиця модифікованого симплекс-методу

$$t_{ij} = t_{i, j-n-1} \quad (i = \overline{1, n+2}, j = \overline{n+2, N}), t_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (i = \overline{1, n+2}, j = \overline{1, n+1}).$$

Внаслідок виконаних дій елементи c_{i1} ($i = \overline{1, n+1}$) у стисненій симплекс-таблиці будуть невід'ємними. Отже, наявний у цій таблиці базисний розв'язок $\eta_j = c_{1j} \geq 0$ ($j = \overline{1, n+1}$), $\eta_{n+k} = 0$ ($k = \overline{2, 2N-n}$) буде і допустимим.

Другий етап алгоритму. У зв'язку зі специфікою опорної таблиці – відсутність в явному вигляді майже половини стовпців – модифікований симплекс-метод має певні особливості реалізації порівняно зі стандартною схемою.

15. Покладається $it = it + 1$ та $p_{r_k} = \sum_{l=1}^{n+2} t_{l, r_k} \cdot c_{n+2, l+1}$ ($k = \overline{n+2, N}$).

16. Якщо для всіх $k = \overline{n+2, N}$ виконується умова $0 \leq p_{r_k} \leq b_{n+2}$, то здійснюється перехід до п. 23, інакше – визначається номер v ключового стовпця матриці T з умови

$$p_v = \max \left\{ \max_{r_k: p_{r_k} < 0} |p_{r_k}|, \max_{r_k: p_{r_k} > b_{n+2}} |b_{n+2} - p_{r_k}| \right\}, \quad (k = \overline{n+2, N}).$$

17. Якщо $1 \leq v \leq N$, то $f_i = t_{i,v}$, інакше – $f_i = b_i - t_{i,v}$, $i = \overline{1, n+2}$.

18. Обчислюються елементи ключового стовпця матриці C за формулою

$$c_{i,n+4} = \sum_{k=1}^{n+2} f_k \cdot c_{i,k+1}.$$

19. Визначається номер μ ключового рядка матриці C , для якого $\frac{c_{\mu,1}}{c_{\mu,n+4}} = \min_{i: c_{i,n+4} > 0} \frac{c_{i,1}}{c_{i,n+4}}$. Якщо таких номерів декілька, то для усунення невідповідності й запобігання зациклювання вибирається найменший [2, с. 295].

20. Виконується жорданове перетворення матриці C з ключовим елементом $c_{\mu,n+4}$ за допомогою процедури *Jordan*.

21. Якщо для елементів матриці C виконуються рівності $\sum_{i=1}^{n+1} c_{ij} = 1$,

$\sum_{i=1}^{n+1} c_{i,n+4} = 1, (j = \overline{1, n+2})$, то здійснюється перехід на п. 22, інакше – видається

повідомлення про помилку в обчисленнях і алгоритм закінчує роботу.

22. Здійснюється підготовка до наступного кроку модифікованого симплекс-методу. Для цього в масивах індексів міняються місцями індекс змінної, яка виводиться з базису, та індекс змінної, яка вводиться з нього. Спочатку покладається $k = d_{\mu}$. Якщо $p_v < 0$, то $d_{\mu} = d_v$, інакше – $d_{\mu} = d_v + N$ для $d_v \leq N$ і $d_{\mu} = d_v - N$ для $d_v > N$. Далі послідовно виконуються присвоєння: $d_v = k$, $k = r_{\mu}$, $r_{\mu} = r_v$, $r_v = k$ і здійснюється перехід на п. 15.

23. Запам'ятовуються номери змінних останнього базису. Якщо $1 \leq d_v \leq N$, то покладається $i_v = d_v$, інакше – $i_v = d_v - N$ ($v = \overline{1, n}$).

На цьому другий етап алгоритму закінчено. Оптимальний розв'язок задачі (8)–(9) знаходиться у масиві P , а значення цільової функції дорівнює значенню елемента $c_{n+2,1}$.

