A METHOD FOR PREDICTIVE ORDER ADAPTATION BASED ON MODEL AVERAGING

Guang Deng, Hua Ye, Slaven Marusic and David Tay

Department of Electronic Engineering, La Trobe University
Bundoora, Victoria 3083, Australia
d.deng@ee.latrobe.edu.au

ABSTRACT

In lossless image coding, it has been demonstrated that using the method of ordinary least squares (OLS) to design a linear predictor for each pixel results in better compression performance than that of the state-of-the-art. In previous studies, the order of the predictor is chosen empirically and £xed for the whole image. Since images are non-stationary signals, the order should be adapted to the local characteristics of the image. In this paper, we tackle this problem by using a model averaging approach. We show that by averaging over a group of OLS predictors, the effective number of parameter of the resultant predictor is adjusted adaptively. We show that the proposed method is robust to changes in the size of the training block. It also leads to better performance than the OLS predictor.

1. INTRODUCTION

1.1. Ordinary least squares based prediction

Predictive coding has been one of the most important techniques for lossless image compression. In recent years, several independent studies have shown that by using the ordinary least squares (OLS) to design a linear predictor for each pixel, better performance than that of state-of-the-art in lossless image compression have been achieved [1, 2, 3]. In a matrix form, the OLS problem can be represented as

$$\mathbf{t} = \mathbf{X}^T\mathbf{w} + \mathbf{e} \tag{1}$$

where $\mathbf{t} = [x(N), x(N-1), ..., x(1)]^T$ is a vector of N pixels in a training block shown in Figure 1, $\mathbf{e}$ is a vector of the prediction errors and $\mathbf{X} = [\mathbf{x}(N), \mathbf{x}(N-1), ..., \mathbf{x}(1)]$ is an $(L \times N)$ matrix. Its column vector $\mathbf{x}(k)$ has L elements, denoted by $x_k(l)$. It represents the graylevel of the lth causal neighbouring pixel of the pixel $x(k)$.

By minimizing the sum of squared prediction errors for pixels in the training block, the OLS solution for the coef£cient vector $\mathbf{w}$ is given by

$$\widehat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{t} \tag{2}$$

For the current pixel $x(N+1)$, its prediction is given by

$$\widehat{x}(N+1) = \mathbf{x}^T(N+1)\widehat{\mathbf{w}} \tag{3}$$

In previous studies, the order of the predictor (L) is chosen empirically and is £xed for the whole image. As the image signal is generally regarded as non-stationary, it is expected that different areas of the image may require predictions of different orders. In an extreme case, there is an optimal order for each pixel. In addition, for a given training block of N pixels, using a larger prediction order usually leads to a smaller training error for the training block. However, there is no guarantee that a predictor with a smaller training error will make a smaller prediction error for the current pixel. In fact, it is well known that a larger order may lead to an over-complex model that can over-£t the training data [4]. As such, the resultant predictor has poor prediction/generalization ability.

Our goal is to determine a predictor with good predictive ability. This predictor can not be determined by the OLS which only minimizes the training error. Therefore, it is necessary to study methods that can be used to determine the best prediction order for each pixel.

1.2. Ridge least squares

A well-established method for changing the model complexity is by using the ridge least squares (RLS) [5] that provides a mechanism to control the model complexity. RLS can be motivated from a number of different perspectives such as: Bayesian interpolation [6], regularization and the bias-variance relationship [4, 5] of a data model. In RLS, the coef£cient vector is obtained by minimizing a cost function

$$J = \beta\mathbf{e}^T\mathbf{e} + \alpha\mathbf{w}^\mathbf{T}\mathbf{w} \tag{4}$$

where α and β are called hyper-parameters which are estimated from the training data. The coef£cient vector is given by

$$\widehat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{t} \tag{5}$$

0-7803-7750-8/03/$17.00 ©2003 IEEE

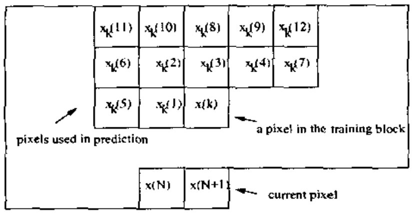

Figure 1. The training block of pixels for the current pixel, denoted by $x(N+1)$. For a pixel $x(k)$, there is an associated vector $\mathbf{x}(k) = [x_k(1), x_k(2), ..., x_k(12)]^T$. The prediction of $x(k)$ is given by $\widehat{x}(k) = \mathbf{x}^T(k)\widehat{\mathbf{w}}$. In this figure, a linear prediction of the order of 12 is used.

where $\lambda = \frac{\alpha}{\beta}$. It can be seen that when $\alpha = 0$, RLS becomes OLS.

