Appendix A

Details of the Implementation

This appendix contains mathematical details regarding the Bayesian neural network implementation described in Chapter 3, and used for the evaluations in Chapter 4. Some features of this implementation are not discussed in these earlier chapters, but are described here for completeness.

Due to the variety of network architectures accommodated, it is necessary here to use a notation that is more systematic, albeit more cumbersome, than that which is used elsewhere. This notation is summarized on the next page.

A.1 Specifications

This section defines the class of network models that are implemented by the software, and explains how they are specified, in an abstract way. (For the detailed syntax of network specifications, and other non-mathematical details, see the documentation that comes with the software.)

A.1.1 Network architecture

The multilayer perceptron networks that this implementation supports consist of a layer of input units, zero or more hidden layers with tanh activation function, and a layer of output units. Units in each hidden layer are connected to units in the preceding hidden layer and to units in the input

Values associated with units

v_i^I Value of ith input unit, before the offset is added

v_i^ℓ Value of ith hidden unit in layer ℓ, before the offset is added

v_i^O Value of ith output unit

u_i^ℓ Value of the input to the ith hidden unit

Parameters of the network

t_i^I Offset for ith input unit

t_i^ℓ Offset for ith hidden unit in layer ℓ

b_i^ℓ Bias for ith unit in hidden layer ℓ

b_i^O Bias for ith output unit

$w_{i,j}^{I,O}$ Weight from ith input unit to jth output unit

$w_{i,j}^{I,\ell}$ Weight from ith input unit to jth unit in hidden layer ℓ

$w_{i,j}^{\ell-1,\ell}$ Weight from ith unit in hidden layer $\ell-1$ to jth unit in hidden layer ℓ

$w_{i,j}^{\ell,O}$ Weight from ith unit in hidden layer ℓ to the jth output unit

Hyperparameters defining priors for parameters

σ_t^I Common sigma for offsets of input units

σ_t^ℓ Common sigma for offsets of units in hidden layer ℓ

σ_b^ℓ Common sigma for biases of units in hidden layer ℓ

σ_b^O Common sigma for biases of output units

$\sigma_w^{I,O}$ Common sigma for weights from input units to output units

$\sigma_w^{I,\ell}$ Common sigma for weights from input units to units in hidden layer ℓ

$\sigma_w^{\ell-1,\ell}$ Common sigma for weights from units in hidden layer $\ell-1$ to units in hidden layer ℓ

$\sigma_w^{\ell,O}$ Common sigma for weights from units in hidden layer ℓ to output units

$\sigma_{w,i}^{I,O}$ Sigma for weights from input unit i to output units

$\sigma_{w,i}^{I,\ell}$ Sigma for weights from input unit i to units in hidden layer ℓ

$\sigma_{w,i}^{\ell-1,\ell}$ Sigma for weights from unit i in hidden layer $\ell-1$ to units in hidden layer ℓ

$\sigma_{w,i}^{\ell,O}$ Sigma for weights from unit i in hidden layer ℓ to output units

$\sigma_{a,i}^O$ Sigma adjustment for weights and biases into output unit i

$\sigma_{a,i}^\ell$ Sigma adjustment for weights and biases into unit i in hidden layer ℓ

layer. Units in the output layer are connected to units in the hidden layers and to units in the input layer. Each of these connections has an associated weight, used to form a weighted sum of inputs to a unit along incoming connections. Each unit in the hidden and output layers has a bias, which is added to this weighted sum of inputs. Each unit in the input and hidden layers has an offset, which is added to its output. Any of these sets of parameters (associated with a particular layer, or pair of layers) may be missing in any particular network, producing the same effect as if their values were zero.

The following formulas define the outputs, v_i^O, of a network for given values of the inputs, v_i^I. Note that the interpretation of these outputs is determined by the data model, described next.

$$u_i^\ell = b_i^\ell + \sum_k w_{k,i}^{I,\ell}(v_k^I + t_k^I) + \sum_k w_{k,i}^{\ell-1,\ell}(v_k^{\ell-1} + t_k^{\ell-1}) \quad \text{(A.1)}$$

$$v_i^\ell = \tanh(u_i^\ell) \quad \text{(A.2)}$$

$$v_i^O = b_i^O + \sum_k w_{k,i}^{I,O}(v_k^I + t_k^I) + \sum_\ell \sum_k w_{k,i}^{\ell,O}(v_k^\ell + t_k^\ell) \quad \text{(A.3)}$$

Here, and subsequently, the summations are over all units in the appropriate layer, or over all hidden layers (for ℓ). The number of layers and the numbers of units in each layer are part of the architecture specification, but these numbers are not given symbols here. The term in the equation for u_i^ℓ involving layer $\ell-1$ is omitted for the first hidden layer.

A.1.2 Data models

Networks are normally used to define models for the conditional distribution of a set of "target" values given a set of "input" values. There are three sorts of models, corresponding to three sorts of targets — real-valued targets (a "regression" model), binary-valued targets (a "logistic regression" model), and "class" targets taking on values from a (small) finite set (a generalized logistic regression, or "softmax" model). For regression and logistic regression models, the number of target values is equal to the number of network outputs. For the softmax model, there is only one target, with the number of possible values for this target being equal to the number of network outputs.

The distributions for real-valued targets, y_j, in a case with inputs v_i^I may be modeled by independent Gaussian distributions with means given by the corresponding network outputs, and with standard deviations given by the hyperparameters σ_j — the "noise levels" for the targets. The probability density for a target given the associated inputs and the network parameters

is then

$$P(y_j \mid \text{inputs, parameters}) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-(y_j - v_j^O)^2/2\sigma_j^2\right) \quad \text{(A.4)}$$

Alternatively, each case, c, may have its own set of standard deviations, $\sigma_{j,c}$, with the corresponding precisions, $\tau_{j,c} = \sigma_{j,c}^{-2}$, being given Gamma distributions with means of τ_j and shape parameter α_2 (called this for reasons that will become clear later):

$$P(\tau_{j,c} \mid \tau_j) = \frac{(\alpha_2/2\tau_j)^{\alpha_2/2}}{\Gamma(\alpha_2/2)} \tau_{j,c}^{\alpha_2/2-1} \exp\left(-\tau_{j,c}\alpha_2/2\tau_j\right) \quad \text{(A.5)}$$

The previous case corresponds to the degenerate Gamma distribution with $\alpha_2 = \infty$. Otherwise, integrating out $\tau_{j,c}$ gives a t-distribution for the target with α_2 "degrees of freedom":

$$P(y_j \mid \text{inputs, parameters}) = \frac{\Gamma((\alpha_2+1)/2)}{\Gamma(\alpha_2/2)\sqrt{\pi\alpha_2}\sigma_j} \left[1 + (y_j - v_j^O)^2/\alpha_2\sigma_j^2\right]^{-(\alpha_2+1)/2} \quad \text{(A.6)}$$

For a logistic regression model, the probability that a binary-valued target, y_j, has the value 1 is given by

$$P(y_j = 1 \mid \text{inputs, parameters}) = \left[1 + \exp(-v_j^O)\right]^{-1} \quad \text{(A.7)}$$

For a softmax model, the probability that a class target, y, has the value j is given by

$$P(y = j \mid \text{inputs, parameters}) = \exp(v_j^O) \Big/ \sum_k \exp(v_k^O) \quad \text{(A.8)}$$

A.1.3 Prior distributions for parameters and hyperparameters

The prior distributions for the parameters of a network are defined in terms of hyperparameters. Conceptually, this implementation provides for one hyperparameter for every parameter, but these lowest-level hyperparameters are not explicitly represented. Mid-level hyperparameters control the distribution of a group of low-level hyperparameters that are all of one type and all associated with the same source unit. High-level (or "common") hyperparameters control the distribution of the mid-level hyperparameters, or of the low-level hyperparameters for parameter types with no mid-level hyperparameters. The same three-level scheme is used for noise levels in regression models.

