High Dimensional Model Representation (HDMR) Based Folded Vector Decomposition

LE Titsya Dıvanıyan
İstanbul Technical University
Informatics Institute
Maslak, 34469 İstanbul TÜRKİYE (TURKEY)
letisya.divanyan@be.itu.edu.tr

Metin Demıralp
İstanbul Technical University
Informatics Institute
Maslak, 34469 İstanbul TÜRKİYE (TURKEY)
metin.demiralp@be.itu.edu.tr

Abstract: The “folded vector” statement which has been accepted by many scientists recently can be employed for naming multiindex arrays some mathematicians prefer to call “tensors” despite their wrong recalling some physical features they do not have here in fact. Folded vectors can be considered as arrays having more than one indices. The purpose of this work is to decompose a folded vector by using High Dimensional Model Representation (HDMR) which has now a quite powerful theory behind it. To this end, the considered array is decomposed to a constant term followed by one index, two indices, three indices, and so on, components. Paper presents the basic conceptual issues to be used for this purpose together with some comments about the practical applications.

Key–Words: Multiindex Arrays, Folded Vectors, High Dimensional Model Representation

1 Arrays With More Than One Indices And Folded Vectors

Especially at the beginning of the new century there has been an increasing tendency to use three and four index entities[1, 2, 3]. These are encountered in signal processing related issues. For example, pictures can be represented by a rectangular array of pixels and can be treated by using the methods of ordinary linear algebra. Whereas an animation or a movie can be considered as the sequence of frames, each of which is represented by matrices of same type, and therefore, their representations can be realized by using three index arrays where one index specifies the frame while the others define the pixel location in the specified frame. A vector can be easily decomposed to some other vectors with much simpler structures. The same thing is also valid for matrices. The important thing in these decompositions is the choice of the simpler vectors we call “basis vectors”. The theory of matrices dictates us the utilization of the spectral items, eigenvalues and eigenvectors. This results in “spectral decomposition”. However this decomposition is restricted to square matrices only. Its extension to nonsquare, or in other words, “rectangular” matrices brings to the fore the idea of singular value decomposition (SVD)[4] which is in fact based on a dual eigenvalue problem between two different dimensional Cartesian spaces via a two way mapping whith one branch specified by the focused matrix while the other branch is in reverse direction and specified by the transpose of the same matrix. SVD is quite popular and its dominant components are used for approximating the original matrix when data compression is needed. Principal Component Analysis and Factor Analysis can be mentioned amongst its utilization possibilities.

It is always possible to convert a vector to an array with more than one indices by reindexing the vector elements. As a simple example we can consider a vector of even number of elements. If we take the upper half of this vector as the first column of a two column matrix then the second column of the same matrix can be taken as lower half of the same vector. So a vector, which is a one index array, has been converted to a matrix, which is a two index array. If the original vector’s number of elements were multiple of four then the uppermost and second uppermost quarters of the vector can be used to construct a two column matrix while the second bottommost and bottommost quarters of the same vector is used to construct another two column matrix. These two matrices can be considered as the subarrays of a three index array. Thus, for this case, a vector, which is a one index array, becomes a three index array. The actions to convert a vector to a more than one index array can be called “folding”. In this sense, a multiindex array can be considered as a folded vector [5, 6]. The reverse action to folding procedure can be called “unfolding”. It is always possible to convert a multiindex array to a vector by using an
appropriate unfolding operation.

The rest of the paper is organised as follows. Next section recalls High Dimensional Model Representation in its plain form. The third section is devoted to the HDMR of a three index entity. While the fine points and certain pitfalls in the applications are given in the fourth section, the fifth section; finalizes the paper with concluding remarks.

2 High Dimensional Model Representation (HDMR)

High dimensional model representation (HDMR) was first proposed by Sobol in its plain form where the unit hypercubic (which is located in positive hyper-quadrant with one corner sitting on the origin) geometry and unit constant weight function are assumed to be used. HDMR’s plain form was extended to the cases where the geometry becomes more flexible by using a rectangular hyperspace and weighting possibility is made available by proposing a product type weight function whose each factor is a univariate function of a distinct independent variable, by Rabitz and his group. Rabitz and his group also defined certain particular types of HDMR[7]. Demiralp and his group developed various new forms of HDMR[8] and brought the very important quality estimators they called “Additivity Measurers”[9]. The introduction of support functions and definition of the broad extension to HDMR, called “Enhanced Multivariance Products Representation (EMPR)” served to create a powerful theory which is almost complete today[10]. Some other authors also published many papers related to HDMR applications.

