Tina Memo No. 1999-006

Internal, rejected by NIPS 1998.

The Application of the Bhattacharra Measure as a Selector of the Point of Optimal Generalisation During Neural Network Training.

P. Rockett and N. A. Thacker.

Last updated
11 /4 / 2005

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

The Application of the Bhattacharyya Measure as a Selector of the Point of Optimal Generalisation During Neural Network Training

Peter Rockett, Dept. of Electronic & Electrical Engineering, University of Sheffield, Mappin St, Sheffield, S1 3JD, UK {p.rockett@shef.ac.uk}

Neil Thacker, Dept. of Medical Biophysics, University of Manchester, Stopford Building, Oxford Rd., Manchester M13 9PT, UK {nat@sv1.smb.man.ac.uk}

Abstract

In this paper we discuss the desirability of selecting a data model with due regard to the errors on the data and particularly the strategy for obtaining maximal similarity between the target and prediction probability density functions (PDFs). We describe the properties of the Bhattacharyya measure which make it suitable for this comparison and outline the methodology for the construction of a suitable prediction PDF for a neural network. We present experimental results showing that the Bhattacharyya measure can accurately locate the optimal stopping point for conventional backpropagation learning on a univariate regression problem.

1 - Introduction

Generalisation ability is a major issue when fitting data with parametric models. Determining which model parameters yield the optimal prediction accuracy remains a challenging problem, as indeed does the question of which is the most appropriate model with which to describe a finite data sample. The model selection problem recurs throughout the physical and statistical sciences in a number of guises but has been termed the *bias-variance dilemma* by Geman *et al* [1]; Bishop [2] also provides a useful discussion of the problem.

One common approach to determining the point of optimal generalisation is to partition the set of labelled examples into a training set and a validation set. Only the training set is explicitly used to adjust the model parameters and the validation set is employed to test the model's accuracy on unseen data. As a limiting case, a single member of the dataset is used for performance testing and the model is repeatedly retrained using each member as the validation set in turn. This latter *leave-one-out* approach is rather time-consuming; dataset partitioning approaches have other drawbacks which mean they are unsatisfactory – see [2].

Reflecting its importance, the issue of generalisation in neural networks has received a great deal of attention including Bayesian treatments [3,4].

In general, neural networks are trained to minimise a squared output error which is equivalent to minimising the *p-2* (or Euclidean) norm of the difference between the target and prediction vectors. The drawback of this statistic is that it ignores the likely errors on the components of these two vectors which can lead to *overfitting* whereby the model becomes overly specialised on the training sample.

In this paper we present a novel approach to finding the point of optimal generalisation, the under-pinning principle of which can be *illustrated* by the problem of drawing a smooth curve through a set of experimentally determined data shown in Figure 1. When a human performs this task of model selection, they take due account of the error bars on the data. For example, the same data points are shown in Figures 1(a) and (b) but with different error bars leading to two different but nonetheless *reasonable* curves. The conclusion is clearly that data with smaller errors mandate a more elaborate predictive model.

In this work we have formalised the notion of taking due account of errors during a fitting process by comparing the probability density functions (PDFs) of the target and predicted quantities. A model with a prediction PDF which is substantially different from the target PDF for a given datum obviously cannot be generalising to the underlying parent dataset in

an optimal fashion. In seeking the point of optimal predictive generalisation we are seeking the greatest degree of similarity between target and prediction distributions. The measure we have used as a gauge of similarity is the Bhattacharyya metric [5,6] which we argue (elsewhere) is the optimal method of comparing frequency-coded data [6].

In this paper we present a derivation of the Bhattacharyya measure to underline the properties which make it attractive for the present application. Next we describe the construction of the prediction PDFs for a neural network and the application of the Bhattacharyya measure. Finally, we show some experimental results and discuss future directions for this research.