These hyperparameters are represented in terms of "sigma" values, σ, but their distributions are specified in terms of the corresponding "precisions", $\tau = \sigma^{-2}$, which are given Gamma distributions. The top-level mean

is given by a "width" value associated with the parameter type. The shape parameters of the Gamma distributions are determined by "alpha" values associated with each type of parameter. An alpha value of infinity concentrates the entire distribution on the mean, effectively removing one level from the hierarchy. The sigma for a weight may also be multiplied by an "adjustment" value that is associated with the destination unit.

This gives the following generic scheme for the priors for weights:

$$P(\tau_w) = \frac{(\alpha_{w,0}/2\omega_w)^{\alpha_{w,0}/2}}{\Gamma(\alpha_{w,0}/2)} \tau_w^{\alpha_{w,0}/2-1} \exp\left(-\tau_w \alpha_{w,0}/2\omega_w\right) \quad \text{(A.9)}$$

$$P(\tau_{w,i} \mid \tau_w) = \frac{(\alpha_{w,1}/2\tau_w)^{\alpha_{w,1}/2}}{\Gamma(\alpha_{w,1}/2)} \tau_{w,i}^{\alpha_{w,1}/2-1} \exp\left(-\tau_{w,i} \alpha_{w,1}/2\tau_w\right) \quad \text{(A.10)}$$

$$P(\tau_{a,j}) = \frac{(\alpha_a/2)^{\alpha_a/2}}{\Gamma(\alpha_a/2)} \tau_{a,j}^{\alpha_a/2-1} \exp\left(-\tau_{a,j} \alpha_a/2\right) \quad \text{(A.11)}$$

For weights from input units to output units, for example, τ_w will equal $\tau_w^{I,O} = [\sigma_w^{I,O}]^{-2}$, and similarly for $\tau_{w,i}$, while $\tau_{a,j}$ will equal $[\sigma_{a,i}^O]^{-2}$. The top-level precision value, ω_w, is derived from the "width" value specified for this type of weight. The positive (possibly infinite) values $\alpha_{w,0}$ and $\alpha_{w,1}$ are also part of the prior specification for input to output weights, while α_a is a specification associated with the output units (note that in this case the "width" value is fixed at one, as freedom to set it would be redundant).

The distribution for a weight from unit i of one layer to unit j of another layer may be Gaussian with mean zero and standard deviation given by $\sigma_{w,i}\sigma_{a,j} = [\tau_{w,i}\tau_{a,j}]^{-1/2}$. That is:

$$P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) = \frac{1}{\sqrt{2\pi}\sigma_{w,i}\sigma_{a,j}} \exp\left(-w_{i,j}^2/2\sigma_{w,i}^2\sigma_{a,j}^2\right) \quad \text{(A.12)}$$

(Here, $w_{i,j}$ represents, for example, $w_{i,j}^{I,O}$, in which case $\sigma_{w,i}$ represents $\sigma_{w,i}^{I,O}$ and $\sigma_{a,j}$ represents $\sigma_{a,j}^O$.)

Alternatively, each individual weight may have its own "sigma", with the corresponding precision having a Gamma distribution with mean $\tau_{w,i}\tau_{a,j}$ and shape parameter given by $\alpha_{w,2}$. The previous case corresponds to the degenerate distribution with $\alpha_{w,2} = \infty$. Otherwise, we can integrate over the individual precisions and obtain t-distributions for each weight:

$$P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) \quad \text{(A.13)}$$

$$= \frac{\Gamma((\alpha_{w,2}+1)/2)}{\Gamma(\alpha_{w,2}/2)\sqrt{\pi\alpha_{w,2}}\,\sigma_{w,i}\sigma_{a,j}} \left[1 + w_{i,j}^2/\alpha_{w,2}\sigma_{w,i}^2\sigma_{a,j}^2\right]^{-(\alpha_{w,2}+1)/2}$$

The same scheme is used for biases, except that for them there are no mid-level hyperparameters. We have

$$P(\tau_b) = \frac{(\alpha_{b,0}/2\omega_b)^{\alpha_{b,0}/2}}{\Gamma(\alpha_{b,0}/2)} \tau_b^{\alpha_{b,0}/2-1} \exp\left(-\tau_b \alpha_{b,0}/2\omega_b\right) \quad \text{(A.14)}$$

where τ_b might, for example, be $\tau_b^O = [\sigma_b^O]^{-2}$, etc.

The distribution of the biases is then either

$$P(b_i \mid \sigma_b, \sigma_{a,i}) = \frac{1}{\sqrt{2\pi}\sigma_b \sigma_{a,i}} \exp\left(-b_i^2/2\sigma_b^2\sigma_{a,i}^2\right) \quad \text{(A.15)}$$

if $\alpha_{b,1} = \infty$, or if not

$$P(b_i \mid \sigma_b, \sigma_{a,i}) = \frac{\Gamma((\alpha_{b,1}+1)/2)}{\Gamma(\alpha_{b,1}/2)\sqrt{\pi\alpha_{b,1}}\,\sigma_b\sigma_{a,i}} \left[1 + b_i^2/\alpha_{b,1}\sigma_b^2\sigma_{a,i}^2\right]^{-(\alpha_{b,1}+1)/2} \quad \text{(A.16)}$$

For the offsets added to input and hidden unit values, there are no mid-level hyperparameters, and neither are "adjustments" used. We have

$$P(\tau_t) = \frac{(\alpha_{t,0}/2\omega_t)^{\alpha_{t,0}/2}}{\Gamma(\alpha_{t,0}/2)} \tau_t^{\alpha_{t,0}/2-1} \exp\left(-\tau_t \alpha_{t,0}/2\omega_t\right) \quad \text{(A.17)}$$

where τ_t might, for example, be $\tau_t^I = [\sigma_t^I]^{-2}$, etc.

The distribution of the offsets is then either

$$P(t_i \mid \sigma_t) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-t_i^2/2\sigma_t^2\right) \quad \text{(A.18)}$$

if $\alpha_{t,1} = \infty$, or if not

$$P(t_i \mid \sigma_t) = \frac{\Gamma((\alpha_{t,1}+1)/2)}{\Gamma(\alpha_{t,1}/2)\sqrt{\pi\alpha_{t,1}}\,\sigma_t} \left[1 + t_i^2/\alpha_{t,1}\sigma_t^2\right]^{-(\alpha_{t,1}+1)/2} \quad \text{(A.19)}$$

The scheme for noise levels in regression models is also similar, with τ_j, the precision for target j, being specified in terms of an overall precision, τ, as follows:

$$P(\tau) = \frac{(\alpha_0/2\omega)^{\alpha_0/2}}{\Gamma(\alpha_0/2)} \tau^{\alpha_0/2-1} \exp\left(-\tau\alpha_0/2\omega\right) \quad \text{(A.20)}$$

$$P(\tau_j \mid \tau) = \frac{(\alpha_1/2\tau)^{\alpha_1/2}}{\Gamma(\alpha_1/2)} \tau_j^{\alpha_1/2-1} \exp\left(-\tau_j\alpha_1/2\tau\right) \quad \text{(A.21)}$$

where ω, α_0, and α_1 are parts of the noise specification. A third alpha (α_2) is needed for the final specification of the noise in individual cases, as described in the Section A.1.2.

A.1.4 Scaling of priors

The top-level mean precisions used in the preceding hierarchical priors (the ω values) may simply be taken as specified (actually, what is specified is the corresponding "width", $\omega^{-1/2}$). Alternatively, for connection weights only (not biases and offsets), the ω for values of one type may be scaled automatically, based on the number of source units that feed into each destination unit via connections of this type. This scaling is designed to produced sensible results as the number of source units goes to infinity, while all other specifications remain unchanged.

