

Converting a Multi-Dimensional Matrix to a One-Dimensional Matrix by Coding Devices

Mohammad Ali Khandan and Mehdi Delkhosh*

Islamic Azad University, Bardaskan Branch, Department of Computer, Bardaskan, Iran

Corresponding author: Mehdi Delkhosh

ABSTRACT: In many sciences, there are multi-dimensional matrices in various dimensions. Such as, image processing, heating problems in physics and mechanics, Objects processing in radar systems, Weather Information Processing, and etc. There are storage problems when you are working with multi-dimensional matrices because the data volume is high. Also, using these matrices in the various algorithms is time-consuming and processing time is very high. In this paper, we will convert multi-dimensional matrix to a one-dimensional matrix (vector) by coding number, in order to reduce storage volume and increase the processing speed of the algorithms.

Keywords: Sorting Data, Multi-Dimensional Matrix, Coding Number, Pairing Function, Gödel Function, Cantor Pairing Function, Coding by Base.

INTRODUCTION

In many sciences, we use data coding for maintaining data security, data storage and data transfer from one computer to another.

In practice, there are different methods of coding, that each of them have advantages, disadvantages and applications.

In this paper, we introduce some famous and useful coding methods, and we'll expression some of the advantages and disadvantages of them, and then we using coding by Base for convert a multi-dimensional matrix to a one-dimensional matrix (vector), in order to reduce storage volume and increase the processing speed of the algorithms.

Introducing Some Coding Functions for Converting Several Numbers to a Unique Number

In many sciences, use data coding for maintaining data security, data storage and data transfer from one computer to another.

In practice, there are different methods of coding, that each of them have advantages, disadvantages and applications.

In this section, we introduce some famous and useful coding methods.

Coding by Pairing Function

By this coding, we can convert two numbers to a unique number [Davis and et al 1994, Nayebi 2013, Gödel 1931]. As follow:

$$\langle x, y \rangle = 2^x (2y + 1) - 1 = z$$

In this method, we just can convert a pair of numbers to a number in each step. For converting three numbers to a unique number, first, we should convert first and second numbers to a number, then code the generated number with third number. Like this:

Suppose, we want to convert three numbers a, b, c to a unique number:

$$\langle a, b \rangle = z$$

$$\langle z, c \rangle = k$$

k is a coded number from numbers a, b, c .

Problems of method:

- 1- Increase amount of arithmetic operations if increase count of numbers. Because we have to convert a pair of numbers and for other numbers do it again.
- 2- In each step, one of numbers appear in a power of 2, so the generated number is exponential and huge.
- 3- Decoding the number in order to get initial numbers is time consuming and difficult.

Coding by Gödel Function

In this coding method we can code several numbers to a unique number simultaneously [Kohlenbach 2010, Doria and Doria 2011, Negadi 2008]. As follow:

$$[a, b, c, d, \dots] = 2^a \cdot 3^b \cdot 5^c \cdot 7^d \dots$$

In this method, each of the numbers, appear to power of one of prime numbers.

The advantage of this method compared to Pairing Function is that, it can code several numbers simultaneously.

Problems of method:

- 1- The achieved number is exponential and huge, and maybe we encounter lack of storage space in the computer variables.
- 2- Obtaining prime numbers, need to a separate algorithm. This action reduces the speed of the Gödel's algorithm. It increases order of Gödel's algorithm.

Coding by Cantor Pairing Function

Cantor Pairing Function can convert a pair of nonnegative integer numbers to a unique number [Cegielskia 1999, Doria and Doria 2011].

Cantor pairing function is defined as follows, and we will be represented by symbol π .

$$\pi : N \times N \rightarrow N$$

$$\pi(k_1, k_2) := \frac{1}{2}(k_1 + k_2)(k_1 + k_2 + 1) + k_2$$

The function takes a pair of nonnegative integer k_1, k_2 and get $\pi(k_1, k_2)$ to output.

Cantor pairing function for more that 2 parameters:

$$\pi^{(n)}(k_1, \dots, k_{n-1}, k_n) := \pi(\pi^{(n-1)}(k_1, \dots, k_{n-1}), k_n)$$

Figure 1 shows numbers that generated by algorithm.

