INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 5)
Available online at: www.ijariit.com

Reduction and reproduction of information using special principle

Nikhil S. Bharadhwaj
nikpersonal011235813@gmail.com

ABSTRACT

The paper written here regards a method which invokes a certain principle to reduce the content of a collection of data. The original content can be replicated in the exact same way. Meaning, there will be no data loss after the procedure. In this method, no data is needed to be stored as guidance/requirement for the reproduction of original data. This means that this method can be repeated to the resultant of the previous procedure done using this method and so on. For multiple processes, the number of times this method have been applied has to be stored (and obviously, the final data set in reduced form) to get the original data set. This method first converts a given data set into another data set which is termed "special" (how is it special?). This special data set can be reduced considerably to another data set of lesser content than the original and so on. The main idea here is to convert to that special form. The note has to be taken here that the reduction is reasonable which fits one practical criterion (that is a reasonable reduction). A particular pattern in these special numbers will be shown.

Keywords— *Space, Special, Errors, Skipping, Extenders*

1. INTRODUCTION

Reduction of data has always been not easy. A reduction method should be such that it can be applied to all possibilities of data. For some possibilities, it's easily possible to reduce because these data sets have some peculiar property associated with them (only ones in it, only zeros in it, alternating ones and zeros, etc.).Note that ones and zeros are used to represent information here as information is ones and zeros. But a procedure is needed which should work on all possibilities. One way is to accept the fact that all possibilities are special and build algorithms which search for that special property. I think that special property can mean patterns ("can"). When we find pattern/s in the data set, we can form a table which contains information about the pattern and note how many times the pattern has been repeated and duplicate the whole dataset into a smaller one. But considering all possibilities, a single loop of the pattern can be the whole data set (meaning there won't be any easily recognizable patterns). So that means there will be no considerable reduction, as table storage will be as big as the data set (or as small as the data set depending on the data set). We can also search for mini patterns (patterns surrounded by random data). We can also search for the frequency of a particular data set in the given data set and map the smallest information possible to the most frequently occurred data set and so on. But in all of these techniques, we have to store some information which is needed by the reproducing algorithms. Patterns in a data set are special. But that pattern won't be the same for all possibilities. Each possibility shows a different pattern (the words explain themselves here). So, there is no specialty in the collection of specials (collection of patterns for all possibilities). Now, how about the following statement:

We convert all the possibilities of data to a special form (each possibility has a different special form otherwise those possibilities cannot be differentiated later) and the special thing about these special forms (plural because of multiple possibilities) is that they are all special "in the same way".

To understand this, refer the following sentences:
This principle is similar to taking all the citizens of a kingdom and giving them a new form of common purpose (soldiers). And one feature of these soldiers will be--that they can go back to their original life with/without serving their purpose.

Back to data reduction, that would eliminate the need of having any table to store data which is needed by the reproducing algorithm because we already know how it is special. So how it is special can be directly written in the reproducing algorithm beforehand.

1.1 Method

Now we just have to create a method which would fit the above criteria and just let it do its magic.

But first, let's read a bit about numbers. To differentiate the possibilities (what possibilities?), we use symbols and arrange them in different orders to map the possibilities (each symbol/ordered symbols maps to one possibility). Let's say that we only know two symbols '1' and '0'.Assuming no reason to explain why (probably history or maybe of the value each symbol represents and arranged in an increasing order),'0' comes before '1'. So '0' maps to one possibility and '1' maps to another possibility (Note that more symbols are used in real life which is the basis of number systems). To map the remaining possibilities, symbols are used collectively and each collective symbol maps to a possibility (Also, the order can be changed to map more possibilities). But one catch is that space (no of collective symbols) will be increased (To represent each possibility now, space required will be more than the previous possibilities).'00','01','10','11' map to the possibilities and so on ('000',....). But in binary number system (meaning number system having only two symbols),'00'and '01' are not used to map the possibilities (Why? not sure. probably because '00' and '01' are considered to be the same as'0' and '1'(Why are they considered same? Value??)). So, only '10' and '11' are used in mapping. The same thing applies when space is increased again ('100','101','110','111'). This means that except for space=1, a collection of symbols should start with '1'.

