Kronecker Product Operations to Find Functions of Higher Dimensions Using Least Squares Method

Dr. Sudad K. Ibraheem  Dr. Amaal A. Mohammed
Al-Mustansiriya University Al-Mustansiriya University
College of Science College of Engineering
Dept. of Mathematics Dept. of Ways & Transportation

Abstract: In this work, we have describe the application of kronecker product operation to interpolation the functions of higher dimensions (3 and 4-variables function) by using least square method.
Several examples are given to illustrate which is used to define and specialize the method. A comparison between the exact function and interpolation function depending on least square errors.

Keywords: Kronecker product, interpolation, least square method.
1. Introduction

Kronecker product, named after German mathematician Leopold Kronecker, is a special operator used in matrix algebra for multiplication of two matrices. This product, written as $\otimes$, gives the possibility to obtain a composite matrix of elements of any pair of matrices. "any" stresses here that Kronecker product works without assumptions on the size of composing matrices, as it is the case with ordinary matrix multiplication [1].

Kronecker product has been successfully used as a framework for understanding different variants of the fast Fourier transform [2]. Van Loan [3,4] has described various interesting properties of Kronecker product and their applications. We shall only briefly review some of the properties of Kronecker product of matrices [5].

Let matrices $A$ be $(m \times n)$ and $B$ be $(m' \times n')$. Let $C = A \otimes B$ (kron(A,B) in matlab notation), then matrix $C$ is size $(m \times n' \times n)$. If matrix $A$ is $3 \times 3$, then

\[
C = \begin{bmatrix}
    a_{11}B & a_{12}B & a_{13}B \\
    a_{21}B & a_{22}B & a_{23}B \\
    a_{31}B & a_{32}B & a_{33}B
\end{bmatrix}
\]

Some of interesting properties of Kronecker products are summarized below,

1. $\left( A \otimes B \right) \left( C \otimes D \right) = AC \otimes BD \quad \ldots(1)$
2. $\left( A + B \right) \otimes C = A \otimes C + B \otimes C \quad \ldots(2)$
3. $\left( A \otimes B \right) \otimes C = A \otimes \left( B \otimes C \right) \quad \ldots(3)$
4. $\left( A \otimes B \right)^{-1} = A^{-1} \otimes B^{-1} \quad \ldots(4)$
5. $\left( A \otimes B \right)^t = A^t \otimes B^t \quad \ldots(5)$

A literature survey shows that most of successful least squares method are for the finite impulse response filters.

The least square method – a very popular technique is used to compute estimation of parameters and to fit data. It is one of the oldest techniques of modern statistics.
Nowadays, the least square is widely used to find or estimate the numerical values of the parameter to fit a function to a set of data and to characterize the statistical properties to estimates.

We give a quick introduction to the basic elements of probability and statistics which we need for the method of least squares, for more details see [6,7,8,9].

In the standard linear model \[ (10) \]

\[
y = X B + \varepsilon
\]

where \( y \) is the \((n \times 1)\) response vector, \( X \) is an \((n \times p)\) model matrix, \( B \) is a \((p \times 1)\) vector of parameters to estimate, and \( \varepsilon \) is an \((n \times 1)\) vector of errors.

Assuming that \( \varepsilon \sim N(0, \delta^2 I_n) \) leads to the familiar ordinary – least squares (OLS) estimator of \( B \)

\[
b_{OLS} = (X^t X)^{-1} X^t y
\]

\[
V_{OLS} = \delta^2 (X^t X)^{-1}
\]

2. Interpolation of the Functions of 3 and 4 – variables:

Kronecker products arise from interpolation of tabulated function values of 2–variables [5,11] and of higher dimensions. For the 3 and 4-variables function indexed as \( F(x, y, z) \) and \( F(w, x, y, z) \) respectively, the corresponding interpolation scheme would be:

\[
F(x, y, z) = \sum_{lpq} C_{lpq} \phi_l(x) \phi_p(y) \phi_q(z)
\]

\[
and
\]

\[
F(w, x, y, z) = \sum_{pqrs} C_{pqrs} \phi_p(w) \phi_q(y) \phi_r(x) \phi_s(z)
\]
where the coefficients $C_{lpq}$ and $C_{pqrs}$ can be computed to satisfy the interpolation conditions:

$$F_{ijk} = \sum_{lpq} C_{lpq} \phi_i(x_i) \phi_p(y_j) \phi_q(z_k) \quad \text{i, j, k} = 1, ..., n \quad \ldots(9)$$

and

$$F_{ijkl} = \sum_{pqrs} C_{pqrs} \phi_p(w_1) \phi_q(x_j) \phi_r(y_k) \phi_s(z_l) \quad \text{i, j, k, l} = 1, ..., n \quad \ldots(10)$$

