EXACT SOUND COMPRESSION WITH OPTIMAL LINEAR PREDICTORS

Simon Fox and Peter Tischer

Department of Computer Science
Monash University

ABSTRACT

The optimal Least-Entropy predictor for DPCM exact sound compression is used to compare and evaluate the performance of predictors computed using such criteria as Least-Squares and Least-Absolute-Deviations. Results indicate that performance of Least-Squares predictors is sensitive to the data particularly as the dynamic range is increased. This has been addressed previously by breaking the data into small blocks and calculating new Least-Squares predictors for each block. We show that while this leads to considerable improvement, a single Least-Entropy predictor over the whole data set often performs favourably. An improvement for the Least-Squares predictor is suggested where a small percentage of the data is discarded in the least-squares modelling. This often gives a predictor considerably closer to optimal. The Least-Absolute-Deviations predictor is found to perform very close to optimal in general, suggesting that the distribution of errors is well approximated by a Laplacian distribution.

INTRODUCTION

In a typical sound waveform, it is expected that there is a significant degree of correlation between nearby waveform samples. If the waveform is encoded in Pulse Code Modulation (PCM) format as a stream of discrete waveform amplitudes, there will be some degree of redundancy in the encoding, and hence a longer message length for the data than is necessary.

Differential Pulse Code Modulation (DPCM) is a method which pre-processes the data to reduce correlation between nearby samples. To encode a particular sample, a prediction is made from previously encoded samples. The difference between this prediction and the actual sample value is encoded as an 'error' value. It is only necessary to transmit this error value, since the decoder can form the same predicted value using previously decoded samples. Some form of entropy coding such as arithmetic coding can then be used to encode the errors (Witten, Radford, Neal and Cleary, 1987).

This paper addresses the problem of how to form a prediction for a given sample. A linear predictor forms a prediction for the current sample as a linear combination of n previous samples. Such as predictor is said to be of order n. The coefficients of a simple linear predictor are fixed for all sound files. The simplest example is the Delta predictor, where the previous sample is the prediction for the current sample. Another example is the Mean predictor where a sample is predicted as the mean of the n previous samples.

Better results are achieved where a predictor's coefficients depend on the given sound file. Coefficients are chosen to minimise some criterion. A common criterion is Least-Squares (LSQ), where the squared sum of errors is minimised. Such a predictor is optimal where the distribution of prediction errors is Gaussian. However, we show that this is often not the case with sound data, and that for 16 bit data Least-Squares predictors are too sensitive to anomalies in the data which can cause large prediction errors. Also, there is no guarantee that increasing the order of a Least-Squares predictor will result in an improvement.

When DPCM is used with sound, often the prediction error distribution appears Laplacian. In this case, the optimal criterion to minimise is the sum of absolute error values. This is the Least-Absolute-Deviations (LAD) predictor. Such a predictor is far more difficult to compute than a Least-Squares predictor however. (Denoel and Solvay, 1985) proposed using a modified version of the Burg algorithm to compute the LAD predictor. In our work we use the Iteratively Re-weighted Least Squares (IRLS) algorithm (Ruzinsky and Olsen, 1989).

If we choose predictor coefficients to minimise the entropy of the prediction errors, the predictor is optimal, no matter what the prediction error distribution is. Since the entropy indicates the number of bits per sample required to encode the errors using entropy encoding, a Least-Entropy (LENT) predictor gives the shortest possible codelength for a given predictor order *n*. This is optimal from a compression point of view, and also in the sense that the predictor coefficients represent the best possible model for the data under the constraint of linear predictive coding and the assumption that prediction errors are independent and identically distributed. Furthermore, it is guaranteed that an increase in predictor order must equal or decrease the entropy of the prediction errors. The LENT predictors are also computed using the IRLS algorithm (Tischer 1993).