The theory behind this scaling concerns the convergence of sums of independent random variables to "stable distributions" (Feller 1966, Samorodnitsky and Taqqu 1994), as discussed in Chapter 2. The symmetric stable distributions are characterized by a width parameter and an index, α, in the range (0,2]. If $X_1, \ldots, X_n$ are independent and each has the same symmetric stable distribution of index α, then $(X_1 + \cdots + X_n)/n^{1/\alpha}$ has this same stable distribution as well. The stable distribution with index 2 is the Gaussian. The sums of all random variables with finite variance converge to the Gaussian, along with some others. Typically, random variables whose moments are defined up to but not including α converge to the stable distribution with index α, for $\alpha < 2$.

This leads to the following scaling rules for producing ω based on the specified base precision, ω_0, the number of source units, n, and the relevant α value (see below):

$$\omega = \begin{cases} \omega_0 n & \text{for } \alpha = \infty \\ \omega_0 n\alpha/(\alpha-2) & \text{for } \alpha > 2 \\ \omega_0 n \log n & \text{for } \alpha = 2 \quad \text{(but fudged to } \omega_0 n \text{ if } n < 3\text{)} \\ \omega_0 n^{2/\alpha} & \text{for } \alpha < 2 \end{cases} \tag{A.22}$$

Here, α is $\alpha_{w,2}$ if that is finite, and is otherwise $\alpha_{w,1}$. The scheme doesn't really work if both $\alpha_{w,1}$ and $\alpha_{w,2}$ are finite. When $\alpha = 2$, the scaling produces convergence to the Gaussian distribution, but with an unusual scale factor, as the t-distribution with $\alpha = 2$ is in the "non-normal" domain of attraction of the Gaussian distribution.

A.2 Conditional distributions for hyperparameters

Implementation of Gibbs sampling for hyperparameters requires sampling from the conditional distribution for one hyperparameter given the values of the other hyperparameters and of the network parameters. This section describes how this is done.

A.2.1 Lowest-level conditional distributions

The simplest conditional distributions to sample from are those for "sigma" hyperparameters that directly control a set of network parameters. This will be the situation for the lowest-level sigmas, as well as for higher-level sigmas when the lower-level sigmas are tied exactly to this higher-level sigma (i.e. when the "alpha" shape parameter for their distribution is infinite). The situation is analogous for sigma values relating to noise levels in regression models, except that the errors in training case are what is modeled, rather than the network parameters.

In general, we will have some hyperparameter $\tau = \sigma^{-2}$ that has a Gamma prior, with shape parameter we will call α, and with mean ω (which may be a higher-level hyperparameter). The purpose of τ is to specify the precisions for the independent Gaussian distributions of n lower-level quantities, z_i. In this situation, the conditional distribution for τ will be given by the following proportionality:

$$P(\tau \mid \{z_i\}, \ldots) \propto \tau^{\alpha/2-1} \exp(-\tau\alpha/2\omega) \cdot \prod_i \tau^{1/2} \exp(-\tau z_i^2/2) \quad \text{(A.23)}$$

$$\propto \tau^{(\alpha+n)/2-1} \exp\left(-\tau(\alpha/\omega + \sum_i z_i^2)/2\right) \quad \text{(A.24)}$$

The first factor in equation (A.23) derives from the prior for τ, the remaining factors from the effect of τ on the probabilities of the z_i. The result is a Gamma distribution that can be sampled from by standard methods (Devroye 1986).

When the distributions of the z_i are influenced by "adjustments", $\tau_{a,i}$, the above formula is modified as follows:

$$P(\tau \mid \{z_i\}, \{\tau_{a,i}\}, \ldots)$$

$$\propto \tau^{(\alpha+n)/2-1} \exp\left(-\tau(\alpha/\omega + \sum_i \tau_{a,i} z_i^2)/2\right) \quad \text{(A.25)}$$

Gibbs sampling for the adjustments themselves is done in similar fashion, using the weighted sum of squares of parameters influenced by the adjustment, with the weights in this case being the precisions associated with each parameter.

A.2.2 Higher-level conditional distributions

Sampling from the conditional distribution for a sigma hyperparameter that controls a set of lower-level sigmas is more difficult, but can be done in the most interesting cases using rejection sampling. This method is generally adequate, but not completely satisfactory. I plan to replace it with a better scheme soon.

Assume that we wish to sample from the distribution for a precision hyperparameter τ, which has a higher-level Gamma prior specified by α_0 and ω, and which controls the distributions of lower-level hyperparameters, τ_i, that have independent Gamma distributions with shape parameter α_1 and mean τ. The conditional distribution for τ is then given by the following proportionality:

$$P(\tau \mid \{\tau_i\}, \ldots)$$

$$\propto \tau^{\alpha_0/2-1} \exp(-\tau\alpha_0/2\omega) \cdot \prod_i \tau^{-\alpha_1/2} \exp(-\tau_i\alpha_1/2\tau) \quad \text{(A.26)}$$

$$\propto \tau^{(\alpha_0-n\alpha_1)/2-1} \exp\left(-\tau\alpha_0/2\omega - (\alpha_1 \sum_i \tau_i) / 2\tau\right) \quad \text{(A.27)}$$

Defining $\gamma = 1/\tau$, we get:

$$P(\gamma \mid \{\tau_i\}, \ldots)$$

$$\propto \tau^2 P(\tau \mid \{\tau_i\}, \ldots) \quad \text{(A.28)}$$

$$\propto \tau^{(\alpha_0-n\alpha_1)/2+1} \exp\left(-\tau\alpha_0/2\omega - (\alpha_1 \sum_i \tau_i) / 2\tau\right) \quad \text{(A.29)}$$

$$\propto \gamma^{(n\alpha_1-\alpha_0)/2-1} \exp\left(-\gamma(\alpha_1 \sum_i \tau_i) / 2\right) \cdot \exp\left(-\alpha_0/2\omega\gamma\right) \quad \text{(A.30)}$$

The first part of this has the form of a Gamma distribution for γ, provided $n\alpha_1 > \alpha_0$; the last factor lies between zero and one. If $n\alpha_1 > \alpha_0$, we can therefore obtain a value from the distribution for γ by repeatedly sampling from the Gamma distribution with shape parameter $n\alpha_1 - \alpha_0$ and mean $(n\alpha_1 - \alpha_0)/(\alpha_1 \sum_i \tau_i)$ until the value of γ generated passes an acceptance test, which it does with probability $\exp(-\alpha_0/2\omega\gamma)$. We may hope that the probability of rejection will be reasonably low if α_0 is small, which is typical.

In some contexts, the values τ_i are not explicitly represented, and must themselves be found by sampling using the method of the previous section.

A.3 Calculation of derivatives

To use the hybrid Monte Carlo method, we must be able to calculate the derivatives of the log of the posterior probability density for the parameter values, which are found by summing the derivatives of the log likelihood and of the log of the prior probability density of the parameter values. This section details how this is done.

A.3.1 Derivatives of the log prior density

For fixed values of the explicitly-represented hyperparameters, one can easily obtain the derivatives of the log of the prior probability with respect to the network weights and other parameters. Generically, if $\alpha_{w,2} = \infty$, we get, from equation (A.12), that

$$\frac{\partial}{\partial w_{i,j}} \log P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) = -\frac{w_{i,j}}{\sigma_{w,i}^2 \sigma_{a,j}^2} \tag{A.31}$$

while otherwise, we get, from equation (A.14), that

$$\frac{\partial}{\partial w_{i,j}} \log P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) = -\frac{\alpha_{w,2}+1}{\alpha_{w,2}\sigma_{w,i}^2\sigma_{a,j}^2} \frac{w_{i,j}}{\left[1 + w_{i,j}^2/\alpha_{w,2}\sigma_{w,i}^2\sigma_{a,j}^2\right]} \tag{A.32}$$

Similar formulas for derivatives with respect to the biases are obtained from equations (A.15) and (A.16) and for derivatives with respect to the offsets from equations (A.18) and (A.19).