HDMR, in its plain form, can be expressed by the following equality for a multivariate function

\[ f(x_1, \ldots, x_N) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{i,j=1}^{N} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots N}(x_1, \ldots, x_N) \tag{1} \]

where the right hand side components \( f_0, f_i(x_i) \) \( f_{ij}(x_i, x_j) \) are the constant term which is a scalar, univariate components, bivariate components respectively.

There are \( 2^N \) number of the components at the right hand side of (1). Hence, HDMR is a finite term decomposition. However, as \( N \) grows, the number of its terms increases very rapidly and makes its utilization quite impractical. This urges us to truncate the right hand side of (1) at some specified level of multivariate. The most preferred truncations are at the levels of constancy, more realistically univariance, or at most bivariance. This need for truncation enforces us to find some scalars to measure the truncation quality. However, before doing so, what we need is to impose some conditions to determine the HDMR components for a given multivariate functions. To this end the basic facility is the Sobol impositions which were a little bit more generalized by Rabitz for the extensions he proposed. The Sobol imposition states that each HDMR component’s integral except the one for the constant term over anyone of its independent variables on the corresponding interval under the corresponding weight factor should vanish. As can be shown by mathematical induction this imposition is sufficient to determine each HDMR component uniquely. On the other hand, these “vanishing integral conditions” result in mutual orthogonality amongst the HDMR components. This orthogonality is the basic issue leading us to define additivity measurers. The \( k \)th additivity measurer is the ratio of the norm square of the sum of the terms up to and including \( k \) level of multivariance to the norm square of the given multivariate function. This definition is based on Hilbert space in which the original function and its components lie. The inner product is weighed by the weight function of HDMR and defined on the orthogonal geometry of HDMR. These additivity measurers form a well ordered sequence which is bounded by 0 (lower bound) and 1 (upper bound).

Although we have given the basics of plain HD-MR only, there have been many developments about HDMR. Amongst these, Factorized HDMR[11], Generalized HDMR[12], Hybrid HDMR [13], Cut and Multicut HDMR, Small Scale and Combined Small Scale HDMR[14] are perhaps the ones investigated in detail. Quite recently, univariate support functions are introduced to elevate each HDMR component to maximum multivariance level. This increases the numerical efficiency in truncation approximations. This decomposition method has been called “Enhanced Multivariate Product Representation (EMPR) and, especially in Demiralp’s group, a remarkably increasing tendency has been shown on the development of this method and its practical applications.

In the studies of Demiralp group HDMR was considered to decompose the multivariate functions whose independent variables were taking values from certain intervals. In other words, the objects at the focus were continuously varying mathematical items. Whereas, a function of a continuous variable, can be given through either an evaluation rule or generally a finite list of function values versus some specified independent variable values. The second case covers
only a discrete set of informations, from which, continuity is provided by interpolation. Data partitioning type studies in our group focused on such problems and HDMR, FHDMR, GHDMR and Hybrid HDMR were successfully used for the obtention of the true or best functional structure. In these works, Dirac delta function products type weight function have been used to discretize the plain or derived HDMR at the focus. Quite recently, we have started to deal with the arrays by directly defining a new HDMR class which can be called “Discrete HDMR Category” while the HDMR varieties on continuous functions are gathered into another class named “Continuous HDMR Category”. Of course, there is another category about the mixed type HDMRs dealing with array valued functions which seem to be our future works from now. In these senses, this work can be considered as a preliminary study for the HDMR construction for arrays.

3 The HDMR Based Decomposition of Three Index Arrays

We focus on three index arrays and their HDMR based decompositions in this section for simplicity. What we develop here can be easily extended to higher than three index arrays. We are going to use \( f \) to symbolize the array for recalling the HDMR. The general term of the array will be denoted by \( f_{i_1,i_2,i_3} \) where \( i_j \) \( (j = 1, 2, 3) \) takes values between 1 and a positive integer, say \( n_j \) and we have used the comma symbol in the indexing for clarity (to prevent the confusion of more than one digits numbers).