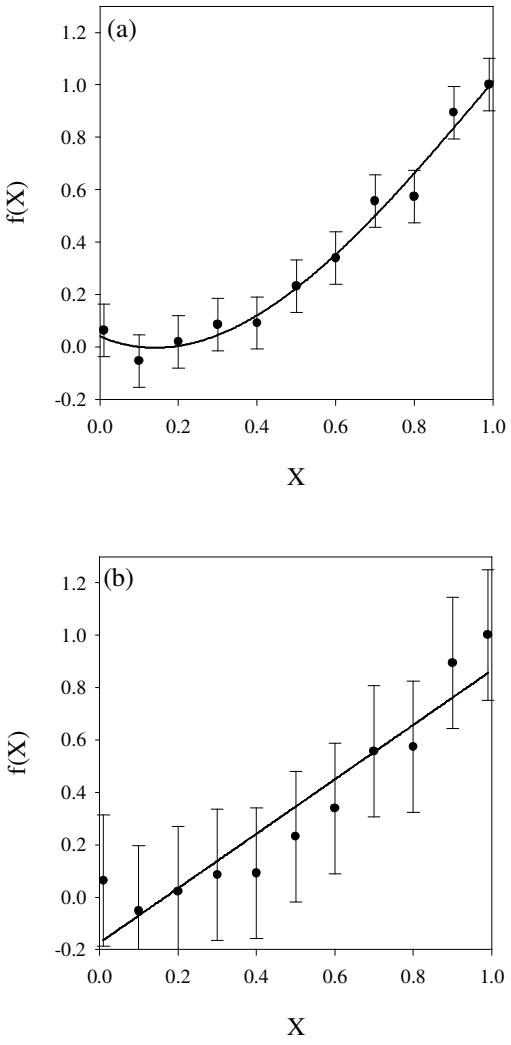

Figure 1: The same data in (a) and (b) but different error bars.

2 - Derivation and Properties of the Bhattacharyya Measure

If we imagine a random variable, X in one-dimension, we wish to measure a distance in the space between two points, x_1 and x_2. In order to enforce statistical significance on our measure we normalise the distance by the standard deviation of X which leads to:

$$d = \int_{x_1}^{x_2} \frac{dX}{\sigma}$$

At this point we have two possible ways forward: We can either assume that σ is constant and can be taken outside the integral which leads to the straightforward Euclidean distance and the χ^2 measure. This is a valid local approximation but for significant distances in pattern space this becomes less and less true.

Alternatively, under the assumption that the errors on X are Poisson distributed, $\mathrm{var}(X) = X$ and so the integral becomes:

$$d = \int_{x_1}^{x_2} \frac{dX}{\sqrt{X}} = 2\left(\sqrt{x_2} - \sqrt{x_1}\right)$$

For measuring distances in N-dimensions it is convenient to consider d^2:

$$d^2 = 4\left(\sqrt{x_2} - \sqrt{x_1}\right)^2$$

which is the Matusita measure [7]. Expanding the square in the above expression yields:

$$d^2 = 4\left(x_2 + x_1 - 2\sqrt{x_2}\sqrt{x_1}\right)$$

If we consider X_i to be the N bin contents of a histogram then from the normalisation condition on a PDF, the sum over all bins of both x_1 and x_2 is unity and therefore the Matusita distance is of the form:

$$d^2 = const \times \left(1 - \sqrt{x_1 x_2}\right)$$

The term, $\sqrt{x_1 x_2}$ is the Bhattacharyya measure and it is clear that minimising the Matusita distance is equivalent to maximising the Bhattacharyya measure; for computational convenience we prefer the Bhattacharyya form of the measure.

Two observations are in order at this point: First, from the derivation it is clear that the Bhattacharyya measure is a *generalised* χ^2 measure but one that properly accounts for the spatial variation across the pattern space of standard deviation under the assumption of Poisson-distributed errors.

Second, in forming the Bhattacharyya metric we take the sum of products of the square roots of the variates, or alternatively, we are taking the vector inner product of two transformed vectors. The procedure is thus akin to the data transformations of classical statistics. In [6] we show that the effect of this transformation is to map the variates into a space in which the variance of the Bhattacharyya measure is constant; it is this important fact that makes the unadjusted Bhattacharyya measure the appropriate statistic for comparing different models. This is in sharp contrast to the χ^2 statistic which requires systematic correction for model comparison [8].

Further empirical justification of the suitability of the Bhattacharyya measure as a model selector has been given by Lacey *et al* [9] in the context of a switchable Kalman filter while Aherne *et al* [6] consider other important properties of the Bhattacharyya measure.

3 - Application of the Bhattacharyya Measure to MLP Prediction

Having established a principled means of comparing two PDFs, we now address the issue of constructing a prediction PDF for a neural network. Broadly we have followed the methodology employed previously in applying the Bhattacharyya measure to the problem of identifying the correct order of polynomial to fit through a set of data [10].

