Efficient Storage Reduction of Frequency of Items in Vertical Data Layout

A.Meenakshi ,
Dept of MCA, Computer Center,
Madurai Kamaraj University,
Madurai, India.
aa_meena@yahoo.com

Dr.K.Alagarsamy,
Associate Professor,
Dept of MCA, Computer Center,
Madurai Kamaraj University,
Madurai, India.
alagarsamymku@gmail.com

Abstract— The digital databases are immersed with large amount of data. The explosive growth of massive amounts of data leads to space complexity, performance degradation, scalability and time complexity. We cannot stop the incoming data, but we can fix some limitations to reduce the massive collection of data. The crux of the matter is how to accommodate all these data in an efficient manner, minimize storage space and lossless data. There are many algorithms which have been developed so far. Even though there is a cut-throat competition to satisfy all these criteria, we have developed a novelty approach in our proposed work to reduce the frequencies of items in vertical data layout .In our work, we have concentrated on Run length encoding method in applying vertical layout and moreover, we have done experimental results in very large databases.

Keywords- Compression , vertical layout, vertical RLE, frequency, splitmatrix.

I. INTRODUCTION

For the past two decades ,database is swelling day by day ; there is a need to churn the data or to find the way of managing the massive collection of data. Many advances have been made in the field of compression . Moreover , the explosive proliferation of information and continuous growth of data applications are outgrowing any technological advances in storage devices and communication tools.Data compression offers an attractive approach to alleviate many of the problems associated with data profileration. Compression algorithm is tailored to utilize the enormous speed and memory size.When there is a large amount of data growth, there is a need for compression of data.Data compression often refers to as “Data Encoding” and data decompression is referred to as “Data decoding” .The essential figure of merit for data compression is the “Compression Ratio” or ratio of the size of a compressed file to the original uncompressed file.Compression is the process of reducing the size of a file[1].The aim of the data compression is to reduce redundancy stored or communicated data, thus increasing effective data density. Datapreprocessing aims at simplifying the relationships to be inferred by a model and also by reducing the number of input variables.

There are a number of data prepocessing techniques.

1) Data Cleaning (to remove noise and inconsistent data)

2) Data Integration (integration of multiple databases, data cubes or files)

3) Data Transformation (Transformation of data into forms suitable for mining)

4) Data reduction (A reduced representation of the dataset i.e. SmallerVolume. A particular form of data reduction is through data compression which has specific emphasis on the use of encoding technique.)

Data compression is a process that reduces the data size, by removing the excessive information.Compressing data in a database system is attractive for two reasons: data storage reduction and performance improvement [2]. There are mainly two types of compression methods such as i) lossless compression ii) lossy compression .In lossless compression , there is no loss of data during compression and decompression.It is mainly used in compressing text documents. Arithmetic coding , Substitutional compressors and Huffman coding are some of the examples for lossless compression. In lossy compression , there is a loss of data during compression.It is used in Video compression , Audio compression and Fractal compression .These are some of the examples for lossy compression.

II. DATASET ORGANIZATION

Dataset organizations can be processed horizontally or vertically. For several decades and especially with the pre-eminence of relational database systems, data is almost always formed into horizontal record structure and then processed vertically. In a horizontal enumerated data organization, each transaction contains only items positively associated with a customer purchase. In a horizontal layout, the database is organized as a set of rows, with each row representing a customer's transaction in terms of the items that are purchased in the transaction [5].There is an alternative approach to this data layout such as vertical layout. It consists of each item associated with a column of values representing the transaction in which it is present. It has smaller effective database size, compact storage of the database and better support of dynamic database.

A market-basket database is a two dimensional matrix where the rows represent individual customer purchase transaction and the columns represent the items on sale.

