ISSN No. 0976-5697

Volume 3, No. 1, Jan-Feb 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Rule Base with Frequent Bit Pattern and Enhanced k-Medoid Algorithm for the Evaluation of Lossless Data Compression.

Nishad P.M.*
Ph.D Scholar, Department Of Computer Science
NGM College, Pollachi, India
nishadpalakka@yahoo.co.in

Dr. N. Nalayini
Associate professor, Department of computer science NGM College Pollachi, Coimbatore, India
sandeep_nalayini@hotmail.com

Abstract: This paper presents a study of various lossless compression algorithms; to test the performance and the ability of compression of each algorithm based on ten different parameters. For evaluation the compression ratios of each algorithm on different parameters are processed. To classify the algorithms based on the compression ratio, rule base is constructed to mine with frequent bit pattern to analyze the variations in various compression algorithms. Also, enhanced K- Medoid clustering is used to cluster the various data compression algorithms based on various parameters. The cluster falls dissentingly high to low after the enhancement. The framed rule base consists of 1,048,576 rules, which is used to evaluate the compression algorithm. Two hundred and eleven Compression algorithms are used for this study. The experimental result shows only few algorithm satisfies the range "**High**" for more number of parameters.

Keywords: Lossless compression, parameters, compression ratio, rule mining, frequent bit pattern, K–Medoid, clustering.

I. INTRODUCTION

Data compression is a method of encoding rules that allows substantial reduction in the total number of bits to store or transmit a file. Two basic classes of data compression are applied in different areas currently that are lossy and lossless compression [l, 2]. This paper evaluates the performance of all possible lossless compression algorithms using popular data mining technique. Rule mining is used with four ranges for ten parameters and to evaluate the performance the frequent bit pattern and clustered using the modified Sorted K- Medoid [3, 4] algorithm is used. Meteorological data mining with finding the hidden pattern is proposed in this paper [5][6][7].

The main objective of this paper is to evaluate the performance of various lossless data compression algorithms based on various ten parameters. To test the compression ability and survey two hundred and eleven and eleven algorithms are used. The parameters considered are various types of files such as Executable files (Exe), English text files (ENG), Log files (LOG), alphabetically sorted list files (ALP), DLL files (DLL), BMP files (BMP), JPEG files (JPG), help files (HLP) Document Files(DOC) and PDF files (PDF) is used and find the average compression ratio for all parameters with every compression algorithm. Initially 1,048,576 are constructed for evaluating the compression ability and performance of algorithms based on the range values and with sorted K-Medoid algorithm the rules are constructed based on the individual parameters.

The entire algorithms average compression ratio is collected based-on all ten parameter and the minimum and maximum compression ratio is collected to construct the new range for the individual parameter. The minimum represent the lowest compression ratio for the parameter and the maximum shows the peek compression ratio of algorithms on various parameters, for example 19.43 is the minimum compression ratio and the 76.84 is the maximum compression ratio for the parameter EXE shown in table-1

Table-1 Maximum and minimum compression ratio of parameters

Parameters	Minimum	Maximum
EXE	19.43	76.84
ENG	35.56	88.94
LOG	31.91	98.75
ALP	34.06	90.51
DLL	15.06	67.31
BMP	25.37	87.01
JPG	-41.18	24.37
HLP	34.48	90.29
DOC	15.07	88.42
PDF	-20.37	21.6

To evaluate the performance and generate the rule four ranges are used that is LOW, AVERAGE, MIDDLE and HIGH. The individual range value is calculated for each parameter by uniformly splitting the minimum and maximum compression ratio that is shown in the table -2. The LOW range for the EXE is 19.43 to 33.7825. After generating the range values the rules are generated. The rules are generated by matching Compression ratio of each algorithm with the range value if the compression ratio falls in any of the range. Then that range is set to construct the rule. If same rule is processed of many algorithms then the rule is treated as unique and only the algorithm is updated. For example in table-3 S no 7 the same rule is generated for the STUFFIT-14 and WINZIP -14 so the rule is uniquely created and the algorithm column alone is updated the process is repeated up to the end of the dataset. So for two hundred and eleven algorithms 46 rules are triaged shown in table-3 and table-4. From the rules the performance of algorithms can be easily evaluated.

© 2010, IJARCS All Rights Reserved

Table -2: Range values for the Parameters.