Considering the *n*-th training datum, we calculate the elements of the inverse covariance matrix of the network parameters from first-order error propagation:

$$c_{ij}^{-1} = \sum \frac{1}{\sigma^2}\left(\frac{\partial f}{\partial a_i}\frac{\partial f}{\partial a_j}\right)$$

where σ^2 is the target variance, $a_{i,j}$ are the model parameters, $f = f(x;\mathbf{a})$ is the model prediction and c_{ij}^{-1} is the *i,j*–th element of the *inverse* covariance matrix. Since we require an estimate of the model prediction at the *n*-th point of the training set, the summation is taken over the whole training set but *omitting* the *n*-th datum. (Clearly if the *n*-th datum were to be included in an estimate of prediction ability at the *n*-th datum, such an estimate would be biased.) At this point we can determine an estimate of the constraint on the prediction, δy , for the *n*-th datum from:

$$\delta y_n^2 = \nabla f_n \mathbf{C}_n \nabla f_n^T$$

where ∇f_n is the column vector of partial derivatives of *f* evaluated at the *n*-th datum and $\mathbf{C}_n$ is the inverse of the matrix whose elements are given by c_{ij}^{-1}. The (expectation of the) model prediction, f_n is known therefore the Bhattacharyya overlap between prediction and target quantities can be calculated from [5]:

$$B_{diag} = \sqrt{\frac{2\sqrt{\delta f_n^2 \times \sigma^2}}{\delta f_n^2 + \sigma^2}} \exp\left(\frac{-(f_n - y_n)^2}{4\left(\delta f_n^2 + \sigma^2\right)}\right)$$

where $\delta f_n^2 = \delta y_n^2 + \sigma^2$ is the prediction error and B_{diag} is the Bhattacharyya overlap at the *n*-th datum. Taken over the whole training set, the total Bhattacharyya measure is given by:

$$B_{tot} = \prod\left(B_{diag}\right)$$

although in practice we sum the common logarithm of B_{diag} for reasons of numerical stability.

The above treatment assumes that the prediction errors at each datum are independent; in practice, the predictions at two data points will both be conditioned on the same set of model parameters and hence the prediction errors will, in general, be correlated. We can accommodate these correlations by estimating the covariance on the parameters by computing the *average* of the leave-one-out parameter covariance matrices:

$$\mathbf{C}_a = \frac{1}{N}\sum_{n=1}^{N} \mathbf{C}_n$$

and determining the prediction covariance matrix from:

$$\mathbf{C}_f = \nabla \mathbf{f} \mathbf{C}_a \nabla \mathbf{f}^T + \sigma^2 \mathbf{I}$$

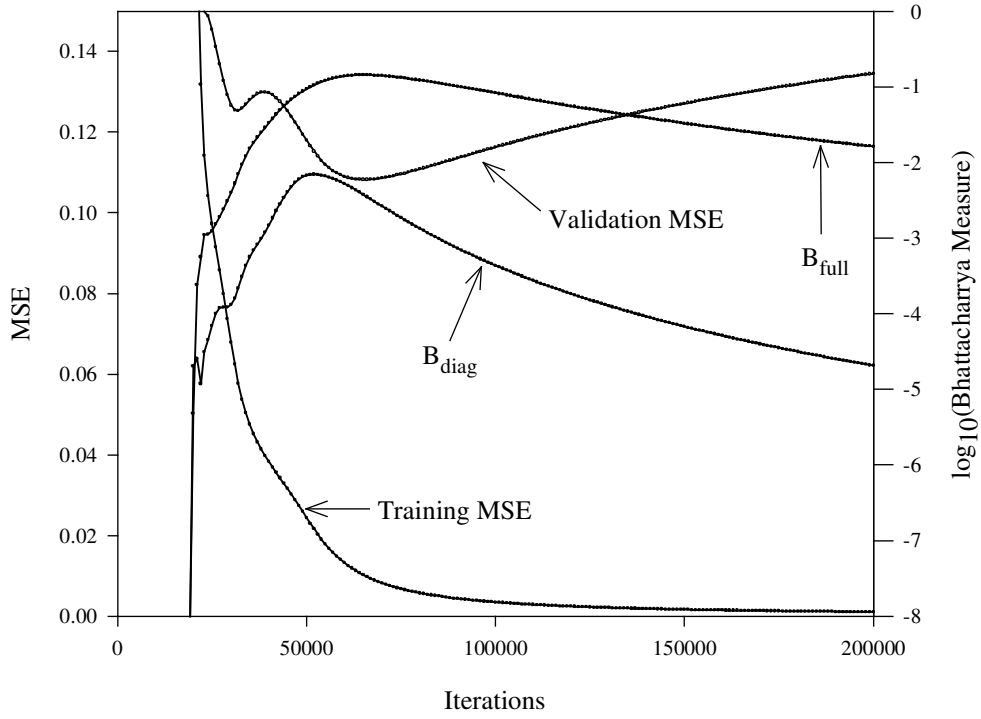

Figure 2: Training MSE, Validation MSE, B_{full} & B_{diag} versus Iteration Number for1:6:1 Network Trained on Sinewave Data. Target variance = 0.05.