TID	LIST OF ITEMS
1	{MILK,SUGAR, WHEAT}
2	{MILK, ICECREAM,SUGAR,FLOUR}
3	{MILK,SUGAR}
4	{ICECREAM,MILK,WHEAT}
5	{ICECREAM,WHEAT,SUGAR}

FIG. 1 TRANSACTIONS DATABASE

TID	LIST OF ITEMS
1	M, S, W
2	M , I, S
3	M, S
4	I, M, W
5	I, W, S

HORIZONTAL LAYOUT

TID	LIST OF ITEMS			
	M	S	I	W
1	1	1	2	1
2	2	4	4	
3	3	5	5	
4	5			

VERTICAL LAYOUT

III . LITERARY REVIEW OF COMPRESSION TECHNIQUES

A. Basic compression algorithms

There are many compression techniques such as Run length Encoding (RLE), Huffman Coding, Arithmetic Coding, LZW(Lempel Ziv), Golomb Coding , LZ-77, Encoding technique, Differencing Coding, Dictionary Coding and Database Compression method using data mining.

B. Discussion of the compression algorithms

Huffman Coding (Huffman, 1951) is a data compression method proposed by David Huffman. It computes the occurrence frequency of each single character in an article to construct a Huffman tree.

Arithmetic Coding (Witten, Ned and Cleary, 1987) is applied to convert an input string into real number between 0 and 1.

LZW(Lempel Ziv) approach deals with dictionary of characters. It replaces strings of characters with single code . LZ77(Ziv and Lempel , 1977) , LZ78(Ziv and Lempel, 1978) and LZW(Welch ,1984) are some of the LZW Compression approaches.

Golomb coding is highly suitable for situations in which the occurrence of small values in the input stream is significantly more likely than large values.

Differencing Coding (Gibson ,1980) is often applied on multimedia data compression. It is coding the difference from one sample to the next rather than coding the actual sample value.

Dictionary coding selects n frequent vocabularies in an article and each vocabulary has a corresponding code .

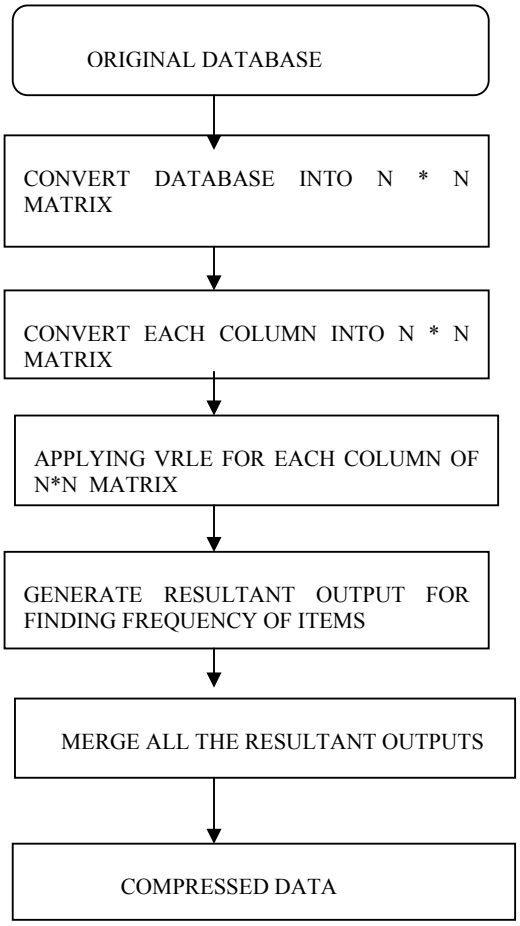
Database Compression method using data mining (Goh . et al 1998), a database compression method based on Differencing Coding (Ng and Ravishankar , 1997) and CIT (Changchein and Lu, 2001) , an association rule mining structure are explored. Some of the techniques are employed for database compression such as 1) Bit Compression (BIT) 2) Adaptive Text Substitution (ATS) 3) Tuple Differential Coding (TDC) .There is a promising technique for reducing the cost of data storage and transmission.