What is that possibility which was mentioned numerous times in the last paragraph? Values. Values are our way to quantitatively define the world around us. Quantitative measurement is required for exact measurement. That's what makes it exact! The symbols above are used to represent these values. These values can be on anything we see or anything we can't see in nature. Let's assume we are talking about tennis ball/s in a box. How do we know how many balls are present in the box? The answer to how many should not be "Pretty much" or "This much" with hand signals. So, to represent the number of tennis balls in the box we use these symbols as substitutes. It's like as if these symbols are substitutes we use to represent anything which we see visually (writing '50' tennis balls is way better than writing 50 circles and coloring them green). So, symbols are used for the quantitative measurement of an object and eliminate the possibility of a mistake (We might forget how many items we saw).

To sum up, the point of this reading above is to get a basic idea of numbers and mapping of symbols to values**.** Each symbol/collection of symbols map a single value. So, right now there are 2 columns. One for symbol, one for value. The proposed method is as follows:

Symbol maps value, as usual, value maps another symbol which maps its corresponding value as per column 1 mapping of column 2. Now, there are 4 columns (symbol, value, symbol, value). The 3rd column (value to symbol) symbol/s is the ones which are termed special. This is one instance where the Special Principle can be applied.

Now let's talk about how the thought process worked to get these special numbers. The objective was a reduction of data. The idea is to take the symbols in any data set 2 at a time (from left to right) and the following rules are applied—replace '00' by '00'

Replace '01' by '01'
Replace '10' by '1'.

This would account for reduction as '10' is replaced by '1' (space is reduced by 1). But there is no mapping given above when '11' occurs.'11' cannot be mapped to '11' as the reproduction algorithm will get confused as to what "11" represents,'11' or '1010'.To prevent this problem from occurring, mapping from the 2nd column to the 3rd column is critical. The mapping will be such that when a row in a 3rd column is taken 2 at a time from left to right,'11' will not occur. This doesn't mean that 2 consecutive 1's must not be present. When taken 2 at a time from left to right,'11' will not occur.'100110' is valid because when taken 2 at a time from left to right, we would not get '11' even though there are consecutive 1's. Those consecutive 1's will be split thus making it valid. So, particular values are termed special when their symbolic equivalent has no '11' when taken 2 at a time taken from left to right.

How the Special principle is used here is left to be understood by the readers (as it's obvious!).
So, the steps are as follows:

First, recognize the symbols. Then, map them accordingly into the 3rd column and apply the above idea (replacement). See the below tables for a better understanding. Note that for value column, decimal equivalent symbols are used to represent value.

Table 1: Conversion Table

Symbols	Replaced with
00	00
01	01
10	1

Table 2: Transition

Symbol	Value	Symbol after change	Value after change
0	0	0	0
1	1	1	1
10	2	10	2
11	3	100	4
100	4	101	5
101	5	1000	8
110	6	1001	9
111	7	1010	10
1000	8	10000	16
1001	9	10001	17
1010	10	10010	18
1011	11	10011	19

1100	12	10100	20
1101	13	10101	21
1110	14	100000	32
1111	15	100001	33
10000	16	100010	34
10001	17	100100	36
10010	18	100101	37
10011	19	100110	38
10100	20	101000	40
10101	21	101001	41
10110	22	101010	42
10111	23	1000000	64
11000	24	1000001	65
11001	25	1000010	66
11010	26	1000011	67
11011	27	1000100	68
11100	28	1000101	69
11101	29	1001000	72
11110	30	1001001	73
11111	31	1001010	74

For the reproducing algorithm, just reverse table 1 and apply it.

Table 3: Conversion to the original information

Symbols	Replaced with
00	00
01	01
1	10

One more advantage here is that we can choose other symbols too to be eliminated. Here, ”11” is eliminated in the symbols which are done for the sole purpose of explaining. Other 3 symbols (“00”, ”01”, ”10”) can also be eliminated to get a new series of numbers (Then “11” should be included). This can be done according to our preference; meaning whichever fits better (depends on how your data set is).

Also, for data sets whose space is odd, the last symbol will have no value to be paired with. For this, a simple rule can be formed here as it is as follows:

Replace ‘0’ with ‘0’
replace ‘1’ with ‘11’ and when ‘11’ occurs as the last pair,’10’ is mapped with it instead of ‘1’.

But, this method has limitations. In practical approaches, following the table above will be slightly not favorable (being kind on myself) because of space increases (from column 1 to column 3). Even though reduction is possible, knowing that it will increase first before decreasing is slightly not likable. Also, column 1 to column 3 mapping is not easy as time becomes an issue (In the application field, we deal with Gigabytes).

Then why was the above table explained? To get a grip on what’s being explained. For practical applications, the above table must be tweaked. Many other easier ways can be found out because technology advancement works better under a group of people than under a single person.

