Journal of the Korean Statistical Society (2002), **31**: *1, pp 63–75*

Efficient Markov Chain Monte Carlo for Bayesian Analysis of Neural Network Models[†]

Paul E. Green[1], Changha Hwang[2] and Sangbock Lee[3]

Abstract

Most attempts at Bayesian analysis of neural networks involve hierarchical modeling. We believe that similar results can be obtained with simpler models that require less computational effort, as long as appropriate restrictions are placed on parameters in order to ensure propriety of posterior distributions. In particular, we adopt a model first introduced by Lee (1999) that utilizes an improper prior for all parameters. Straightforward Gibbs sampling is possible, with the exception of the bias parameters, which are embedded in nonlinear sigmoidal functions. In addition to the problems posed by nonlinearity, direct sampling from the posterior ditributions of the bias parameters is compounded due to the duplication of hidden nodes, which is a source of multimodality. In this regard, we focus on sampling from the marginal posterior distribution of the bias parameters with Markov chain Monte Carlo methods that combine traditional Metropolis sampling with a slice sampler described by Neal (1997, 2001). The methods are illustrated with data examples that are largely confined to the analysis of nonparametric regression models.

Keywords: Feedforward neural networks (FFNN), MCMC sampling, Bayesian prediction.

1. Introduction

Feedforward neural networks have recently received considerable attention due to their successful use in a wide variety of statistical applications including regression and classification problems. Ripley (1996) covers a broad range of

[†] University Research Program supported by the Ministry of Information & Communications in South Korea

[1] Department of Statistics, Seoul National University, Seoul 151-747, Korea

[2] Department of Statistical Information, Catholic University of Taegu, Taegu 712-702, Korea

[3] Department of Statistical Information, Catholic University of Taegu, Taegu 712-702, Korea

topics involving neural networks and discusses various applications related to pattern recognition. More recently, the Bayesian method has gained acceptance as a plausible framework for neural network modeling largely due to the compatibility of the prior specification with the resulting posterior predictive performance.

Initially, the major impediment to Bayesian implementation of neural networks was the calculation of high-dimensional integrals. Bishop (1995b) and MacKay (1995) provide detailed accounts of Bayesian methods for neural networks and use Gaussian approximations to summarize information contained in posterior distributions. Neal (1996) considers a Bayesian method that uses a Markov chain Monte Carlo (MCMC) sampler to sample directly from posterior distributions, and argues for the use of appropriate priors as the number of hidden nodes increases.

Most Bayesian attempts of neural networks involve hierarchical models at several levels using proper priors. The priors are usually chosen, not based on any real prior information, but for mathematical tractibility and ease of computation. Another justification for using proper priors is that propriety of posterior distributions is guaranteed. Still, these models tend to be complicated with numerous hyperparameters, making estimation computationally expensive. In addition, assessing priors is not always straightforward since eventually at some level hyperparameters are required to be fixed in advance.

The focus of this work is to consider Bayesian methods for neural network models that require less computational effort, yet produce desirable results, using MCMC sampling. In this regard, we adopt a model first introduced by Lee (1999) that uses improper priors for all parameters. Then all that is needed to ensure propriety of posterior distributions is to require that the columns of the design matrix are linearly independent, and that the bias parameters are bounded.

Even though the posterior distribution is guaranteed to be proper by placing restrictions on the bias parameters, fitting neural networks using the output from Markov chain samplers requires careful consideration. First, the exchangeability of hidden nodes is a source of multimodality, as discussed in Müller and Rios Insua (1998). Constructing an efficient sampler that can jump between multiple modes is an important task. In addition, neural network models are highly nonlinear, due to the bias parameters which are embedded in sigmoidal functions.

