Організація рекурентно-паралельних обчислень в комбінаторному алгоритмі МГУА для задач індуктивної побудови моделей

Єфіменко С.М., Степашко В.С.

Відділ інформаційних технологій індуктивного моделювання
Міжнародний науково - навчальний центр інформаційних технологій та систем
syefim@ukr.net, stepashko@irtc.org.ua

Анотація

Розроблено теоретичні основи розпаралелювання операцій в комбінаторному алгоритмі COMBI МГУА з рекурентним оцінюванням параметрів моделей. Це дозволяє радикально підвищити ефективність структурно-параметричної ідентифікації при забезпеченні рівномірного навантаження будь-якої заданої кількості наявних процесорів кластерної системи.

1. Вступ

МГУА [1] як основний інструмент теорії індуктивного моделювання належить до найсучасніших методів обчислювального інтелекту і м'яких обчислень. Цей метод, розроблений академіком О.Г.Івахненком, є оригінальним і ефективним засобом розв'язання широкого спектру задач штучного інтелекту, в тому числі ідентифікації та прогнозування, розпізнавання образів і кластеризації, інтелектуального аналізу даних і пошуку закономірностей.

Важливим критерієм ефективності програмних засобів, що ґрунтуються на методах індуктивного моделювання, є час отримання результату моделювання. Рекурентні обчислення та розпаралелювання операцій при цьому є одними з найбільш ефективних засобів досягнення високої продуктивності таких програмних продуктів.

Оскільки рекурентні обчислення забезпечують істотне зменшення кількості операцій, а паралельні обчислення – в залежності від числа процесорів, то поєднання цих двох потужних апаратів дозволяє отримати ефект у вигляді недосяжного раніше підвищення продуктивності алгоритмів МГУА.

2. Задача структурно-параметричної ідентифікації

У загальному випадку задача структурно-параметричної ідентифікації, полягає у формуванні за даними вибірки деякої множини моделей різної структури виду

$$\hat{y}_f = f(X, \hat{\theta}_f) \tag{1}$$

і пошукові оптимальної моделі за умовою

$$f^* = \underset{f \in \Im}{\operatorname{argmin}} CR(y, f(X, \hat{\theta}_f)), \tag{2}$$

причому оцінки параметрів в (2) для кожної моделі $f \in \Im$ є розв'язком ще однієї екстремальної задачі виду

$$\hat{\theta}_f = \underset{f \in \Im}{\operatorname{argmin}} QR(y, X, s_f), \tag{3}$$

де s_f називається складністю моделі f і дорівнює кількості ненульових компонент у моделі (2); QR – критерій якості розв'язання задачі параметричної ідентифікації кожної окремої моделі, що генерується в задачі структурної ідентифікації, де моделі порівнюються за критерієм CR.

3. Комбінаторний алгоритм COMBI МГУА на основі рекурентно-паралельних обчислень

3.1. Комбінаторний алгоритм МГУА

У комбінаторному алгоритмі, призначеному для пошуку (шляхом повного перебору всіх можливих варіантів) кращої регресії, що містить найбільш інформативну підмножину вхідних змінних (регресорів), є такі основні блоки: перетворення даних згідно з вибраним класом структур моделей, лінійних за параметрами; формування моделей різної складності; обчислення значень зовнішніх критеріїв якості і відбір кращих моделей; оцінка якості отриманих моделей.

У випадку лінійного об'єкта з m входами в процесі повного перебору порівнюються моделі виду

$$\hat{y}_\nu = X_\nu \hat{\theta}_\nu, \; \nu = \overline{1, \; ..., \; 2^m - 1}, \tag{4}$$

де десятковому числу ν ставиться у відповідність двійкове число d_ν, одиничні елементи якого вказують на включення в модель регресорів з відповідними номерами, а нульові – на виключення. Тоді формування структур частинних моделей формалізується за допомогою двійкового структурного вектора $d_\nu = \{d_1, d_2, ..., d_m\}$, $d_k = \{0;1\}$, $k = \overline{1, m}$.