1.2 Improvement 1

Note that the special numbers are also present in column 1. So assume that in practical applications, these special numbers will not occur as the original data set (meaning as space increases in original data set, tell me that we won’t get “11” in application fields!!That we deal with gigabytes of data with no occurrence of ‘11’ when taken from left to right!!!). Let's call those data sets which have “11” when taken 2 at a time from left to right as errors (since they cause errors when table 1 is applied). Idea is to map those errors to the special series (starting from 0!). How the errors are mapped will be according to your preference (most probable errors will be given special numbers of lowest space/errors in increasing order). Observe the table below. I’ll assume the space of original data set as 6 and map the errors to the special numbers.

Table 4: Transition

Error datasets	Value	After mapping	Value
100011	35	0	0
100111	39	1	1
101011	43	10	2
101100	44	100	4
101101	45	101	5

101110	46	1000	8
101111	47	1001	9
110000	48	1010	10
110001	49	10000	16
110010	50	10001	17
110011	51	10010	18
110100	52	10011	19
110101	53	10100	20
110110	54	10101	21
110111	55	100000	32
111000	56	100001	33
111001	57	100010	34
111010	58	100100	36
111011	59	100101	37
111100	60	100110	38
111101	61	101000	40
111110	62	101001	41
111111	63	101010	42

The advantages of this table above are that if we luck out and find our error data set early in the table, the mapping will be of lesser space dataset plus that data set can be reduced because it’s a special number. But we have to store the space of the original data set (Unless we can mix it with the reduced data set and reduce further with some fixed rules formed beforehand so that space can be extracted later by the reproducing algorithm without any additional data for guiding it. Try your best to not store any kind of data to be used as guidance by the reproducing algorithm). The above table is when the space of the original data set is 6. But practically, it can go to Gigabytes. Meaning we can give assurance that that original dataset will be of error type (Otherwise, we can make it a rule that “11” will be added to the end every time). Thus making mapping much easier than the previous table (where special numbers are mapped to special numbers). Let's check out another table when space=7.

Table 5: Transition for space=7

Error datasets	Value	After mapping	Value
1000110	70	0	0
1000111	71	1	1
1001110	78	10	2
1001111	79	100	4
1010110	86	101	5
1010111	87	1000	8
1011000	88	1001	9
1011001	89	1010	10
1011010	90	10000	16
1011011	91	10001	17
1011100	92	10010	18
1011101	93	10011	19
1011110	94	10100	20
1011111	95	10101	21
1100000	96	100000	32
1100001	97	100001	33
1100010	98	100010	34
1100011	99	100100	36
1100100	100	100101	37
1100101	101	100110	38
1100110	102	101000	40
1100111	103	101001	41
1101000	104	101010	42
1101001	105	1000000	64
1101010	106	1000001	65
1101011	107	1000010	66
1101100	108	1000011	67
1101101	109	1000100	68
1101110	110	1000101	69
1101111	111	1001000	72
1110000	112	1001001	73
1110001	113	1001010	74
1110010	114	1001011	75
1110011	115	1001100	76
1110100	116	1001101	77

1110101	117	1010000	80
1110110	118	1010001	81
1110111	119	1010010	82
1111000	120	1010011	83
1111001	121	1010100	84
1111010	122	1010101	85
1111011	123	10000000	128
1111100	124	10000001	129
1111101	125	10000010	130
1111110	126	10000100	132
1111111	127	10000101	133

In the end, instead of ‘10000000’ which reduces by only 1, we can have other special numbers which would reduce more (‘10101010’). Again, this method can be tweaked in many ways to suit the user appropriately. One solution for that space increase problem (from column 1 to column 3) is this-- When the space of data set (in column 3) exceeds the space of original data set, a new algorithm kicks in which produces special numbers (of space=space of original number+1) of decreasing capability of reduction (unlike before which was producing in an increasing order of magnitude). So, it’s better if the original data set occurs at the starting or the ending of the table.

As stated before, better ways of optimization to suit practical criteria can be found out.

Practically, the mapping from column 1 to column 3 is most problematic. Traversing through the table is not easily favorable. But it’s the bold letters from page 1 that guides us in a proper direction.

1.3 Improvement 2

One way where we can seriously solve the space problem is by including all possibilities in column 3. For example, only “10” is included for space=2.”00” and “01” are not included. If we include them, the space problem can be solved to an excellent extent. The ratio will increase.