Due to these considerations, we propose sampling from the marginal posterior distribution of the bias parameters by combining traditional Metropolis sampling with a slice sampler as described by Neal (1997, 2001). The slice sampler has been shown to possess the ability to jump between various modes, making it a useful sampler for neural network models. As regards the bias parameters, a Metropolis sampler is used to sample from the intercept parameters jointly, while the slice sampler samples from the slope parameters one at a time. Conditional on the slope parameters, the intercept parameters are jointly close to normal, making Metropolis a reasonable choice. Conversely, conditional on the intercept parameters, the slope parameters can be multimodal and ill-behaved, in which case we use the slice sampler.

Several works have appeared that consider a model selection procedure whereby the number of hidden nodes is determined by incorporating a reversible jump Markov chain (Green, 1995) into the MCMC sampler (see, for example, Rios Insua and Müller 1998, or Holmes and Mallick 1998). In this work we assume that the model architecture is fixed in advance, and focus mainly on the development of efficient MCMC sampling techniques.

In Section 2 the neural network model which defines the likelihood and the improper prior are described. The procedure for using the sample generated from an MCMC sampler to perform nonparametric regression is addressed in Section 3. Some calculations that are required to implement MCMC are presented in Section 4 and the MCMC sampling methods are described in Section 5. Finally, data examples illustrate the use of feedforward neural networks for conducting nonparametric regression analysis in Section 6. In what follows, the terms *distribution* and *density* are used interchangeably, and random variables are denoted by uppercase letters while observed realizations are denoted by lowercase letters.

2. A Bayesian Neural Network Model for Nonparametric Regression

The focus of this work is to develop an efficient MCMC sampler to perform nonparametric regression using neural network models. We consider a feedforward neural network model with one hidden layer and M hidden nodes. In particular, the model is

$$Y_i = \beta_0 + \sum_{j=1}^{M} \beta_j \, \Phi(\mathbf{x}_i^T \gamma_j) + \epsilon_i \qquad \epsilon_i \sim N(0, \sigma^2), \tag{2.1}$$

where Y_i , $i = 1, \ldots, N$ is the observed response, $\mathbf{x}_i^T = (1, x_{i1}, x_{i2}, \ldots, x_{ip})$ are the explanatory variables, $\beta^T = (\beta_0, \beta_1, \ldots, \beta_M)$ are called the *weight* parameters,

$\gamma_j^T = (\gamma_{j0}, \gamma_{j1}, \ldots, \gamma_{jp})$ are called the *bias* parameters, and

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \, dy$$

is the *activation* function. Any smooth sigmoidal function can be used as the activation function. We choose the standard normal cumulative distribution function to force narrow alternatives for the bias parameters. Other commonly used activation functions include the standard logistic cumulative distribution function and the hyperbolic tangent function (see, for example, Bishop 1995a).

In a Bayesian framework it is necessary to specify the likelihood and prior distributions. Recently, most Bayesian methods that have appeared for feedforward neural networks involve several levels of hierarchical modeling. In an attempt to alleviate the computational burden that is associated with assessing models with large numbers of parameters, Lee (1999) developed a simpler model based on an improper prior. From (2.1) the likelihood is

$$Y_i \,|\, \beta, \gamma, \sigma^2 \sim N(\lambda_i, \sigma^2),$$

where the normal mean satisfies

$$\lambda_i = \beta_0 + \sum_{j=1}^{M} \beta_j \, \Phi(\mathbf{x}_i^T \gamma_j)$$

and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_M)$. The prior, however, is improper for all parameters and is given by

$$p(\beta, \gamma, \sigma^2) \propto (\sigma^2)^{-1}.$$

Later, some restrictions will be imposed on γ to ensure that the posterior distribution is proper. However, the prior for β is flat and improper, and the prior for σ^2 is the usual improper prior for a scale parameter.

