Neural Network Models for Conditional Distribution Under Bayesian Analysis

Tatiana Miazhynskaia
tmiazhyn@pop.tuwien.ac.at
Institute of Management Science, Vienna University of Technology,
A-1040 Vienna, Austria

Sylvia Frühwirth-Schnatter
sylvia.fruehwirth-schnatter@jku.at
Institute for Applied Statistics, Johannes Kepler University Linz,
A-4040 Linz, Austria

Georg Dorffner
georg.dorffner@meduniwien.ac.at
Austrian Research Institute for Artificial Intelligence and Department of Medical
Cybernetics and Artificial Intelligence, Medical University of Vienna, Vienna, Austria

We use neural networks (NN) as a tool for a nonlinear autoregression to predict the second moment of the conditional density of return series. The NN models are compared to the popular econometric GARCH(1,1) model. We estimate the models in a Bayesian framework using Markov chain Monte Carlo posterior simulations. The interlinked aspects of the proposed Bayesian methodology are identification of NN hidden units and treatment of NN complexity based on model evidence. The empirical study includes the application of the designed strategy to market data, where we found a strong support for a nonlinear multilayer perceptron model with two hidden units.

1 Introduction

Finance is one of the most frequent areas of neural network (NN) applications. NN modeling provides a very general framework for financial modeling, is sufficiently flexible, and can easily encompass a wide range of securities and fundamental asset price dynamics. The empirical studies by Donaldson and Kamstra (1997), Darrat and Zhong (2000), Yao and Tan (2001), and Dunis and Jalilov (2002) provide strong support for NNs as a potentially useful device for modeling and predicting financial markets. In these studies, the authors use probabilistic models describing the mean of the (conditional) distribution of market returns with help of NN. At the same time, the second moment of the return distribution plays a predominant role in modern financial markets. Volatility, measured by the standard deviation of returns, is the key variable for the pricing of financial instruments and for financial risk estimation. Modeling the volatility is of great importance in the financial literature, and NNs provide a good tool to model nonlinearity in the volatility process (see, e.g., Locarek-Junge & Prinzler, 1998; Schittenkopf, Dorffner, & Dockner, 2000; Dunis & Huang, 2001).

In this letter, we consider NN volatility models in a Bayesian framework. The key contributions of our study are a fully Bayesian analysis of autoregressive NNs models, including prior specification, Markov chain Monte Carlo (MCMC) simulation, model selection, and predictive analysis, with an application of this methodology to financial data; an objective strategy for NN hidden units identification; and the Bayesian treatment of neural network complexity based on model evidences.

The Bayesian inference on NN models was first implemented using a gaussian approximation of the posterior distribution (MacKay, 1992; Bishop, 1995). Neal (1996) introduced the hybrid MCMC algorithm, which makes efficient use of gradient information to reduce random walk behavior. Müller and Insua (1998) used a multivariate Metropolis-Hastings algorithm and marginalized over some parameters to improve performance. Lee (1999) provides a complete review of Bayesian model selection together with model-averaging methods for neural network regressions. Bayesian learning for NN models was also discussed by Lampinen and Vehtari (2001) and Vehtari and Lampinen (2002).

The MCMC algorithm presented in this letter improves these approaches, which are of limited use for the financial models we have in mind. The application of the hybrid MCMC algorithm to our models is practically limited because of the recurrent structure of the models and the consequently rather expensive computation of the energy gradient. Moreover, the autoregressive structure of the variance equation implies no separability between the model parameters, and, hence, marginalization as in Müller and Insua (1998) is not possible. In our Bayesian implementation we combine Gibbs sampling of hyperparameters (Neal, 1996) with Metropolis-Hastings simulations of model parameters.

A further open question in the NN community is the selection of the optimal size of the network. Marrs (1998), Holmes and Mallick (1998), Andrieu, de Freitas, and Doucet (2000), and Menchero, Diez, and Insua (2005) applied the reversible-jump MCMC algorithm to obtain joint estimates of the number of neurons and weights. In our study, we use the bridge sampling technique (Meng & Wong, 1996) to compute model evidence for different NN sizes and choose the optimal size of the network based on posterior model probabilities. This strategy allows handling models independently and allows the comparison of models that differ not only in the number of neurons but also in the model structure and distributional assumptions.

The efficiency of the MCMC simulations for the NN parameters is closely connected with the identifiability problem of the network hidden units. Unlike the known literature (Müller & Insua, 1998), we select identifiability constraints a posteriori, after the random permutation of the hidden nodes parameters. This strategy follows an approach introduced for finite mixture models by Frühwirth-Schnatter (2001).

The letter is organized as follows. In section 2 we present the NN volatility model we are working with. Section 3 includes the short overview of main concepts of the MCMC posterior simulations and Bayesian model selection. The detailed implementation of the Bayesian methodology together with empirical issues is given in section 4. Section 5 concludes the paper.

2 Models

To model return series, it is usual to split returns into a predictable deterministic component and a stochastic error. The deterministic component μ_t is the conditional mean of the returns at time t, that is, $\mu_t = \mathbf{E}(r_t|I_{t-1})$, where $\mathbf{E}$ denotes the expectation operator and I_{t-1} is an information set (history) at time $t-1$. The error process e_t is assumed to satisfy $\mathbf{E}(e_t|I_{t-1}) = 0$ and $\mathbf{E}(e_t^2|I_{t-1}) = h_t$. Thus, h_t is the conditional variance of the return series at time t, given information up to time $t-1$. If h_t is time dependent, the return series model is called heteroskedastic.

