Parametric Probability Models for Lossless Coding of Natural Images

Hua Ye, Guang Deng and John C Devlin
Department of Electronic Engineering, La Trobe University
Bundoora,Victoria 3083, Australia.
Phone: +61 3 9479 2036; Fax:+61 3 9471 0524.
email: h.ye@ee.latrobe.edu.au

ABSTRACT

Parametric modelling based entropy coding is an alternative technique to the non-parametric, contex-based techniques that are widely used in current image coding algorithms. Compared to the non-parametric techniques, the proposed parametric technique is more tractable mathematically and is easier to understand and manuplate. In this paper, we examine a number of issues related with the application of this technique to image coding. We discuss the strong and weak points of several probability distributions that can be used in our algorithm. We then present a working framework that can perform close to the state-of-the-art lossless image compression techniques.

1 Introduction

In a predictive lossless image compression system, there are often two distinctive stages: adaptive prediction and adaptive entropy coding. In the first stage, the system will predict the gray level of the current pixel with information about the past pixels. In the second stage, the prediction error will be coded with an entropy coder. This paper is mainly concerned with the second stage.

Entropy coding is a process in which the input symbols are converted into bit streams according to a probability model that specifies the likelihood for each possible symbol to occur. Naturally, the performance of entropy coding is dependent on the probability model being used. If the probability model is close to the true probability distribution, then good compression performance is expected. Otherwise, if it does not reflect the true distribution, poor compression or even data expansion will happen. There are generally two types of approaches to build the probability model: non-parametric ones and parametric ones.

The non-parametric approach is widely used in many current lossless image coding algorithms like the well-known CALIC [1]. Basically, the probability model is the frequency table of the symbols learnt on the fly of the coding process. If the signal source is stationary, then this easily built model will also be the best possible model we can get. However, natural images are normally non-stationary signals, meaning that different areas in an image may have different probability distributions. Context modelling, which is based on composite source models [6], has proved itself to be an effective technique to tackle this problem. In a context modelling scheme, pixels are classified into different categories (contexts) according to information on their neighbourhoods. They are then encoded using the probability models pertaining to their contexts. Although very effective, most published context modelling techniques are developed based on heuristics. Another important issue related to context-based coding schemes is that the prediction error often have to be quantised to avoid big distortions to the learnt probability distributions at the early stages of the coding process. Again, most quantisation schemes are also based on heuristics.

The parametric approach has been less popular in lossless compression community when compared to the non-parametric one. A well-known published algorithm using it is TMW [2]. In this approach, a symbol is encoded according to a pre-defined probability distribution rather than a frequency table. There are two questions that need to be answered before this method can produce satisfactory results. The first question is: what kind of probability distribution should be used? As is discussed above, natural images are generally non-stationary. Therefore, whatever probability distribution is used, it must be able to change according to local characteristics. Once we have established the probability distribution, the second question is: how do we estimate the parameter(s) of this probability distribution from the local characteristics?

In this paper, we present an experimental study to address these problems. Section 2 first compares several probability distribution functions (pdf) that are commonly used in image modelling and statistics, then describs a simple method to estimate the parameters for these pdf's on the fly of the coding process. Section 3 demonstrates the performance of the proposed schemes when coupled with an adaptive linear predictor.

2 Parametric probability models

In this section, we describe and compare a number of probability distributions that are commonly used in image modelling. These distributions include several one-parameter distribution and a two-parameter one. We will also propose a simple method to estimate these parameters.