3. Using the Output from an MCMC Sampler

The output from an MCMC sampler will be used to perform nonparametric regression using a feedforward neural network with one hidden layer and M hidden nodes. The highly nonlinear nature of a neural network model induced by the bias parameters γ makes estimation and running an efficient MCMC sampler a challenging task. Nevertheless, once a sample

$$(\beta^t, \gamma^t), \; t = 1, \ldots, K \; \text{ from the posterior } \; p(\beta, \gamma \,|\, y)$$

has been generated, predictive means can be calculated using

$$(\hat{\lambda}_i \mid y) = \frac{1}{K}\sum_{t=1}^{K}(\lambda_i \mid \beta = \beta^t, \gamma = \gamma^t, y).$$

The method used for generating the sample is the focus of Section 5 and is illustrated in the data examples in Section 6. First, however, we consider some posterior calculations required for implementing the MCMC sampler.

4. Some Posterior Calculations

By combining the normal likelihood with the improper prior it is straightforward to show that the posterior distribution can be written as

$$\begin{aligned} &p(\beta, \gamma, \sigma^2 \mid y) \\ &\propto \ (\sigma^2)^{-(N/2+1)} \exp\left[-\tfrac{1}{2\sigma^2}\left[(N-M-1)s^2 + (\beta - \hat{\beta})^T Z^T Z(\beta - \hat{\beta})\right]\right], \end{aligned} \tag{4.1}$$

where $\hat{\beta}$ and s^2 resemble the usual estimates from least squares regression.

$$\hat{\beta} = (Z^T Z)^{-1} Z^T y, \qquad s^2 = \frac{1}{N-M-1}(y - Z\hat{\beta})^T (y - Z\hat{\beta}),$$

and the design matrix Z is given by

$$Z = \begin{bmatrix} 1 & \Phi(\mathbf{x}_1^T \gamma_1) & \dots & \Phi(\mathbf{x}_1^T \gamma_M) \\ \vdots & & & \vdots \\ 1 & \Phi(\mathbf{x}_N^T \gamma_1) & \dots & \Phi(\mathbf{x}_N^T \gamma_M) \end{bmatrix}.$$

Note that the γ parameters are embedded in the matrix Z and that $\hat{\beta}$ and s^2 also depend on γ through Z.

By holding γ and σ^2 fixed in (4.1), the conditional distribution for β is multivariate normal

$$\beta \mid \gamma, \sigma^2, y \sim N_{M+1}(\hat{\beta}, \sigma^2 (Z^T Z)^{-1}).$$

Next, we integrate out β from (4.1) to get

$$p(\gamma, \sigma^2 \mid y) \ \propto \ |Z^T Z|^{-1/2} \, (\sigma^2)^{-((N-M-1)/2+1)} \exp\left[-\frac{(N-M-1)s^2}{2\sigma^2}\right]. \tag{4.2}$$

Then, by holding γ fixed in (4.2), the conditional for σ^2 is scaled inverse chi-square

$$\sigma^2 \mid \gamma, y \sim \text{Inv-}\chi^2(N-M-1, s^2).$$

Thus, the conditionals for β and σ^2 are standard densities and are easy to sample from. The conditional for γ, however, is complicated and does not resemble any standard distribution. This should not be surprising since γ is embedded in a nonlinear sigmoidal function. In fact, depending on the data, it is the presence of the bias parameters and the duplication of hidden nodes that leads to multimodality in the posterior surface. In order to understand how γ influences estimation, it may be useful to examine the marginal posterior density.

The marginal posterior density of γ can be obtained by integrating out σ^2 from (4.2) to get

$$p(\gamma \,|\, y) \;\propto\; \frac{1}{|Z^T Z|^{1/2}\, s^{N-M-1}}.$$

Two conditions ensure that this distribution is proper. Let $c_1 > 0$ and $c_2 > 0$ be real constants. The first condition, $|Z^T Z| > c_1$, ensures that the columns of Z are linearly independent so that the design matrix is not ill-conditioned. The second condition, $|\gamma_j| < c_2$, for all elements of γ_j for each j, ensures that the integral is finite. These are exactly the same two conditions imposed by Lee. In practice, c_2 should be chosen large enough so that most of the posterior density is concentrated away from c_2. Centering the covariates helps to alleviate this problem (see, for example, the discussion in the simulated data example in Section 6.3).