The HDMR of \( f \) can be elementwisely expressed as follows

\[
\begin{align*}
    f_{i_1,i_2,i_3} &= f^{(0)} + \sum_{j=1}^{3} f^{(j)}_{i_j} + \sum_{j,k=1 \atop j < k}^{3} f^{(j,k)}_{i_j,i_k} \\
    &+ f^{(1,2,3)}_{i_1,i_2,i_3}, \\
    &1 \leq i_j \leq n_j, \quad j = 1, 2, 3
\end{align*}
\] (2)

which is impressed from the plain HDMR on the continuous functions. Even though, this is in fact the very specific case of the HDMR on arrays, the extension to arrays with any given number of indices is quite straightforward. The independent variables are changed by the indices which are discrete entities and their domains are no longer intervals but some finite sets whose elements are taken from the first elements of the positive integer set. The multivariance is now related to the number of indices appearing in each component. Once again the components are ordered in ascending multivariance, that is, ascending number of indices, starting from the constant term whose multivariance is zero.

To determine the right hand side components in (2) we can use the sums instead of integrals in plain HDMR. However, we may need to use also weights under the sums to give different importances to different indices. This urges us to seek how the weight is defined for this case. As we know from plain HDMR, the integrals used in the component determination can also be related to inner products for an appropriately defined Hilbert space. Inner products can have weight functions in plain HDMR. On the other hand one of the basic definitions of an inner product is its match with the norm definition when its both arguments are same. This leads us to seek how to appropriately use a weight in norm definition. All impressions and experiences from plain HDMR point to the following norm definition for a given vector \( a \) with \( n \) elements

\[
\|a\| \equiv (a^T Wa)^{1/2} = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} a_i W_{i,j} a_j \right)^{1/2}
\] (3)

where \( W \) stands for a matrix of \( n \times n \) as can be immediately noticed. The matrix \( W \) must be symmetric to provide the inner product’s symmetry over real vectors, and, positive definite to fulfill the requirement that the norm of a nonzero vector never vanishes. All these requirements leads us to use the following definition for weight matrix

\[
W \equiv \Omega^T \Omega
\] (4)

where \( \Omega \) stands for a nonsingular arbitrary matrix. This facilitates the construction of a true weight matrix. Of course, the simplest weight matrix is the one proportional to unit matrix even though the diagonal matrices with positive elements can serve for the same level of simplicity.

If the vector \( a \) is replaced by a folded vector of three indices then we can write

\[
\|a\|^2 = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} a_{i_1,i_2,i_3} \times W_{i_1,j_1}^{(1)} W_{i_2,j_2}^{(2)} W_{i_3,j_3}^{(3)} a_{i_1,i_2,i_3}
\] (5)

where the total weight array \( W \) which is a six index entity can be expressed as follows

\[
W_{i_1,i_2,i_3,j_1,j_2,j_3} \equiv W_{i_1,j_1}^{(1)} W_{i_2,j_2}^{(2)} W_{i_3,j_3}^{(3)}
\] (6)

where the subsindices of the overall weight array are deliberately partitioned to two groups, one including \( i_1, i_2, i_3 \) and the other including \( j_1, j_2, j_3 \). This is for...
the fact that \( j \)'s play the role of column indices while \( i \)'s behave like row indices, and, the overall array can be considered as a mapping from the space of three index entities whose indices vary between 1 and \( n_1 \) inclusive, between 1 and \( n_2 \) inclusive, and, between 1 and \( n_3 \) inclusive respectively. Hence, \( W \) the overall weight array can be considered as a folded matrix with the same folding order, 3, in column and row indices.

The identity in (6) specifies the overall weight array to product type which is composed from three separate weight matrices through a three way outer product. Each matrix factor, \( W^{(1)} \), \( W^{(2)} \), and, \( W^{(3)} \) is responsible for a mapping over an individual index, either \( j_1 \) or \( j_2 \) or \( j_3 \). This in fact corresponds to the product type structure of the weight matrix in plain HDMR.