Table 8: Calculation of ratio for Improvement 2

Space	Number of Special numbers	Number of values	Ratio
1	2	2	1
2	3	2	1.5
3	6	4	1.5
4	9	8	1.125
5	18	16	1.125
6	27	32	0.84375

Improvement in ratio can be achieved by making more assumptions which are obvious (Otherwise, it would not be perfect). Clearly, there are more and better assumptions than ‘11’ will always occur in the original data set.

2. CHARACTERISTICS

The graph can be drawn for the ratio of a number of special numbers to the number of possible numbers for a given space. But this portion is just a show-off of basic mathematical skills. So, the tables are enough. Consider the below table.

Table 6: Calculation of ratio

Space	Number of special numbers	Number of Values	Ratio
1	2	2	1
2	1	2	0.5
3	2	4	0.5
4	3	8	0.375
5	6	16	0.375
6	9	32	0.28125
7	18	64	0.28125
8	27	128	0.2109375
9	54	256	0.2109375

To explain this, for space=3 only 4 and 5 are special numbers (‘100’,’101’) while the number of values is 4(“100”, 101”, ”110”, ”111”). A recognizable pattern for the number of special numbers is the following:
*2, *1.5, *2, *1.5, *2…

As we can see, the ratio decreases as space increases. This is the reason for that “space problem” from column 1 to column 3.Lower numbers look up to higher numbers to map themselves and this lower number feasting of higher numbers increase as space increases denoted by the ratio. For practical purposes, we have to keep that increase as low as possible. An expression which follows will relate the ratio to space.

Let n be space.

When n is odd: ratio = 2 * 3 ^ ((n – 3) / 2) / 2 ^ (n – 1)

When n is even: ratio = 3 ^ ((n – 2) / 2) / 2 ^ (n – 1)

In general, ratio = (3 ^ ((n – 2) / 2) / 2 ^ (n – 1)) * (2 / (3) ^ 0.5) ^ (n % 2)

Where % indicates remainder operation.

The ratio decrease can also be explained in this way. Due to alternate multiplication by 1.5 instead of 2 (unlike denominator which multiplies by 2 every time), ratio decreases.

Also, values are taken assuming that it always starts with 1(Binary number system).

The following table shows special numbers until space =8.

Table 7: Acknowledgment of special numbers

Space	Special numbers	Extenders
1	1	-
2	2	-
3	4,5	-
4	8,9,10	-
5	16,17,18,19,20,21	19
6	32,33,34,36,37,38,40,41,42	38
7	64,65,66,67,68,69,72,73,74,75,76,77,80,81,82,83,84,85	67,75,76,77,83
8	128,129,130,132,133,134,136,137,138,144,145,146,148,149,150,152,153, 154,160,161,162,164,165,166,168,169,170	134,150,152,153,154,166

There is a pattern to realize these numbers too (Special numbers). The first digit will always be 2^ (n-1) where n is space. Now, see that till 5(space) there are no skips in number flow (arithmetic progression with d=1). Skipping starts from 6. Skipping uses the below series.

1, 3, 6,11,22,43...
(*2,*2-1*2,*2-1 ...)(Except the first though)

For a particular space, the pattern of (space-2) is repeated 3 times with the appropriate skipping number in between.

For example, when space=6, the pattern of space=4 is repeated 3 times with the first skipping number 1 in between. For space=7, the pattern of space=5 is repeated 3 times with the second skipping number 3 in between. And so on.

FOR SPACE=6
32, 33, 34
36, 37, 38
40, 41, 42

FOR SPACE=7
64, 65, 66,67,68,69
72, 73, 74,75,76,77
80, 81, 82,83,84,85

Extenders are those numbers which contain consecutive 1's but still a special number. Since they "extend" the range of special numbers, extenders seem appropriate.

3. CONCLUSION

Many other better ways (optimum) ways can be found out to solve the column 1-column 3 transition problem. The paper here gives some insight into the thought process of the author. The main principle is the bold letters on page 1 which guide us in a proper direction to think. In other words, the direction of the vector (which compresses data effectively) is pointed by this paper. The magnitude can be increased by a group of people instead of a single entity. Let's just say that if the initial magnitude increase was substantial, the author would be standing outside a patent office first instead of writing a paper on it.

Also, in my opinion figuring out the basic laws of nature should not be overlooked by the younger generations. To understand this, refer the following sentences:

In war, there are a set of weapons available. We can both take the presented weapons and improve on it or we can spend some time thinking to create new weapons.

4. REFERENCES

[1] Micheal Spivak - Calculus
[2] https://en.wikipedia.org/wiki/Lossless_compression