A WEIGHTED LEAST SQUARES METHOD FOR ADAPTIVE PREDICTION IN LOSSLESS IMAGE COMPRESSION

Hua Ye, Guang Deng and John C. Devlin

Department of Electronic Engineering, La Trobe University
Bundoora, Victoria 3083, Australia

{h.ye, d.deng, j.devlin}@ee.latrobe.edu.au

ABSTRACT

This paper is concerned with the application of the least squares method in predictive lossless image compression. Based on the observation of the limitations of ordinary least squares (OLS) method that is employed in most published algorithms, a unique weighted least squares (WLS) method is proposed. This WLS method gives a weight to the contribution of an observed sample according to the similarities between the feature at this sample and that at the pixel to be predicted. Experimental results have confirmed the superiority of the proposed method over both the OLS based method and WLS methods using alternative weighting schemes. It has been demonstrated that when coupled with other components of a lossless image compression algorithm, the proposed method can achieve a compression performance better than any other algorithms known to the authors.

1. INTRODUCTION

In predictive lossless image coding, an accurate predictor is always desirable because it is the prediction error that will be entropy coded. The simplest and most widely used prediction method is linear prediction. However, for a linear predictor to be able to accurately model image data which are usually nonstationary, some kind of adaptation scheme must be developed to let the predictor change according to local characteristics. In the past few years, the least squares (LS) method has found applications in this kind of adaptation and has been shown to perform very well [1][2][3][4].

The least squares method employed by most image compression algorithms published so far is often referred to as ordinary least squares (OLS), which gives equal weight to the contribution of each observation. This method, however, may not be suitable for at least some areas of a natural image. It can be easily seen that if the pixel to be predicted is in a smooth area, a sample in an edge area will provide much less information than a sample in the same smooth area. One solution to this problem is the use of weighted least squares (WLS) that gives each observation weight according to the relative amount of information it provides. More recently, WLS has also been applied to lossless image compression. For example, the Glicbawls [5][6] proposed by Meyer and Tischer employed a WLS based predictor that leads to a noticeably better compression performance than those with a single OLS based predictor.

Obviously, the central issue in using WLS in lossless image compression is to develop an effective weighting scheme. In Glicbawls, the weight for an observed pixel mainly depends on how close the pixel is to the current pixel. In this paper, we propose a unique weighting scheme that gives a weight according to the similarities between the feature at the observed pixel and that at the current pixel. Our experimental results show that this weighting scheme outperforms several alternative schemes, including one which is similar to that used in Glicbawls. The effectiveness of the proposed WLS method is also confirmed by coupling it with other components of a lossless image compression system [7], which produces a compression performance that is better than any other algorithms known to the authors.

2. THE PROPOSED WLS METHOD

Suppose that the desired value (target) of the output of a predictor is y. With a vector of inputs $\mathbf{x} = [x_1, \ldots, x_J]^T$, the output of a linear predictor can be defined as

$$\widehat{y} = \boldsymbol{\beta}^T \mathbf{x}, \tag{1}$$

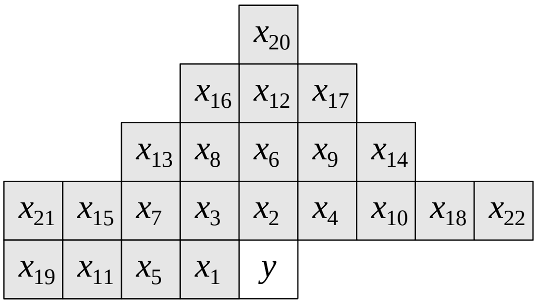

Fig. 1. Position relationship between the input pixels of adaptive linear predictors and the pixel to be predicted. For a Jth order predictor, $x_1, \ldots, x_J$ will be the input pixels to form a prediction of pixel y.