Now having the defined weight matrices we can proceed to the determination of the HDMR components. To this end, we can multiply the both sides of (2) by the overall weight array’s general element and then we can sum the resulting indexed entities over all the possible values of all indices. This gives

\[
\sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f_{j_1,j_2,j_3} = \sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(0)} + \sum_{k=1}^{n_k} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(k)} + \sum_{k_1,k_2=1}^{n_{k_1}} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(k_1,k_2)} + \sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(1,2,3)}
\]

where we have changed the indices of (2) by some other symbols for preventing the confusion with the \( i \) and \( j \) indices of the weight factors, and we have used the following shorthand summation notation

\[
\sum_{(i,j)} \equiv \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3}
\]

As can be seen easily scaling of any weight factor in (7) does not destroy the equality. This enables us to impose the following sum normalizations

\[
\sum_{i_k=1}^{n_k} W^{(k)}_{i_k,j_k} = 1, \quad k = 1, 2, 3
\]

which allow us to rewrite (7) as follows

\[
\sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f_{j_1,j_2,j_3} = f^{(0)} + \sum_{k=1}^{n_k} n_k W^{(k)}_{i_k,j_k} f^{(k)} + \sum_{k_1,k_2=1}^{n_{k_1}} n_{k_1} W^{(k_1)}_{i_1,j_1} W^{(k_2)}_{i_2,j_2} f^{(k_1,k_2)} + \sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(1,2,3)}
\]

This equation is analogous to the constant component determining equation of plain HDMR. It contains 7 unknown array type entities in addition to the very beginning constant term. This means flexibilities which may be used to impose certain conditions for unique component determination. Since the plain HDMR counterpart of (10) contains only constant component as unknown we can do the same thing and impose the following conditions for the right hand side components of (2)

\[
\sum_{i_k=1}^{n_k} \sum_{j_k=1}^{n_k} W^{(k)}_{i_k,j_k} f^{(k)} = 0, \quad k = 1, 2, 3;
\]

\[
\sum_{i_k=1}^{n_k} \sum_{j_k=1}^{n_k} \sum_{j_{k_2}=1}^{n_{j_{k_2}}} \sum_{j_{k_3}=1}^{n_{j_{k_3}}} W^{(k_1)}_{i_{k_1},j_{k_1}} W^{(k_2)}_{i_{k_2},j_{k_2}} f^{(k_1,k_2)} = 0, \quad k_1, k_2 = 1, 2, 3, \quad k_1 < k_2;
\]

\[
\sum_{(i,j)} W^{(1)}_{i_1,j_1} W^{(2)}_{i_2,j_2} W^{(3)}_{i_3,j_3} f^{(1,2,3)} = 0
\]

which are seven conditions for seven unknowns. The first three equations here are completely analogous to the “Vanishing Integral Conditions” of Sobol. The second three equations can however be considered as derived from the following conditions.

\[
\sum_{i_{k_3}=1}^{n_{i_{k_3}}} \sum_{j_{k_3}=1}^{n_{j_{k_3}}} W^{(k_{1,k_2})}_{i_{k_1},j_{k_1};j_{k_2}} f^{(k_1,k_2)} = 0, \quad k_1, k_2 = 1, 2, 3, \quad k_1 < k_2, \quad k_3 = k_1, k_2
\]

(12) and the first group conditions in (11) imply that the following “Vanishing Sum Conditions” can be imposed for discrete HDMR components: Any Discrete HDMR component except the constant term vanishes
under the summation with respect to one of its indices over the relevant domain and under the related weight matrix factor. This is absolutely analogous to the Vanishing Integral Conditions of HDMR.

The vanishing sums conditions permit us to rewrite (10) as follows

$$f^{(0)} = \sum_{(i,j)}^* W_{i_1,j_1}^{(1)} W_{i_2,j_2}^{(2)} W_{i_3,j_3}^{(3)} f_{j_1,j_2,j_3}$$

(13)

which is apparently a unique relation to determine the constant component of Discrete HDMR.