2.1 One-parameter distributions

The one-parameter distributions that will be used are:

- Laplacian distribution, given by

$$f(x) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}}; \tag{1}$$

- Gaussian distribution, given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; \tag{2}$$

- t-distribution, given by

$$f(t) = K\left(1 + \frac{t^2}{n}\right)^{-\frac{n+1}{2}}, \tag{3}$$

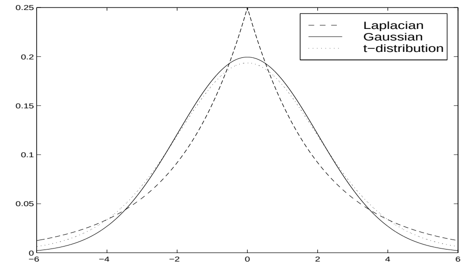
  where K is a normalisation factor and the positive integer n is called the degree of freedom. The t-distribution is quite close to the Gaussian distribution when $\mu = 0$ and $\sigma = 1$, especially for a large n. Therefore, it can not be used in applications where σ can be any value. In [3], a modified t-distribution given by

$$f(t) = K\left(1 + \frac{t^2}{13\sigma^2}\right)^{-\frac{13}{2}} \tag{4}$$

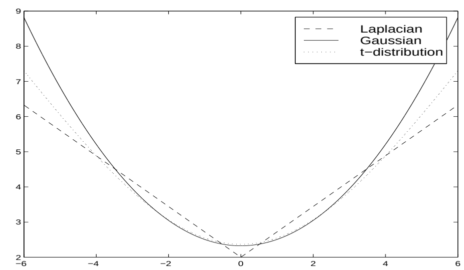
  is applied. In the rest of this paper, we will refer to (4) as the t-distribution.

Actually, in equations (1) and (2), there are two parameters: the centre (mean) μ and the standard deviation σ. But one can usually assume that the distributions are centred at the predicted value. So μ is known once a prediction is made. For the t-distribution, we can define $t = x - \mu$.

Fig. 1 shows the three probability distributions when $\sigma = 2$. From Fig. 1(a), one can see that among these three, Laplacian gives the highest probabilities near $x - \mu = 0$ but gives the lowest probabilities when $x - \mu$ is at the mid-range ($\sim \sigma$). The t-distribution is just a little bit lower than Gaussian at the centre. However, when x is further away from μ, t-distribution gives slightly higher probabilities. To see the numbers of bits needed for different distributions at different areas, Fig. 1(b) is obtained by taking $-\log_2(\cdot)$ of the probability density values. Although the difference between



(a) Probability density in linear scale



(b) Probability desity taken -log2(.).

Figure 1: The probability distributions when $\sigma = 2$.

Gaussian and t-distribution is still very little around the centre ($-\sigma$ to σ), it is much more significant at remote areas. This means that we will need much more bits if we use Gaussian instead of t-distribution when the amplitude of prediction error ($x - \mu$) is big.

2.2 Two-parameter distribution – generalised Gaussian

From the above discussion we can see that each distribution has its strong and weak points. Therefore a distribution that is good for one area in an image may not be good for another area. We should be able to get better performance if a distribution can change its shape according to local statistics. One such distribtution is the generalised Gaussian [4]. It is defined as

$$f(x) = ae^{-(b|x-\mu|)^{\gamma}}, \tag{5}$$

where μ, σ^2, and γ are mean, variance and shape parameter of the distribution, respectively. The positive constants a and b are given by

$$a = \frac{b\gamma}{2\Gamma(1/\gamma)}$$

and

$$b = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}}$$

where $\Gamma(\cdot)$ is the Gamma function defined by

$$\Gamma(z) = \int_0^{\infty} t^{z-1}e^{-t}dt. \tag{6}$$

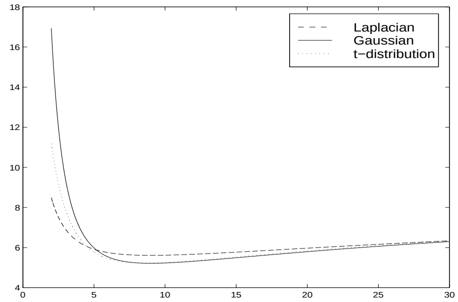


Figure 2: Sensitivities of different distributions to estimation errors of the variance. The value to code is $e = x - \mu = 9$. To illustrate the impact to the final bit-rate, the bit number is shown on the vertical axis.