Range	EXE	ENG	LOG	ALP	DLL	BMP	JPG	HLP	DOC	PDF
LOW	19.43	35.56	31.91	34.06	15.06	25.37	-41.18	34.48	15.07	-20.37
	33.7825	48.905	48.62	48.1725	28.1225	40.78	-24.7925	48.4325	33.4075	-9.8775
AVERAGE	33.7826	48.906	48.63	48.1726	28.1226	40.79	-24.7924	48.4326	33.4076	-9.8774
	48.135	62.25	65.33	62.285	41.185	56.19	-8.405	62.385	51.745	0.614999
MIDDLE	48.136	62.26	65.34	62.286	41.186	56.2	-8.404	62.386	51.746	0.615
	62.4875	75.595	82.04	76.3975	54.2475	71.6	7.982499	76.3375	70.0825	11.1075
HIGH	62.4876	75.596	82.05	76.3976	54.2476	71.7	7.9825	76.3376	70.0826	11.1076
	76.84	88.94	98.75	90.51	67.31	87.01	24.37	90.29	88.42	21.6

Table -3 Rules for two hundred and eleven algorithms based on the four ranges

R NO	EXE	ENG	LOG	ALP	DLL	BMP	JPG	HLP	DOC	PDF
1	HIGH	HIGH	LOW	LOW	HIGH	LOW	LOW	LOW	LOW	HIGH
2	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	MIDDLE	HIGH	HIGH	HIGH
3	HIGH	HIGH	HIGH	HIGH	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
4	MIDDLE	HIGH	HIGH	HIGH	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
5	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
6	HIGH	HIGH	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
7	HIGH	HIGH	HIGH	HIGH	MIDDLE	HIGH	HIGH	HIGH	HIGH	HIGH
8	MIDDLE	HIGH	HIGH	HIGH	AVERAGE	HIGH	MIDDLE	HIGH	HIGH	HIGH
9	MIDDLE	HIGH	HIGH	HIGH	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	MIDDLE
10	MIDDLE	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
11	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	MIDDLE
12	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	AVERAGE	HIGH
13	MIDDLE	HIGH	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
14	MIDDLE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
15	MIDDLE	HIGH	HIGH	MIDDLE	AVERAGE	HIGH	MIDDLE	HIGH	HIGH	HIGH
16	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
17	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
18	MIDDLE	MIDDLE	HIGH	HIGH	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
19	MIDDLE	HIGH	HIGH	MIDDLE	MIDDLE	HIGH	AVERAGE	HIGH	HIGH	AVERAGE
20	AVERAGE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
21	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
22	AVERAGE	MIDDLE	HIGH	AVERAGE	LOW	MIDDLE	LOW	MIDDLE	HIGH	LOW
23	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH	MIDDLE
24	MIDDLE	MIDDLE	HIGH	HIGH	AVERAGE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
25	MIDDLE	MIDDLE	HIGH	AVERAGE	AVERAGE	MIDDLE	MIDDLE	HIGH	HIGH	HIGH
26	MIDDLE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	AVERAGE	HIGH	HIGH	MIDDLE
27	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	MIDDLE	AVERAGE	HIGH	HIGH	MIDDLE
28	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE
29	MIDDLE	AVERAGE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	HIGH	MIDDLE
30	MIDDLE	MIDDLE	HIGH	MIDDLE	AVERAGE	HIGH	MIDDLE	MIDDLE	HIGH	MIDDLE
31	MIDDLE	AVERAGE	HIGH	MIDDLE	MIDDLE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH
32	AVERAGE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH
33	MIDDLE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH
34	MIDDLE	AVERAGE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	HIGH	HIGH
35	AVERAGE	MIDDLE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	MIDDLE	MIDDLE
36	MIDDLE	AVERAGE	HIGH	MIDDLE	AVERAGE	MIDDLE	AVERAGE	MIDDLE	HIGH	MIDDLE
37	AVERAGE	LOW	HIGH	AVERAGE	LOW	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE
38	AVERAGE	AVERAGE	HIGH	MIDDLE	AVERAGE	MIDDLE	MIDDLE	MIDDLE	HIGH	MIDDLE
39	AVERAGE	LOW	HIGH	AVERAGE	LOW	LOW	MIDDLE	MIDDLE	HIGH	MIDDLE
40	AVERAGE	AVERAGE	HIGH	HIGH	LOW	HIGH	MIDDLE	MIDDLE	MIDDLE	AVERAGE
41	AVERAGE	AVERAGE	HIGH	MIDDLE	AVERAGE	MIDDLE	AVERAGE	MIDDLE	HIGH	MIDDLE
42	LOW	LOW	HIGH	LOW	LOW	LOW	MIDDLE	AVERAGE	HIGH	MIDDLE
43	AVERAGE	AVERAGE	HIGH	MIDDLE	LOW	MIDDLE	LOW	MIDDLE	MIDDLE	LOW
44	AVERAGE	LOW	MIDDLE	AVERAGE	LOW	AVERAGE	MIDDLE	AVERAGE	MIDDLE	MIDDLE
45	MIDDLE	HIGH	MIDDLE	MIDDLE	MIDDLE	HIGH	MIDDLE	HIGH	HIGH	HIGH
46	LOW	LOW	LOW	LOW	LOW	LOW	MIDDLE	LOW	LOW	MIDDLE