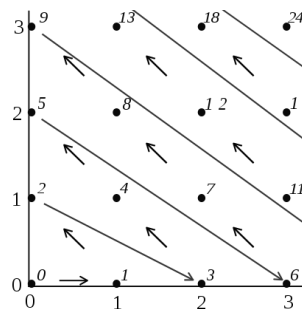


Figure 1. the numbers generated by cantor pairing function's algorithm

In order to decode cantor pairing function we can do as follows:

$$z = \pi(x, y) = \frac{(x + y)(x + y + 1)}{2} + y$$

$$w = x + y$$

$$t = \frac{w(w + 1)}{2}$$

$$z = t + y$$

$$w^2 + w - 2t = 0$$

$$w = \frac{\sqrt{8t+1}-1}{2}$$

$$t \leq z = t + y < t + (w+1) = \frac{(w+1)^2 + (w+1)}{2}$$

$$w \leq \frac{\sqrt{8z+1}-1}{2} < w+1$$

$$w = \left\lceil \frac{\sqrt{8z+1}-1}{2} \right\rceil$$

So, to calculate x, y from z , we do as follow:

$$t = \frac{w(w+1)}{2}$$

$$y = z - t$$

$$x = w - y$$

Problems of method:

- 1- Increase amount of arithmetic operations if increase count of numbers.
- 2- Achieved number from more than two numbers is very large.

Coding by Base

This new coding method uses base. The function takes a set of numbers and the base number for this set as input, and get a unique number as output.

Note that all numbers in the set must be smaller than base number. In other words, advantage of the method is that we could specify base number according to biggest number that can appear to the set, and by this way the coding number becomes small.

The coding function by Base can be defined as follows:

$$B(a_1, a_2, \dots, a_n; b) = \begin{cases} a_1b^0 + a_2b^1 + \dots + a_nb^{n-1} & \text{if } (a_1 < b, a_2 < b, \dots, a_n < b,) \\ -1 & \text{otherwise} \end{cases}$$

Multi-Dimensional Matrices

Introduction of Multi-Dimensional Matrices

Normal matrices that used in mathematical applications have 2 dimensions. Usually, we display rows by i and columns by j .

In the multi-dimensional matrices, dimensions of matrix is more than 2, so working with them is more difficult than normal matrices.

Matrices are used to store tabular data, which often depend on several factors. For example, the stored data in two-dimensional matrices are dependent on two factors, or we can say that, we need two factors for defining an entry of a matrix.

Suppose we have a sequence of teachers of a university and a sequence of courses are taught at this university. We want to know what courses each teacher can teach. That university to decide what each courses will be presented to one professor. To display this information, we need a two-dimensional matrix that in its row we have name of teachers and in its column, name of courses (Table 1). Also we can do it for matrices with more than two dimensions. For example: if we want to show time of class that teacher teach a course, we can use a 3 dimensions matrix. Each of dimensions of matrix shows one of its information, courses - teachers-time classes (Figure 2).

Table 1. This matrix represents relations between teachers and courses of a university

	Analysis	Graph	Chemistry	Mathematical
Delkhosh	0	1	0	1
Darbandi	0	0	1	0
Mahdian	0	0	1	1
Khandan	1	1	1	0

Experience has shown, in most cases, when length of rows increases, the ratio of zero data or useless data grows. For example, in above matrix (Figure 2), assume we have courses of various fields, and teachers from various fields, where each teacher can only teach courses in his fields, so we can see a lot of data in matrix have zero value.

Sparse matrix is a matrix with many zero data. The important property in sparse matrix is that locations of useless data are specified. But in many of matrices, especially in the multi-dimensional matrices, locations of useless data are not specified exactly. So in many cases, we cannot use sparse matrices. Such as above example: offering courses in a university, until, all course not offered, and the professors are not set, locations of unnecessary data are unclear exactly.

Now imagine matrices with more than 2 dimensions and great length for each row. You can see zero data are growing and we store this data but in practice we do not use of them.

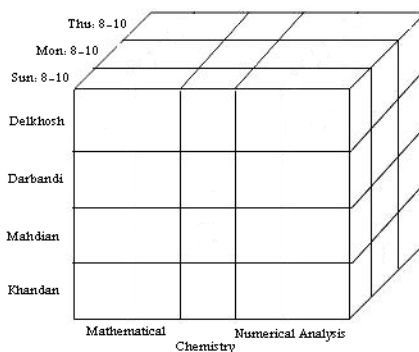


Figure 2. three-dimensional matrix with dimensions courses - teachers - time classes

The amount of memory consumed by multi-dimensional matrices in computer

In language C++, the amount of memory consumption for an *integer number* is 2 bytes. Assume length of each row is 100 (in practice, we have many matrices with length of rows more than 100, as example above: number of courses in each university has more than 700). We show it in Table 2.