To evaluate the univariate (one index array) Discrete HDMR component depending on the index $i_1$ we can elementwisely multiply both sides of (2) by the weight matrices $W^{(2)}$ and $W^{(3)}$ (for this purpose some symbols are changed not to confuse with the indices of the matrices and $i_1$), and then, take the sums over all indices except $i_1$. This gives

$$f_{i_1}^{(1)} = \sum_{i_2=1}^{n_2} \sum_{j_2=1}^{n_2} \sum_{i_3=1}^{n_3} \sum_{j_3=1}^{n_3} W_{i_2,j_2}^{(2)} W_{i_3,j_3}^{(3)} f_{i_1,j_2,j_3} - f^{(0)}, \quad i_1 = 1, 2, ..., n_1$$

(14)

where we have used the vanishing sums conditions. The other univariate Discrete HDMR components can be determined through the same way and determination, as seen, is unique.

In the determination of the constant term for the Discrete HDMR we have used six fold sums while the univariate components required four fold sums. It is not hard to guess that two fold summations suffice for the determination of the bivariate terms. After the determination of all terms except the trivariate one which remains as the only unknown, we can directly evaluate the trivariate component from the Discrete HDMR equality.

### 4 Possible Pitfalls and Certain Issues

We have shown that the vanishing sums conditions take us to the unique determination of the Discrete HDMR. The basic reason why these conditions lead to the unique Discrete HDMR component determination is the orthogonality in the geometry and the product type weight arrays. The orthogonal geometry comes from the domain definitions of the indices. If the domain of one index were dependent on the values of the other indices then the geometry would not be orthogonal. The orthogonality also implies the separability of the sums. The other reason for the separability is the product type weight. The use of nonproduct type weight functions together with the vanishing sums conditions causes some inconsistencies in the component determination. Either orthogonal geometry and product type weights should be used or some other type impositions instead of the vanishing sums conditions should be found to remove the inconsistencies. These are important pitfalls in the definition of the Discrete HDMR as their counterparts are in Continuous HDMR.

The formulae of the previous section are quite complicated and this complication may become untreatable when the number of the indices in the array increases. Basic reason for this is the utilization of the weight arrays which have been kept quite general. However, the utilization of diagonal type matrices as weight array factors reduces the complexity because of the disappearance of certain sums whose number is halved by the diagonality.

Animations or videos are perhaps the best examples for the three index arrays (as we have mentioned previously) amongst the cases encountered in practice. For compression purposes or for easy and healthy interpretations these arrays are generally treated by singular value decomposition (SVD) based techniques. SVD is a decomposition so is the Discrete HDMR. Hence, it is a good idea to compare certain implementation results for both approaches. SVD is more rigid in comparison with the Discrete HDMR where the weight array gives the opportunity of monitoring the truncation approximations such that the numerical convergence rate, that is, the approximation quality can be made more controllable.

### 5 Conclusion

We have constructed the Discrete HDMR as an array decomposing method in this work. We restricted our investigations to the arrays having three indices for simplicity. However this does not exclude the chance of practical applicability of the decomposition we have developed. Nevertheless, animation or similar type applications are possible.

The representation we have developed here is analogous to the plain HDMR. All features encountered in Continuous HDMR seem to find their counterparts in Discrete HDMR. The vanishing sums conditions, orthogonal geometry, product type weight are amongst them.

We have not focused on the possibility of defining the additivity measurers, or in other words, quality measurers for the truncation approximants as it is done in Continuous HDMR. The vanishing sums conditions imply the orthogonality amongst the components via an inner product defined through weighed sums. This orthogonality brings separability to com-
ponents in norm square so enables us to define certain scalars as the ratios of norm squares of truncations at certain level multivariance to the norm square of the original array. These scalars are upperbounded by 1 and lowerbounded by 0 and form a well ordered sequence. Hence, it is possible to make good estimations about the performance of the truncations in numerical approximations.

The extension of the presented representation to the cases of more than three index entities seems to be quite straightforward. Hence we do not want to get into further details.

The performance in numerical approaches by using Discrete HDMR truncations can be increased by using either support arrays or transformations such that the focus is shifted to another HDMR with comparably high level of constancy. These and other similar issues are now under intense studies and the results would be reported possibly in our future publications.

Acknowledgements: The second author is grateful to Turkish Academy of Sciences, where he is a principal member, for its support and motivation.

References :