© 2010, IJARCS All Rights Reserved

Table: 4 Algorithms or Compressors for the Corresponding Rules

R NO	Algorithms Or Compressors
1	PAQ8PX
2	7-Zip 9.22 + ASH 07 + BIT 0.7 + CCM 1.30c + CMM4 0.2b + COMPRESSIA 1.0b + CTXf 0.75 b1 + DURILCA 0.5 + ENC 0.15 + EPM r9 + FreeARC 0.666 + HIPP 0.5819 + LPAQ8 + NanoZip 0.08a + PAQAR 4.5 + PIMPLE2 + PPMonstr J rev.1 + RK 1.04.1 + RKC 1.02 + RZM 0.07h + SL
3	777 0.04b1 + BALZ 1.15 + BEE 0.7.9 + BIX 1.00b7 + BruteCM 0.1d + CABARC 1.00.0106 + FlashZIP 0.99b8 + GRZIP 0.7.3 + LZPM 0.16 + LZPX(J) 1.2h + LZTurbo 0.95 + Ocamyd 1.66test1 + PIM 2.90 + PPMN 1.00b1 km + Quark 0.95r + RKUC 1.04 + UFA 0.04b1 + WinRAR 4.01
4	ARHANGEL 1.40 + BSC 2.7.0 + BSSC 0.95a + CTW 0.1 + GRZipII 0.2.4 + HOOK 1.4 + ICEOWS 4.20b + LZAP 0.20.0b + LZXQ 0.4 + M1 0.3b + PPMVC 1.2 + PPMX 0.07 + PPMY SSE (9A9) + PPMd rev J + QAZAR 0.0pre5 + QC 0.050 + QUANTUM 0.97 + Quad 1.12 + RINGS 1.6 + SZIP 1
5	12Ghosts 7.0 + ABC 2.4 + ACB 2.00c + AI 1.1 + ASD 0.2.0 + BA 1.01 + BBB ver1 + BCM 0.12 + BICOM 1.01 + BOA 0.58b + BWMonstr 0.02 + BWTZIP + BZIP 0.21 + BZIP2 1.0.5 + BioArc 1.9 + CHILE 0.5 + DACT 0.8.42 + DARK 0.51 + DC 0.99.307b + DCGA b8 + DGCA 1.10 + D
6	ACE 2.6 + BMA 1.35b + Blizzard 0.24b + CSC 3.2a6 + RKIVE 1.92 + SBC 0.970 rev3 + WINIMP 1.21 + WinACE 2.69
7	STUFFIT 14 + WINZIP 14
8	SR3a
9	NNTC
10	BZP 0.3
11	PSA 0.91a + ZAP32 0.15.0b
12	BAR 1.1.2
13	HuffComp 1.3
14	BJWFLATE 1.54 + Chaos Comp 3.0 + DeepFreezer 1.06 + Etincelle RC2 + LZOP 1.02rc1 + RAX 1.02
15	LZ2A
16	JAR 1.02
17	AIN 2.32 + ALZip 7.0 + AMG 2.2 + ARJ 2.85 + BCArchive 1.08.7 + DCA 1.0.1b + DZIP 2.90 + EAZEL 1.0 + ESP 1.92 + File2Pack 2.0 + GZIP 1.3.5 + HIT 2.10 + LHA 2.67 + LHA32 1.88.3.14 + LHARK 0.4d + LIMIT 1.2 + LZA 1.01 + PKZIP 2.50 + SEMONE 0.6 + SLUG X + THOR
18	CODEC 3.21 + HA 0.999b + KZIP 14-APR-2007 + LGHA 1.1g
19	BVI 1.70
20	Archiver 1.0 + ULZ 0.0.2
21	ARI 2.2 + RDMC 0.06c
22	LCW 0.2
23	JCALG1 5.32
24	SYMBRA 0.2
25	LCSSR 0.2
26	HiP beta 1
27	XPA 1.0.2 + aPLib 0.43
28	HAP 3.06
29	HYPER 2.5
30	LZC 0.08
31	SAR 1.0 + ZOO 2.1
32	QuickLZ 1.40b9
33	Zhuff 0.2
34	ARX 1.0 + Secura 1.7
35	CODER 1.1
36	CA-ZIP 3.4
37	LZ 1.0
38	QPress 0.38b
39	LZP2 0.7d
40	BigCrunch 0.4a1
41	BriefLZ 1.04
42	LZBW1 0.8
43	SRANK 1.0
44	LZRW1
45	ERI 5.1fre
46	SHcodec 1.0.1 + Shindlet