where the basis functions $\phi_\alpha$ can be chosen to be

$$\phi_\alpha(x) = x^{\alpha-1} \quad \alpha = 1, ..., n \quad \ldots(11)$$

The interpolation conditions in Eq.(9) and Eq.(10) can be expressed as a kronecker products and Eq.(6),

$$F = (T_z \otimes (T_y \otimes T_x)) C_{OLS} + \varepsilon \quad \ldots(12)$$

and

$$F = ((T_z \otimes T_y) \otimes (T_x \otimes T_w)) C_{OLS} + \varepsilon \quad \ldots(13)$$

where

$$T_x = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_n(x_1) \\
\vdots & \ddots & \vdots \\
\phi_1(x_n) & \cdots & \phi_n(x_n) \end{bmatrix}, \quad T_y = \begin{bmatrix} \phi_1(y_1) & \cdots & \phi_n(y_1) \\
\vdots & \ddots & \vdots \\
\phi_1(y_n) & \cdots & \phi_n(y_n) \end{bmatrix}$$

$$T_z = \begin{bmatrix} \phi_1(z_1) & \cdots & \phi_n(z_1) \\
\vdots & \ddots & \vdots \\
\phi_1(z_n) & \cdots & \phi_n(z_n) \end{bmatrix} \quad \text{and} \quad T_w = \begin{bmatrix} \phi_1(w_1) & \cdots & \phi_n(w_1) \\
\vdots & \ddots & \vdots \\
\phi_1(w_n) & \cdots & \phi_n(w_n) \end{bmatrix} \quad \ldots(14)$$

The coefficient $C_{OLS}$ can be computed using the property of kronecker products and least square method as following,

Firstly, from (Eq.(12)) we have
\[ F - (T_z \otimes (T_y \otimes T_x)) C_{OLS} = \varepsilon \]  

(15)

Then, multiplication Eq.(15) by \( \varepsilon^t \) to the both sides, we obtain

\[ [F - (T_z \otimes (T_y \otimes T_x)) C_{OLS}]^t [F - (T_z \otimes (T_y \otimes T_x)) C_{OLS}] = \varepsilon^t \varepsilon \]

that is

\[ F^t F - [(T_z \otimes (T_y \otimes T_x)) C_{OLS}]^t F - F^t [(T_z \otimes (T_y \otimes T_x)) C_{OLS}] + [(T_z \otimes (T_y \otimes T_x)) C_{OLS}]^t [(T_z \otimes (T_y \otimes T_x)) C_{OLS}] = S \]

(16)

We derives in the above equation with respect to \( C_{OLS} \) and equal to zero, we have

\[ \frac{\partial S}{\partial C_{OLS}} = -2[T_z \otimes (T_y \otimes T_x)]^t F + 2[T_z \otimes (T_y \otimes T_x)]^t [(T_z \otimes (T_y \otimes T_x)) C_{OLS}] \]

\[ = 0 \]

then

\[ [T_z \otimes (T_y \otimes T_x)]^t F = [T_z \otimes (T_y \otimes T_x)]^t [(T_z \otimes (T_y \otimes T_x)) C_{OLS}] \]

Now the \( C_{OLS} \) can be efficiently computed using the properties of inverse and associative:

\[ C_{OLS} = ([T_z \otimes (T_y \otimes T_x)]^t [T_z \otimes (T_y \otimes T_x)])^{-1} [T_z \otimes (T_y \otimes T_x)]^t F \]

\[ = [T_z \otimes (T_y \otimes T_x)]^{-1} \{ ([T_z \otimes (T_y \otimes T_x)]^t)^{-1} [T_z \otimes (T_y \otimes T_x)]^t \} F \]

Then

\[ C_{OLS} = [T_z \otimes (T_y \otimes T_x)]^{-1} F \]

(18)

Or by using properties of the kronecker product, we obtain

\[ C_{OLS} = [T_z^{-1} \otimes (T_y^{-1} \otimes T_x^{-1})] F \]

(19)