5. Markov Chain Monte Carlo Methods

The following relation is useful for developing a scheme to sample from the posterior distribution

$$\begin{aligned} p(\beta, \gamma \,|\, y) &= \int p(\beta, \gamma, \sigma^2 \,|\, y)\, d\sigma^2 \\ &= \int p(\beta \,|\, \gamma, \sigma^2, y)\, p(\sigma^2 \,|\, \gamma, y)\, p(\gamma \,|\, y)\, d\sigma^2. \end{aligned}$$

The joint posterior density can be seen to be a mixture of a multivariate normal, a scaled inverse chi-square, and the marginal posterior density of γ. This joint density will be proper whenever the marginal density for γ is proper. Thus, the two conditions outlined in Section 4 are necessary to ensure propriety of the joint posterior distribution. The following sampling scheme is presented to implement the MCMC sampler.

Sample γ from $p(\gamma \,|\, y)$.

Sample σ^2 from Inv-$\chi^2(N{-}M{-}1, s^2)$.

Sample β from $N_{M+1}(\hat{\beta}, \sigma^2(Z^TZ)^{-1})$.

Repeat for the desired number of iterations.

The density $p(\gamma \mid y)$ can be very complicated with multiple modes. We combine traditional Metropolis sampling with a slice sampler (Neal 1997, 2001) to sample from $p(\gamma \mid y)$. In particular, Metropolis sampling is used to sample jointly from the intercept parameters γ_{j0}, and slice sampling is used to sample from the slope parameters $(\gamma_{j1}, \ldots, \gamma_{jp})$. The motivation for this strategy is that conditional on the slope parameters, the intercept parameters are jointly close to normal so that Metropolis sampling with a normal candidate should perform well. On the other hand, conditional on the intercept parameters, the slope parameters can have densities with multiple modes, which makes slice sampling attractive, due to its ability to sample from multimodal densities. In Section 6 some examples illustrate these points.

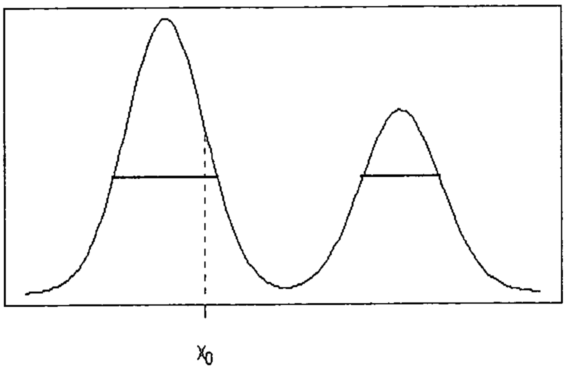

FIGURE 5.1. *Slice sampling*

The slice sampler is an MCMC sampler that works by sampling uniformly from the support under a density function, say $f(x)$. It works particularly well for multimodal densities and, unlike Metropolis sampling, for example, it does not require specification of a candidate distribution. In general, the idea behind slice sampling from a multimodal density can be illustrated with reference to Figure 5.1. The user first chooses a real starting value, say X_0. A real value y, is drawn uniformly from $(0, f(X_0))$ (shown as the dashed line), which defines the *horizontal slice S* (the bold horizontal line). An interval around X_0 is found that

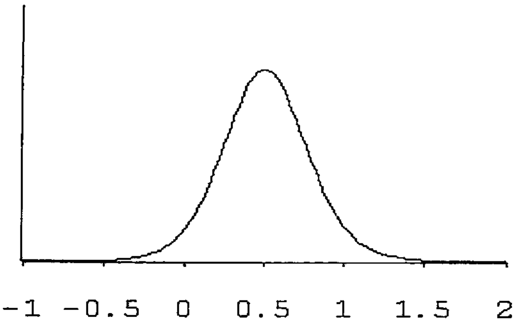

FIGURE 6.1. *Metropolis to sample from* $p(\gamma_{10}|\gamma_{11}, y)$

contains much or all of the slice. The new point X_1 is drawn from the part of the slice contained within the interval. Neal (1997, 2001) provides several algorithms for sampling the new value X_1.