In table 2, we assume each entries of matrix are an integer number. Note that when we increase the dimensions of matrix, the amount of memory to store the matrix grows. Consider in 5-dimensional matrix, the amount of memory consumed is 18.63 GB, where there is not such RAM memory for a home computer, and in 6-dimensional matrix, the amount of memory consumed is 1.82 TB, where there is not such RAM memory on any computer.

Dimension	The length of each row	Total number of entries	Consumer space(Byte)	Consumer space (large unit)
1	100	10^2	2×10^2	0.20 Kb
2	100	10^4	2×10^4	19.53 Kb
3	100	10^6	2×10^6	1.91 Mb
4	100	10^8	2×10^8	190.73 Mb
5	100	10^{10}	2×10^{10}	18.63 Gb
6	100	10^{12}	2×10^{12}	1.82 Tb

Table 2. The amount of memory consumed by multi-dimensional matrices in computer

Therefore, using of large-dimensional matrices and store them on computer with usual methods are not feasible and economical.

Here we assume that type of matrix entries is *int*. If we assume that type of matrix entries is *long or double or object of a class or etc* then the amount of memory consumed by a computer will grown.

As previously mentioned, most multi-dimensional matrix entries are zero or useless. One of techniques that will help storing large matrices is that we do not need store useless entries and just save entries that will be perform operations on them, in order to reduce volume of stored data.

In this paper we represent a method by above technique that it can store huge multi dimensional matrices in RAM of computer easily and perform processing on them.

Storing Multi-Dimensional Matrices by Coding Devices in a One Dimensional Matrix (Vector or List)

According to above, if we can save only entries that have information and ignore useless entries, then we can reduce volume of stored multi-dimensional matrix.

To do this, we introduce a vector modeling of multi-dimensional matrix.

If we consider a 4-dimensional matrix, we can say each entries of a 4-dimensional matrix include 5 properties:

- 4 properties for represent index number of rows.
- A property for value of stored data in entry.

We can convert number of rows to a number by use of coding operation or converting several numbers to a unique number. Now there are only two properties for each entry of matrix. Thus, we can store entries by a vector or list. And if we sort the values of list according to coding number that obtained from converting number of rows, then we have a *list* or *vector* or *one-dimensional matrix* that sorted according to all number of rows.

But, it is important which coding function is suitable where :

- 1- All coded numbers, be unique.
 - 2- Obtained numbers are not too big, in order to store them in computer memory.
 - 3- Coding and decoding of numbers do not need long and complicated calculations.
- To answer this question, we have examined coding functions, which we introduced in the beginning.

Pairing Function

Paring Function takes two numbers convert them to a number. It always gives us a unique number. So far it is appropriate for our system. But, for converting three numbers to a unique number, first two numbers have converted to a number, and then code generated number with third number, where this raises volume computation. Paring Function was defined as follow:

$$\langle x, y \rangle = 2^x (2y + 1) - 1$$

Note that in this method, the first number reaches to the power of 2, that this can generate a large number. Now, for example, we will check number that produced for three numbers 32, 57 and 15

$$\langle 32, 57, 15 \rangle = \langle \langle 32, 57 \rangle, 15 \rangle$$

$$\langle 32, 57 \rangle = 2^{32} (2 \times 57 + 1) - 1 = 493,921,239,039$$

$$\langle 493921239039, 15 \rangle = 2^{493921239039} (2 \times 15 + 1) - 1 = ?????????$$

If you want to calculate the answer of last equation, even in advanced mathematics software like *Mathematica* and *Matlab* gives *overflow* error.

Therefore, this method will produce a large number and it is not suitable.

Gödel Function

This coding method can simultaneously converts several numbers to a unique number. As follow:

$$[a, b, c, d, \dots] = 2^a . 3^b . 5^c . 7^d \dots$$

Now, for example, we will check number that produced for three numbers 32, 57 and 15

$$[32, 57, 15] = 2^{32} . 3^{57} . 5^{15}$$

$$= 205,788,662,868,486,601,000,948,146,241,536,000,000,000,000$$

As you can see, the obtained number is very large, and even it can not be stored in the *long* and *double* variables. Also, it is very difficult to decode and work with large number.