The HIGH range represents the good compression ratio and the LOW range represents the poor compression ratio so the algorithms compression ratio falls HIGH for any parameter then the performance of the algorithm is good for those parameters. For example PAQ8PX is quite good for the parameters EXE, ENG, DLL and PDF and it may not give the best result for the parameters LOG,ALP and BMP(table -3 S no -1)

To simplify the evaluation of algorithm another methodology is used in this paper called frequent bit pattern in this stage the generated rule dataset is processed for finding the frequent bit paten. Initially set the desired values for each parameter. For example EXE is 'HIGH' and ENG is 'HIGH' and LOG is 'HIGH' and ALP is 'HIGH' and DLL is 'HIGH' and BMP is 'HIGH' and JPG is 'HIGH' and HLP is 'HIGH' and DOC is 'HIGH' and PDF is 'HIGH' in this 'HIGH' is set to all the parameters.

So the frequent bit pattern evaluates what are the algorithms which satisfies the best of good compression ratio for individual or group of parameters. For example

© 2010, IJARCS All Rights Reserved

only based on the first parameter EXE five rules are triggered that is.

a. If EXE is 'HIGH' And ENG is 'HIGH' And LOG is 'LOW' And ALP is 'LOW' And DLL is 'HIGH' And BMP is 'LOW' And JPG is ' LOW' And HLP is 'LOW' And DOC is 'LOW' And PDF is 'HIGH' then PAQ8PX
b. If EXE is 'HIGH' And ENG is 'HIGH' And LOG is 'HIGH' And ALP is 'MIDDLE' And DLL is ' MIDDLE' And BMP is 'HIGH' And JPG is ' MIDDLE' And HLP is 'HIGH' And DOC is 'HIGH' And PDF is 'HIGH' then ACE 2.6 + BMA 1.35b + Blizzard 0.24b + CSC 3.2a6 + RKIVE 1.92 +SBC 0.970 rev3 + WINIMP 1.21 + WinACE 2.69
c. If EXE is 'HIGH' And ENG is 'HIGH' And LOG is 'HIGH' And ALP is 'HIGH' And DLL is ' MIDDLE' And BMP is 'HIGH' And JPG is ' MIDDLE' And HLP is 'HIGH' And DOC is 'HIGH' And PDF is 'HIGH' then .7.9 + BIX 1.00b7 + BruteCM 0.1d +CABARC 1.00.0106 + FlashZIP 0.99b8 + GRZIP 0.7.3 + LZPM 0.16 +LZPX(J) 1.2h + LZTurbo 0.95 + Ocamyd 1.66test1 + PIM 2.90 + PPMN1.00b1 km + Quark 0.95r + RKUC 1.04 + UFA 0.04b1 + WinRAR 4.01 +YZX 0.04
d. If EXE is 'HIGH' And ENG is 'HIGH' And LOG is 'HIGH' And ALP is 'HIGH' And DLL is ' MIDDLE' And BMP is 'HIGH' And JPG is 'HIGH' And HLP is 'HIGH' And DOC is 'HIGH' And PDF is 'HIGH' then STUFFIT 14 + WINZIP 14
e. If EXE is 'HIGH' And ENG is 'HIGH' And LOG is 'HIGH' And ALP is 'HIGH' And DLL is 'HIGH' And BMP is 'HIGH' And JPG is ' MIDDLE' And HLP is 'HIGH' And DOC is 'HIGH' And PDF is 'HIGH' then 7-Zip 9.22 + ASH 07 + BIT 0.7 + CCM 1.30c + CMM4 0.2b +COMPRESSIA 1.0b + CTXf 0.75 b1 + DURILCA 0.5 + ENC 0.15 + EPM r9+ FreeARC 0.666 + HIPP 0.5819 + LPAQ8 + NanoZip 0.08a + PAQAR4.5 + PIMPLE2 + PPMonstr J rev.1 + RK 1.04.1 + RKC 1.02 + RZM0.07h + SLIM 0.23d + SQUEEZ 5.63 + TC 5.2 dev2 + UHARC 0.6b + Ultra7z Opt 0.05 + WinRK 3.1.2 + ZPAQ 2.05

Frequent pattern based on group of parameters such as EXE, ENG, LOG, ALP, DLL, BMP, HLP, DOC,PDF for this only one rule is triggered

a. If EXEis 'HIGH'And ENGis 'HIGH'And LOGis 'HIGH'And ALPis 'HIGH'And DLL is'HIGH'And BMPis 'HIGH'And JPG is' MIDDLE'And HLPis 'HIGH'And DOCis 'HIGH'And PDFis 'HIGH' then 7-Zip 9.22 + ASH 07 + BIT 0.7 + CCM 1.30c + CMM4 0.2b +COMPRESSIA 1.0b + CTXf 0.75 b1 + DURILCA 0.5 + ENC 0.15 + EPM r9+ FreeARC 0.666 + HIPP 0.5819 + LPAQ8 + NanoZip 0.08a + PAQAR4.5 + PIMPLE2 + PPMonstr J rev.1 + RK 1.04.1 + RKC 1.02 +

Frequent pattern based on group of parameters such as EXE, ENG, LOG, ALP, DLL, BMP, JPG HLP, DOC, and PDF for this only one rule is triggered.