Зміну станів вектора d можна організувати багатьма способами. Досить зручною для повного перебору є схема зміни вектора d за принципом двійкового лічильника, в останній розряд якого додається одиниця. При цьому є однозначна відповідність між порядковим номером чергової моделі і станом структурного вектора.

При переборі складність частинних моделей змінюється від 1 до m. Загальне число варіантів при повному переборі складе 2^m–1 різних структур. Зокрема, для випадку трьох аргументів послідовність усіх комбінацій виглядає так:

$$y_1 = a_1x_1$$
$$y_2 = a_2x_2$$
$$y_3 = a_1x_1 + a_2x_2$$
$$y_4 = a_3x_3$$
$$y_5 = a_1x_1 + a_3x_3$$
$$y_6 = a_2x_2 + a_3x_3$$
$$y_7 = a_1x_1 + a_2x_2 + a_3x_3$$

У таблиці 1 подано залежність часу моделювання за допомогою найшвидшого варіанту комбінаторного алгоритму (без розпаралелювання обчислень) від кількості аргументів при повному переборі.

Таблиця 1: Час повного перебору

Кількість аргументів	Кількість моделей	Час моделювання
20	1 048 576	1 сек
21	2 097 152	2 сек
22	4 194 304	4 сек
...	...	...
36	7E+10	~ 1 день
...	...	...
45	4E+13	~ 1 рік

При кількості аргументів, більшій за 30, перебір усіх варіантів на персональному комп'ютері за прийнятний для моделювання час стає недоцільним. Одним з найефективніших способів прискорення комбінаторного алгоритму є оптимальне поєднання рекурентних обчислень [2, 3] з їх розпаралелюванням на кластерних комплексах [4, 5].

3.2. Основи рекурентно-паралельних обчислень в алгоритмі COMBI

Комбінаторний алгоритм має особливість, яка дозволяє зробити висновок про високу ефективність його розпаралелювання. Так, всі блоки алгоритму, представлені на рисунку 1, можуть виконуватися кожним процесором автономно, без необхідності міжпроцесорної взаємодії. Головний процесор лише збирає результати окремих процесорів і відбирає кращу модель.

Далі описується розроблена схема розпаралелювання комбінаторного алгоритму з рекурентним обчисленням параметрів моделей за допомогою модифікованого алгоритму Гаусса [6] розв'язування систем лінійних рівнянь.

Схема використовує генерацію двійкових чисел, які відповідають послідовним десятковим числам. Всі можливі комбінації та послідовність двійкових структурних векторів, для випадку трьох аргументів, можна записати таким чином:

{0, 0, 0}
{0, 0, 1}
{0, 1, 0}
{0, 1, 1}
{1, 0, 0}
{1, 0, 1}
{1, 1, 0}
{1, 1, 1}.

Цю схему пропонується використати для розпаралелювання комбінаторного алгоритму на багатокластерні системи таким чином.