A.3.2 Log likelihood derivatives with respect to unit values

The starting point for calculating the derivatives of the log likelihood with respect to the network parameters is to calculate the derivative of the log likelihood due to a particular case with respect to the network outputs. For the regression model with $\alpha_2 = \infty$, we get from equation (A.4) that

$$\frac{\partial}{\partial v_j^O} \log P(y \mid v_j^O) = -\frac{y_j - v_j^O}{\sigma_j^2} \tag{A.33}$$

When α_2 is finite, we get from equation (A.6) that

$$\frac{\partial}{\partial v_j^O} \log P(y \mid v_j^O) = -\frac{\alpha_2+1}{\alpha_2\sigma_j^2} \frac{y_j - v_j^O}{\left[1 + (y_j - v_j^O)^2/\alpha_2\sigma_j^2\right]} \tag{A.34}$$

For the model of binary targets given by equation (A.7), we get the following, after some manipulation:

$$\frac{\partial}{\partial v_j^O} \log P(y \mid v_j^O) = y_j - \left[1 + \exp(-v_j^O)\right]^{-1} \tag{A.35}$$

$$= y_j - P(y_j = 1 \mid v_j^O) \tag{A.36}$$

For the many-way "softmax" classification model of equation (A.8), we get the following (where $\delta(y, j)$ is one if $y = j$ and zero otherwise):

$$\frac{\partial}{\partial v_j^O} \log P(y \mid \{v_k^O\}) = \delta(y, j) - \frac{\exp(v_j^O)}{\sum_k \exp(v_k^O)} \tag{A.37}$$

$$= \delta(y, j) - P(y = j \mid \{v_k^O\}) \qquad \text{(A.38)}$$

Let L be the log likelihood due to a single training case — that is, $L = \log P(y \mid \text{inputs, parameters}) = \log P(y \mid \text{outputs})$. Once the derivatives of L with respect to the output unit values are known, its derivatives with respect to the values of the hidden and input units can be found by the standard backpropagation method. From equations (A.1), (A.2), and (A.3):

$$\frac{\partial L}{\partial v_i^\ell} = \sum_j w_{i,j}^{\ell,O} \frac{\partial L}{\partial v_j^O} + \sum_j w_{i,j}^{\ell,\ell+1} \frac{\partial L}{\partial u_j^{\ell+1}} \qquad \text{(A.39)}$$

$$\frac{\partial L}{\partial u_i^\ell} = \left(1 - [v_i^\ell]^2\right) \frac{\partial L}{\partial v_j^\ell} \qquad \text{(A.40)}$$

$$\frac{\partial L}{\partial v_i^I} = \sum_j w_{i,j}^{I,O} \frac{\partial L}{\partial v_j^O} + \sum_\ell \sum_j w_{i,j}^{I,\ell} \frac{\partial L}{\partial u_j^\ell} \qquad \text{(A.41)}$$

In (A.39), the second term is not present when ℓ is the last hidden layer.

A.3.3 Log likelihood derivatives with respect to parameters

The derivatives of L with respect to the network parameters (with explicitly represented noise sigmas fixed) are obtained using the derivatives with respect to unit values and unit inputs found in the previous section, as follows:

$$\frac{\partial L}{\partial b_i^O} = \frac{\partial L}{\partial v_i^O} \qquad \text{(A.42)}$$

$$\frac{\partial L}{\partial b_i^\ell} = \frac{\partial L}{\partial u_i^\ell} \qquad \text{(A.43)}$$

$$\frac{\partial L}{\partial t_i^\ell} = \frac{\partial L}{\partial v_i^\ell} \qquad \text{(A.44)}$$

$$\frac{\partial L}{\partial t_i^I} = \frac{\partial L}{\partial v_i^I} \qquad \text{(A.45)}$$

$$\frac{\partial L}{\partial w_{i,j}^{\ell,O}} = \frac{\partial L}{\partial v_j^O} (v_i^\ell + t_i^\ell) \qquad \text{(A.46)}$$

$$\frac{\partial L}{\partial w_{i,j}^{\ell-1,\ell}} = \frac{\partial L}{\partial u_j^\ell} (v_i^{\ell-1} + t_i^{\ell-1}) \qquad \text{(A.47)}$$

$$\frac{\partial L}{\partial w_{i,j}^{I,\ell}} = \frac{\partial L}{\partial u_j^\ell} (v_i^I + t_i^I) \qquad \text{(A.48)}$$

$$\frac{\partial L}{\partial w_{i,j}^{I,O}} = \frac{\partial L}{\partial v_j^O} (v_i^I + t_i^I) \qquad \text{(A.49)}$$

The derivatives found in this way for each training case are summed over the full training set, and added to the derivatives with respect to the log prior density, to give the derivatives with respect to the log posterior probability density, which control the hybrid Monte Carlo dynamics.

A.4 Heuristic choice of stepsizes

Stepsizes for dynamical trajectory computations and for Metropolis updates are heuristically chosen based on the values of the training inputs and the current values of the hyperparameters. These stepsize choices are made on the assumption that the system is near equilibrium, moving about in an approximately Gaussian hump of the posterior distribution. If the axes of this hump were aligned with the coordinate axes, the optimal stepsize along each axis would be in the vicinity of the standard deviation along that axis. Since the axes of the bowl may not be aligned with the coordinate axes, the actual stepsizes may have to be less than this. On the other hand, the estimates used are in some respects conservative. Any overall adjustment of the stepsizes to account for these factors must be done manually by the user.

Estimates of the posterior standard deviations along the axes are based on estimates of the second derivatives of the log posterior probability density along the axes. These second derivatives are estimated using estimates of the second derivatives of the log likelihood with respect to the values of units in the network.

Letting L be the log likelihood for a single training case, we get the following for real-valued targets, with $\alpha_2 = \infty$, using equation (A.33):

$$-\frac{\partial^2 L}{\partial (v_j^O)^2} = \frac{1}{\sigma_j^2} \tag{A.50}$$

while for finite α_2, we get from equation (A.34) that

$$-\frac{\partial^2 L}{\partial (v_j^O)^2} = \frac{\alpha_2 + 1}{\alpha_2 \sigma_j^2} \left[\left(1 + \frac{(v_j^O)^2}{\alpha_2 \sigma_j^2}\right)^{-1} + \frac{2(v_j^O)^2}{\alpha_2 \sigma_j^2} \left(1 + \frac{(v_j^O)^2}{\alpha_2 \sigma_j^2}\right)^{-2} \right] \tag{A.51}$$

This is estimated by its maximum value, which occurs at $v_j^O = 0$:

$$-\frac{\partial^2 L}{\partial (v_j^O)^2} \approx \frac{\alpha_2 + 1}{\alpha_2 \sigma_j^2} \tag{A.52}$$

For binary-valued targets, equation (A.36) gives

$$-\frac{\partial^2 L}{\partial (v_j^O)^2} = \frac{1}{[1 + \exp(v_j^O)][1 + \exp(-v_j^O)]} \approx \frac{1}{4} \tag{A.53}$$

Again, the estimate is based on the maximum possible value, which occurs when $v_j^O = 0$.

We get a similar estimate for a class target, using equation (A.38):

$$-\frac{\partial^2 L}{\partial (v_j^O)^2} = \frac{\exp(v_j^O)}{\sum_k \exp(v_k^O)} \left[1 - \frac{\exp(v_j^O)}{\sum_k \exp(v_k^O)} \right] \approx \frac{1}{4} \tag{A.54}$$

These estimates for the second derivatives of L with respect to the outputs are propagated backward to give estimates for the second derivatives of L with respect to the values of hidden and input units.