The most famous model widely used in financial practice is the GARCH model (Bollerslev, 1986), where the conditional variance is governed by a linear autoregressive process of past squared returns and variances. In our empirical analysis we use the GARCH(1,1) model with the conditional normal distribution and the AR(1) mean specification,

$$\begin{cases} r_t = a_0 + a_1 r_{t-1} + e_t, & t = 1, 2, \ldots, N \\ e_t \mid I_{t-1} \sim \mathbf{N}(0, h_t), & \\ h_t = \alpha_0 + \alpha_1 e_{t-1}^2 + \beta_1 h_{t-1}, & \end{cases}$$

where $\mathbf{N}(0, h_t)$ denotes the gaussian distribution with mean 0 and variance h_t. This model captures several stylized facts of asset returns: heteroskedasticity, volatility clustering, and excess kurtosis. But it frequently fails to capture highly irregular phenomena, such as market crashes and other unanticipated events that can lead to significant structural changes.

One popular direction to extend the classical GARCH is to allow nonlinear dependencies in the conditional variance. As a tool for introducing nonlinearity, we use neural network–based modeling (NN), describing the conditional variance by a multilayer perceptron (MLP) as in Schittenkopf et al. (2000).

To model the conditional variance h_t, we adapt the dynamics of the MLP in a recurrent fashion:

$$h_t = \sum_{j=1}^{H} v_j \Psi\left(w_j e_{t-1}^2 + \gamma_j h_{t-1} + c_j\right) + \alpha_0 + \alpha_1 e_{t-1}^2 + \beta_1 h_{t-1}, \tag{2.1}$$

with the logistic activation function of the hidden units $\Psi(z) = \exp(z)/(\exp(z)+1)$ and unrestricted output activation function. We assume positivity of the parameters $\alpha_0, \alpha_1, \beta_1$, and $v_1, \ldots, v_H$ to guarantee positivity of the conditional variance. We call such a nonlinear volatility model the MLP model. Its complete specification reads

$$\begin{cases} r_t = a_0 + a_1 r_{t-1} + e_t, \\ e_t \mid I_{t-1} \sim \mathbf{N}(0, h_t), \\ h_t \text{ is given by equation 2.1} \end{cases}$$

In such a definition, the MLP model is a nonlinear generalization of the GARCH(1,1) model, where the GARCH specification for the conditional variance is replaced by the recurrent MLP defined in equation 2.1. At that, we use the same notations for the common parameters α_0, α_1, and β_1 as the GARCH parameters and the linear parameters (shortcuts and bias) in the MLP specification.

With respect to the size of the NN, we will test the MLP model with different values of H and perform Bayesian model selection to define the optimal NN size.

3 Bayesian Inference: Main Concepts

In this section, we review fundamental concepts of Bayesian model selection as well as of Monte Carlo Markov chain (MCMC) posterior simulations.

The distinctive feature of the Bayesian framework (compared to the classical analysis) is its use of probability to express all forms of uncertainty. Thus, in addition to specifying a stochastic model for the observed returns $Y = (r_1, \ldots, r_N)$ given a vector of unknown parameters θ, we suppose that θ is a random quantity as well. The dependency of Y on θ is defined in the form of the likelihood $L(Y|\theta)$. Our subjective beliefs we may have about θ before having looked at the data Y are expressed in a prior distribution $\pi(\theta)$.

At the center of the Bayesian inference is a simple and extremely important expression known as Bayes' rule:

$$p(\theta|Y) = \frac{p(Y,\theta)}{p(Y)} = \frac{L(Y|\theta)\pi(\theta)}{\int L(Y|\theta)\pi(\theta)d\theta}. \tag{3.1}$$

Thus, having observed Y, our initial views about θ are updated by the data to get the distribution of θ conditional on Y. It is called the posterior distribution of θ. The normalizing constant $p(Y)$ of the posterior density, equation 2.1, is called the model evidence.

Suppose that we have a set of models $\mathbf{M} = \{M_i\}$ and some preliminary knowledge about model probabilities $\pi(M_i)$. Interest lies in obtaining the posterior probabilities $p(M_i|Y)$ for every model M_i in consideration of arriving at a single best model, or determining weights for model averaging. The model evidence yields posterior model probabilities as

$$p(M_i|Y) = \frac{p(Y|M_i) \cdot \pi(M_i)}{\sum_{k=1}^{n} p(Y|M_k)\pi(M_k)}. \tag{3.2}$$

The main problem is that the integral to compute $p(Y)$ in equation 3.1 is analytically tractable in only certain restricted problems, and sampling-based methods must be used to obtain estimates of the model evidences. Many different approaches have been suggested in the literature. The most widely used is the group of direct methods: the harmonic mean estimator of Newton and Raftery (1994), importance sampling (Frühwirth-Schnatter, 1995), reciprocal importance estimator (Gelfand & Dey, 1994), and bridge sampling (Meng & Wong, 1996; Frühwirth-Schnatter, 2004). Chib (1995) and Chib and Jeliazkov (2001) proposed an indirect method for estimating model evidences based on the MCMC output. A different approach to compute posterior model probabilities using MCMC is to include the model indicator as a parameter in the sampling algorithm itself. Green (1995) introduced a reversible-jump MCMC strategy for generating from the joint posterior distribution. (For a complete review of different Bayesian model selection methods, see, e.g., Carlin & Louis, 1996, or Chen, Shao, & Ibrahim, 2000.)