Run Length Encoding (RLE) is one of the simplest forms of data compression ; it is sometimes known as "Run Length Limiting "(RLL) [4].It is often used as a preprocessor for other compression algorithm. The large runs of consecutive identical data values are replaced by a simple code with the data value and length of the run. It is a very primitive way of compressing data.

If a data item d occurs n consecutive times in the input stream, it replaces the n occurrences with the single pair nd. The n consecutive occurrences of a data item are called as run length of n and this approach to data compression is called as Run length Encoding or RLE.

IV . PROPOSED WORK

Most of the RLE concepts were applied only for Image Compression. But in our proposed work , we are concentrating on RLE for compressing frequent occurrence of very large databases.RLE replaces sequences of the same data values within a file by a count number and a single value .It offers decent compression ratios in specific types of data. Our main objective is to implement RLE in Vertical layout. Vertical orientation offers better compression than horizontal one. Column lengths are proportional to size of database, whereas row lengths are proportional to size of schema. According to many authors' suggestions, this vertical layout outperforms significantly the horizontal layout. So we have taken into account the vertical layout by applying RLE. The following figure represents the steps of the proposed system.



A . STEPS FOR GENERATING VERTICAL RUNLENGTH ENCODING (VRLE)

1) Input the large databases into n *n matrix as a file.

2) Split the matrix by columnwise according to the vertical data layout.
3) Apply RLE for each column matrix
4) Generate output for each column
5) Combine all the outputs to give a final compressed output.

B.. ***FUNCTION FOR SPLITTING EACH MATRIX***

The input database (D) is converted into n*n matrix. Before applying VRLE, we have to split the database in a vertical manner. Each column is taken into account and split into m*m matrix. For splitting this column, we have to follow the procedure of vertical data layout and the following source code is applied.

```
void splitmat (int tmat,int t,int fname)
  {
   int r=0,ccnt=0;
   char ch,name[15];
  sprintf(name,"temp%d.txt",fname);
  fp=fopen("matrix.txt","r");
  fs=fopen(name,"w");
  if(fp==NULL)  {
   puts("File cannot open the file");
   exit(1);  }
  while((ch=fgetc(fp))!=EOF)  {
  while(ch!='\n'&&ch!=EOF)  {
   if(ccnt==tmat)  {
   if(r<t)  {
   fputc(ch,fs);
   while(((ch=fgetc(fp))!=EOF)&&(ch!='\n')&&(ch!=' '))  {
   fputc(ch,fs);  }
   r++;  }
   else  {
  fputc(ch,fs);
  while(((ch=fgetc(fp))!=EOF)&&(ch!='\n')&&(ch!=' '))  {
   fputc(ch,fs);  }
  fprintf(fs,"\n");
  r=0;  }
```

```
 if(r!=0)   {
fprintf(fs," ") ;    }
 }
 if(ch==' ')
ccnt++;
if(ch=='\n')
break;
ch=fgetc(fp);
}
ccnt=0;
}
 fclose(fp);
 fclose(fs);
}
```

C. *ALGORITHM FOR VRLE*

After splitting the database as matrix format , we can do the necessary steps for VRLE . For each column is converted into m*m matrix column format.Apply RLE for each column matrix.The main advantage of RLE is to remove the redundancy of items; it leads to efficient storage of space.

```
[Read the Database D and create the tid lists for each of the items
 S= VerticalPartitionDatabase(D)
 N= Number of partitions(columns)
 For I = 1 to N do begin
 Spiltmat =genmultmat(i)
 Sm' = genmultmat(sm_i ,  p)
 End
 Perform steps for VRLE
 For c = 0 to col
T=*(p+c)
Count =1
For r=1 to row
```

If(t= *(p+c+r*col) U

Count++

Else

T=*(p+c+r*col)

Count=1

Perform steps for combining all frequencies of items

L^g = Sm'

For i=2 to n do begin

Com^g=combine local large(l^g,sm')

For all col C com^g

Lg= l^g U sm' U{CƐcom^g}

End

V. EXPERIMENTAL RESULTS

VRLE is coded in C language and all experiments run on a pc of Intel ®Pentium ® Dual CPU T2410 @2.00 GHZ, with 2 GB Main Memory, Windows Vista Operating system. We have presented the following datasets which are publicly available at FIMI repository.[**http://fimi.cs.helsinki.fi/data**] ;Workshop on Frequent Item set Mining Implementations(FIMI'04).