No algorithm satisfy High Compression ratio for all parameters so the rule triggered is zero. Here, using this evaluation methodology the effectiveness of various algorithms based on single or multiple parameters can be easily evaluated. For ten parameters 1024 combinations of frequent bit pattern can be evaluated. The table -5 shows the rulers trigged for 1024 combinations of frequent bit pattern. For example in the table 1 row 1 and column 1 gives 46 rules because the bit pattern is performed unconditionally in row 1 column 2 gives five that is based on the first parameter Exe= "HIGH". And in the last row and last column gives Zero because that is evaluated using the all parameters, so easily clarify that no algorithm terns best compression ratio for all parameters.

The modified K-Mediod Algorithm is used to evaluate and cluster the algorithms based on the parameters. After finding the mediod values that are sorted dissentingly. So this leads to the clarity that is the first cluster indicates that compression algorithm gives the best output. Generated rule is fetched and assigned value 1,2,3,4 to the ranges LOW, AVERAGE, MIDDLE, HIGH respectively so the numeric equivalent is passed to the range and the range data set is processed to the K-Medoid algorithm for each parameter three clusters are generated. That is shown in the table-6 the clusters generated in the ratio (Percentage) is shown in table -7.

Table -5 Frequent bit pattern number of rules triggered for the parameters EXE is HIGH and ENG is HIGH and LOG is HIGH and ALP is HIGH and DLL is HIGH and BMP is HIGH and JPG is HIGH and HLP is HIGH and DOC is HIGH and PDF is HIGH

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	46	5	16	5	42	4	14	4	10	3	6	3	10	3	6	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
2	18	4	13	4	17	4	12	4	8	3	6	3	8	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	25	4	14	4	24	4	13	4	9	3	6	3	9	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	14	4	12	4	13	4	11	4	7	3	6	3	7	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	38	4	14	4	37	4	13	4	9	3	6	3	9	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	16	4	12	4	15	4	11	4	7	3	6	3	7	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	25	4	14	4	24	4	13	4	9	3	6	3	9	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	14	4	12	4	13	4	11	4	7	3	6	3	7	3	6	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	25	5	13	5	23	4	11	4	8	3	5	3	8	3	5	3	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
18	12	4	10	4	11	4	9	4	6	3	5	3	6	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	19	4	11	4	18	4	10	4	8	3	5	3	8	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	11	4	9	4	10	4	8	4	6	3	5	3	6	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	23	4	11	4	22	4	10	4	8	3	5	3	8	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	11	4	9	4	10	4	8	4	6	3	5	3	6	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	19	4	11	4	18	4	10	4	8	3	5	3	8	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	11	4	9	4	10	4	8	4	6	3	5	3	6	3	5	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

© 2010, IJARCS All Rights Reserved

Table -6 Clustered rules based on K-Medoid Algorithm based on Parameters.