When doing this backward propagation, we need an estimate of the second derivative of L with respect to the summed input to a tanh hidden unit, given its second derivative with respect to the unit's output. Letting the hidden unit output be $v = \tanh(u)$, we have

$$\frac{d^2 L}{du^2} = \frac{d}{du} \left[(1 - v^2) \frac{dL}{dv} \right] \tag{A.55}$$

$$= (1 - v^2)^2 \frac{d^2 L}{dv^2} - 2v(1 - v^2) \frac{dL}{dv} \tag{A.56}$$

$$\approx (1 - v^2)^2 \frac{d^2 L}{dv^2} \approx \frac{d^2 L}{dv^2} \tag{A.57}$$

The first approximation assumes that since $2v(1-v^2)(dL/dv)$ may be either positive or negative, its effects will (optimistically) cancel when averaged over the training set. Since v is not known, the second approximation above takes the maximum with respect to v. The end result is that we just ignore the fact that the hidden unit input is passed through the tanh function.

The backward propagation also ignores any interactions between multiple connections from a unit. Since the stepsizes chosen are not allowed to depend on the actual values of the network parameters, the magnitude of each weight is taken to be equal to the corresponding sigma hyperparameter, multiplied by the destination unit adjustment, if present. This gives the following generic estimate:

$$\frac{\partial^2 L}{\partial (v_i^S)^2} \approx \sum_D \sum_j (\sigma_{w,i}^{S,D} \, \sigma_{a,j}^D)^2 \frac{\partial^2 L}{(v_j^D)^2} \tag{A.58}$$

Here, S is the source layer, D goes through the various layers receiving connections from S, $\sigma_{w,i}^{S,D}$ is the hyperparameter controlling weights to layer D out of unit i in S, and $\sigma_{a,j}^D$ is the sigma adjustment for unit j in D.

The second derivative of L with respect to a weight, $w_{i,j}^{S,D}$, can be expressed as follows:

$$\frac{\partial^2 L}{\partial (w_{i,j}^{S,D})^2} = (v_i^S)^2 \frac{\partial^2 L}{\partial (v_j^D)^2} \tag{A.59}$$

When the weight is on a connection from an input unit, $v_i^S = v_i^I$ is the ith input for this training case, which is known. If the weight is on a connection from a hidden unit, $(v_i^S)^2$ is assumed to be one, the maximum possible value.

Second derivatives with respect to biases and offsets are simply equal to the second derivatives with respect to the associated unit values.

These heuristic estimates for the second derivatives of L due to each training case with respect to the various network parameters are summed for all cases in the training set. To these are added estimates of the second derivatives of the log prior probability density with respect to each parameter, giving estimates of the second derivatives of the log posterior density.

For the second derivative of the log prior density with respect to weight $w_{i,j}$, we have

$$-\frac{\partial^2}{\partial w_{i,j}^2} \log P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) = \frac{1}{\sigma_{w,i}^2 \sigma_{a,j}^2} \tag{A.60}$$

if α_2 is infinite, while for finite α_2, we use an estimate analogous to equation (A.52):

$$-\frac{\partial^2}{\partial w_{i,j}^2} \log P(w_{i,j} \mid \sigma_{w,i}, \sigma_{a,j}) \approx \frac{\alpha_2 + 1}{\alpha_2 \sigma_{w,i}^2 \sigma_{a,j}^2} \tag{A.61}$$

Biases and offsets are handled similarly.

Finally, the stepsize used for a parameter is the reciprocal of the square root of minus the estimated second derivative of the log posterior with respect to that parameter.

A.5 Rejection sampling from the prior

In addition to the Monte Carlo implementation based on Markov chain sampling, a simple Monte Carlo procedure using rejection sampling has also been implemented. This procedure is very inefficient; it is intended for use only as a means of checking the correctness of the Markov chain implementations.

The rejection sampling procedure is based on the idea of producing a sample from the posterior by generating networks from the prior, and then accepting some of these networks with a probability proportional to the likelihood (for the given training data) of the generated parameter and

hyperparameter values. For data models with discrete targets, this idea can be implemented directly, as the likelihood is the probability of the targets in the training set, which can be no more than one. For regression models, the likelihood is the probability density of the targets, which can be greater than one, making its direct use as an acceptance probability invalid. If the noise levels for the targets are fixed, however, the likelihood is bounded, and can be used as the acceptance probability after rescaling. For a Gaussian noise model (equation (A.4)), this is accomplished by simply ignoring the factors of $1 / \sqrt{2\pi}\sigma_j$ in the likelihood; the analogous procedure can be used for noise from a t-distribution (equation (A.6)).

When the noise levels are variable hyperparameters, a slightly more elaborate procedure must be used, in which the noise levels are not generated from the prior, but rather from the prior multiplied by a bias factor that gives more weight to higher precisions (lower noise). This bias factor is chosen so that when it is cancelled by a corresponding modification to the acceptance probability, these probabilities end up being no greater than one.

Specifically, the overall noise precision, τ, and the noise precisions for individual targets, the τ_j, are sampled from Gamma distributions obtained by modifying the priors of equations (A.20) and (A.21) as follows:

$$f(\tau) \propto \tau^{nm/2} P(\tau) \tag{A.62}$$

$$\propto \tau^{(\alpha_0+nm)/2-1} \exp\left(-\tau\alpha_0/2\omega\right) \tag{A.63}$$

$$f(\tau_j \mid \tau) \propto \tau_j^{n/2} P(\tau_j \mid \tau) \tag{A.64}$$

$$\propto \tau^{-(\alpha_1+n)/2} \tau_j^{(\alpha_1+n)/2-1} \exp\left(-\tau_j\alpha_1/2\tau\right) \tag{A.65}$$

Here, n is the number of training cases and m is the number of targets. The resulting joint sampling density is

$$f(\tau, \{\tau_j\}) = f(\tau) \prod_{j=1}^{m} f(\tau_j \mid \tau) \propto P(\tau, \{\tau_j\}) \prod_{j=1}^{m} \tau_j^{n/2} \tag{A.66}$$

Since this sampling density is biased in relation to the prior by the factor $\prod_{j=1}^{m} \tau_j^{n/2}$, when constructing the acceptance probability we must multiply the likelihood by the inverse of this factor, $\prod_{j=1}^{m} \tau_j^{-n/2} = \prod_{c=1}^{n} \prod_{j=1}^{m} \sigma_j$. This cancels the factors of $1/\sigma_j$ in the target probabilities of equations (A.4) and (A.6), leaving an acceptance probability which is bounded, and can be adjusted to be no more than one by ignoring the remaining constant factors.

Appendix B

Obtaining the software

The implementation of Bayesian learning for neural networks described in Appendix A is available free of charge for research and educational purposes. This implementation is written in C, and currently is designed for use only on Unix systems. It does not require any special graphics or user interface environment. The software also does not use any special Unix facilities, but it is nevertheless likely that various modifications would be required in order for it to run in some other environment, and I cannot undertake to provide assistance with any such conversion.

Potential users should note that this software is intended to support research in Bayesian neural network learning, not as a tool for routine data analysis.

The software is available over the Internet, via my World Wide Web home page, at URL

`http://www.cs.utoronto.ca/~radford/`

It can also be obtained by anonymous ftp to `ftp.cs.utoronto.ca`, directory `pub/radford`. Look in the `README` file there for further instructions.

Unfortunately, it is difficult to say for how long the above instructions will remain valid. If you encounter difficulties, you should be able to find an up-to-date link at Springer-Verlag's Web page, which is currently located at URL

`http://www.springer-ny.com/`

Bibliography

ACKLEY, D. H., HINTON, G. E., AND SEJNOWSKI, T. J. (1985) "A learning algorithm for Boltzmann machines", *Cognitive Science*, vol. 9, pp. 147-169.

ANDERSEN, H. C. (1980) "Molecular dynamics simulations at constant pressure and/or temperature", *Journal of Chemical Physics*, vol. 72, pp. 2384-2393.

BALDI, P. AND CHAUVIN, Y. (1991) "Temporal evolution of generalization during learning in linear networks",*Neural Computation*, vol. 3, pp. 589-603.