We based our decision with respect to the proper model selection method on the results of a previous simulation study using the GARCH(1,1) model (Miazhynskaia & Dorffner, 2006). In that study five methods were compared: (1) the harmonic mean estimator (Newton & Raftery, 1994) (2) Chib's candidate etimator (Chib, 1995; Chib & Jeliazkov, 2001), (3) reciprocal importance estimators (Gelfand & Dey, 1994), (4) the bridge sampling estimator (Meng & Wong, 1996), and (5) reversible-jump MCMC (Green, 1995). While all five methods proved to be feasible for model selection in this domain, a slight preference can be given to bridge sampling, since it tends to produce higher evidence (than, for instance, the reversible-jump algorithm) and use less computational time (than, for instance, Chib's candidate estimators). Therefore, it seemed a sensible choice to use bridge sampling throughout the work presented in this letter. We note, however, that we consider this a good choice mainly since we are dealing only with neural network models of relatively low model complexity. For larger models, reversible-jump estimation might prove superior, despite convergence problems observed in practice (Menchero et al., 2005).

A further important issue is that in the study reported here, we deal with nonlinear models, where the identifiability problem of the network units brings irregularity, in particular multimodality, into posterior analysis. For the similar case of mixture models, Frühwirth-Schnatter (2004) found Chib's estimator (Chib, 1995) to be sensitive to the labeling switching problem and proposed the random permutation approach in the context of bridge sampling methods. The bridge sampling technique, together with random permutation methods, is easily extended to deal with Bayesian selection with respect to the size of the NN.

As an input for the bridge sampling algorithm, we need a sample from the posterior distribution of the parameters $p(\theta|Y)$. Since $p(\theta|Y)$ is analytically intractable for our models, we adopt MCMC sampling as a tool to obtain posterior samples. The idea of MCMC is based on the construction of an irreducible and aperiodic Markov chain with realizations $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}, \ldots$ in the parameter space, equilibrium distribution $p(\theta|Y)$, and a transition probability $K(\theta'', \theta') = \pi(\theta^{(t+1)} = \theta'' \,|\, \theta^{(t)} = \theta')$, where θ' and θ'' are the realized states at time t and $t+1$, respectively. Under appropriate regularity conditions, asymptotic results guarantee that as $t \to \infty$, $\theta^{(t)}$ tends in distribution to a random variable with density $p(\theta|Y)$. The best-known MCMC procedures are Gibbs sampling, when we have completely specified full conditional distributions, and the Metropolis-Hastings (MH) algorithm, which provides a more general framework. (For an introduction to MCMC simulation methods, we refer to Chib & Greenberg, 1996, and Geweke, 1999.)

4 Bayesian Inference: Implementation and Illustration

4.1 Priors. The starting point for Bayesian inference is a prior distribution over unknown model parameters. Unfortunately, because of the autoregressive structure of our variance dynamics, there is no property of conjugacy for all parameters, and we have to choose prior specifications just according to our subjective believes.

All model parameters are assumed to be independent a priori. In the case of a neural network, it is difficult to provide any meaningful distribution for a particular weight since the correspondence between this weight and the observed data is not easily identifiable. Following the standard approach in the NN literature (Neal, 1996; Menchero et al., 2005), we adopt a hierarchical prior structure that enables us to treat the priors' parameters, also called hyperparameters, as random variables drawn from suitable distributions, the so-called hyperpriors. A convenient form for these hyperpriors is a vague inverse gamma distribution with fixed shape and mean parameters. To guarantee positivity of the conditional variance, we use the logarithmic transformation of the parameters α_0, α_1, β_1, and $v_1, \ldots, v_H$. By considering the GARCH model as a particular case of the nonlinear MLP model, we adopt the same hierarchical prior structure for the GARCH parameters α_0, α_1, and β_1.

With respect to the priors of the mean parameters a_0 and a_1, it seems natural to take fixed symmetrical gaussian priors with some big variance to be rather flat over the interval of variation of the data.

Summarizing, the following prior structure is employed:

- $\mathbf{N}(0, A_0)$ for the mean parameters a_0 and a_1
- $\mathbf{logN}(\kappa_i, \frac{1}{\tau_i})$, $i = 1, 2, 3$, for three linear variance parameters α_0, α_1, β_1
- $\mathbf{N}(0, \frac{1}{\tau_4})$ for any hidden weight w, $\mathbf{N}(0, \frac{1}{\tau_5})$ for any parameter γ, and $\mathbf{N}(0, \frac{1}{\tau_6})$ for any bias parameter c
- $\mathbf{logN}(0, \frac{1}{\tau_7})$ for any hidden-output weight v
- $\mathbf{Ga}(\xi_j, \omega_j)$ for the hyperparameters τ_j, $j = 1, \ldots, 7$

The prior variance A_0 of the mean parameters a_0 and a_1 is fixed at $A_0 = 10$. For the linear variance parameters, the priors' centers are fixed at $(\kappa_1, \kappa_2, \kappa_3) = (-3.0, -2.3, -0.2)$, reflecting our idea about the GARCH parameters.