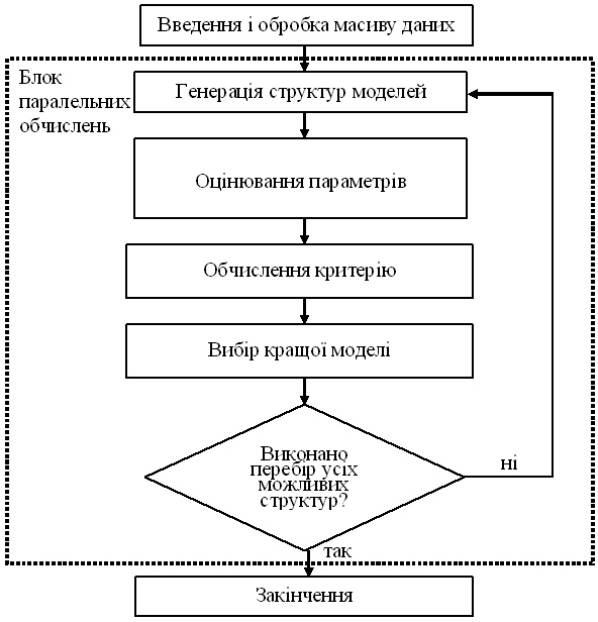

Рисунок 1: Блок-схема розпаралелювання алгоритму COMBI

Очевидно, що перша половина повної послідовності двійкових структурних векторів є дзеркальним відображенням другої половини, у якій виконується інвертування елементів (нульові елементи замінюються на одиничні і навпаки). Так, для згаданого раніше випадку трьох аргументів першому вектору {0, 0, 0} відповідає останній {1, 1, 1}, другому {0, 0, 1} – передостанній {1, 1, 0} і т.д. Така закономірність справджується для випадку будь-якої складності моделей (загальної кількості аргументів) і дозволяє розв'язати задачу рівномірного розподілу кількості моделей (а також забезпечити однакову сумарну кількість аргументів) на задану кількість процесорів кластерної системи. Покажемо, як можна розпаралелити задачу для випадку повного перебору серед трьох аргументів на кластерній системі, що складається з двох процесорів. Отримаємо моделі з двійковими структурними векторами:

{0, 0, 0}
{0, 0, 1}
{1, 1, 0}
{1, 1, 1},

які будує перший процесор (4 моделі із загальною кількістю аргументів, рівною 6) та моделі з двійковими структурними векторами:

{0, 1, 0}
{0, 1, 1}
{1, 0, 0}
{1, 0, 1},

які будує другий процесор (також маємо 4 моделі із загальною кількістю аргументів, рівною 6).

Запишемо алгоритм знаходження послідовного набору двійкових структурних векторів при кількості

аргументів, рівній m, для k-го процесора кластерної системи ($k = \overline{1, K}$) у вигляді послідовності кроків:

крок 1: за формулою $P(m) = \sum_{s=1}^{m} C_m^s = 2^m$ обчислюємо загальну кількість моделей, що будуються комбінаторним алгоритмом;

крок 2: знаходимо кількість моделей, що їх будуватиме кожен процесор: $P_k(m) = P(m)/K$. У випадку, якщо $P(m)$ не ділиться націло на K, на перший процесор припадатиме вище обчислювальне навантаження, яке відрізнятиметься від решти процесорів не більше, ніж на K додаткових моделей;

крок 3: генеруємо $P_\kappa(m)/2$ послідовних структур (починаючи з тієї, яка відповідає десятковому числу $(\kappa-1)P_\kappa(m)/2$ та закінчуючи тією, яка відповідає десятковому числу $(\kappa-1)P_\kappa(m)/2 + P_\kappa(m)/2 = \kappa P_\kappa(m)/2$);

крок 4: знаходимо структуру, інвертовану до тієї, яка відповідає десятковому числу $\kappa P_\kappa(m)/2$ (всі одиниці у двійковому представленні числа $\kappa P_\kappa(m)/2$ замінюємо на нулі, а всі нулі – на одиниці);

крок 5: генеруємо $P_\kappa(m)/2$ послідовних структур, починаючи з тієї, яка отримана на попередньому кроці.

3.3. Оцінка ефективності розпаралелювання рекурентних обчислень

Знайдемо теоретичну оцінку ефективності розробленої схеми розпаралелювання комбінаторного алгоритму з рекурентним оцінюванням параметрів.

Нехай, маємо m аргументів для повного перебору на 2^k процесорах кластерної системи, що передбачає побудову 2^m моделей. Тоді на кожен процесор припадає додаткове навантаження (обумовлене необхідністю оцінювати параметри першої моделі нерекурентно) у вигляді не більше ніж m моделей для першої половини послідовності двійкових структурних векторів та не більше ніж m моделей для другої половини послідовності, що отримуються інвертуванням попередніх (див. кроки 4 та 5 наведеного вище алгоритму).

Кожен процесор побудує 2^{m-k} моделей рекурентно та додатково не більше ніж $2m$ моделей, що становить $2m/2^{m-k}=m/2^{m-k-1}$ частину навантаження.