Since the LSQ criterion is far less expensive to compute, we attempt to use it to generate predictors closer to optimal. The method suggested in (Tischer 1993) is to discard the small percentage of data that leads to large errors in the LSQ modelling process (Censored Least Squares or CLS). In most cases, this produces a predictor much closer to optimal. Finally we observe the effects of breaking the data into blocks and calculating new LSQ predictors for each block.

LEAST SQUARES PREDICTORS AND BLOCKING

It is well known that LSQ regression can be overly sensitive to some small amount of data that does not fit the body of the data (Rousseeuw and Leroy, 1987). In the case of LSQ predictors, a very small percentage of noisy samples can badly effect performance. This is particularly true when the dynamic range of the data is increased. For sixteen bit data, two adjacent samples can differ by up to 65535. Such large differences are badly predicted with a linear predictor and cause large errors. This can lead to predictors that perform poorly since the weight of an observation depends on the square of the prediction error.

To counter this problem in the past, blocking has been used (Robinson, Garofolo and Fiscus, 1994). The data is split into small blocks usually of about 128-256 samples. For each block, a new Least-Squares predictor is computed and a block of prediction errors produced. We assume that the set of blocks of prediction errors are treated as a single set of prediction errors and are entropy encoded as such. Note that there is the overhead of transmitting new predictor coefficients for each block.

If better predictors are used, it is possible to increase the size of the blocks and thus reduce the overhead of transmitting coefficients. In the extreme example, a single LENT predictor over the entire data set will usually perform comparably to LSQ predictors using blocking. However, LSQ predictors are very easy to compute. If we can improve the LSQ predictor to be less sensitive to anomalies in the data, we can use larger blocks without losing performance, and reduce overhead.

A way to overcome the sensitivity of LSQ regression is a method called Censored Least Squares (CLS) (Tischer, 1993). A small percentage of the data that appears to cause large prediction errors is discarded during the LSQ modelling. A simple criteria for rejection of data is to discard samples beyond some arbitrary distance from the mean value. Any sample that is predicted using a previously rejected sample is also rejected. This way, the effect of outliers in the data is reduced. The results in section 5 were generated using a distance of 2.6 x the standard deviation of the data. In most cases, this lead to a rejection of 5-10% of the data. We attempt to show that using CLS predictors may allow us to increase blocksizes from 256 to 4096, without any significant loss in performance. Therefore, the cost of transmitting block predictor coefficients is reduced sixteen-fold.

TEST DATA

Three distinct classes of sound data are tested, namely speech, music and artificial sound effects. Speech data from the "NIST Speech Disc 1-1.1 October 1990" CD-ROM is used for testing. It is all sampled at 16,000 Hz. and is 16 bits per sample. File length varies from about 35,000 to 80,000 samples. A speech file has a male or female voice speaking a whole sentence.

To test music data, we sampled music from the following compact discs: Dire Straits- "Money For Nothing" (DS), Matt Finish- "Short Note" (MF), Nirvana- "Never Mind" (NV) and The Cure-

"Disintegration" (CU). The sampling was done at a variety of rates and in both 8 and 16 bit formats. File sizes are generally large, often greater than 500,000 samples.

Thirdly, some artificial sound data found on the Silicon Graphics Indy is used. These sounds include a range of car noises called "Screech" and "Crash". These sound effects are of interest because we suspect they are badly modelled by a linear predictor.

RESULTS OF SINGLE PREDICTORS

The following tables show the results of tests on the data described in section 3. The predictors tested are LSQ, LAD and LENT. Each predictor is tested with order 3,6 and 9. Included for each file tested is its original entropy and the resulting entropy when the simple order 1 Delta predictor is used. For comparisons, some tables contain results in brackets from other tables.