The hyperparameters τ_j follow a gamma distribution with shape parameter ξ_j and mean ω_j, for $j = 1, \ldots, 7$. After preliminary tuning, we fixed the hyperprior shape parameters at $\xi_j = 10$ for all hyperparameters. The means ω_j of the hyperprior, controlling the width of the prior, were chosen to reach the maximal posterior model probability. We choose $\omega_j = 2$ for the transformed linear variance parameters $\log \alpha_0$, $\log \alpha_1$, and $\log \beta_1$; $\omega_j = 0.2$ for the input-hidden NN weights, $\gamma_1, \ldots, \gamma_H$, and biases; and $\omega_7 = 1$ for the hidden-output NN weights $\log v_1, \ldots, \log v_H$.

Note that the first three hyperparameters, τ_1, τ_2, τ_3, control a single model parameter, whereas the hyperparameters $\tau_4, \ldots, \tau_7$ control H parameters. Subsequently the number of parameters controlled by hyperparameter τ_j will be denoted by k_j.

4.2 MCMC Posterior Simulation. Having specified the prior distribution, we may now perform MCMC simulations to obtain random draws from the posterior distributions of the model parameters. The posterior simulations for all model parameters are complicated by the fact that the lagged residuals and variances are included in the recurrent variance specifications, so there is no separability between the regression parameters and the heteroskedastic parameters. Moreover, having included the hyperparameters in our models, we have to perform simulations from the full conditional distribution $p(\tau|Y, \theta)$ as well.

In this way, our MCMC algorithm iterates through the following steps:

- Sampling the mean parameters a_0 and a_1
- Sampling the variance parameters α_0, α_1, β_1 and the NN weights $w_1, \ldots, w_H, \gamma_1, \ldots, \gamma_H, c_1, \ldots, c_H, v_1, \ldots, v_H$
- Sampling the hyperparameters $\tau_1, \ldots, \tau_7$

4.2.1 Metropolis-Hastings Simulations for the Mean Parameters. To get inference for the mean parameters $a = (a_0, a_1)'$, we use the MH algorithm with a bivariate proposal distribution, constructed to mimic the true full conditional posterior distribution, following ideas introduced in Müller and Insua (1998), Nakatsuma (2000), and Kaufmann and Frühwirth-Schnatter (2002). The gaussian proposal density reads

$$\begin{aligned} q\left(a|a^{(old)}\right) \sim\ & \mathbf{N}\big(m\big(a^{(old)}\big), M\big(a^{(old)}\big)\big), \\ & m\big(a^{(old)}\big) = M\big(a^{(old)}\big) \cdot Z'\hat{\Sigma}\left(a^{(old)}\right)^{-1} Y, \\ & M\big(a^{(old)}\big) = \left[Z'\hat{\Sigma}\left(a^{(old)}\right)^{-1} Z + A_0^{-1} I_2\right]^{-1}, \end{aligned}$$

where $\hat{\Sigma}(a^{(old)}) = diag(h_2(a^{(old)}, \theta_{-a}), \ldots, h_N(a^{(old)}, \theta_{-a}))$, where θ_{-a} are the remaining model parameters being fixed at their most recent draw, Z is the regression matrix with rows $[1\ r_{t-1}]$, $t = 2, \ldots, N$, A_0 denotes the prior variances, and I_2 is an identity matrix.

The resulting MH step appears to be rather efficient, showing an acceptance rate of around 80%.

4.2.2 Random Walk Metropolis Algorithm for the Variance Parameters. The variance parameters include the linear (GARCH) parameters $\alpha_0, \alpha_1, \beta_1$ and the NN weights $w_1, \ldots w_H, \gamma_1, \ldots, \gamma_H, c_1, \ldots, c_H$, and $v_1, \ldots, v_H$. Again, because of the recurrent structure of the variance equations, the variance h_t depends on these parameters directly as well as indirectly through h_{t-1}.

To sample from the posterior, we adopt in a first run a random walk single move MH algorithm with a gaussian candidate density $\mathbf{N}(\theta_i^{(old)}, \sigma_i^2)$ for each variance parameter. The proposal variance σ_i^2 uses the current value of the hyperparameter τ_i, controlling the prior of the corresponding parameter θ_i, that is, $\sigma_i^2 = c_i/\tau_i$, where c_i is a tuning constant, chosen so as to reach some reasonable acceptance rate. This choice of the proposal variance is motivated by an intuitive sense that the posterior distribution of the hyperparameters reflects the variation in the corresponding parameter posterior. To obtain an effective simulation scheme, we proceed in an adaptive way: starting simulations with rather large tuning constants c_i, after approximately 500 to 1000 iterations we adapt them to get an acceptance rate of about 20% to 40% (see Carlin & Louis, 1996, for details). These initial iterations are then discarded.

After this first run, we checked for posterior correlation between the parameters. As it is well known (Hills & Smith, 1992), this phenomenon reduces the performance of MCMC algorithms. In a second run, we collected highly correlated parameters in one block and updated them by a multivariate random walk Metropolis step with a gaussian proposal density. The variance-covariance matrix of this proposal is selected as $\Sigma = C \cdot R \cdot C$, where $C = diag(\sigma_1, \dots, \sigma_k)$ and R is a fixed correlation matrix estimated from the single-move MH draws. (For other possible strategies to tune multivariate MH algorithm, see Chib & Greenberg, 1995.)