The model complexity for RLS is represented by the effective number of parameters [5] (ENP) which is defined as

$$\gamma = tr(\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T) \tag{6}$$

For RLS, the ENP is less than L. It is a function of λ for a given matrix $\mathbf{X}$. For OLS, $\lambda = 0$ and the ENP is L which is the same as the order of the linear predictor. Therefore, we can tune the parammater λ such that the ENP of the predictor is adapted to the training data in order to achieve improved predictive performance.

The hyper-parameters α and β can be estimated by either one of the re-estimation algorithms: the EM algorithm [7] and the DM algorithm [6].

EM algorithm:

$$\frac{1}{\beta} = \frac{\mathbf{e}^T\mathbf{e} + \gamma/\beta}{N} \tag{7}$$

$$\frac{1}{\alpha} = \frac{\widehat{\mathbf{w}}^T\widehat{\mathbf{w}} + (L-\gamma)/\alpha}{L} \tag{8}$$

DM algorithm:

$$\frac{1}{\beta} = \frac{\mathbf{e}^T\mathbf{e}}{N-\gamma} \tag{9}$$

$$\frac{1}{\alpha} = \frac{\widehat{\mathbf{w}}^T\widehat{\mathbf{w}}}{\gamma} \tag{10}$$

It has been shown by Orr [7] and Mackay [6] that after a number of iterations, both algorithms converge. This is also confirmed in our experiments using images.

1.3. Outline of the proposed approach

In this paper, we address the problem of how to change the order of the predictor adaptively from a model averaging point of view. We show that under certain assumptions, a linear predictor can be expressed as a linear combination of a group of linear predictors of different order determined by the OLS method. The combination coefficients are the posterior probability of the component predictors given the training data. By regarding the combination coefficient as a measure of the performance of the component predictor, we study four other combination methods which use the local prediction error, the generalized cross validation criteria, the Bayesian information criteria and simple averaging, respectively.

2. THE MODEL AVERAGING APPROACH

2.1. Problem formulation

We formulate the prediction problem as the the following. Given the observed data $D = \{t(n), \mathbf{x}(n)\}_{n=1:N}$ and a group of M models. Each model, denoted by H_m, is represented by the parameters $\{\alpha_m, \beta_m, \mathbf{w}_m, L_m\}$. We want to predict the target $t(N+1)$ for a new input $\mathbf{x}(N+1)$. The probability distribution for the target is given by

$$\begin{aligned} &P(t(N+1)|\mathbf{x}(N+1), D) \\ &= \textstyle\sum_{m=1}^{M} P(t(N+1)|\mathbf{x}(N+1), H_m)P(H_m|D) \end{aligned} \tag{11}$$

The prediction is the minimum mean square error estimate which is the conditional mean

$$\begin{aligned} \widehat{t}(N+1) &= E(t(N+1)|\mathbf{x}(N+1), D) \\ &= \textstyle\sum_{m=1}^{N} P(H_m|D)E(t(N+1)|\mathbf{x}(N+1), H_m) \\ &= \textstyle\sum_{m=1}^{N} P(H_m|D)\widehat{t}_m(N+1) \end{aligned} \tag{12}$$

where

$$\widehat{t}_m(N+1) = E(t(N+1)|\mathbf{x}(N+1), H_m) \tag{13}$$

is the conditional mean of each model. Therefore, using the proposed method, the prediction is formed by a weighted average of the predictions made by component models.

We note that a full treatment of this problem requires two major tasks: (1) the estimation of model parameters and the conditional mean of each model, and (2) the calculation of the posterior probability of each model given the data.