BARNETT, V. (1982) *Comparative Statistical Inference*, Second Edition, New York: John Wiley.

BERGER, J. O. (1985) *Statistical Decision Theory and Bayesian Analysis*, New York: Springer-Verlag.

BERNARDO, J. M. AND SMITH, A. F. M. (1994) *Bayesian Theory*, New York: John Wiley.

BISHOP, C. M. (1995) *Neural Networks for Pattern Recognition*, Oxford University Press.

BOX, G. E. P. AND TIAO, G. C. (1973) *Bayesian Inference in Statistical Analysis*, New York: John Wiley.

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. (1984) *Classification and Regression Trees*, Belmont, California: Wadsworth.

BRIDLE, J. S. (1989) "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition", in F. Fouglemann-Soulie and J. Héault (editors) *Neuro-computing: Algorithms, Architectures and Applications*, New York: Springer-Verlag.

BUNTINE, W. L. AND WEIGEND, A. S. (1991) "Bayesian back-propagation", *Complex Systems*, vol. 5, pp. 603-643.

CREUTZ, M. AND GOCKSCH, A. (1989) "Higher-order hybrid Monte Carlo algorithms", *Physical Review Letters*, vol. 63, pp. 9-12.

CYBENKO, G. (1989) "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, vol. 2, pp.303-314.

DUANE, S., KENNEDY, A. D., PENDLETON, B. J., AND ROWETH, D. (1987) "Hybrid Monte Carlo", *Physics Letters B*, vol. 195, pp. 216-222.

DEGROOT, M. H. (1970) *Optimal Statistical Decisions*, New York: McGraw-Hill.

DEVROYE, L. (1986) *Non-uniform Random Variate Generation*, New York: Springer-Verlag.

FALCONER, K. (1990) *Fractal Geometry: Mathematical Foundations and Applications*, Chichester: John Wiley.

FELLER, W. (1966) *An Introduction to Probability Theory and its Applications, Volume II*, New York: John Wiley.

FUNAHASHI, K. (1989) "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, vol. 2, pp. 183-192.

GELFAND, A. E. AND SMITH, A. F. M. (1990) "Sampling-based approaches to calculating marginal densities", *Journal of the American Statistical Association*, vol. 85, pp. 398-409.

GELMAN, A., CARLIN, J. B., STERN, H. S., AND RUBIN, D. B. (1995) *Bayesian Data Analysis*, London: Chapman & Hall.

GEMAN, S., BIENENSTOCK, E., AND DOURSAT, R. (1992) "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, vol. 4, pp. 1-58.

GEMAN, S. AND GEMAN, D. (1984) "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741.

GEYER, C. J. AND THOMPSON, E. A. (1995) "Annealing Markov chain Monte Carlo with applications to ancestral inference", *Journal of the American Statistical Association*, vol. 90, pp. 909-920.

GRENANDER, U. (1981) *Abstract Inference*, New York: John Wiley.

HARRISON, D. AND RUBINFELD, D. L. (1978) "Hedonic housing prices and the demand for clean air", *Journal of Environmental Economics and Management*, vol. 5, pp. 81-102.

HASTIE, T. J. AND TIBSHIRANI, R. J. (1990) *Generalized Additive Models*, London: Chapman & Hall.

HERTZ, J., KROGH, A., AND PALMER, R. G. (1991) *Introduction to the Theory of Neural Computation*, Redwood City, California: Addison-Wesley.

HINTON, G. E. AND VAN CAMP, D. (1993) "Keeping neural networks simple by minimizing the description length of the weights", *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, Santa Cruz, 1993*, pp. 5-13.

HORNIK, K., STINCHCOMBE, M., AND WHITE, H. (1989) "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, pp. 359-366.

HOROWITZ, A. M. (1991) "A generalized guided Monte Carlo algorithm", *Physics Letters B*, vol. 268, pp. 247-252.

JEFFREYS, W. H. AND BERGER, J. O. (1992) "Ockham's razor and Bayesian analysis", *American Scientist*, vol. 80, pp. 64-72. See also the discussion in vol. 80, pp. 212-214.

KENNEDY, A. D. (1990) "The theory of hybrid stochastic algorithms", in P. H. Damgaard, *et al.* (editors) *Probabilistic Methods in Quantum Field Theory and Quantum Gravity*, New York: Plenum Press.

KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. (1983) "Optimization by simulated annealing", *Science*, vol. 220, pp. 671-680.

LE CUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W., AND JACKEL, L. D. (1990) "Handwritten digit recognition with a back-propagation network", in D. S. Touretzky (editor) *Advances in Neural Information Processing Systems 2*, pp. 396-404, San Mateo, California: Morgan Kaufmann.

LIU, Y. (1994) "Robust parameter estimation and model selection for neural network regression", in J. D. Cowan, G. Tesuaro, and J. Alspector (editors) *Advances in Neural Information Processing Systems 6*, pp. 192-199. San Mateo, California: Morgan Kaufmann.

MACKAY, D. J. C. (1991) *Bayesian Methods for Adaptive Models*, Ph.D thesis, California Institute of Technology.

MACKAY, D. J. C. (1992a) "Bayesian interpolation", *Neural Computation*, vol. 4, pp. 415-447.

MACKAY, D. J. C. (1992b) "A practical Bayesian framework for backpropagation networks", *Neural Computation*, vol. 4, pp. 448-472.

MACKAY, D. J. C. (1992c) "The evidence framework applied to classification networks", *Neural Computation*, vol. 4, pp. 720-736.

MACKAY, D. J. C. (1994a) "Bayesian non-linear modeling for the energy prediction competition", *ASHRAE Transactions*, vol. 100, pt. 2, pp. 1053-1062.

MACKAY, D. J. C. (1994b) "Hyperparameters: Optimise, or integrate out?", in G. Heidbreder, editor, *Maximum Entropy and Bayesian Methods, Santa Barbara, 1993*, Dordrecht: Kluwer.

MACKENZIE, P. B. (1989) "An improved hybrid Monte Carlo method", *Physics Letters B*, vol. 226, pp. 369-371.

MCCULLAGH, P. AND NELDER, J. A. (1983) *Generalized Linear Models*, London: Chapman & Hall.

MARINARI, E. AND PARISI, G. (1992) "Simulated tempering: A new Monte Carlo Scheme", *Europhysics Letters*, vol. 19, pp. 451-458.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. (1953) "Equation of state calculations by fast computing machines", *Journal of Chemical Physics*, vol. 21, pp. 1087-1092.

NEAL, R. M. (1992a) "Bayesian mixture modeling", in C. R. Smith, G. J. Erickson, and P. O. Neudorfer (editors) *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis, Seattle, 1991*, Dordrecht: Kluwer Academic Publishers.

NEAL, R. M. (1992b) "Bayesian training of backpropagation networks by the hybrid Monte Carlo method", Technical Report CRG-TR-92-1, Dept. of Computer Science, University of Toronto.

NEAL, R. M. (1993a) "Bayesian learning via stochastic dynamics", in C. L. Giles, S. J. Hanson, and J. D. Cowan (editors), *Advances in Neural Information Processing Systems 5*, pp. 475-482, San Mateo, California: Morgan Kaufmann.

NEAL, R. M. (1993b) "Probabilistic inference using Markov Chain Monte Carlo methods", Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto. Available in Postscript via the World Wide Web, at URL `http://www.cs.utoronto.ca/~radford/`

NEAL, R. M. (1994) "An improved acceptance procedure for the hybrid Monte Carlo algorithm", *Journal of Computational Physics*, vol. 111, pp. 194-203.

NEAL, R. M. (in press) "Sampling from multimodal distributions using tempered transitions", to appear in *Statistics and Computing*.

PEITGEN, H.-O. AND SAUPE, D. (editors) (1988) *The Science of Fractal Images*, New York: Springer-Verlag.