where $\nabla \mathbf{f}$ is now the matrix of partial derivatives at each datum in the training set. The prediction error is now described by an N-dimensional Gaussian distribution where N is the number of training patterns. We have derived the Bhattacharyya overlap integral for two N-dimensional Gaussian distributions with arbitrary covariance matrices to be:

$$B_{full} = \int_{-\infty}^{+\infty} \sqrt{\phi_1(\mathbf{x})\phi_2(\mathbf{x})}\, d\mathbf{x}$$
$$= \sqrt{\frac{2^N}{\left|\boldsymbol{\Sigma}_1\right|^{\frac{1}{2}}\left|\boldsymbol{\Sigma}_2\right|^{\frac{1}{2}}\left|\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1}\right|}} \exp\left(\frac{-\gamma}{4}\right)$$

where:

$$\gamma = \mathbf{M}_1^T\boldsymbol{\Sigma}_1^{-1}\mathbf{M}_1 + \mathbf{M}_2^T\boldsymbol{\Sigma}_2^{-1}\mathbf{M}_2$$
$$-\left(\mathbf{M}_1^T\boldsymbol{\Sigma}_1^{-1} + \mathbf{M}_2^T\boldsymbol{\Sigma}_2^{-1}\right)$$
$$\times\left(\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1}\right)\left(\boldsymbol{\Sigma}_1^{-1}\mathbf{M}_1 + \boldsymbol{\Sigma}_2^{-1}\mathbf{M}_2\right)$$

and $\mathbf{M}_{1,2}$ are the mean vectors and $\boldsymbol{\Sigma}_{1,2}$ are the covariance matrices. Since this form of the prediction uses a full covariance matrix – as opposed to a diagonal approximation – we refer to this measure as B_{full}. We thus have two measures of similarity, B_{diag} and B_{full} with which to compare the target and prediction PDFs for a network.

4 - Experimental Results

To validate our approach we have considered regression rather than classification problems since a classifier has only to perform well in the region near the decision surface to give favourable results. For a regression problem, on the other hand, the prediction has to be acceptable over the *entire* domain. Thus regression poses a stiffer test of generalisation.

We have use conventional MLP networks with sigmoidal non-linearities and linear output nodes. For the results on the problem reported here our evaluation methodology has been to generate a small training set and a rather large validation set which approximates the universe of the underlying parent distribution. Both training and validation data were corrupted by zero-mean, additive Gaussian noise. We have trained using standard error backpropagation with a very small, constant learning rate in order to densely map the error space. In parallel with the training we calculate the mean-squared error (MSE) over the validation set as well as the Bhattacharyya measures (B_{diag} and B_{full}) and plot these measures as a function of training iteration number. If the

Bhattacharyya measure truly does predict the point of best generalisation we should observe a maximum in the Bhattacharyya metric corresponding to the minimum in the validation MSE. We believe this methodology to be a straightforward and unambiguous means of validating optimal generalisation algorithms.

Figure 2 shows the result for training a 1:6:1 network on the function:

$$f(x) = \sin(2\pi x)$$

where we have used a training set of 10 members and validation set of 10,000. Both training and validation sets have been corrupted with Gaussian noise of a variance of 0.05. Our choice of such a complex network to map such a simple problem was to guarantee eventual overfitting of the training data.

From Figure 2 training and validation MSEs follow a well-worn pattern with the validation curve displaying a minimum at around 60,000 iterations. The B_{diag} measure has a peak close to the point of best generalisation but due to the approximations in its formulation it is slightly in error. The Bhattacharyya measure calculated with the full covariance matrix (B_{full}), however, has a peak which exactly coincides with the point of minimum validation MSE and is thus able to correctly identify the point of optimal generalisation. Interestingly, it seems from the plot of validation MSE that the error surface for this training problem exhibits a false minimum at around 30,000 iterations which the B_{full} measure completely ignores.