LSQ Predictor on 16 bit data

FILENAME	FILE TYPE	ORIGINAL ENTROPY	DELTA ENTROPY	LSQ PREDICTORS		
				ORDER 3	ORDER 6	ORDER 9
MF	Music 22 kh	12.803	11.118	12.329	11.653	11.409
NV	Music 44 kh	14.646	12.988	12.366	13.108	12.356
DS	Music 22 kh	13.809	12.581	13.617	15.604	13.348
DS	Music 44 kh	13.851	11.826	11.649	13.415	12.634
sa1	Speech	9.548	8.099	7.943	7.873	7.822
si1467	Speech	10.280	9.291	9.115	9.070	9.070
sx325	Speech	9.448	8.014	7.878	7.894	7.924
Crash	Effect	10.090	9.991	13.253	12.714	13.284
Screech	Effect	13.320	10.430	8.845	10.830	9.594

TABLE 1. LSQ Predictors: Prediction Error Entropies for 16 bit data

These results indicate that the LSQ predictor badly models 16 bit music data. In many cases, increasing the order from 3 to 6 results in a significant loss in performance. In some cases even, better performance is achieved using the Delta predictor. For speech data, LSQ generally performs better. However, there are still cases where increasing predictor order reduces performance. For the sound effects data, the Least-Squares predictor fails badly possibly due to the occurrence of large prediction errors. Note the improvement in performance when the sampling rate of the file DS is increased from 22khz to 44khz.

LSQ Predictor on 8 bit data

FILENAME	FILE TYPE	ORIGINAL ENTROPY	DELTA ENTROPY	LSQ PREDICTORS		
				ORDER 3	ORDER 6	ORDER 9
MF	Music 22 kh	4.865	3.259	3.270	3.250	3.265
NV	Music 44 kh	6.740	5.050	4.298	4.066	4.032
DS	Music 22 kh	5.902	4.660	4.662	4.656	4.657
DS	Music 44 kh	5.902	3.880	3.634	3.568	3.569
CU	Music 8 kh	4.866	4.153	4.151	4.029	4.002

TABLE 2. LSQ Predictors: Prediction Error Entropies for 8 bit data

Again there are cases where the LSQ predictor fails to improve on the Delta predictor. However, in most cases a reasonable improvement is made. It is rarely beneficial to increase the predictor order beyond 6.

LAD Predictor on 16 bit data

FILENAME	FILETYPE	LAD PREDICTORS ORDER 3		ORDER 6		ORDER 9	
MF	Music 22 kh	10.919	(12.329)	10.709	(11.653)	10.732	(11.409)
NV	Music 44 kh	12.183	(12.366)	11.821	(13.108)	11.655	(12.356)
DS	Music 22 kh	12.520	(13.617)	12.503	(15.604)	12.504	(13.348)
DS	Music 44 kh	11.368	(11.649)	11.223	(13.415)	11.113	(12.634)
sa1	Speech	7.884	(7.943)	7.819	(7.873)	7.767	(7.822)
si1467	Speech	9.094	(9.115)	9.034	(9.070)	9.027	(9.070)
sx325	Speech	7.686	(7.878)	7.694	(7.894)	7.660	(7.924)
Crash	Effect	12.459	(13.253)	12.492	(12.714)	12.506	(13.284)
Screech	Effect	5.571	(8.845)	5.355	(10.830)	5.276	(9.594)

TABLE 3. LAD Predictors: Prediction Error Entropies for 16 bit data
(LSQ Entropies shown in brackets for comparison)

The LAD predictor significantly outperforms the LSQ predictor in all tests with 16 bit data. It displays robustness to noisy data (see Crash and Screech). In the music data where the LSQ predictor failed, LAD performs reliably. Increasing predictor order usually produces equal or better results. Even with speech data where the LSQ predictor performed well, the LAD predictor still makes some improvement.

LAD Predictor on 8 bit data

FILENAME	FILE TYPE	LAD PREDICTORS ORDER 3		ORDER 6		ORDER 9	
MF	Music 22 kh	3.160	(3.270)	3.153	(3.250)	3.151	(3.265)
NV	Music 44 kh	4.295	(4.298)	4.064	(4.066)	4.028	(4.032)
DS	Music 22 kh	4.601	(4.662)	4.587	(4.656)	4.585	(4.657)
DS	Music 44 kh	3.581	(3.634)	3.516	(3.568)	3.504	(3.569)
CU	Music 8 kh	4.151	(4.151)	4.024	(4.029)	3.996	(4.002)

TABLE 4. LAD Predictors: Prediction Error Entropies for 16 bit data
(LSQ Entropies shown in brackets for comparison)

Again, the LAD predictor always improves on the LSQ predictor for 8 bit data. However, in many cases the improvement is small.