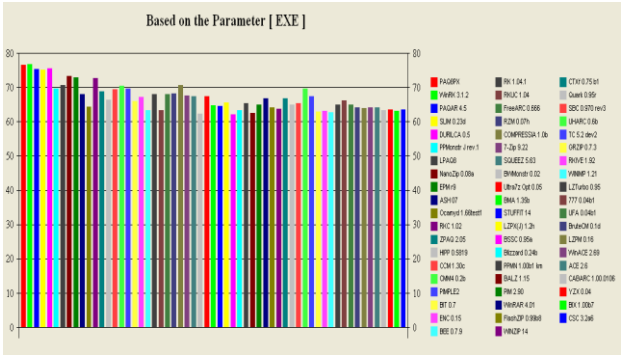
S No	Exe	Eng	LOG	ALP	DLL	BMP	JPG	HLP	DOC	PDF
1	1	1	3	3	1	3	3	2	3	1
2	1	1	1	1	1	1	2	1	1	1
3	1	1	1	1	2	1	2	1	1	1
4	2	1	1	1	2	1	2	1	1	1
5	2	1	1	2	2	1	2	1	1	1
6	1	1	1	2	2	1	2	1	1	1
7	1	1	1	1	2	1	1	1	1	1
8	2	1	1	1	3	1	2	1	1	1
9	2	1	1	1	2	1	2	1	1	2
10	2	2	1	1	2	2	2	1	1	1
11	2	1	1	2	2	1	2	1	1	2
12	2	1	1	2	2	1	2	2	3	1
13	2	1	1	2	3	2	2	1	1	1
14	2	2	1	2	3	2	2	1	1	1
15	2	1	1	2	3	1	2	1	1	1
16	2	1	1	2	2	2	2	1	1	1
17	2	2	1	2	2	2	2	1	1	1
18	2	2	1	1	2	1	2	1	1	1
19	2	1	1	2	2	1	3	1	1	3
20	3	2	1	2	3	2	2	1	1	1
21	2	2	1	2	2	1	2	1	1	1
22	3	2	1	3	3	2	3	2	1	3
23	2	2	1	2	2	2	2	1	1	2
24	2	2	1	1	3	2	2	1	1	1
25	2	2	1	3	3	2	2	1	1	1
26	2	2	1	3	3	2	3	1	1	2
27	2	2	1	3	2	2	3	1	1	2
28	2	2	1	3	2	1	2	2	1	2
29	2	3	1	3	3	2	2	2	1	2
30	2	2	1	3	3	1	2	2	1	2
31	2	3	1	3	2	2	2	2	1	1
32	3	2	1	3	3	2	2	2	1	1
33	2	2	1	3	3	2	2	2	1	1
34	2	3	1	3	3	2	2	2	1	1
35	3	2	1	3	3	2	2	2	2	2
36	2	3	1	3	3	2	3	2	1	2
37	3	3	1	3	3	2	3	2	2	2
38	3	3	1	2	3	2	2	2	1	2
39	3	3	1	3	3	3	2	2	1	2
40	3	3	1	1	3	1	2	2	2	1
41	3	3	1	2	3	2	3	2	1	2
42	3	3	1	3	3	3	2	3	1	2
43	3	3	1	3	3	2	3	2	2	3
44	3	3	2	3	3	3	2	3	2	2
45	2	1	2	2	2	1	2	1	1	1
46	3	3	3	3	3	3	2	3	3	2

Table -7 cluster percentage

	Clusters		
Parameter	**High**	**Average**	**Low**
ALP	21.73913	34.78261	43.47826
BMP	39.13043	50.0	10.86957
DLL	4.347826	41.30435	54.34783
DOC	82.6087	10.86957	6.521739
ENG	34.78261	36.95652	28.26087
EXE	10.86957	60.86957	28.26087
HLP	54.34783	39.13043	6.521739
JPG	2.173913	78.26087	19.56522
LOG	91.30435	4.347826	4.347826
PDF	56.52174	36.95652	6.521739

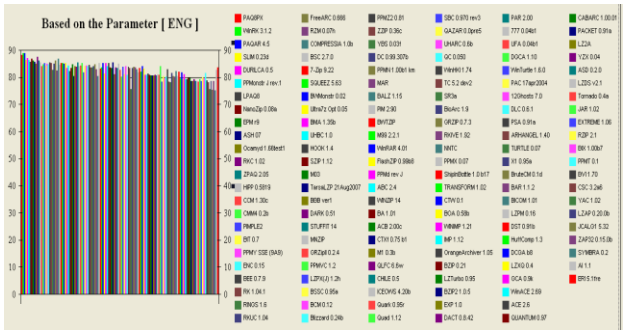
The clustered percentage ratio indicates that only the 21.73913 % algorithm gives best result for ALP, 39.13043% algorithm gives best result for BMP, 4.347826% algorithm gives best result for DLL, 82.6087% algorithm gives best result for DOC, 34.78261% algorithm gives best result for ENG, 10.86957% algorithm gives best result for EXE, 54.34783% algorithm gives best result for HLP, 2.173913% algorithm gives best result for JPG, 91.30435% algorithm gives best result for LOG and 56.52174% algorithm gives best result for PDF. Three clusters is created for each parameter so totally thirty cluster is created for ten parameters is shown in table -5

Graph -1 to graph -10 represents the compression ratio of the compression algorithms on various parameters which satisfies the Range 'HIGH'. Only few compression algorithms satisfy the 'HIGH' for more number of parameters. No algorithm falls in HIGH range for all parameters shown in table-4.