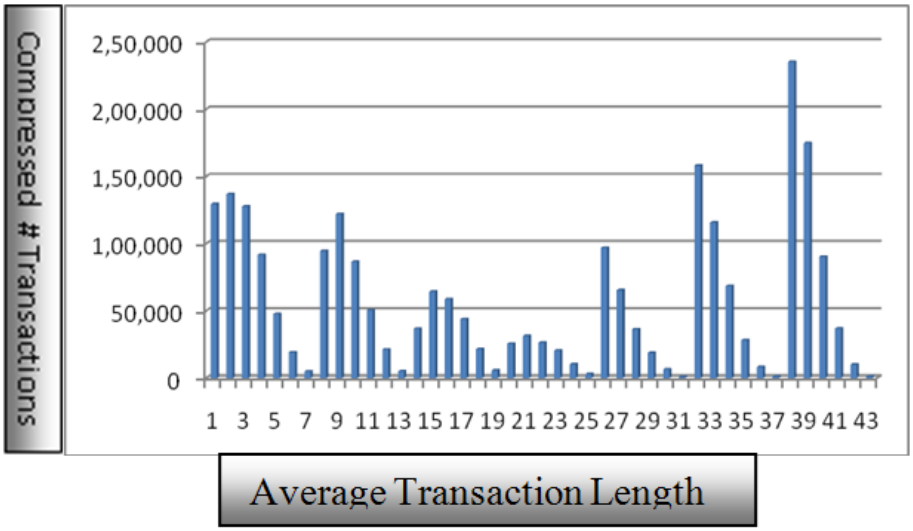
A Main features of datasets

DATASET	NO. OF ITEMS	AVERAGE TRANSACTION LENGTH	#TRANSACTIONS	SIZE (KB/MB)
CHESS	76	37	3196	336 K.B.
MUSHROOM	120	119	8124	581 K.B.
PUMSB	7117	74	49046	16.6M.B.
CONNECT	130	43	67557	9.21 M.B.

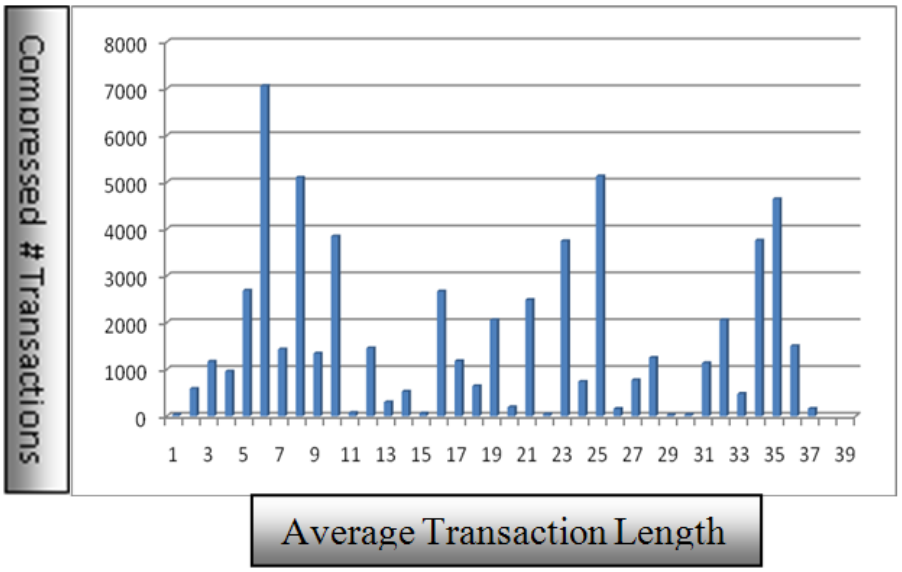
In order to show how VRLE performs when it is allowed to run to complete execution, we have chosen four large datasets for compression .The following figure shows the graphical representation of the compressed data sets. It has also resulted in the reduction of the memory space. The graph has been plotted with the results clearly about efficiently compressed datasets. The X-axis represents the average transactions of the dataset and Y-axis represents the # compressed transactions.