При заданій кількості аргументів та зростаючій кількості процесорів теоретична ефективність розпаралелювання буде зменшуватися. Так, для m=25 аргументів (для меншої кількості аргументів немає сенсу розпаралелювати алгоритм) та 2^7=128 процесорів матимемо 0.0002 (або 0.02%) втрат, тобто теоретична ефективність становитиме 99.98 відсотків. При більшій кількості аргументів вона буде ще вищою.

Крім того, додаткові втрати матиме перший процесор у вигляді не більше, ніж 2^{k+1} додаткових моделей (див. крок 2 алгоритму). Це означає додаткове навантаження на нього, що становить $2^{k+1}/2^{m-k}=2^{2k-m+1}$ частину. Для випадку m=25 аргументів та 2^7=128 процесорів матимемо 0.001 (або 0.1%) втрат, тобто загальна теоретична ефективність становитиме 99.88 відсотків.

4. Результати експериментів з рекурентно-паралельних обчислень

4.1. Експеримент 1

З метою експериментального визначення ефективності розробленої схеми комбінаторного алгоритму на основі рекурентно-паралельних обчислень було проведено тестовий експеримент по розв'язанню задачі структурно-параметричної ідентифікації за допомогою комбінаторного алгоритму з рекурентно-паралельними обчисленнями (для кількості аргументів, рівної 25) з розділення всіх кроків на 5 потоків і послідовним їх виконанням на одному процесорі.

Результат цього експерименту є наближеним до теоретичного, оскільки у ньому виключено міжпроцесорну взаємодію.

Експеримент виконувався таким чином: генерувалася матриця плану X розміром 45×25 (45 точок для 25 аргументів) для системи умовних рівнянь $X\theta = y$. Вектор y формувався як лінійна комбінація п'яти аргументів $y = x_{11} + x_{12} + x_{13} + x_{14} + x_{15}$. Виконувалася структурно-параметрична ідентифікація з вибором кращої моделі за критерієм регулярності [7] та вимірювався час моделювання при розпаралелюванні на різну кількість потоків.

Результат, представлений на рисунку 2 у вигляді діаграми часу виконання, демонструє ефективність використання розробленої схеми.

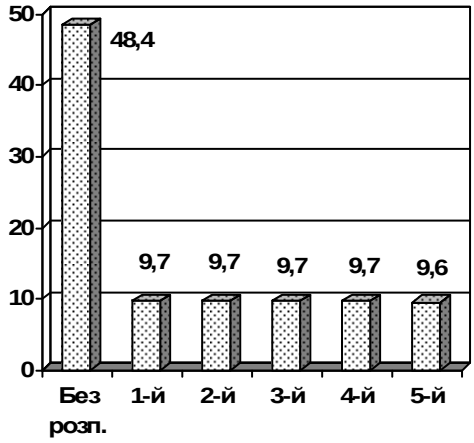

Рисунок 2: Час (в секундах) виконання алгоритму COMBI на основі рекурентно-паралельних обчислень

Ефективність розпаралелювання E досягла 99,7%, а рівномірність навантаження P дорівнювала 99,2%, які визначалися за формулами:

$$E = \frac{T_1}{5 \times T_{5\max}} \times 100\%, \qquad (5)$$

$$P = (1 - \frac{T_{5\max} - T_{5\min}}{T_{5\max}}) \times 100\%, \qquad (6)$$

де T_1 – час виконання алгоритму з одним потоком (тобто, без розпаралелювання), T_{5max} – час виконання алгоритму з розпаралелюванням на 5 потоків (визначається як максимальний серед п'яти потоків час виконання програми), T_{5min} – мінімальний серед п'яти потоків час виконання програми.

4.2. Експеримент 2

Метою тестового експерименту було визначити ефективність схеми розпаралелювання комбінаторного алгоритму з рекурентними обчисленнями по відношенню до паралельного алгоритму з нерекурентним оцінюванням параметрів. Ефективність визначалася як відношення часу виконання відповідних програм (для кількості аргументів від 21 до 25):

$$E_{rec} = \frac{T_{nonrec}}{T_{rec}} \quad (1)$$

Результат експерименту, представлений на рисунку 3, показує зростання значення E_{rec} зі збільшенням числа аргументів.