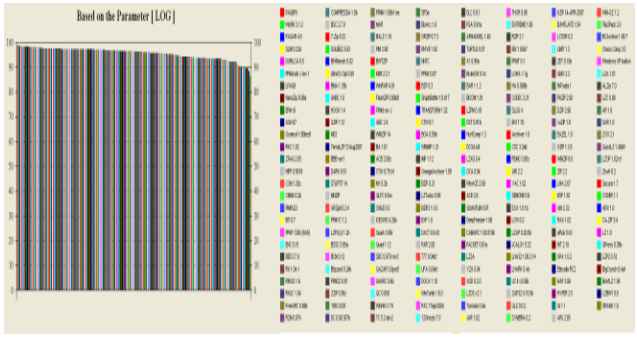


Graph 1 –Compression algorithms within the High Range for EXE

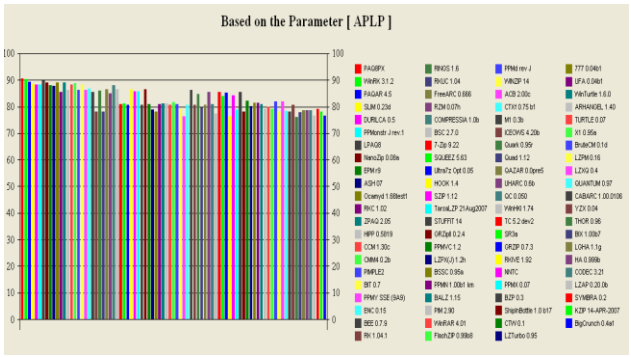
© 2010, IJARCS All Rights Reserved



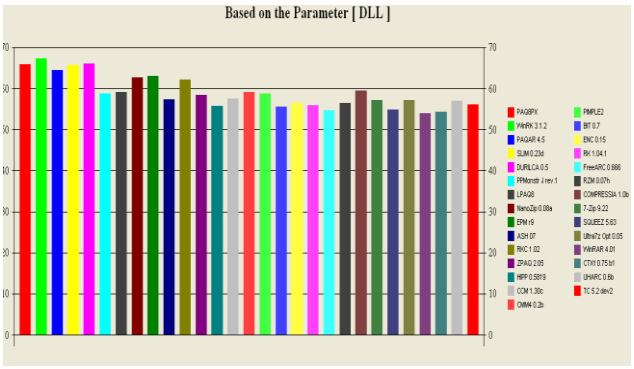
Graph 2 –Compression algorithms within the High Range for ENG



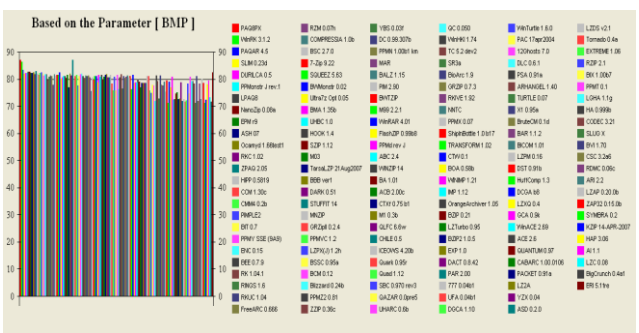
Graph 3 –Compression algorithms within the High Range for LOG



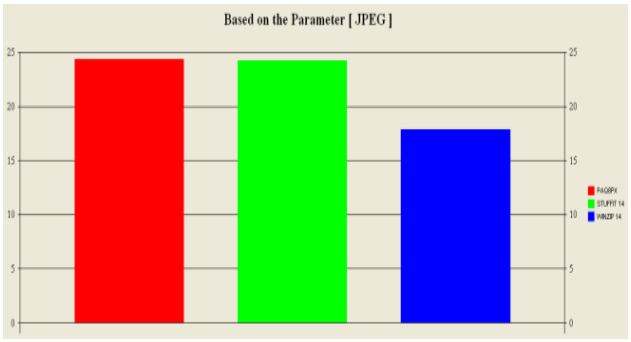
Graph 4 –Compression algorithms within the High Range for ALP



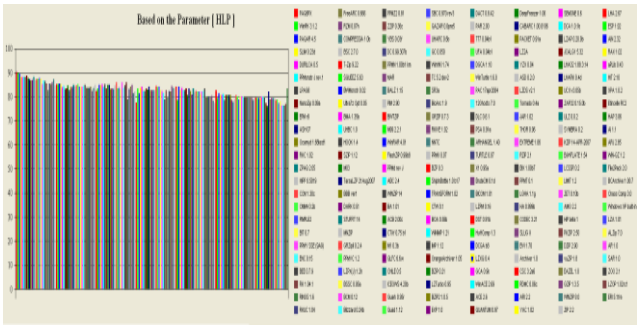
Graph 5 –Compression algorithms within the High Range for DLL



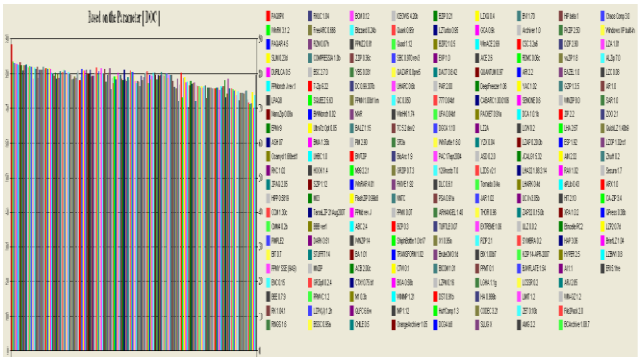
Graph 6 –Compression algorithms within the High Range for BMP