Assume

\[ C_{lpq} = Q_q P_p L_l \quad l, p, q = 1,...,n \]
And
\[ F_{ijk} = K_k J_j I_i \quad i, j, k = 1, \ldots, n \]
Then
\[ C_{OLS} = (C_{lpq})_{n^3 \times 1} = ((Q_q)_{n \times 1} \otimes (P_p)_{n \times 1}) \otimes (L_l)_{n \times 1} \quad l, p, q = 1, \ldots, n \]
\[ F = (F_{ijk})_{n^3 \times 1} = ((K_k)_{n \times 1} \otimes (J_j)_{n \times 1}) \otimes (I_i)_{n \times 1} \quad i, j, k = 1, \ldots, n \]
At the same method, from Eq.(13), we have
\[ C_{OLS} = [(T_z \otimes T_y) \otimes (T_x \otimes T_w)]^{-1} F \quad \ldots(20) \]
Or
\[ C_{OLS} = [(T_z^{-1} \otimes T_y^{-1}) \otimes (T_x^{-1} \otimes T_w^{-1})] F \quad \ldots(21) \]
Assume
\[ C_{pqrs} = S_s R_r Q_q P_p \quad p, q, r, s = 1, \ldots, n \]
And
\[ F_{ijkl} = L_1 K_k J_j I_i \quad i, j, k, l = 1, \ldots, n \]
Then
\[ C_{OLS} = (C_{pqrs})_{n^4 \times 1} = ((S_s)_{n \times 1} \otimes (R_r)_{n \times 1}) \otimes ((Q_q)_{n \times 1} \otimes (P_p)_{n \times 1}) \]
\[ p, q, r, s = 1, \ldots, n \]
\[ F = (F_{ijkl})_{n^4 \times 1} = ((L_1)_{n \times 1} \otimes (K_k)_{n \times 1}) \otimes ((J_j)_{n \times 1}) \otimes (I_i)_{n \times 1} \]
i, j, k, l = 1, \ldots, n

3. Examples:
The performance of the proposed approach described in the above section will be tested it on the following examples.

**Example (1):**
Consider the following tabulated function values.
Kronecker Product Operations.. Dr. Sudad K., Dr. Amaal A. Issue No. 31/2013

<table>
<thead>
<tr>
<th></th>
<th>x_n</th>
<th>y_n</th>
<th>z_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F</th>
<th>z_1</th>
<th>z_2</th>
<th>z_3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>y_1</td>
<td>y_2</td>
<td>y_3</td>
</tr>
<tr>
<td>x_1</td>
<td>5</td>
<td>21</td>
<td>9.5</td>
</tr>
<tr>
<td>x_2</td>
<td>0</td>
<td>10</td>
<td>1.5</td>
</tr>
<tr>
<td>x_3</td>
<td>10</td>
<td>32</td>
<td>17.5</td>
</tr>
</tbody>
</table>

where \((x_i, y_j, z_k)\) and \(F(x_i, y_j, z_k)\), \(i, j, k = 1, 2, 3\)

Let

\[
F(x, y, z) = \sum_{q=1}^{3} \sum_{p=1}^{3} C_{ijp} \phi_i(x) \phi_j(y) \phi_q(z) \quad ...(22)
\]

where \(\phi_i(x), \phi_j(y)\) and \(\phi_q(z)\) are define as Eq.(11).

At first, find \(T_x, T_y\) and \(T_z\) using Eq.(14)

\[
T_x = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{pmatrix}, \quad T_y = \begin{pmatrix} 1 & y_1 & y_1^2 \\ 1 & y_2 & y_2^2 \\ 1 & y_3 & y_3^2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}
\]

and

\[
T_z = \begin{pmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ 1 & z_3 & z_3^2 \end{pmatrix} = \begin{pmatrix} 1 & 0.25 & 0.0625 \\ 1 & -1 & 1 \end{pmatrix}
\]

Then, by using Eq.(18) or Eq.(19) and Matlab program we find:

\[
(C_{OLS})^i = (0 5 0 0 0 0 0.5 0 0 0 0 0 -2 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
\]

Finally, substituting these values in Eq.(22), we get the interpolation function as:

\[
F(x, y, z) = 5x + 0.5y^2 - 2yz + 3xyz + 3y^2z
\]
Example (2):

Let us consider the following table lists data:

<table>
<thead>
<tr>
<th>n</th>
<th>x_n</th>
<th>y_n</th>
<th>z_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>( \pi/3 )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-1</td>
<td>( \pi/6 )</td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
<td>0</td>
<td>( \pi/4 )</td>
</tr>
</tbody>
</table>

where \((x_i, y_j, z_k)\), \(i, j, k = 1, 2, 3, 4\) and the exact function \(F_{ex}(x, y, z) = x \cdot y \cdot \sin(z)\).

Let

\[
F(x, y, z) = \sum_{q=1}^{4} \sum_{p=1}^{4} \sum_{l=1}^{4} C_{l,p,q} \phi_l(x) \phi_p(y) \phi_q(z) \quad \ldots (23)
\]

Then, find \(T_x\), \(T_y\) and \(T_z\) from Eq.(14)

\[
T_x = \begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 2 & 4 & 8 \\
1 & 1 & 1 & 1 \\
1 & -2 & 4 & -8
\end{pmatrix},
T_y = \begin{pmatrix}
1 & 3 & 9 & 27 \\
1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 \\
1 & 0 & 0 & 0
\end{pmatrix}
&
T_z = \begin{pmatrix}
1 & \pi/3 & \pi^2/9 & \pi^3/27 \\
1 & \pi/2 & \pi^2/4 & \pi^3/8 \\
1 & \pi/6 & \pi^2/36 & \pi^3/216 \\
1 & \pi/4 & \pi^2/16 & \pi^3/64
\end{pmatrix}
\]