2.2. Using an OLS model

The first task can be simplified by assuming a regression model so that

$$t(n) = \widehat{t}_m(n) + e_m(n). \tag{14}$$

With a ridge regression model, the model parameters can be estimated from the data by using an iterative algorithm. This has been briefly described in section 1.2. In this paper, we use an OLS model and set the order of the mth model as L_m $(L_i \neq L_j, i \neq j)$, such that we only require to determined the vector $\widehat{\mathbf{w}}_m$. The computational requirement is thus reduced compared to that of the RLS. The prediction using the *m*th model is given by

$$\widehat{t}_m(N+1) = \mathbf{x}_m^T(N+1)(\mathbf{X}_m^T\mathbf{X}_m)^{-1}\mathbf{X}_m^T\mathbf{t}_m \qquad (15)$$

where the index *m* is used to indicate that the respective vectors and matrix are formed by using the *m*th model.

2.3. The posterior probability for each model

The second task can be simpli£ed by assuming that the prior probability for each model is the same so that

$$P(H_m|D) \propto P(D|H_m)P(H_m) \propto P(D|H_m) \qquad (16)$$

This probability can be determined using the following approximation that is based on the assumption that the probability distribution $P(D|\mathbf{w}_m, H_m)$ is sharply centered at the OLS solution of the coef£cient vector $\widehat{\mathbf{w}}_m$:

$$P(D|H_m) \propto P(D|\widehat{\mathbf{w}}_m, H_m)P(\widehat{\mathbf{w}}_m|H_m)\sigma_{\mathbf{w}|D} \qquad (17)$$

The last term

$$\sigma_{\mathbf{w}|D} = \frac{(2\pi\sigma_m^2)^{\frac{L_m}{2}}}{\sqrt{\det \mathbf{X}^T\mathbf{X}}} \qquad (18)$$

is the error-bar for the estimation of $\mathbf{w}_m$, where σ_m^2 is the variance of prediction errors due to the *m*th model. The prior probability $P(\widehat{\mathbf{w}}_m|H_m)$ can be calculated by assuming that each coef£cient is independent to others and its prior probability is proportional to its error-bar:

$$\begin{aligned} P(\widehat{\mathbf{w}}_m|H_m) &= \textstyle\prod_{k=1}^{L_m} P(\widehat{w}_m(k)|H_m) \\ &= \textstyle\prod_{k=1}^{L_m} \frac{1}{a\sigma_m^2(k)} \end{aligned}$$

where *a* is a scaling factor and $\sigma_m^2(k)$ is the variance for the estimation of $\widehat{w}_m(k)$. This variance is the *k*th diagonal element of the matrix: $\frac{1}{\sigma_m^2}(\mathbf{X}^T\mathbf{X})^{-1}$. The probability $P(D|\widehat{\mathbf{w}}_m, H_m)$ can be calculated by assuming an i.i.d. Gaussian distribution with the data *D*. In our experiment, this method will be called "MA-EVI".

Since the above method is very computation intensive, we consider another simpli£cation by regarding equation (12) as a linear combination of the component predictions. As such, we have

$$\widehat{t}(N+1) = \sum_{m=1}^{M} \theta_m \widehat{t}_m(N+1) \qquad (19)$$

where $\sum_{m=1}^{M} \theta_m = 1$ and $\theta_m \geq 0$. The combination coef£cient θ_m can be interpreted in different ways. For example, it may be proportional to a measure of the performance of the mth predictor. Better performing predictors should have larger coef£cients. For a model, we use the average squared prediction errors of four immediate neighbouring pixels as a measure of its performance

$$\theta_m \propto \frac{1}{\sum_{k=0}^{3} e_m^2(N-k)} \qquad (20)$$

In our experiment, this method will be called "MA-ERR".

The combination coef£cients may also be inversely proportional to an estimate of the prediction error given by equation (14). We can use a number of well established methods such as the generalized cross validation (GCV) and the Bayes information criterion (BIC) to estimate the prediction error[5][8]. Let s_m represent the sum of square errors for the data *D* using the *m*th model, then the GCV and BIC estimates of the prediction error are given by the following two formulas, respectively

$$e_{GCV} = \frac{N}{(N-\gamma_m)^2} s_m \qquad (21)$$

and

$$e_{BIC} = \frac{N + \gamma_m(\ln(N) - 1)}{N(N-\gamma_m)} s_m \qquad (22)$$

where γ_m is the effective number of parameters of the *m*th model. Since we use an OLS model, $\gamma_m = L_m$. We can set the combination coef£cient as

$$\theta_m \propto 1/e_{GCV}$$

or

$$\theta_m \propto 1/e_{BIC}.$$

In our experiment, these two methods will be called "MA-GCV" and "MA-BIC", respectively.