Even for the multivariate proposal density, the MCMC simulations exhibited high autocorrelation in the posterior chains, which will increase the variance of posterior estimates (Raftery & Lewis, 1992). This rate of dependence can be characterized by the inefficiency factor.[1] Based on the computed inefficiency factors, we define the length of the chain. Moreover, to reduce correlation in the posterior sample (and, consequently, reduce the MCMC variance), we resave only one drawing out of every, say L, iterations ($L \approx 1/\{\text{average inefficiency factor}\}$). This strategy, known as thinning (Raftery & Lewis, 1992), is also useful to reach savings in storage and computational time. In our empirical study, we reached the inefficiency factor of 1 for the mean parameters. For the variance parameters, it was between 2 and 8 for the linear weights and 1 to 5 for the nonlinear weights. The final posterior samples are of size 1000.

4.2.3 Gibbs Sampling for Hyperparameters. Because of the conjugate hyperprior form, we obtain the conditional distribution of every hyperparameter τ_i to be again the gamma distribution with parameters

$$\begin{cases} \text{shape } \xi_i + 1 \text{ and mean } \frac{\xi_i + 1}{\xi_i/\omega_i + (\theta_i - \kappa_i)^2}, & \text{for } i = 1, 2, 3 \\ \text{shape } \xi_i + H \text{ and mean } \frac{\xi_i + H}{\xi_i/\omega_i + \sum_{j=1}^{H} (\theta_i^{(j)})^2}, & \text{for } i = 4, \dots, 7, \end{cases}$$

where $\theta_i = (\theta_i^{(1)}, \dots, \theta_i^{(k_i)})$ are the model parameters controlled by the hyperparameter τ_i. This posterior is what is needed for Gibbs sampling, since given θ_i the value of τ_i is independent of the other parameters, hyperparameters, and target values.

[1] Inefficiency factor κ is the factor by which we have to increase the run length of the MCMC sampler compared to independent and identically distributed sampling. It accounts for the whole serial dependence in the sampled values (Geweke, 1992) in the following way:

$$\kappa = 1 + 2 \sum_{l=1}^{L} \left(1 - \frac{l}{L+1}\right) \rho(l),$$

where $\rho(l)$ represents the autocorrelation at lag l of the sampled parameter values. The bandwidth L is chosen such that $\rho(l)$, $l = 1, \dots, L$, significantly contributes to the serial dependence of the sampled values.

4.3 Model Evidence Calculation. We use the MCMC posterior draws as inputs to the bridge sampling algorithm to compute the model evidence. To reduce the dimension of integration, we integrated out the hyperparameters. This leads to a multivariate Student-t prior for the model parameters.

Let us denote the posterior draws by $\{\theta^{(m)}\}_{m=1}^{M}$. Assume further that we can find some simple approximation to the posterior density, some known density $h(\theta)$. By $\{\tilde{\theta}^{(l)}\}_{l=1}^{L}$ we denote a sample from this approximating density. Then we can apply the following iterative procedure to find the bridge sampling estimator of the model evidence,

$$\hat{p}_{BS}^{(t)}(Y) = \hat{p}_{BS}^{(t-1)}(Y) \frac{\frac{1}{L}\sum_{l=1}^{L} \frac{\hat{p}(\tilde{\theta}^{(l)}|Y)}{Lh(\tilde{\theta}^{(l)})+M\hat{p}(\tilde{\theta}^{(l)}|Y)}}{\frac{1}{M}\sum_{m=1}^{M} \frac{\hat{h}(\theta^{(m)})}{Lh(\theta^{(m)})+M\hat{p}(\theta^{(m)}|Y)}}, \tag{4.1}$$

where we used the normalized posterior as $\hat{p}(\theta|Y) = \frac{1}{\hat{p}_{BS}^{(t-1)}} L(Y|\theta)\pi(\theta)$. To start the iteration, we can use the importance density estimator:

$$p_{BS}^{(0)}(Y) = \frac{1}{L}\sum_{l=1}^{L} \frac{L(Y|\tilde{\theta}^{(l)})\pi(\tilde{\theta}^{(l)})}{h(\tilde{\theta}^{(l)})}.$$

(See Meng & Wong, 1996, and Frühwirth-Schnatter, 2004, for more details on the bridge sampling method.)

In order to calculate the model evidence by iteration 4.1, we have to choose an approximating importance density $h(\theta)$. The algorithm will reach high efficiency if $h(\theta)$ roughly matches the posterior density $p(\theta|Y)$. Following the suggestion of Gelfand and Dey (1994), we used a multivariate normal importance density with the mean and covariance estimated from the MCMC posterior sample. We want to stress that we use the logarithmic transformation of the parameters α_0, α_1, β_1, and $v_1, \ldots, v_H$ to construct the importance density $h(\cdot)$. To define the importance density $h(\theta)$ for the MLP model with $H > 1$, we used the posterior output constrained with respect to labeling of the hidden units (see section 4.4.1). We need not concern ourselves with the tail behavior of $h(\cdot)$ in comparison to the posterior density, because the optimal bridge sampling estimator is robust in this respect (Frühwirth-Schnatter, 2004).

The convergence of the iterative procedure in equation 4.1 was very fast, usually reached in the third to fourth iteration. After computing the evidence for all models in the analysis, we calculated the posterior model probabilities, equation 3.2.