6. Data Examples

6.1. The Bates and Watts data

The following data appear in Bates and Watts (1988). This is a simple example of a nonlinear model, but illustrates the advantages of a slice sampler when sampling from the slope parameters. Consider a feedforward neural network with one hidden node ($M = 1$) and one explanatory variable. The model is

$$Y_i = \beta_0 + \beta_1\, \Phi(\gamma_{10} + \gamma_{11}x_i) + \epsilon_i\,, \qquad \epsilon_i \sim N(0, \sigma^2).$$

To get a sample from $p(\gamma_1 \,|\, y) = p(\gamma_{10}, \gamma_{11} \,|\, y)$, Metropolis sampling is used to sample from the intercept $p(\gamma_{10} \,|\, \gamma_{11}, y)$, and slice sampling is used to sample from $p(\gamma_{11} \,|\, \gamma_{10}, y)$. Figure 6.1 is a plot of the intercept $p(\gamma_{10} \,|\, \gamma_{11}, y)$ when $\gamma_{11} = 1.4$. Note that when the slope is fixed, this density is close to normal and Metropolis with a normal candidate can be expected to perform well. Figure 6.2 is a side-by-side plot of the slope $p(\gamma_{11} \,|\, \gamma_{10}, y)$ when $\gamma_{10} = 0.5$. This density appears fairly well behaved, except at 0, where there is a large spike. The right plot in Figure 6.2 is a close-up at 0 of the left plot. The reason there is no density in a small neighborhood of 0 is because this is where the Z matrix is ill-conditioned. We use a slice sampler to overcome the difficulty of sampling from this density. When considering the slope parameters in problems of higher dimension, with larger numbers of hidden nodes, there can be several disjoint regions where there

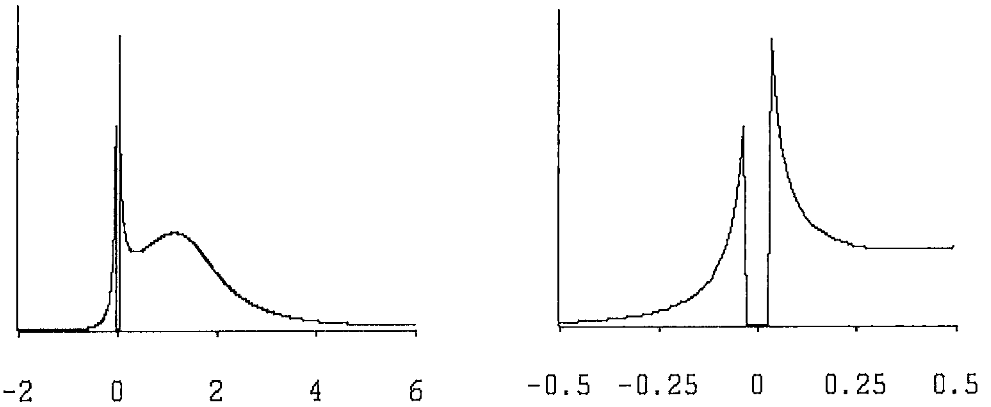

FIGURE 6.2. *Slice sampler samples from $p(\gamma_{11}|\gamma_{10}, y)$. Right plot:Close-up at* 0.

is no density. Finally, Figure 6.3 is a plot of the data and the fitted regression curve obtained by averaging the regression function over the sampled values. For this problem we sampled 10,000 values and discarded the initial 3,000 for burn-in, for an effective sample size of 7,000.