Третій етап алгоритму.

24. Обчислюються коефіцієнти узагальненого полінома і величина $\rho (= z_{n+1})$ за формулами:

$$z_j = h_{n+2,j} + \sum_{k=1}^{n+1} p_k \cdot h_{kj} \quad (j = \overline{1, n}), \quad z_{n+1} = c_{n+2,1}.$$

25. Визначаються величини A і L

$$A = \min_{1 \leq v \leq n+1} \left| \sum_{j=1}^n z_j \varphi_j \left(X^{(i_v)} \right) - f \left(X^{(i_v)} \right) \right|; \quad L = \max_{1 \leq v \leq n+1} \left| \sum_{j=1}^n z_j \varphi_j \left(X^{(i_v)} \right) - f \left(X^{(i_v)} \right) \right|.$$

26. Обчислюється похибка δ у визначенні величини наближення ρ за формулою $\delta = L - A$.

27. Виводяться на друк такі результати:

z_1, \dots, z_n – коефіцієнти узагальненого полінома;

z_{n+1} – величина наближення ρ ;

δ – похибка у визначенні величини найкращого рівномірного наближення ρ ;

i_1, \dots, i_n – номери змінних останньої базисної системи.

В описаному алгоритмі для виконання перетворення Жордана довільної матриці A розмірністю $m \times n$ з відомим ключовим елементом a_{lk} служить процедура *Jordan*. У цій процедурі послідовно виконуються перетворення ключового елемента $a_{lk} = \frac{1}{a_{lk}}$, елементів ключового рядка $a_{lj} = a_{lj} \cdot a_{lk}$ ($j = \overline{1, n}, j \neq k$), ключового стовпця $a_{ik} = -a_{ik} \cdot a_{lk}$ ($i = \overline{1, m}, i \neq l$) та елементів поза ключовим хрестом $a_{ij} = a_{ij} - a_{ik} \cdot a_{lj}$ ($i = \overline{1, m}, j = \overline{1, n}, i \neq l, j \neq k$).

Приклади обчислень. Для оцінки ефективності розробленого алгоритму наближення функцій багатьох змінних узагальненими поліномами було проведено порівняння з іншими аналогічними алгоритмами на тестових прикладах.

Приклад 1. За розробленим алгоритмом знайдено поліном найкращого рівномірного наближення $P_4(x, y)$ вигляду:

$$P_4(x, y) = z_1 + z_2x + z_3y + z_4x^2 + z_5xy + z_6y^2 + z_7x^3 + z_8x^2y + z_9xy^2 + z_{10}y^3 + z_{11}x^4 + z_{12}x^3y + z_{13}x^2y^2 + z_{14}xy^3 + z_{15}y^4, \quad (10)$$

який апроксимує функцію $\cos x \cdot \sin y$ у на сітці $S = \{x_i, y_j\}$ ($x_i = 0.1i; y_j = 0.1j; i = \overline{0, 10}; j = \overline{0, 10}$) з абсолютною похибкою $\rho = 0.0002732$. Для порівняння, похибка наближення цієї функції поліномом вигляду (10), побудованим за алгоритмом [9], дорівнює $\rho = 0.0002735$. Гірше наближення дає також поліном $P_4(x, y)$, знайдений за алгоритмом [7], причому для цього виявилось необхідним 43 кроки алгоритму, водночас як у нашому алгоритмі – 25 кроків.