where $\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_J]^T$ are the predictor coefficients. In lossless image compression, the input vector for a pixel is often formed by pixels in its causal neighbourhood. Fig. 1 shows the way of selecting input pixels in this paper. Given N previous observations target $\{y(n-i)\}_{i=1}^{N}$ and the corresponding input vectors $\{\mathbf{x}(n-i)\}_{i=1}^{N}$ (collectively referred to as the training data block, or training window), the OLS method find the coefficients $\boldsymbol{\beta}_{\text{OLS}}$ that minimise the sum of square prediction errors

$$\text{SSE} = \sum_{i=1}^{N} [y(n-i) - \widehat{y}(n-i)]^2 . \tag{2}$$

The solution is given by

$$\boldsymbol{\beta}_{\text{OLS}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}, \tag{3}$$

where matrix

$$\mathbf{X} = [\mathbf{x}(n-1) \ \cdots \ \mathbf{x}(n-N)]^T$$

and column vector

$$\mathbf{y} = [y(n-1) \ \ldots \ y(n-N)]^T .$$

From the above description, we can see that the OLS solution (3) will be optimal for the training data block. However, our goal is to make a good predictor for the current pixel, not for the whole training window. Therefore, even when a set of coefficients fits the training window well, it is still possible that it is not good for our goal at all. When we apply the resultant model to a new data sample, which is outside the training window, we have to assume that this pixel has the same characteristics as those in the training window. An additional assumption is that all the data in the training window are roughly from a single model. This latter assumption is largely dependent on a proper choice of the training window. For statistically nonstationary image data, it is not unreasonable to expect these assumptions to be invalid. If either one of these assumptions is not true, the OLS based predictor will give poor prediction for the current pixel. For example, if the data in the training window belong to two very different models, OLS can only learn a model somewhere between these two. Therefore, no matter which of these two models the current pixel belongs to, the learned model will not be a good model for it.

A WLS method attempts to alleviate the problem by giving different weights to the contributions of different pixels so that a weighted sum of squared errors is minimised:

$$S_{\text{weighted}} = \sum_{i=1}^{N} w_i \left[y(n-i) - \boldsymbol{\beta}^T\mathbf{x}(n-i)\right]^2 , \tag{4}$$

where w_i is the weight assigned to the observation $\{\mathbf{x}(n-i)\,,\ y(n-i)\}$. It can be easily shown that the resultant coefficient vector will be [8]

$$\widehat{\boldsymbol{\beta}}_{\text{WLS}} = \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y}, \tag{5}$$

where $\mathbf{W}$ is a diagonal matrix

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_N \end{bmatrix}.$$

Based on the previous discussion, the weight w_i should be determined according to the similarity between the features at $y(n-i)$, a pixel in the training window, and $y(n)$, the current pixel. From Fig. 1, we can see that the feature at a pixel may be well defined by its corresponding input pixels. Therefore, the similarity between the features at $y(n-i)$ and $y(n)$ can be be measured by similarity between the two corresponding input vectors $\mathbf{x}(n-i)$ and $\mathbf{x}(n)$. The similarity between two vectors is usually measured by the distance between them. The smaller the distance is, the more similar they are to each other. Following this idea, we define the weight as

$$w_i = \frac{1}{\|\mathbf{x}(n-i) - \mathbf{x}(n)\|^2 + \Delta}, \tag{6}$$

where Δ is a quantity introduced to avoid division-by-zero errors. In our implementation, we set $\Delta = 1$.

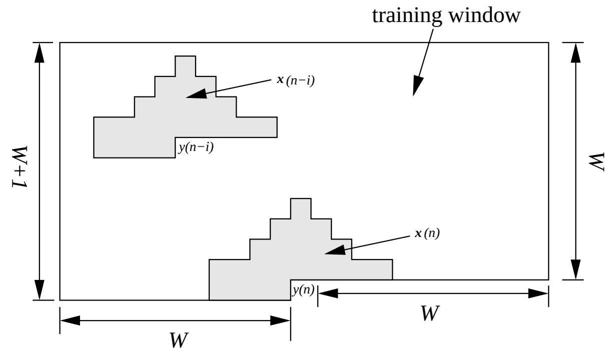

Fig. 2. The arrangement of the least squares training of the coefficients of a linear predictor for the current pixel (target) $y(n)$. The training window size is defined by W.

3. EXPERIMENTAL EVALUATION OF THE PROPOSED TECHNIQUE

In this section, we evaluate the effectiveness of the proposed WLS method by comparing it with the OLS method as well as WLS methods using two alternative weighting schemes. We also present the coding results of using the proposed WLS method as a component of a lossless image compression algorithm.

The setting of the training window is shown in Fig. 2. The training window size is defined by parameter W. Obviously, all the pixels in the training are taken from the previously scanned pixels (using the raster scan order). This ensures that no overhead information is needed to be transmitted to the decoder.