4.4 Illustration. In the empirical illustration, we use daily closing values of the Japan index NIKKEI 225. The index series were obtained online from public sources (http://finance.yahoo.com). The time interval taken for all data sets was 17 years from January 1986 to December 2002, where the last year was left to test our models out of sample. All data were transformed into continuously compounded returns r_t in the standard way by the natural logarithm of the ratio of consecutive daily closing levels.

4.4.1 Identification of Hidden Units in MLP Model. The efficiency of the MCMC simulations for the NN parameters is closely connected to the identifiability problem of the network hidden units, often termed label switching in the context of mixture models (Stephens, 2000). That is, the likelihood $L(Y|\theta)$ is invariant with respect to relabeling of the units. Since our prior of the NN parameters is also invariant against permutations of the units, the posterior distribution is similarly invariant with $H!$ symmetric sets of modes, H being the number of hidden units. This symmetry in the posterior causes problems when attempting to perform inference regarding the individual NN units or to represent the posterior.

Since this issue is directly related to mixture models with their component identification, we adopt the permutation approach of Kaufmann and Frühwirth-Schnatter (2002). Thus, parallel to the MCMC simulations, we perform a random permutation of the hidden unit weights. This helps to produce efficient posterior samples with perfect mixing between the units. When analyzing the posterior sample, we produce scatter plots of the NN parameters to define identification constraints and reorder the posterior sample according to these restrictions. This helps to avoid multimodality in the posterior due to arbitrary labeling of the units and helps in choosing an approximating density to calculate the model evidence.

Our approach in dealing with the NNs identification problem differs from the one known in literature (Menchero et al., 2005), where an identification condition is defined before observing posteriors. Such formal identifiability introduces an obvious bias toward an NN structure with assumed number of hidden units, when the true number of units is smaller. Moreover, even if the number of units is correct, clear separability of all groups of parameters is not always the case.

For the returns from NIKKEI 225 index, the pairwise posterior scatter plots for the MLP model with two hidden units ($H = 2$) are presented in Figure 1. The plots show two separated nodes with respect to the hidden-output parameters v_j. To remove multimodality, we reorder the posterior output according to the condition $v_1 < v_2$. The marginal posterior plots of the nonlinear parameters w_j, γ_j, c_j, v_j before and after identification are given in Figure 2. Note that it is hard or even impossible to identify the nodes based on constraints on parameter other than v_j. It is evident from Figure 1 that a constraint on w_j, such as $w_1 < w_2$, would not remove multimodality due to units' labeling.

When we increase the number of hidden units in the MLP model and take $H = 3$, the pairwise posterior scatter plots in Figure 3 give no identification for the third hidden unit, and in the posterior in Figure 4, we see only two

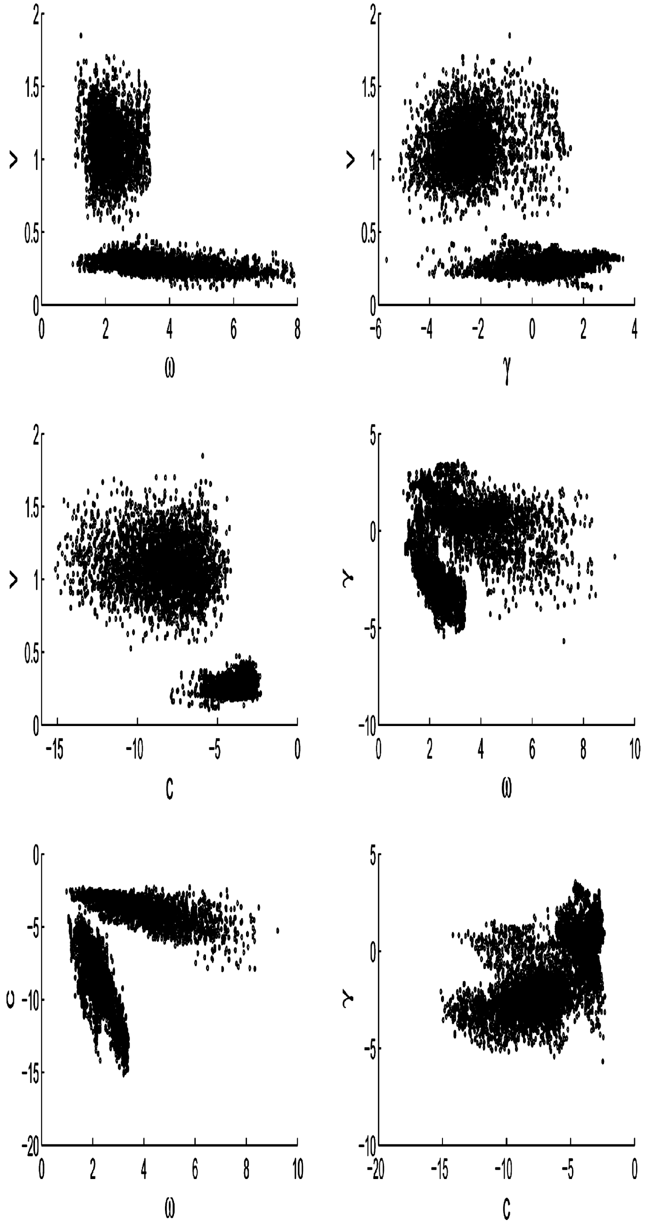

Figure 1: Posterior paired plots of the nonlinear parameters of the MLP model with $H = 2$, showing two well-separated modes with respect to the parameter υ.