Now, by using Eq.(18) and Matlab program we obtain the coefficients \(C_{l,p,q}\), \(l, p, q = 1, 2, 3, 4\):

\[
(C_{l,p,q})^t_{4^3 \times 1} = (0 0 0 0 0 -0.0190 0 -0.0001 0 0 0 0 0 -0.0001 0 0.00002 0 0 0 0 0 1.0872 0 0.0004 0 0 0 0 0 0.0003 0 -0.0001 0 0 0 0 0 -0.1354 0 -0.0004 0 0 0 0 0 -0.0002 0 0.0001 0 0 0 0 0 -0.0915 0 0.0001 0 0 0 0 0 0.0001 0 -0.00002)^t
\]

Substituting these values of Eq.(23), we get the interpolation function:
F(x, y, z) = -0.0190 x y - 0.0001 x^3 y - 0.0001 x y^3 + 0.00002 x^3 y^3 + 1.0872 x y z + 0.0004 x^3 y z + 0.0003 x y^3 z - 0.0001 x^3 y^3 z - 0.1354 x y z^2 - 0.0004 x^3 y z^2 - 0.0002 x y^3 z^2 + 0.0001 x^3 y^3 z^2 - 0.0915 x y z^3 + 0.0001 x^3 y z^3 + 0.0001 x y^3 z^3 - 0.00002 x^3 y^3 z

A comparison between the exact function and interpolation function depending on least square errors for all values of \( F_{i,j,k} \) i, j, k = 1, 2, 3, 4 is

\[ \text{L.S.E.} = 2.294873 \times 10^{-08} \]

**Example (3):**

Let the following table lists data for the 4-variables:

<table>
<thead>
<tr>
<th>n</th>
<th>x_n</th>
<th>y_n</th>
<th>z_n</th>
<th>w_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>1</td>
</tr>
</tbody>
</table>

where \( (x_i, y_j, z_k, w_l) \), i, j, k, l = 1, 2, 3 and the exact function \( F_{\text{ex.}}(w, x, y, z) = w e^z - x^2 y + 3 w x y \).

Let

\[ F(w, x, y, z) = \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{l=1}^{3} C_{pqrs} \phi_p(w) \phi_q(x) \phi_r(y) \phi_s(z) \]

...(24)

Find \( T_w \), \( T_x \), \( T_y \) and \( T_z \) from Eq.(14)

\[ T_w = \begin{pmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, T_x = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}, T_y = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } T_z = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & -2 & 4 \end{pmatrix} \]

Now, by using Eq.(20) or Eq.(21) and Matlab program we obtain the coefficients \( C_{pqrs} \) \( p, q, r, s = 1, 2, 3 \):

\[ (C_{pqrs})_{3 \times 4}^t = (0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3 \ 0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -0.000006 \ 1.28965 \ 0.000006 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \]
Substituting these values of Eq.(24), we get the interpolation function:

\[ F(w, x, y, z) = w + 3 w x y - x^2 y - 0.000006 z + 1.28965 w z + 0.000006 w^2 z - 0.000006 z^2 + 0.42865 w z^2 + 0.000006 w^2 z^2 \]

A comparison between the exact function and interpolation function depending on least square errors for all values of \( F_{i,j,k,l} \), \( i, j, k, l = 1, 2, 3 \) is

\[ \text{L.S.E.} = 4.803638 \times 10^{-8} \]

### 4. Conclusions

In this work, we have presented an application of a kronecker product in interpolation. The interpolation with kronecker products uses the basis functions as \( \phi_\alpha \) functions to find unknown function (3 and 4 - variables function) for the tabulated function values by using least square method. Matlab program (ver.6.5) are used to calculates the inverse matrix as well as the kronecker product of matrices. Efficiency of this method for solving these problems have approved by some examples and found the interpolation functions by this method approximate for the exact function.
References


عمليات ضرب كرونكر المتعدد لإيجاد دوال ذات الأبعاد العليا باستخدام
طريقة المربعات الصغرى

المستخلص

في هذا العمل، نصف تطبيق عملية ضرب كرونكر لبناء الدوال ذات الأبعاد العليا (دوال ل 3 و 4 متغيرات) باستخدام طريقة المربعات الصغرى.

عدد أمثلة أعطيت بشكل خاص لتفصيل التطبيقات التي استعملت لتعريف الطريقة.

المقارنة بين الدالة المضبوطة والدالة المبنية اعتمدت على أقل الأخطاء المبرعة.