3.1. Comparison with OLS

To compare the performances of the proposed WLS method and the OLS method, we apply an 18th order WLS (labelled WLS-0 to differentiate it from WLS methods using other weighting schemes) based predictor and an 18th order OLS based predictor on the JPEG 9-image test set.[1] The results are presented as the average entropy of prediction errors versus the size of the training window in Fig. 3. It can be seen that when the training window is small ($W < 9$), OLS is better than WLS-0; however, when the training window is large enough, WLS-0 performs better.

This can be explained by splitting the contribution of each observation (an input/output pair) in the training window towards accurate prediction of current pixel into two parts: the relevant part and the irrelevant part. The relevant part helps define the “true” feature where the current pixel is and thus makes a positive contribution. The irrelevant part, on the other hand, defines a different feature from the “true” feature for the current pixel and thus makes a negative contribution.

The proposed weighting scheme attempts to assign a smaller weight to an observation with a more significant irrelevant part. Unfortunately this scales down the relevant part of the observation at the same time. When the training window size W is relatively small, the drawback due to the scaled down relevant part may override the benefit due to the scaled down irrelevant part because there is too little relevant information available. However, when W is large, the benefit may override the drawback because there will be enough relevant information even after the weighting process.

[1]These test images are available from: ftp://ftp.csd.uwo.ca/pub/from_wu/images/.

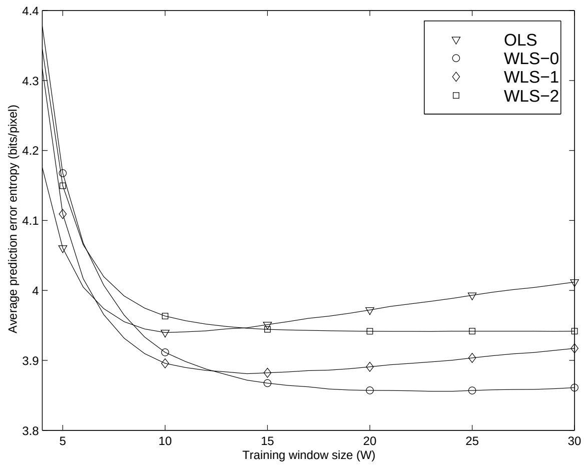

Fig. 3. Performance comparison between the proposed WLS based predictor (labelled WLS-0) and an OLS based predictor (OLS) as well as WLS predictors using two alternative weighting methods (labelled WLS-1 and WLS-2). All the predictors are of 18th order. The performance is measured by the average entropy of prediction errors for the JPEG 9-image test set.

3.2. Comparison with WLS Using Other Weighting Methods

To verify the effectiveness of the proposed weighting method, we compare it with two alternative weighting methods, which are labelled WLS-1 and WLS-2, in Fig. 3.

* WLS-1: We also assign the weight according to similarities between the input vectors, but use a

image	Glicbawls	OLS	WLS-0	TMW^{Lego}	multi-WLS-0
balloon	2.64	2.69	2.64	2.60	2.60
barb	3.92	3.94	3.90	3.84	3.75
barb2	4.31	4.31	4.25	4.24	4.18
board	3.39	3.39	3.33	3.27	3.27
boats	3.63	3.64	3.58	3.53	3.53
girl	3.56	3.58	3.50	3.47	3.45
gold	4.28	4.27	4.25	4.22	4.20
hotel	4.18	4.16	4.08	4.01	4.01
zelda	3.54	3.56	3.53	3.50	3.51
Average bit-rate	*3.72*	*3.73*	3.67	*3.63*	*3.61*

Table 1. Compression results (in bits/pixel) on the JPEG 9-image test set.

measure different from (6) . The weight is defined as

$$w_i = \frac{1}{d_m^{(i)} + \Delta}, \quad (7)$$

where $d_m^{(i)}$ is the Manhattan distance between $\mathbf{x}(n-i)$ and $\mathbf{x}(n)$:

$$d_m^{(i)} = \sum_{j=1}^{J} |x_j(n-i) - x_j(n)|,$$

and $\Delta = 1$.

∗ WLS-2: In general, we can assume that the closer a pixel is to the current pixel geometrically, the closer their features are to each other. Therefore, geometrically closer pixels should make more contributions towards an accurate prediction than pixels further away. Based on this consideration, we define the weights as

$$w_i = \frac{1}{d_g^{(i)} + \Delta}, \quad (8)$$

where $d_g^{(i)}$ is the Manhattan distance between $y(n)$, the pixel to be predicted, and $y(n-i)$, a pixel in the training window. In Glicbawls, a weighting scheme based on the Manhattan distance $d_g^{(i)}$ is also used. In this scheme, the weight is given by $w_i = 0.8^{d_g^{(i)}}$. Obviously, this weight decays much more quickly than (8) as $d_g^{(i)}$ increases.