6.2. The Galaxy data

The following data originally appeared in Buta (1987) and a subset using the first 80 observations are kindly provided by Müller (personal communication). Müller and Rios Insua (1998) fit Bayesian feedforward neural network models to these data, illustrating some issues of multimodality that occur when the number of nodes M is too large. They discuss other concerns associated with traditional MCMC samplers, such as Metropolis, that are related to the difficulty of sampling from multimodal densities. In other work, Rios Insua and Müller (1998) use reversible jump Markov chain Monte Carlo (Green, 1995) as a form of model selection to automatically choose the number of hidden nodes, in order to alleviate some difficulties associated with multimodality. In this paper, we focus on sampling from multimodal densities using a slice sampler. Reversible jump Markov chain Monte Carlo could be combined with slice sampling to improve efficiency, but is not considered here. In particular, the model for a feedforward neural network with $M = 2$ hidden nodes is

$$Y_i = \beta_0 + \beta_1 \Phi(\gamma_{10} + \gamma_{11} x_i) + \beta_2 \Phi(\gamma_{20} + \gamma_{21} x_i) + \epsilon_i , \qquad \epsilon_i \sim N(0, \sigma^2).$$

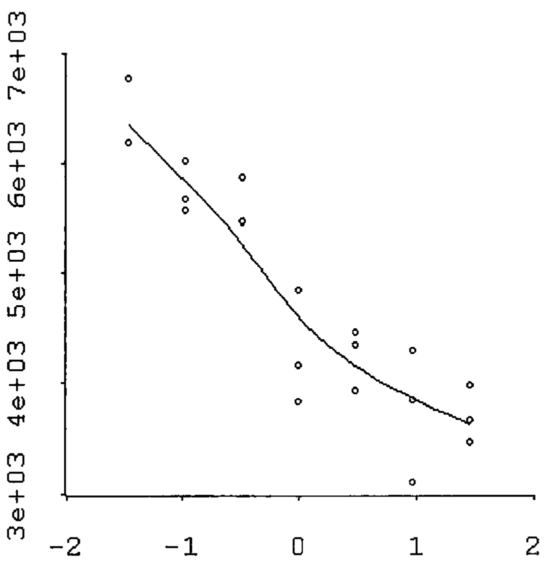

FIGURE 6.3. *Bates and Watts data with regression curve*

We use a Metropolis sampler to sample from $(\gamma_{10}, \gamma_{20})$ jointly, and a slice sampler to sample from γ_{11} and γ_{21} separately. The data and fitted regression curve are given in Figure 6.4. Due to the exchangeability of hidden nodes, the identifiability conditions imposed by Müller and Rios Insua of ordering the slope parameters $(\gamma_{11} < \gamma_{21})$ is followed.

6.3. The simulated data

In order to assess the ability of the proposed Bayesian neural network model with an improper prior to perform nonparametric regression based on the underlying assumptions, 50 observations were simulated from the model

$$Y_i = -2 + 10\,\Phi(-1 - 4x_i) - 5\,\Phi(2 - x_i) + 8\,\Phi(1.5 + 4x_i) + \epsilon_i\,, \qquad \epsilon_i \sim N(0, 0.3^2).$$

The data were then fit to a model with $M = 3$ hidden nodes. Two conditions that ensure propriety of the posterior distribution are that $|Z^T Z| > c_1 > 0$ and $|\gamma| < c_2$ for some real constants c_1 and $c_2 > 0$. For the first condition we adopt Lee's suggestion of setting $c_1 = 1/N = 1/50$. For the second condition the absolute values of all bias parameters are restricted to be less than $c_2 = 12$. Note that in all data examples, the explanatory variables are scaled to have mean equal to 0 and standard deviation equal to 1. This facilitates choosing c_2 since $\mathbf{x}_i^T \gamma_j$ is