Cantor Pairing Function

In cantor pairing function as Paring Function, we need to code data two by two.

Now, for example, we will check number that produced for three numbers 32, 57 and 15

$$\pi^3(32, 57, 15) = \pi(\pi^2(32, 57), 15)$$

$$\pi^2(32, 57) = \frac{1}{2} (32 + 57)(32 + 57 + 1) + 57 = 4062$$

$$\pi^2(4062,15) = \frac{1}{2}(4062+15)(4062+15+1)+15 = 8,313,018$$

Coded number is much smaller than the value obtained in two previous methods.

But, the problem of this method is that, when increase count of numbers (more than 3), grows the coded number and increase amount of computation.

Coding by Base

In coding by base, we should already know the numbers (the row index) is not greater than a certain value.

In example of three numbers 32, 57 and 15, we assume that the number indexes can not be more than 99, so we consider the base number 100.

$$B(32,57,15;100) = 32 \times 100^0 + 57 \times 100^1 + 15 \times 100^2 = 155,732$$

As you can see, the obtained number is smaller than 3 previous methods, and the amount calculations are less.

Also, we know the generated codes will not be more than a certain number (as defined in array, we must first determine the length of array). That is, according to the base number and numbers, we can calculate the upper bound of the coding. For example, in above sample, the upper bound for three numbers with base 100 will be:

$$B(99,99,99;100) = 99 \times 100^0 + 99 \times 100^1 + 99 \times 100^2 = 999,999$$

I.e. biggest generated code number for three numbers with base 100 is 999999, where it can store in a long integer easily.

As another example, See Figure 3, data in a 3-dimensional matrix with the coding base was converted to a vector, and is stored in a less memory in a computer.

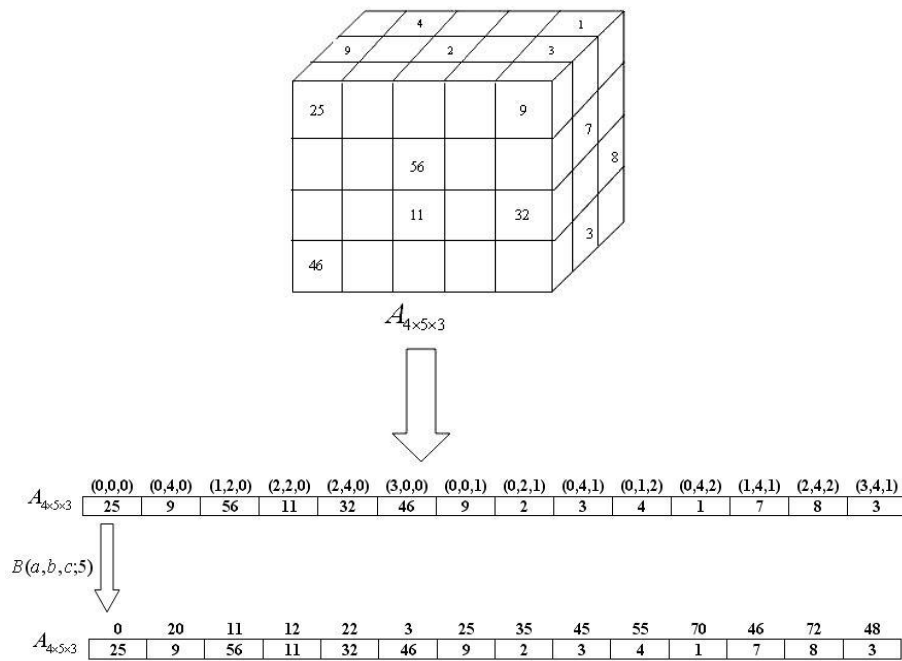


Figure 3. Converting a 3-Dimensional Matrix to a vector by Base Coding

**Some of Applications of Vector Matrix (Matrix into Vector Conversion)
Image Processing**

In Image Processing, we are dealing with data that includes three properties length, width, height and color. Also the data are considered for an image at the pixel level. Thus, for an image, we have a lot of data that must be processed. We can use four-dimensional matrix for displaying and saving data, where each row represents one of the properties of length, width, height and color. Because we work with a lot of data, we can use matrices that convert to a vector that it has less volume and working with them are easier. (Due to lower data)

Solving the system of equations

Solving the system of equations $Ax = b$ such that A is a two-dimensional matrix has been talked a lot in mathematics. Now suppose that A is a multidimensional matrix whose count of dimensions is greater than two. In this case, to solve the system with current methods used for two-dimensional matrices are not suitable. Also, due to the high data, if we work with current methods in matrices, Processing time increases and high consumption of memory needed to store. But, if we convert the multi-dimensional matrix to a vector, and then we get the methods to work with this vector, which gives us the same answer, then either comes down processing time of algorithm and requires less memory than previous methods.