The shape parameter γ determines the decay rate of the curve. When $\gamma = 2$, it becomes the Gaussian distribution. When $\gamma = 1$, it is the Laplacian distribution. It can be much sharper than the Laplacian or be much more spread out than the Gaussian.

2.3 Parameter estimation

For one-parameter distributions, the task will be to estimate the variance σ^2. It can easily be shown that if the probability distribution is centred at the predicted value, the optimal value σ that gives the highest probability for prediction error $e = x - \mu$ will be $\sigma_{opt} = |e|$. Since e can not be known when it is coded, there is almost certainly some difference between the estimated σ and σ_{opt} . To examine the sensitivity of different distributions to this kind of estimation error, we plot the probability density values at $e = 9$ for different values of σ in Fig. 2. From this figure, we can see that (1) it is generally better to make an "over estimate" than an "under estimate"; (2) the t-distribution seems to be more tolerant of estimation errors; (3) although the Laplacian distribution is the best when there is an "under estimate", its values for optimal σ fall short of those of Gaussian and t-distribution.

To make a reliable estimate, the sample must be large. But since images are usually non-stationary, the sample should not be too large. Otherwise pixels having different statistics would be included in the same sample. Through experiments, we found that the template shown in Fig. 3 seems to be able to produce good estimates for many natural image.

The shape parameter γ in the generalised Gaussian distribution will need more efforts to find. In [4], a rather straight-forward method is introduced. The variance σ^2 and the mean of the abssolute values $E\left[|x-\mu|\right]$ are first determined. The ratio of these two values will meet the following equation:

$$r(\gamma) = \frac{\sigma^2}{E^2\left[|x-\mu|\right]} = \frac{\Gamma(1/\gamma)\Gamma(3/\gamma)}{\Gamma^2(2/\gamma)}. \qquad (7)$$

From (7), we can then find the shape parameter γ.

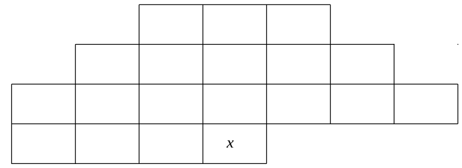


Figure 3: The templet used to estimate the parameters of the probability model. x represents the pixel to be coded.

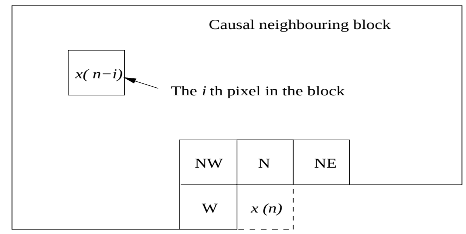


Figure 4: The causal pixel block used for prediction.

2.4 An implementation issue

The discussion so far are entirely on continuous distribution. But the pixel values for an image are all integers. Therefore, we need to find a way to specify the probability for each value to occur. One method is to do integration of the probability density in the interval $[x - 0.5,\ x + 0.5)$ for the probability of x, as is described in [3]. Another method is to use the probability density value at x as the probability of x (of course, we need to multiply it with a normalisation factor). Our experiment shows that the difference in the results of using these two methods is actually very small. Therefore, whenever an analytical expression of the integral is not available, we use the second method.

3 Experimental results

In this section, we briefly describe the adaptive predictor that is used to test the proposed entropy coding schemes. This is followed by the experimental results.

3.1 The adaptive predictor [5]

In the proposed algorithm, the final prediction is a linear combination of a set of six fixed sub-predictors shown in Table 1. Each of these predictors uses one or more of the four causally neighbouring pixels as is shown in Fig. 4. Let $p_k(n)(k = 1, 2, ..., 7)$ represent the kth causal sub-predictor for the current pixel, then the proposed predictor is:

$$P(n) = \frac{1}{D}\sum_{k=1}^{7}\alpha_k p_k(n) \qquad (8)$$

where α_k are the coefficients and $D = \sum_{k=1}^{7}\alpha_k$ is a normalization factor. The prediction is truncated to its nearest integer in actual implementation.

$$p_1 = W$$
$$p_2 = N$$
$$p_3 = N + W - NW$$
$$p_4 = NE$$
$$p_5 = (N + W)/2$$
$$p_6 = NW$$
$$p_7 = (NE + N)/2$$

Table 1: List of fixed predictors. The location of the causal neighbouring pixels is shown in Fig. 4.