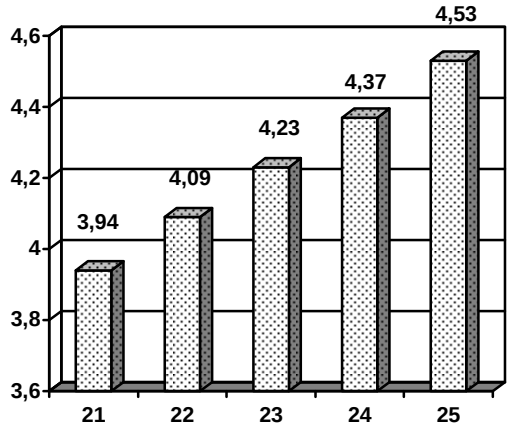

Рисунок 3: Ефективність алгоритму з рекурентно-паралельними обчисленнями.

5. Висновки

Розроблено теоретичні основи розпаралелювання операцій у комбінаторному алгоритмі COMBI GMDH з рекурентним обчисленням параметрів моделей. Зокрема, запропоновано схему розпаралелювання, що ґрунтується на основі алгоритму генерації стандартного двійкового лічильника (що відповідає послідовним десятковим числам).

Особливість схеми полягає у тому, що перед початком моделювання кожен процесор обчислювального кластера самостійно розраховує початковий та кінцевий структурний вектор для кожної складності моделей, чим забезпечується практично рівномірне навантаження на всі елементи кластерної системи загалом та відпадає необхідність у міжпроцесорній взаємодії.

Для задачі оцінювання параметрів за МНК використано рекурентну модифікацію алгоритму Гауса розв'язування систем лінійних рівнянь.

За допомогою тестового експерименту показано, що використання запропонованої схеми забезпечує однакову сумарну кількість моделей та оцінюваних параметрів, що приходяться на кожен процес. Ефективність застосування схеми при кількості процесів, рівній 5, та кількості аргументів рівній 20, становить 95%, а рівномірність навантаження – 98%.

Зі збільшенням кількості аргументів зростає ефективність комбінаторного алгоритму з рекурентно-паралельними обчисленнями по відношенню до паралельного алгоритму з нерекурентним оцінюванням параметрів.

6. Література

[1] Ивахненко А.Г., Степашко В.С. Помехоустойчивость моделирования. – Киев: «Наук. думка», 1985. – 216 с.

[2] Степашко В.С., Єфіменко С.М. Аналіз особливостей рекурентного методу послідовного оцінювання параметрів моделей // Відбір і обробка інформації. – 2004. – № 21. – С. 91–96.

[3] Stepashko V. S. , and Efimenko S. N. . Sequential Estimation of the Parameters of Regression Model // Cybernetics and Systems Analysis, Springer New York, July, 2005, Vol. 41, Num. 4, pp.631-634.

[4] Степашко В.С., Єфіменко С.М., Розенблат О.П. Про застосування паралельних обчислень в задачах моделювання на основі індуктивного підходу // Праці П'ятої міжнародної науково-практичної конференції з програмування „УкрПрог'2006", Київ, 23-25 травня 2006 р. // Проблеми програмування. – 2006. – № 2–3. – С. 170–177.

[5] Stepashko V.S., Yefimenko S.M. Optimal paralleling for solving combinatorial modelling problems // Proceedings of the 2nd International Conference on Inductive Modelling ICIM 2008. – Kyiv, 2008. – P. 172-175.

[6] Єфіменко С.М., Степашко В.С. Рекурентний алгоритм методу Гаусса для розв'язання систем лінійних рівнянь у задачі оцінювання параметрів регресійних моделей // Відбір і обробка інформації. Міжвідомчий збірник наукових праць. – №36 (112). – 2012. – С. 48-55.

[7] Івахненко О.Г., Коппа Ю.В. Алгоритми методу групового врахування аргументів (МГВА) з лінійними операторами // Автоматика. – 1969.– № 4.– С.69–78.