B. ***Results with graphical diagrams***

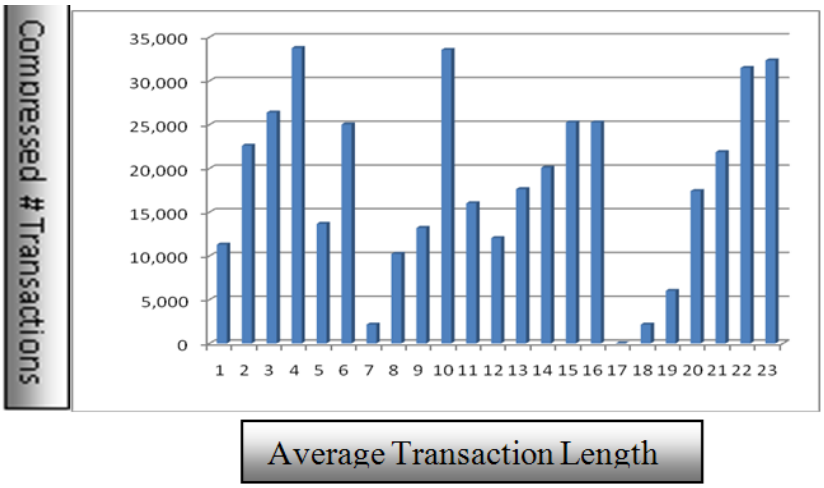
CONNECT DATABASE



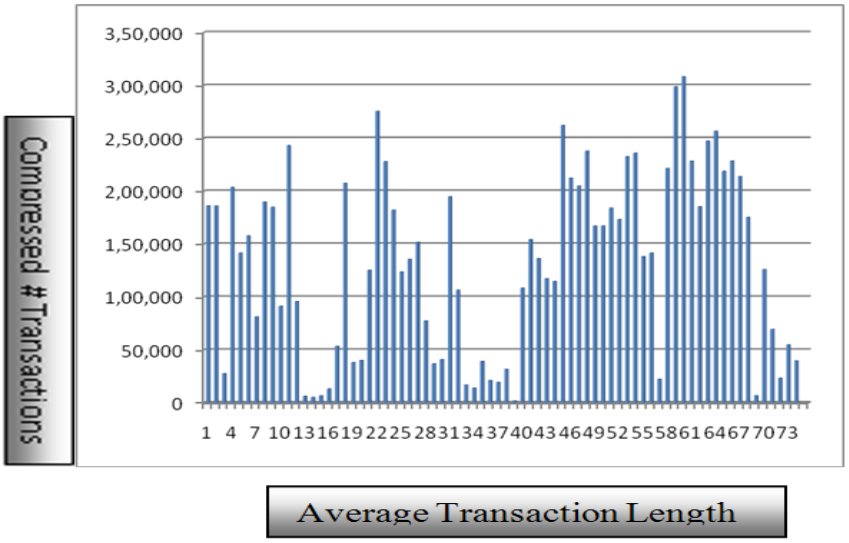
CHESS DATABASE



MUSHROOM DATABASE



PUMSB DATABASE



C. COMPRESSED DATASETS WITH SIZE OF ALL THE DATASETS

The following figure represents the compressed datasets with the memory usage of all the datasets. These datasets are very large database which contains repetition of data items .Our proposed work portrays the compression in an efficient manner.