Physical problems

In physical problems, Such as the problem of heat transfer in solids, we are dealing with data that includes four properties length, width, height and time. To solve them, we can use of Elements method. Thus, for an object, we have a lot of data that must be stored and processed. We can use a four-dimensional matrix for displaying and saving data, where each row represents one of the properties of length, width, height and color.

Planning Courses

The technique described in this paper, is used in software that is written by authors. Suppose that, as the head of college or university, you're interested in scheduling the semester classes for your college or university. Now, you should check the many restrictions (Constraints such as: the count of classes, teaching duty hours, distance between each session, courses interaction in majors, courses interaction in terms, empty time of teacher, courses that proposed by teacher, count of session for each course, calculating the time of class of teachers and etc). These operations manually are time consuming. We stored this information in a four dimensional matrix. Any dimension of matrix in this case was considered: Courses, teachers, empty time teachers for courses, and Fields of study in semester. Then, using technique in this paper, the information is stored in memory and with genetic algorithm to find the optimum answer.

It is clear that:

The length of rows in some of dimensions is more than 1000.

The complexity is very high. (Sipser, 2004) has written in his book: "Computer problems come in different varieties; some are easy, and some are hard. For example, the sorting problem is an easy one. Say that you need to arrange a list of numbers in ascending order. Even a small computer can sort a million numbers rather quickly. Compare that to a scheduling problem. Say that you must find a schedule of classes for the entire university to satisfy some reasonable constraints, such as that no two classes take place in the same room at the same time. The scheduling problem seems to be much harder than the sorting problem. If you have just a thousand classes, finding the best schedule may require centuries, even with a supercomputer.")

According to, structure and relationship between restrictions, we cannot solve the problem with blocking or direct methods.

For processing and problem solving, we should load entries of five-dimensional matrix in computer memory, and then perform operations and calculations on it, that it is not possible. Because if we want to use standard methods to store and process information, we will need More than 300000 GB of RAM in a computer.

However, by the technique described in this paper, this problem can be ran and processed with a normal computer (With 2 GB RAM). Furthermore, because the generated code for each entry of the matrix are not very large (see Figure 3), they can easily to store in a variable of type long.

CONCLUSION

We can use the techniques described in this paper for convert a multi-dimensional matrix to a vector. With this work, we can reduce the memory required to processing and storing a large multi-dimensional matrix, and we can solve many application problems (Such as Image Processing, Data sending, Wireless data sending, Study of physical problems in three dimensions and above, and etc.) and even solving some of the problems that are hard to solve (Such as Planning Courses).

REFERENCES

Cegielskia P and Richard D. 1999. On arithmetical first-order theories allowing encoding and decoding of lists, Theoretical Computer Science, Vol. 222, Issues 1–2, pp. 55–75.

- Davis MD, Sigal R and Weyuker EJ. 1994. Computability, Complexity, and Languages Fundamentals of Theoretical Computer Science, ACADEMIC PRESS LIMITED.
- Doria FA and Doria M. 2011. Einstein, Godel, and the Mathematics of Time,, Find out how to access preview-only content Brazilian Studies in Philosophy and History of Science , Boston Studies in the Philosophy of Science, Vol. 290, pp. 269-279 , Springer.
- Godel K. 1931. Uber formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme I, Monatsheft für Math. und Physik, Vol. 38, pp. 173–198.
- Kohlenbach U. 2010. Godel functional interpretation and weak compactness, Annals of Pure and Applied Logic, Vol. 163, Issue 11, November 2012, pp. 1560–1579, Kurt Goedel Research Prize Fellowships.
- Nayebi A. 2013. on the elimination of the bounded universal quantifier for Diophantine predicates, Logic Seminar, Stanford.
- Negadi T. 2008. The genetic code Via Godel encoding, Submitted on 6 May 2008, <http://arxiv.org/abs/0805.0695>.
- Sipser M. 2004. Introduction to the theory of computation (Second edition), Cambridge, Massachusetts.