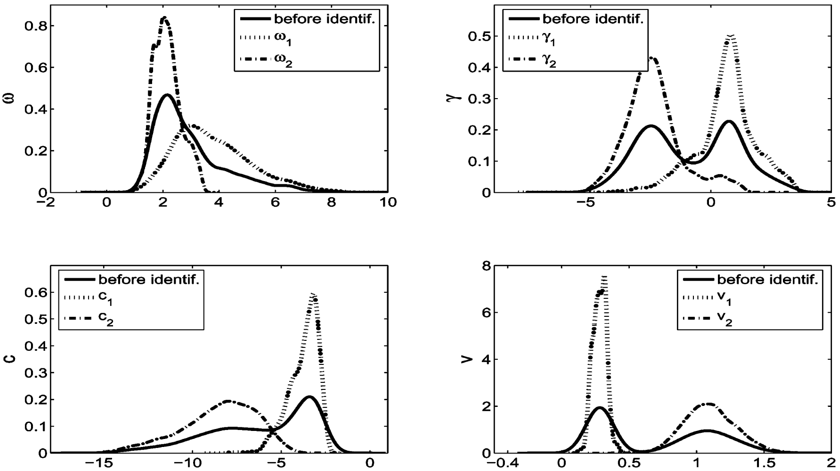

Figure 2: Posterior distribution of the nonlinear parameters before and after identification based on the condition $v_1 < v_2$. MLP model with $H = 2$. The solid line denotes the posterior before applying the identification condition. The dotted and dashed lines correspond to the posterior of the parameters of the first and second hidden units, respectively, after identification.

modes. It seems safe to conclude that we are faced with node duplication, and two hidden units are enough to describe the nonlinearity being present in the volatility structure of the NIKKEI 225 data. As will be shown in the next section, this empirical finding is supported by formal model selection based on model evidence.

4.4.2 Model Selection. Now we can test for the optimal size of the neural network describing the market data. We compute model evidence (ME) and posterior model probabilities (MP) exhibited by the linear GARCH and the nonlinear MLP model for $H = 1, \dots, 4$ (see Table 1). We see a clear preference for nonlinearity for the NIKKEI 225 return series. The posterior model probability of the linear GARCH model is close to 0%. Introducing one hidden unit leads to some improvement in the model performance. The MLP model with two hidden units, however, dominates with a probability of 91%. The models with three and four hidden units turn out to be too complex and are punished by a much smaller model evidence.

4.4.3 Bayesian Predictive Analysis. To complete the fully Bayesian procedure, we evaluate the predictive performance of our models. We are going to compare the models with respect to their ability to predict future returns, given training data $Y_{1:N}$ (years 1986–2001). The test data contain the year 2002.

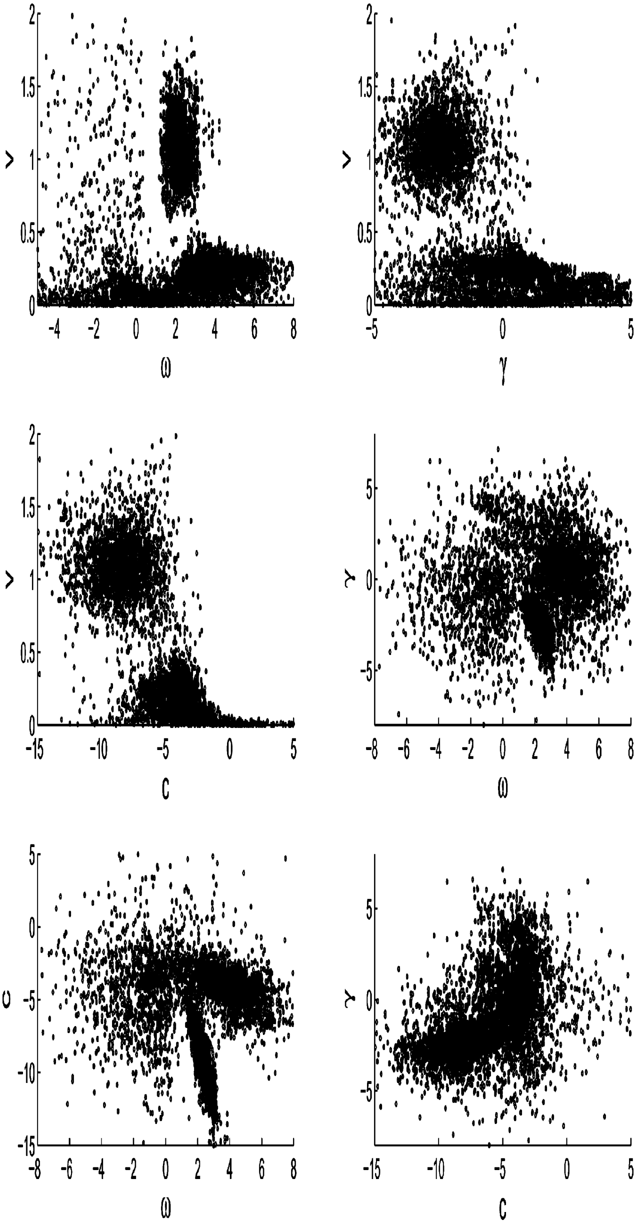

Figure 3: Posterior paired plots of the nonlinear parameters of the MLP model with $H = 3$.