LENT Predictor on 16 bit data

FILENAME	FILETYPE	LENT PREDICTORS ORDER 3		ORDER 6		ORDER 9	
MF	Music 22 kh	10.710	(10.919)	10.638	(10.709)	10.636	(10.732)
NV	Music 44 kh	12.164	(12.183)	11.804	(11.821)	11.649	(11.655)
DS	Music 22 kh	12.520	(12.520)	12.503	(12.503)	12.504	(12.504)
DS	Music 44 kh	11.360	(11.368)	11.223	(11.223)	11.078	(11.113)
sa1	Speech	7.881	(7.884)	7.816	(7.819)	7.767	(7.767)
si1467	Speech	9.094	(9.094)	9.034	(9.034)	9.027	(9.027)
sx325	Speech	7.686	(7.686)	7.658	(7.694)	7.660	(7.660)
Crash	Effect	9.991	(12.459)	9.991	(12.492)	9.991	(12.506)
Screech	Effect	5.571	(5.571)	5.355	(5.355)	5.276	(5.276)

TABLE 5. LENT Predictors: Prediction Error Entropies for 16 bit data
(LAD Entropies shown in brackets for comparison)

The above table shows a comparison between LAD and LENT predictors. For speech and sound effects data, these are virtually the same. Only for 16 bit music data does LENT show some improvement over LAD. In some cases the improvement is reasonable, but generally quite small.

LENT Predictor on 8 bit data

FILENAME	FILE TYPE	LENT PREDICTORS		
		ORDER 3	ORDER 6	ORDER 9
MF	Music 22 kh	3.155 (3.160)	3.138 (3.153)	3.138 (3.151)
NV	Music 44 kh	4.294 (4.295)	4.063 (4.063)	4.028 (4.028)
DS	Music 22 kh	4.601 (4.601)	4.586 (4.587)	4.585 (4.584)
DS	Music 44 kh	3.578 (3.581)	3.515 (3.516)	3.504 (3.504)
CU	Music 8 kh	4.151 (4.151)	4.024 (4.024)	3.995 (3.995)

TABLE 6. LENT Predictors: Prediction Error Entropies for 8 bit data
(LAD Entropies shown in brackets for comparison)

For 8 bit music data, the LENT and LAD predictors produce nearly equivalent results.

RESULTS OF BLOCKING AND CLS PREDICTORS

Single CLS Predictor Results on 16 bit data

FILENAME	FILE TYPE	CLS PREDICTORS		
		ORDER 3	ORDER 6	ORDER 9
NV	Music 44 kh	12.207 (12.366)	11.848 (13.108)	11.681 (12.356)
MF	Music 22 kh	11.113 (12.329)	10.984 (11.653)	10.935 (11.409)
DS	Music 22 kh	12.559 (13.617)	12.549 (15.604)	12.549 (13.348)
DS	Music 44 kh	11.514 (11.649)	11.356 (13.415)	11.230 (12.634)
Screech	Effect	5.572 (8.845)	5.358 (10.830)	5.281 (9.593)

TABLE 7. CLS Predictors: Prediction Error Entropies for 16 bit data
(LSQ Entropies shown in brackets for comparison)

For 16 bit music data and sound effects data, the CLS predictors show significant improvement over LSQ. In some cases above, the improvements are remarkable considering the simplicity of the censoring process. For the 16 bit speech data, we were unable to achieve improvements since the LSQ predictor performed very well on this data anyway.