PRESS, S. J. (1989) *Bayesian Statistics: Principles, Models, and Applications*, New York: John Wiley.

QUINLAN, R. (1993) "Combining instance-based and model-based learning", *Machine Learning: Proceedings of the Tenth International Conference, Amherst, Massachusetts, 1993*, Morgan Kaufmann.

RASMUSSEN, C. E. (1996) "A practical Monte Carlo implementation of Bayesian learning", in D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (editors) *Advances in Neural Information Processing Systems 8*, MIT Press.

RIPLEY, B D. (1987) *Stochastic Simulation*, New York: John Wiley.

RIPLEY, B. D. (1981) *Spatial Statistics*, New York: John Wiley.

RIPLEY, B. D. (1994a) "Flexible non-linear approaches to classification", in V. Cherkassky, J. H. Friedman, and H. Wechsler (editors) *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, pp. 105-126, Springer-Verlag.

RIPLEY, B. D. (1994b) "Neural networks and related methods for classification" (with discussion), *Journal of the Royal Statistical Society B*, vol. 56, pp. 409-456.

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

RISSANEN, J. (1986) "Stochastic complexity and modeling", *Annals of Statistics*, vol. 14, pp.1080-1100.

ROBERT, C. P. (1995) *The Bayesian Choice*, New York: Springer-Verlag.

ROSSKY, P. J., DOLL, J. D., AND FRIEDMAN, H. L. (1978) "Brownian dynamics as smart Monte Carlo simulation", *Journal of Chemical Physics*, vol. 69, pp. 4628-4633.

RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. (1986a) "Learning representations by back-propagating errors, *Nature*, vol. 323, pp. 533-536.

RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. (1986b) "Learning internal representations by error propagation", in D. E. Rumelhart and J. L. McClelland (editors) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, Cambridge, Massachusetts: MIT Press.

RUMELHART, D. E., MCCLELLAND, J. L., AND THE PDP RESEARCH GROUP (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, Cambridge, Massachusetts: MIT Press.

SAMORODNITSKY, G. AND TAQQU, M. S. (1994) *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, New York: Chapman & Hall.

SCHMITT, S. A. (1969) *Measuring Uncertainty: An Elementary Introduction to Bayesian Statistics*, Reading, Massachussets: Addison-Wesley.

SMITH, A. F. M. AND ROBERTS, G. O. (1993) "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods" (with discussion), *Journal of the Royal Statistical Society B*, vol. 55, pp. 3-23 (discussion, pp. 53-102).

STONE, M. (1974) "Cross-validatory choice and assessment of statistical predictions" (with discussion), *Journal of the Royal Statistical Society B*, vol. 36, pp. 111-147.

SZELISKI, R. (1989) *Bayesian Modeling of Uncertainty in Low-level Vision*, Boston: Kluwer.

SZU, H. AND HARTLEY, R. (1987) "Fast simulated annealing", *Physics Letters A*, vol. 122, pp. 157-162.

TIERNEY, L. (1994) "Markov chains for exploring posterior distributions", *Annals of Statistics*, vol. 22, pp. 1701-1762.

THODBERG, H. H. (1996) "A review of Bayesian neural networks with an application to near infrared spectroscopy", *IEEE Transactions on Neural Networks*, vol. 7, pp. 56-72.

TOUSSAINT, D. (1989) "Introduction to algorithms for Monte Carlo simulations and their application to QCD", *Computer Physics Communications*, vol. 56, pp. 69-92.

WAHBA, G. (1990) *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics.

WILLIAMS, C. K. I. AND RASMUSSEN, C. E. (1996) "Gaussian processes for regression", in D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (editors) *Advances in Neural Information Processing Systems 8*, MIT Press.

WOLPERT, D. H. (1993) "On the use of evidence in neural networks", in C. L. Giles, S. J. Hanson, and J. D. Cowan (editors), *Advances in Neural Information Processing Systems 5*, pp. 539-546, San Mateo, California: Morgan Kaufmann.

YOUNG, A. S. (1977) "A Bayesian approach to prediction using polynomials", *Biometrika*, vol. 64, pp. 309-317.

VAPNIK, V. (1982) *Estimation of Dependencies Based on Empirical Data*, translated by S. Kotz, New York: Springer-Verlag.

Index

Lecture Notes in Statistics

For information about Volumes 1 to 67 please contact Springer-Verlag

Vol. 68: M. Taniguchi, Higher Order Asymptotic Theory for Time Series Analysis. viii, 160 pages, 1991.

Vol. 69: N.J.D. Nagelkerke, Maximum Likelihood Estimation of Functional Relationships. V, 110 pages, 1992.

Vol. 70: K. Iida, Studies on the Optimal Search Plan. viii, 130 pages, 1992.

Vol. 71: E.M.R.A. Engel, A Road to Randomness in Physical Systems. ix, 155 pages, 1992.

Vol. 72: J.K. Lindsey, The Analysis of Stochastic Processes using GLIM. vi, 294 pages, 1992.

Vol. 73: B.C. Arnold, E. Castillo, J.-M. Sarabia, Conditionally Specified Distributions. xiii, 151 pages, 1992.

Vol. 74: P. Barone, A. Frigessi, M. Piccioni, Stochastic Models, Statistical Methods, and Algorithms in Image Analysis. vi, 258 pages, 1992.

Vol. 75: P.K. Goel, N.S. Iyengar (Eds.), Bayesian Analysis in Statistics and Econometrics. xi, 410 pages, 1992.

Vol. 76: L. Bondesson, Generalized Gamma Convolutions and Related Classes of Distributions and Densities. viii, 173 pages, 1992.

Vol. 77: E. Mammen, When Does Bootstrap Work? Asymptotic Results and Simulations. vi, 196 pages, 1992.

Vol. 78: L. Fahrmeir, B. Francis, R. Gilchrist, G. Tutz (Eds.), Advances in GLIM and Statistical Modelling: Proceedings of the GLIM92 Conference and the 7th International Workshop on Statistical Modelling, Munich, 13-17 July 1992. ix, 225 pages, 1992.

Vol. 79: N. Schmitz, Optimal Sequentially Planned Decision Procedures. xii, 209 pages, 1992.

Vol. 80: M. Fligner, J. Verducci (Eds.), Probability Models and Statistical Analyses for Ranking Data. xxii, 306 pages, 1992.

Vol. 81: P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search. xxiii, 526 pages, 1993.

Vol. 82: A. Korostelev and A. Tsybakov, Minimax Theory of Image Reconstruction. xii, 268 pages, 1993.

Vol. 83: C. Gatsonis, J. Hodges, R. Kass, N. Singpurwalla (Editors), Case Studies in Bayesian Statistics. xii, 437 pages, 1993.

Vol. 84: S. Yamada, Pivotal Measures in Statistical Experiments and Sufficiency. vii, 129 pages, 1994.

Vol. 85: P. Doukhan, Mixing: Properties and Examples. xi, 142 pages, 1994.

Vol. 86: W. Vach, Logistic Regression with Missing Values in the Covariates. xi, 139 pages, 1994.

Vol. 87: J. Müller, Lectures on Random Voronoi Tessellations.vii, 134 pages, 1994.

Vol. 88: J. E. Kolassa, Series Approximation Methods in Statistics. Second Edition, ix, 183 pages, 1997.

Vol. 89: P. Cheeseman, R.W. Oldford (Editors), Selecting Models From Data: AI and Statistics IV. xii, 487 pages, 1994.

Vol. 90: A. Csenki, Dependability for Systems with a Partitioned State Space: Markov and Semi-Markov Theory and Computational Implementation. x, 241 pages, 1994.

Vol. 91: J.D. Malley, Statistical Applications of Jordan Algebras. viii, 101 pages, 1994.

Vol. 92: M. Eerola, Probabilistic Causality in Longitudinal Studies. vii, 133 pages, 1994.