Приклад 2. Алгоритм застосовувався для наближення залежності квадрату щільності B (в умовних одиницях) від температури t (в $^{\circ}\text{C}$) і концентрації C (в г/л) солі в розчині за допомогою узагальненого полінома за системою трьох базисних функцій $\varphi_1 = C^{1.736}$, $\varphi_2 = t$, $\varphi_3 = C \cdot t$ [10]. Отримана емпірична формула $B^2 = 0.02618C^{1.736} + 0.03547t - 0.006912 \cdot C \cdot t$ дозволила наблизити дані експерименту з абсолютною похибкою 0.073 (відносна похибка – 5%). Зазначимо, що знайдена за методом середніх емпірична формула $B^2 = 0.02644 C^{1.736} + 0.0352 \cdot t - 0.006999 C \cdot t$ дає гірші результати, а саме, абсолютна похибка наближення становить 0.0924 (відносна – 14%) [10].

Висновки. У порівнянні з іншими алгоритмами наближення функцій багатьох змінних запропонований алгоритм дозволяє отримати найкращі рівномірні наближення таких функцій узагальненими поліномами з меншою похибкою та,

як правило, за значно меншу кількість ітерацій. Це вдалося досягти за рахунок удосконалення обчислювальної схеми алгоритму за такими основними напрямками:

- процес зведення задачі наближення до максимум-задачі лінійного програмування проводиться з орієнтацією на отримання початкової симплекс-таблиці з якомога більшим значенням цільової функції, завдяки чому оптимальний розв'язок максимум-задачі знаходиться за відносно малу кількість кроків;

- для розв'язання задачі лінійного програмування застосовується модифікований симплекс-метод, в ході якого жорданові перетворення застосовуються тільки до стисненої таблиці, при цьому опорна таблиця залишається незмінною. У порівнянні зі стандартною симплекс-таблицею стиснена таблиця є майже удвічі меншою, але рівноцінною за представленою інформацією, і вже містить допустимий базисний розв'язок;

- у процесі проведення розрахунків додатково здійснюється контроль правильності виконання обчислень.

Запропонований алгоритм також може ефективно використовуватись для визначення параметрів емпіричних формул, які є представленнями відповідних залежностей між експериментальними даними.

1. *Вакал Л.П., Каленчук-Порханова А.О.* Аналітична обробка даних на основі чебишовської апроксимації // Математичні машини і системи. – 2006. – № 2. – С. 15–24.
2. *Ремез Е.Я.* Основи численних методів чебышевского приближения. – К.: Наук. думка, 1969. – 623 с.
3. *Stiefel E.* Note on Jordan elimination, linear programming and Tschebyscheff approximation // Numerische Math. – 1960. – 2. – N 1. – P. 1–17.
4. *Александренко В.Л.* Алгоритм построения приближённого равномерно-наилучшего решения системы несовместных линейных уравнений // Алгоритмы и алгоритмические языки. – М.: ВЦ АН СССР, 1968. – Вып. 3. – С. 57–74.
5. *Каленчук-Порханова А.А.* Аппроксимация функций одной и многих переменных // Численные методы для многопроцессорного вычислительного комплекса ЕС. – М.: Изд-во ВВИА им. Н.Е. Жуковского, 1987. – С. 366–395.
6. *Демьянов В.Р., Малозёмов В.Н.* Введение в минимакс. – М.: Наука, 1972. – 368 с.
7. *Кондратьев В.П.* Алгоритм наилучшего приближения функций многих переменных // Программы оптимизации (приближение функций). – Свердловск: ИММ УНЦ АН СССР. – 1972. – Вып. 3. – С. 20–48.
8. *Каленчук-Порханова Ж., Вакал Л.* Найкраще рівномірне наближення функцій багатьох змінних // Тези доповідей Міжнарод. математ. конф. ім. В.Я. Скоробогатка.– Дрогобич-Львів: ІППММ НАН України. – 2004. – С. 90.
9. *Петрак Л.В.* Программа построения приближающего полинома для функций многих переменных // Программы оптимизации (приближение функций).– Свердловск: ИММ УНЦ АН СССР. – 1975. – Вып. 6. – С. 145–157.
10. *Батунер Л.М., Позин М.Е.* Математические методы в химической технике. – Л.: Химия, 1968. – 823 с.

Отримано 10.04.2007