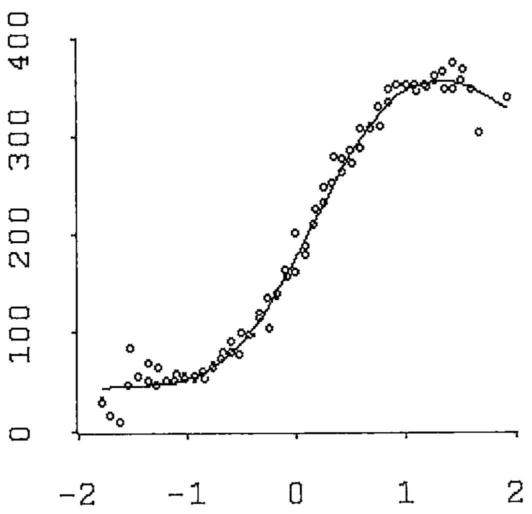

FIGURE 6.4. *Galaxy data with regression curve*

an argument to the activation function. The data and resulting fitted regression curve are presented in Figure 6.5.

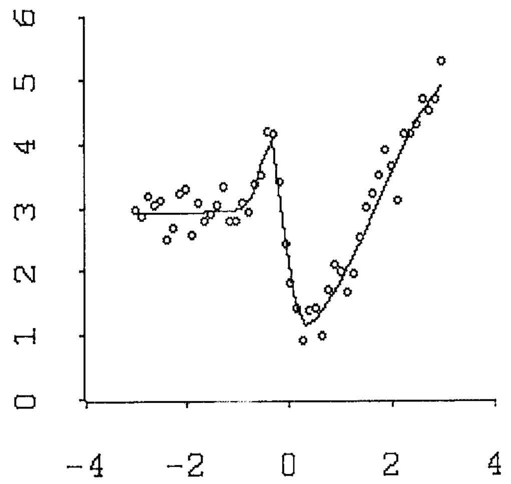

FIGURE 6.5. *Simulated data with regression curve*

7. Concluding Remarks

The Bayesian method for feedforward neural networks provides a useful framework for performing nonparametric regression using MCMC sampling. The idea proposed here combines traditional Metropolis sampling with slice sampling. Both samplers are incorporated into the MCMC scheme in a way that takes advantage of their strengths. The slice sampler is particularly suited for multimodal densities. The Metropolis sampler with a normal candidate is used when conditional densities are close to normal. In general, slice samplers can be multidimensional, but only the one-dimensional case is considered here.

Acknowledgement

This work is part of the University Research Program supported by the Ministry of Information and Communication in South Korea. The authors wish to thank the editor and two anonymous referees for providing constructive suggestions and helpful comments.

References

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*, Wiley, New York.

Bishop, C. M. (1995a). *Neural Networks for Pattern Recognition*, Oxford University Press, 126-127.

Bishop, C. M. (1995b). "Bayesian methods for neural networks", Technical Report NCRG95009. Department of Computer Science and Applied Mathematics, Aston University, Birmingham, U.K.

Buta, R. (1987)."The structure and dynamics of ringed galaxies, III", *Astrophysical Journal Supplement Series*, **64**, 1-37.

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination", *Biometrika*, **82**, 711-732.

Holmes, C. C. and Mallick, B. K. (1998). "Bayesian radial basis functions of variable dimension", *Neural Computation*, **10**, 1217-1233.

Lee, H. K. H. (1999). *Model Selection and Model Averaging for Neural Networks*, Ph.D. thesis, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, U.S.A.

MacKay, D. J. C. (1995). "Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks", review paper to appear in *Network, IDPP.*

Müller, P. and Rios Insua, D. (1998)."Issues in Bayesian analysis of neural network models", *Neural Computation*, **10**, 749-770.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Springer-Verlag, New York.

Neal, R. M. (1997). "Markov chain Monte Carlo methods based on 'slicing' the density function", Technical Report No. 9722, Department of Statistics, University of Toronto, Toronto, Canada.

Neal, R. M. (2001). "Slice sampling", to appear in *The Annals of Statistics.*

Rios Insua, D. and Müller, P. (1998). "Feedforward neural networks for nonparametric regression" in *Practical Nonparametric and Semiparametric Bayesian Statistics* (Dey, D., Müller, P. and D. Sinha, eds.), 181-194, Springer-Verlag, New York.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press.