Vol. 93: Bernard Van Cutsem (Editor), Classification and Dissimilarity Analysis. xiv, 238 pages, 1994.

Vol. 94: Jane F. Gentleman and G.A. Whitmore (Editors), Case Studies in Data Analysis. viii, 262 pages, 1994.

Vol. 95: Shelemyahu Zacks, Stochastic Visibility in Random Fields. x, 175 pages, 1994.

Vol. 96: Ibrahim Rahimov, Random Sums and Branching Stochastic Processes. viii, 195 pages, 1995.

Vol. 97: R. Szekli, Stochastic Ordering and Dependence in Applied Probability. viii, 194 pages, 1995.

Vol. 98: Philippe Barbe and Patrice Bertail, The Weighted Bootstrap. viii, 230 pages, 1995.

Vol. 99: C.C. Heyde (Editor), Branching Processes: Proceedings of the First World Congress. viii, 185 pages, 1995.

Vol. 100: Wlodzimierz Bryc, The Normal Distribution: Characterizations with Applications. viii, 139 pages, 1995.

Vol. 101: H.H. Andersen, M.Højbjerre, D. Sørensen, P.S.Eriksen, Linear and Graphical Models: for the Multivariate Complex Normal Distribution. x, 184 pages, 1995.

Vol. 102: A.M. Mathai, Serge B. Provost, Takesi Hayakawa, Bilinear Forms and Zonal Polynomials. x, 378 pages, 1995.

Vol. 103: Anestis Antoniadis and Georges Oppenheim (Editors), Wavelets and Statistics. vi, 411 pages, 1995.

Vol. 104: Gilg U.H. Seeber, Brian J. Francis, Reinhold Hatzinger, Gabriele Steckel-Berger (Editors), Statistical Modelling: 10th International Workshop, Innsbruck, July 10-14th 1995. x, 327 pages, 1995.

Vol. 105: Constantine Gatsonis, James S. Hodges, Robert E. Kass, Nozer D. Singpurwalla(Editors), Case Studies in Bayesian Statistics, Volume II. x, 354 pages, 1995.

Vol. 106: Harald Niederreiter, Peter Jau-Shyong Shiue (Editors), Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. xiv, 372 pages, 1995.

Vol. 107: Masafumi Akahira, Kei Takeuchi, Non-Regular Statistical Estimation. vii, 183 pages, 1995.

Vol. 108: Wesley L. Schaible (Editor), Indirect Estimators in U.S. Federal Programs. viii, 195 pages, 1995.

Vol. 109: Helmut Rieder (Editor), Robust Statistics, Data Analysis, and Computer Intensive Methods. xiv, 427 pages, 1996.

Vol. 110: D. Bosq, Nonparametric Statistics for Stochastic Processes. xii, 169 pages, 1996.

Vol. 111: Leon Willenborg, Ton de Waal, Statistical Disclosure Control in Practice. xiv, 152 pages, 1996.

Vol. 112: Doug Fischer, Hans-J. Lenz (Editors), Learning from Data. xii, 450 pages, 1996.

Vol. 113: Rainer Schwabe, Optimum Designs for Multi-Factor Models. viii, 124 pages, 1996.

Vol. 114: C.C. Heyde, Yu. V. Prohorov, R. Pyke, and S. T. Rachev (Editors), Athens Conference on Applied Probability and Time Series Analysis Volume I: Applied Probability In Honor of J.M. Gani. viii, 424 pages, 1996.

Vol. 115: P.M. Robinson, M. Rosenblatt (Editors), Athens Conference on Applied Probability and Time Series Analysis Volume II: Time Series Analysis In Memory of E.J. Hannan. viii, 448 pages, 1996.

Vol. 116: Genshiro Kitagawa and Will Gersch, Smoothness Priors Analysis of Time Series. x, 261 pages, 1996.

Vol. 117: Paul Glasserman, Karl Sigman, David D. Yao (Editors), Stochastic Networks. xii, 298, 1996.

Vol. 118: Radford M. Neal, Bayesian Learning for Neural Networks. xv, 183, 1996.

Vol. 119: Masanao Aoki, Arthur M. Havenner, Applications of Computer Aided Time Series Modeling. ix, 329 pages, 1997.

Vol. 120: Maia Berkane, Latent Variable Modeling and Applications to Causality. vi, 288 pages, 1997.

Vol. 121: Constantine Gatsonis, James S. Hodges, Robert E. Kass, Robert McCulloch, Peter Rossi, Nozer D. Singpurwalla (Editors), Case Studies in Bayesian Statistics, Volume III. xvi, 487 pages, 1997.

Vol. 122: Timothy G. Gregoire, David R. Brillinger, Peter J. Diggle, Estelle Russek-Cohen, William G. Warren, Russell D. Wolfinger (Editors), Modeling Longitudinal and Spatially Correlated Data. x, 402 pages, 1997.

Vol. 123: D. Y. Lin and T. R. Fleming (Editors), Proceedings of the First Seattle Symposium in Biostatistics: Survival Analysis. xiii, 308 pages, 1997.

Vol. 124: Christine H. Müller, Robust Planning and Analysis of Experiments. x, 234 pages, 1997.

Vol. 125: Valerii V. Fedorov and Peter Hackl, Model-oriented Design of Experiments. viii, 117 pages, 1997.

Vol. 126: Geert Verbeke and Geert Molenberghs, Linear Mixed Models in Practice: A SAS-Oriented Approach. xiii, 306 pages, 1997.

Vol. 127: Harald Niederreiter, Peter Hellekalek, Gerhard Larcher, and Peter Zinterhof (Editors), Monte Carlo and Quasi-Monte Carlo Methods 1996, xii, 448 pages, 1997.

Vol. 128: L. Accardi and C.C. Heyde (Editors), Probability Towards 2000, x, 356 pages, 1998.

Vol. 129: Wolfgang Härdle, Gerard Kerkyacharian, Dominique Picard, and Alexander Tsybakov, Wavelets, Approximation, and Statistical Applications, xvi, 265 pages, 1998.

Vol. 130: Bo-Cheng Wei, Exponential Family Nonlinear Models, ix, 240 pages, 1998.

Vol. 131: Joel L. Horowitz, Semiparametric Methods in Econometrics, ix, 204 pages, 1998.

Vol. 132: Douglas Nychka, Walter W. Piegorsch, and Lawrence H. Cox (Editors), Case Studies in Environmental Statistics, viii, 200 pages, 1998.

Vol. 133: Dipak Dey, Peter Müller, and Debajyoti Sinha (Editors), Practical Nonparametric and Semiparametric Bayesian Statistics, xv, 408 pages, 1998.

Vol. 134: Yu. A. Kutoyants, Statistical Inference For Spatial Poisson Processes, vii, 284 pages, 1998.

Vol. 135: Christian P. Robert, Discretization and MCMC Convergence Assessment, x, 192 pages, 1998.

Vol. 136: Gregory C. Reinsel, Raja P. Velu, Multivariate Reduced-Rank Regression, xiii, 272 pages, 1998.

Vol. 137: V. Seshadri, The Inverse Gaussian Distribution: Statistical Theory and Applications, xi, 360 pages, 1998.

Vol. 138: Peter Hellekalek, Gerhard Larcher (Editors), Random and Quasi-Random Point Sets, xi, 352 pages, 1998.

Vol. 139: Roger B. Nelsen, An Introduction to Copulas, xi, 232 pages, 1999.

Vol. 140: Constantine Gatsonis, Robert E. Kass, Bradley Carlin, Alicia Carriquiry, Andrew Gelman, Isabella Verdinelli, Mike West (Editors), Case Studies in Bayesian Statistics, Volume IV, xvi, 456 pages, 1999.

Vol. 141: Peter Müller, Brani Vidakovic (Editors), Bayesian Inference in Wavelet-Based Models, xi, 394 pages, 1999.