We have repeated this experiment several tens of times restarting the training with different random initial weight values and the B_{full} measure consistently predicts the point of optimal generalisation every time although the value of MSE predicted by B_{diag} varies somewhat.

We emphasise the only role of the validation set in this work was as an independent estimator of generalisation ability.

5 - Discussion & Future Work

We believe the significance of this work is that we have been able to demonstrate the determination of the optimal stopping point *using the training data alone*. In many respects, the present technique is an analytic formulation of cross-validation. Thus it is of particular value on problems where the amount of data available is small and one would like to use it all for training rather than partition it into separate training and validation sets.

It is appropriate to ask how we have managed to seemingly circumvent the bias-variance dilemma [1]: clearly the additional information which has been employed is the errors on the target values - recall Figure 1 and the related discussion. This error information can be obtained either by direct observation or can be estimated by careful consideration of the measurement process involved in gathering the target values.

From a practical point, the calculation of both forms of the Bhattacharyya similarity measure require the inversion of an inverse covariance matrix on the network parameters. Since the weights in a network can be expected to interact quite strongly we should expect the inverse covariance matrix to be rank-deficient; in practice this matrix is often quite seriously so. We have thus computed the inverse by approximating it with a matrix of (lower but) full rank using singular value decomposition (SVD). In the course of this work we have observed that for certain combinations of dataset and architecture the matrix inversion stage is taxing even for SVD.

As to future areas for the research, we have demonstrated here that for a single training run we can accurately predict the optimal stopping point. Neural network practitioners usually train a range of networks repeatedly and select what appears to be the 'best' network. We plan to address the task of exploring whether the Bhattacharyya measure can be used select the best model from a range of possibilities - in essence, the system identification problem from control engineering.

Finally, although we have here employed the Bhattacharyya measure to find the optimal stopping point for a training algorithm which minimises the MSE over the training set, we believe it should be possible to devise a training algorithm which *directly* maximises the Bhattacharyya integral as a means of locating the optimal generalisation point for a given neural architecture.

6 - Conclusions

In this paper we have discussed the desirability of selecting a data model with due regard to the errors on the data and particularly the strategy of obtaining maximal similarity between the target and prediction PDFs. We have described the properties of the Bhattacharyya measure which make it suitable for this PDF comparison.

We have described the construction of a prediction PDF for a neural network and presented experimental results demonstrating that the Bhattacharyya measure can indeed accurately locate the optimal stopping point for conventional backpropagation learning of a univariate regression problem.

7 - Acknowledgements

The financial support of EPSRC under the 'Neural Networks – The Key Questions' programme is gratefully acknowledged.

References

[1] "Neural Networks and the Bias/variance Dilemma" - S.Geman, E.Bienenstock & R.Doursat, *Neural Computation* **4**(1), 1 (1992)

[2] "Neural Networks for Pattern Recognition" - C.M.Bishop, Oxford University Press (1995)

[3] "A Practical Bayesian Framework for Backpropagation Networks" - D.J.C.McKay, *Neural Computation* **4**(3), 448 (1992)

[4] "Bayesian Training of Backpropagation Networks by the Hybrid Monte-Carlo Method" - R.M.Neal, Technical Report CRG-TR-92-1, Dept. of Computer Science, University of Toronto, Canada

[5] "On a Measure of Divergence Between Two Statistical Populations Defined by Their Probability Distributions" – A.Bhattacharyya, *Bull. Calcutta. Math.* Soc. **35**, 99 (1943)

[6] "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency-Coded Data" - F.J.Aherne, N.A.Thacker & P.I.Rockett, *Kybernetika* **32**(4), 363 (1997)

[7] "Decision Rules Based on Distance for Problems of Fit, Two Samples and Estimation" - K.Matusita, *Ann. Mathematical Statistics* **26**, 631 (1955)

[8] "A New Look at the Statistical Model Identification" - H.Akaike*, IEEE Transactions on Automatic Control,* **AC-19**(6), 716 (1974)

[9] "Feature Tracking and Motion Classification Using a Switchable Model Kalman Filter" - A.J.Lacey, N.A.Thacker & N.L.Seed, *Proc. British Machine Vision Conference 94*, York, September 1994

[10] "B-fitting: A statistical estimation technique with automatic parameter selection" - N.A.Thacker, D.Prendergast & P.I.Rockett, *Proc. British Machine Vision Conference 96*, Edinburgh, September 1996