5.2. LSQ vs. CLS Predictor Results Using Blocking

FILENAME	FILE TYPE	BLOCK SIZE = 256 (LSQ)	BLOCK SIZE = 4096 (LSQ)	BLOCK SIZE = 4096 (CLS)
MF	Music 22 kh	10.378	10.854	10.579
DS	Music 22 kh	12.398	13.327	12.417
DS	Music 44 kh	11.177	11.864	11.209
Screech	Effect	5.659	7.167	5.515

TABLE 8. Order 3 LSQ Predictors and CLS Predictors:
Prediction Error Entropies for 16 bit data (Using Blocksizes of 256 and 4096)

This table firstly shows how the LSQ predictor deteriorates as the blocksize increases from 256 to 4096. However, it then shows that for blocks of size 4096, CLS predictors can be used to produce results similar to LSQ predictors with blocksize 256. Furthermore, the reduction in overhead is significant. The following table shows the same thing, but with order 6 predictors.

FILENAME	FILE TYPE	BLOCK SIZE = 256 (LSQ)	BLOCK SIZE = 4096 (LSQ)	BLOCK SIZE = 4096 (CLS)
MF	Music 22 kh	10.152	10.829	10.398
DS	Music 22 kh	12.379	13.799	12.376
DS	Music 44 kh	10.923	12.057	10.963
Screech	Effect	5.358	6.890	5.251

TABLE 9. Order 6 LSQ Predictors and CLS Predictors:
Prediction Error Entropies for 16 bit data (Using Blocksizes of 256 and 4096)

CONCLUSIONS

The LSQ predictor is found to perform unreliably where large errors may occur, particularly with 16 bit data. A better, but more computationally expensive predictor is the LAD predictor. In nearly all cases, it is found to perform very closely to the optimal LENT predictor. This verifies our observations that prediction error distributions can be well approximated by a Laplacian distribution.

Since the LSQ predictor is easy to compute, it is worth attempting to improve it's performance. The CLS predictor is shown to give remarkable improvements in some cases, simply by discarding data that appears to cause large prediction errors. This predictor can be used with blocking to allow larger blocks to be used without losing performance, and thus reduce ovehead.

FURTHER WORK

In future research, we intend to study techniques for encoding the prediction errors. Rice encoding (Rice, 1991) is a simple coding technique that is appropriate for prediction errors with a Laplacian distribution. Better results should be achieved using arithmetic coding on individual blocks of prediction errors. It is first necessary to study ways of accurately and compactly describing a block's prediction error distribution, since for each block, this information must be transmitted. In particular, the grey-encoded bitplane method (Tischer, Worley, Maeder and Goodwin, 1993) for encoding a distribution will be used.

REFERENCES

Denoel E. and Solvay J. (1985) *Linear Prediction of Speech with a Least Absolute Error Criterion*, IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-33, No.6, December

Rice, Robert F. (1991) *Some Practical Noiseless Coding Techniques {Part 2, Module PSI14,K+}*, JPL Publication, Number 91-3, Jet Propulsion Laboratories, November

Robinson T, Garofolo J, and Fiscus J. (1994) *The Development of File Formats for Very Large Speech Corpora: SPEECH and Shorten*, ICASSP

Rousseeuw P.J and Leroy A.M. (1987) *Robust Regression and Outlier Detection*, (Wiley, New York)

Ruzinsky S.A. and Olsen E.T. (1989) *L_1 and L_∞ Minimization Via a Variant of Karmarkar's Algorithm*, IEEE Transactions on Acoustics, Speech, and Signal Processing. ASSP-37, No.2, February

Tischer P. Worley R. Maeder A. and Goodwin M. (1993) *Context-based Lossless Image Compression*, The Computer Journal, Vol. 36, No. 1

Tischer P. (1993) *Optimal Predictors for Image Compression*, Monash Technical Report Number 93/189, Under review for IEEE

Witten, Ian H. Radford, Neal M. and Cleary, John G. (1987), *Arithmetic Coding For Data Compression*, Communications of the ACM, June, Volume 30, Number 6.