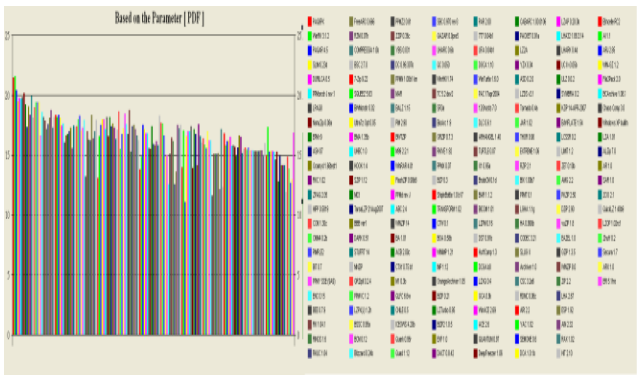
Graph 7 –Compression algorithms within the High Range for JPG



Graph 8 –Compression algorithms within the High Range for HLP



Graph 9 –Compression algorithms within the High Range for DOC



Graph 10 –Compression algorithms within the High Range for PDF

Graph -1 to graph -10 represents the compression ratio of the compression algorithms on various parameters which satisfies the Range 'HIGH'. Only few compression algorithms satisfy the 'HIGH' for more number of parameters. No algorithm falls in HIGH range for all parameters shown in table-4.

II. CONCLUSION AND FUTURE ENHANCEMENT

In this study, how efficiently and effectively rule mining can be applied to evaluate the performance of various data

© 2010, IJARCS All Rights Reserved

compression algorithm on different types of files is shown. The major contribution of this work is 1,048,576 number of rules are framed and the implemented using VB script. The most important focus of this paper is how pattern bit can be applied in the rule mining. The primary aim of reducing time in searching of rules in a large rule-base with 1,048,576 rules is achieved hundred percentage. Also K-Medoid clustering algorithm is applied to group the rules based on the performance of lossless compression algorithms. This clustering helps to reduces the time ratio of rules triggering. This work can be further extended using any other algorithm.

III. REFERENCES

[1]. T. C. Bell, J. G. Cleary, and I. H. Witten, "Text Compression English word" C1iffs:N. J. Prentice-Hall, 1990.

[2]. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing, Reading, Massachusetts": Addison-Welsley Publishing Company, 1992.

[3]. Velmurugan and T. Santhanam "A commiserative analysis between K-medoid methods and Fuzzy C-means clustering algorithms for statistically distributed data points" JTAIT 2011

[4]. Park. Hae-sang; Lee. Jong-seok; Jun. Chi-hyuck. "A K-means-like Algorithm for K medoids Clustering and Its Performance ". Proceedings of the 36th CIE Conference on Computers & Industrial Engineering.. Taipei. Taiwan. p.1222-1231 Jun. 20-23 (2006).

[5]. Sarah N. Kohail and Alaa M. El-Halees "Implementation of Data Mining Techniques for Meteorological Data Analysis" International Journal of Information and Communication Technology Research. 2010\

[6]. Bartok J., Habala O., Bednar P., Gazak M., and Hluch L., "Data mining and integration for predicting significant meteorological phenomena" Procedia Computer Science, p.37 – 46. 2010

[7]. Berkhin P., "Survey of clustering data mining techniques, Accrue Software, San Jose", CA, Tech. Rep., 2002

[8]. Artiles, J.; Gonzalo, J. and Sekine, S. WePS 2 Evaluation Campaign: "overview of the Web People Search Clustering Task". 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference. 2009.

[9]. Park. Hae-sang; Lee. Jong-seok; Jun. Chi-hyuck. "A K-means-like Algorithm for Kmedoids Clustering and Its Performance". Proceedings of the 36th CIE Conference on Computers & Industrial Engineering.. Taipei. Taiwan. p.1222-1231 Jun. 20-23 (2006)

Short Bio Data for the Author

*__Nishad PM__ M.Sc., M.Phil. Seven months Worked as a project trainee in Wipro in 2005, five years experience in teaching, one and half year in JNASC and Three and half year in MES Mampad College. He published one paper national level conference and three international conferences and also presented three national level seminars. Now pursuing Ph.D in Computer Science at NGM College Pollachi.

Dr. N. NALAYINI MCA., M.Phil., Ph.D is presently working as associate professor, Department of computer science NGM College, Pollachi, Coimbatore, India (affiliated to Bharathiar university, Coimbatore) She has published many papers in international/national Journals: her area of interest include Fuzzy logic, data mining, Data compression, Network security, she has to credit 21 years of teaching and research experience.

© 2010, IJARCS All Rights Reserved