From Fig. 3, we can see that the comparison between the proposed WLS-0 and WLS-1 is quite similar to that between WLS-0 and OLS. When the training window size W is smaller than a certain value (about 13), WLS-1 is better than WLS-0. This may be due to the obviously heavier reduction in the contribution of each pixel in the training window in the WLS-0 method (considering the squared terms in (6)). However, when the training window is large enough, WLS-0 is clearly the better method. Actually, the performance of WLS-1 becomes worse with an increasing W (when $W > 15$), indicating that it does not do enough to take out irrelevant contributions to the LS calculation from the observations. Except for very small training window sizes ($W < 6$), the performances of WLS-0 and WLS-1 are consistently better than that of WLS-2. This confirms that feature similarity is a better weighting criterion than geometric position (distance).

3.3. Coding Results Using the Proposed WLS Method

To demonstrate the benefit of using the proposed WLS (WLS-0) method, in Table 1, we show the actual coding results of using an 18th order WLS-0 based predictor and compare them with those using an OLS based predictor of the same order as well as the results of Glicbawls. The training window size W is fixed at 15. A context-based entropy coding scheme, similar to that used in [4], is employed to code the prediction errors. From the table, we can see that WLS-0 clearly outperforms both OLS and Glicbawls. As a further confirmation of the effectiveness of the proposed WLS, we have also included results by using a combination of multiple WLS-0 based predictors (labelled multi-WLS-0 in Table 1).[7] It can be seen that multi-WLS-0 has a performance better than that of TMW^{Lego} [9], arguably the best performing lossless image compression algorithm.

4. CONCLUSION

This paper has presented a new prediction technique for lossless image compression. The proposed prediction technique is based on a unique WLS method which gives a weight to to the contribution of an observed sample according to the similarities between the feature at this sample and that at the pixel to be predicted. The effectiveness of this WLS method has been confirmed by experiment results, which show that the proposed WLS outperforms both the OLS method and WLS methods using some alternative weighting schemes. Its superior performance has been further demonstrated by the actual coding results, which show that when using a combination of a group of WLS-based predictors, we can have a compression performance better than any other algorithms known to us.

5. REFERENCES

[1] N. Kuroki, T. Nomura, M. Tomita, and K. Hirano, “Lossless image compression by two-dimensional linear prediction with variable coefficients,” *IEICE Trans. Fundamentals*, vol. E75-A, pp. 882–889, July 1992.

[2] X. Wu, K. U. Barthel, and W. Zhang, “Piecewise 2D autoregression for predictive image coding,” in *Proc. IEEE International Conference on Image Processing*, vol. 3, (Chicago), pp. 901–904, Oct. 1998.

[3] X. Li and M. Orchard, “Edge-directed prediction for lossless compression of natural images,” *IEEE Trans. Image Processing*, vol. 10, pp. 813–817, June 2001.

[4] H. Ye, G. Deng, and J. C. Devlin, “Adaptive linear prediction for lossless coding of greyscale images,” in *Proc. IEEE International Conference on Image Processing*, vol. 1, (Vancouver, Canada), pp. 128–131, Sept. 2000.

[5] B. Meyer and P. Tischer, “Glicbawls-grey level image compression by adaptive weighted least squares,” tech. rep., School of Computer Science and Software Engineering, Monash University, 1997.

[6] B. Meyer and P. Tischer, “Glicbawls – grey level image compression by adaptive weighted least squares,” in *Proc. IEEE Data Compression Conference*, (Snowbird, Utah), Mar. 2001.

[7] H. Ye, *A Study on Lossless Compression of Greyscale Images*. PhD thesis, Department of Electronic Engineering, La Trobe University, Oct. 2002.

[8] A. Sen and M. Srivastava, *Regression Analysis: Theory, Methods, and Applications*. New York: Springer-Verlag, 1990.

[9] B. Meyer and P. Tischer, “TMW^{Lego} – an object oriented image modelling framework,” in *Proc. IEEE Data Compression Conference*, (Snowbird, Utah), Mar. 2001.