DATA SET	SIZE (K.B. / M.B.)
CHESS	164 K.B.
MUSHROOM	460 K.B.
PUMSB	9.58 M.B.
CONNECT	2.41 M.B.

VI . CONCLUSION

In the modern digital world, there is a tremendous increase of large collection of data. All the data are noisy, missing and inconsistent due to their very large sizes reaching terabytes and petabytes and the trend towards further increase. In this paper, we have concentrated on compressing integer data of very large databases. We have implemented RLE in vertical data layout. The results of graphical diagrams of four datasets of very large databases such as chess, mushroom, pumsb and connect have been proved in an efficient way. In future, we can compare it with existing compression method and also implement it with other datasets also.

REFERENCES

[1] Chin-Feng Lee et. al. A Data Mining Approach to Database compression;Springer Scienc , 2006.

[2] Guy E.Blelloch, Introduction to Data Compression , sept 25, 2010.

[3] Storer, J.A. Data Compression: Methods and Theory; Computer Science Press: New York, NY, USA, 1988.

[4] El Gamal, A.; Orlitsky, A. Interactive data compression. In Proceedings of the 25th Annual Symposium on Foundations of Computer Science, Singer Island, FL, USA, October 24–26, 1984; pp. 100–108. Algorithms **2010**, 3 **75**

[5] Cilibrasi, R. Statistical Inference through Data Compression; ILLC Dissertation Series DS-2007-01; ILLC Publications: Amsterdam, The Netherlands, 2007.

[6] S.W.Golomb Runlength Encoding . IEEE Transaction on Information Theory . 12(3), July 1996.

[7] Pradeep Shenoy et al, Turbo-Charging Vertical Mining of large Databases , IISC, Technical report, DSL,2000

[8] T.Dasu and T.Johnson .Exploratory Data Mining and Data Cleaning , John Wiley and Sons, 2003.

[9] H.V.Jagadish et al. , Special Issue on Data Reduction Techniques.Bulletin of the Technical Committee on Data Engineering , 20(4) , December 1997.

[10] K.Wu , E.J. Otoo , A. Shoshani , and H.Nordberg , "Notes on design and implementation of compressed bit vectors" , Tech. rep. LBNL / PUB-3161 , Berkeley , CA.

[11] K.Wu , E.J. Otoo and A.Shoshani , "Optimizing bitmap indices with efficient compression " , ACM TODS , 2006.

[12] C.Lucchese , et. al. , "Fast and Memory Efficient Mining of Frequent Closed Itemsets " , TKDE , 2006.

[13] Mohammed Al-Laham and Ibrahiem M.M.EI Emary et. al. , Comparitive study between various algorithms of Data Compression techniques, Proceedings of the World Congress on Engineering and Computer Science 2007, WCECS 2007 , October 24 -26 , 2007, U.S.A.

[14] H.V.Jagadish et. al. , Special Issue on Data Reduction Techinques , Bulletin of the technical committee on Data Engineering , 20(4) December 1997.

AUTHOR'S PROFILE

Mrs.A.Meenakshi is currently working in the dept. of ComputerCentre, Madurai Kamaraj University , Madurai, India.She has secured M.C.A. and M.Phil.degree in Computer Science from Madurai Kamaraj University.She worked as a project trainee in ISRO for a period of six months.She published an International paper in IJCA journal and presented a paper in International Symbosium in P.S.G.K.S.R.,Coimbatore and another paper in National Conference in K.L.N. College, Madurai. She has put in 11 years of teaching experience and guided projects for M.C.A. students.

Dr. K.Alagarsamy is working as an associate professor in the department of computer centre, Madurai Kamaraj University (M.K.U.), Madurai , India.He secured his M.C.A., M.Phil. and Doctoral degree in Computer Science from M.K.University,Madurai .He has published 18 International Papers and participated in 4 International seminars.He has 28 years of teaching experience and published 2 books.His current research includes Data Mining and Software Engineering.