In an simplest case, we can set all the combination coef£cients to the same value: $\theta_m = 1/M$. This can greatly reduce the computational complexity. In fact, such a setting re□ects our ignorance of the predictive performance of any predictor. In our experiment, this method will be called "MA-AVE".

2.4. The effective number of parameters

We note that the proposed model averaging method is actually an alternative way to change the effective number of parameters. It can be shown that the effective number of parameters of the proposed predictor is given by

$$\gamma = \sum_{m=1}^{M} \theta_m L_m \leq M \qquad (23)$$

Adapting the effective number of parameters to the training data leads to better predictive ability of the predictor. Model averaging also reduces the variance of the prediction error. This can be seen from the regression model expressed in equation (14). If we assume that the prediction errors produced by different models are independent, having variances σ_m^2 $(m = 1 : M)$, then it can be easily shown that by model averaging the resultant predictor will produce prediction errors whose variance is smaller than that of any one of the component predictors.

3. EXPERIMENTAL RESULTS

In this section, we present experimental results using the "airport" image which is a radar image of an airport. It is an 8-bit 512×512 image. Results of other images are consistent with this image. In our experiment, we used 9 OLS predictors whose orders are from 4 to 12. We tested different ways for calculating the combination coefficients against the size of the training data which is a rectangular block of pixels.

For comparison, we also tested the OLS algorithm of prediction order of 12 and the two representative RLS algorithms which are briefly described in section 1.2 and are called "RLS-EM" and "RLS-DM", respectively. We used the entropy of the prediction errors for the comparison, since our goal is compare the predictive performance of different prediction methods.

We can make the following observations from Table 1. The performance of the proposed model averaging approach and the RLS is quite robust to the size of the training data. When the size is small, its performance does not deteriorate as much as that of the OLS. This clearly demonstrates the benifit of adapting the prediction order. When the size is large, all methods produce similar results. It is interesting to note that the performance of "MA-AVE" is nearly the same as other methods. Although other proposed combination methods, such as MA-EVI, MA-GCV and MA-BIC, have their appealing theoretical support, they are derived based on some assumptions that may not be valid everywhere in an image. Since in practical applications, it is not easy to account for the uncertainty about the assumptions, the simplest approach "MA-AVE" seems a reasonable one that reflects our uncertainty/ignorance about the performance of the component predictors. Thus "MA-AVE" is a promising method for applications where there are certain requirements for low computational complexity.

4. SUMMARY

In this paper, we have addressed the problem of adapting the prediction order in the least squares based predictor design. We have used model averaging as a tool to change the effective number of parameters. We have also investigated a number of ways to calculate the model combination coefficients. Experimental results show that the proposed model averaging method is an effective way of adapting the effective number of parameters to achieve improved predictive performance. We note that the proposed model averaging approach can be applied to other signal modeling problems, since the formulation of the problem described in Section 2.1 is quite general and is not restricted to lossless image coding applications.

	40	84	144	220
MA-EVI	6.6852	6.6253	6.6000	6.5844
MA-ERR	6.7356	6.6225	6.5848	6.5671
MA-GCV	6.7213	6.6191	6.5833	6.5660
MA-BIC	6.7189	6.6182	6.5830	6.5660
MA-AVE	6.7227	6.6188	6.5830	6.5660
OLS-12	6.8783	6.6820	6.6162	6.5854
RLS-EM	6.6505	6.6005	6.5769	6.5631
RLS-DM	6.6493	6.6003	6.5769	6.5631

Table 1. The entropy (bits/pixel) of prediction errors using the "airport" image. Each column presents results of different method using the same size of the training block.

5. REFERENCES

[1] X. Li and M. Orchard, "Edge-directed prediction for lossless compression of natural images," *IEEE Trans. Image Processing*, vol. 10, no. 6, pp. 813–817, June 2001.

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

[3] 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, Oct. 1998, pp. 901–904.

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*. London: Oxford University Press, 1995.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[6] D. J. C. Mackay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[7] M. J. L. Orr, "Recent advances in radial basis function networks," www.anc.ed.ac.uk/~mjo/papers/recad.ps.gz, 1999.

[8] ——, "Introduction to radial basis function networks," www.anc.ed.ac.uk/~mjo/papers/intro.ps, 1996.