	CALIC	Heu	Lap	Gau	t	GG
balloon	2.78	2.75	2.83	2.76	2.75	2.76
barb2	4.46	4.45	4.59	4.54	4.53	4.52
barb	4.31	4.27	4.42	4.34	4.34	4.33
board	3.51	3.49	3.61	3.54	3.53	3.53
boats	3.78	3.76	3.88	3.82	3.81	3.80
girl	3.72	3.67	3.78	3.71	3.71	3.70
gold	4.35	4.33	4.41	4.37	4.35	4.35
hotel	4.18	4.19	4.31	4.28	4.27	4.24
zelda	3.69	3.67	3.81	3.73	3.73	3.73
ABR	*3.86*	*3.84*	3.96	*3.90*	3.89	3.88

Table 2: Compression results (bits/pel) and average bit rate (ABR) for the JPEG test image set. The "Heu" column presents the results using the heuristic-based context modelling and non-parametric entropy coding. The "Lap", "Gau", "t" and "GG" columns presents the results using the Laplacian, Gaussian, t-distribution and the generalised Gaussian distribution functions, respectively.

3.2 Experimental results

In this section, we present the experimental results using the original 8 bits/pixel JPEG test image set which contains nine images whose size is 576x720.

Table 2 presents experimental results for the heuristic-based context modelling and the proposed algorithms. As a bench mark, the results using CALIC are also presented. The results for Laplacian distribution is clearly not as good as those for other distributions. This seems to show that although it is widely accepted the global probability distribution of prediction errors is very close to Laplacian, the instant (local) distribution tend to be more Gaussian-like. The t-distribution performs a little bit bette than the Gaussian. This is probably because the t-distribution is more tolerant of large errors. As expected, the generalised Gaussian is the best among the four listed since it has the most flexibility to adapt itself to different statistics.

It can be seen that the compression performance of proposed parametric modelling algorithms is not as good as the heuristic-based algorithm due to the fact that the coding parameters are not fine-tuned enough for these images. It can also bee seen that with the proposed context modelling algorithm the compression performance of the algorithm is quite close to that of CALIC.

4 Summary and future work

In this paper, we have presented an experimental study on issues related to parametric probability modelling for entropy coding. We have compared several popular distribution functions for suitability to our application–image coding. We also incorporated the proposed entropy coding scheme with an adaptive predictor to test its effectiveness. Compared to the current heuristics based algorithms, the proposed algorithm is more tractable mathematically and is easier to understand and manuplate. The performance of the proposed schemes is also comparable to the state-of-the-arts techniques.

In the future, studies will be carried out on how to model a more complicated probability distribution that can occur in practice. Combination of several different simple distributions seems to be one possible solution.

References

[1] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp.437-444, Apr. 1997.

[2] B. Meyer and P. Tischer, "TMW–A new method for lossless image compression," in *Proc. Int. Picture Coding Symp.*, Berlin, Germany, Sept. 1997.

[3] B. Meyer and P. Tischer, "Glicbawls–grey level image compression by adaptive weighted least squares," Technical Report, School of Computer Science and Software Engineering, Monash University, Australia, 2001.

[4] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video," *IEEE Trans. Circuit and Sys. for Video Tech.*, vol. 5, pp. 52–56, Feb. 1995.

[5] G. Deng and Y. Hua, "Lossless image compression using adaptive predictor combination, symbol mapping and context filtering," *Proc. IEEE Int. Conf. Image Processing*, CDROM, 1999

[6] N. D. Memon and K. Sayood, "Lossless image compression: A comparative study," *Still image compression, Proc. SPIE*, v. 2148, pp. 8-20, 1995.