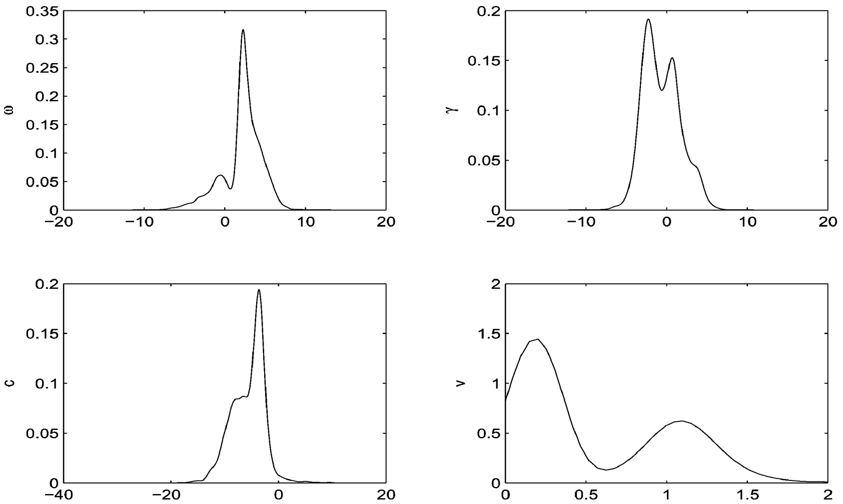

Figure 4: Posterior distribution of the nonlinear parameters for the MLP model with $H = 3$. No identification for the third hidden unit.

Table 1: Model Evidence (ME) and Posterior Model Probabilities (MP).

Model	Log ME	MP
GARCH(1,1)	−6809.8	0.0000
MLP$_{H=1}$ model	−6794.5	0.0034
MLP$_{H=2}$ model	**−6788.9**	**0.9133**
MLP$_{H=3}$ model	−6791.3	0.0829
MLP$_{H=4}$ model	−6796.5	0.0005

Note: The bold type signifies the model with the largest model evidence.

Bayesian prediction is based on $p(y_{N+1}, \ldots, y_{N+T} | Y_{1:N})$, being the posterior predictive density of future returns $y_{N+1}, \ldots, y_{N+T}$, given actual observations $Y_{1:N} = \{y_1, \ldots, y_N\}$. The posterior predictive density for $y_{N+1}, \ldots, y_{N+T}$ is given by

$$p(y_{N+1}, \ldots, y_{N+T} | Y_{1:N}) = \int L(y_{N+1}, \ldots, y_{N+T} | \theta)\, p(\theta | Y_{1:N}) d\theta \qquad (4.2)$$

due to the conditional independence of $y_{N+1}, \ldots, y_{N+T}$ and $Y_{1:N}$. This distribution summarizes the information concerning likely values of future observations given the likelihood, the prior, and the data we have observed so far. (For more information on the Bayesian predictive analysis, see, e.g., Geweke, 1989; Gilks, Richardson, & Spiegelhalter, 1996; or Vehtari & Lampinen, 2002.)

Table 2: Mean Squared Errors of Bayesian Predictions from One Day Ahead to One Year Ahead, Averaged over the Corresponding Period.

Model	1 Day	1 Week	1 Month	6 Months	Year
GARCH(1,1)	0.605	0.963	1.240	2.077	2.813
MLP$_{H=1}$ model	0.610	0.960	1.202	1.813	2.028
MLP$_{H=2}$ model	0.598	0.948	1.171	1.838	2.048
MLP$_{H=3}$ model	0.601	0.949	1.172	1.832	2.042
MLP$_{H=4}$ model	0.605	0.954	1.185	1.838	2.040
Model averaging	**0.594**	**0.943**	**1.153**	**1.765**	**1.959**

Note: The "best" values within every prediction period are in bold.

To sample from the predictive density, equation 4.2, we proceed in the following way. For every value $\theta^{(g)}$ drawn from the posterior distribution $p(\theta|Y_{1:N})$, we generate a random sequence $Y^{(g)}_{N+1:N+T}$ of future returns until T periods ahead. By analogy with classical model checking, we evaluate the predictive performance of a particular model through the mean squared error between the actual observations and the expectations obtained from the predictive densities. These expectations are estimated by taking the average over the sequence $Y^{(g)}_{N+1:N+T}$.

The resulting mean squared errors are given in Table 2 for NN models of different size and different forecasting horizons: 1 day, 1 week, 1 month, 6 months, and 1 year ahead. In the last row, we present the results of model averaging where individual predictions are weighted according to the estimated posterior model probabilities (see Lee, 1999, for more details on model averaging). To remove the sensitivity of the results to the day-specific actual return, we averaged the MSE over the corresponding period.

The results in the Table 2 support our conclusion that the nonlinear models outperform the linear GARCH model with the MLP model, with two hidden units being among the best. The Bayesian predictions weighted according to the posterior model probabilities deliver the best result in terms of minimum of mean squared error.

5 Conclusion

We applied a fully Bayesian approach for autoregressive NN volatility models, including hierarchical prior specifications, MCMC posterior simulations, model evidence computation, and predictive analysis. We proposed a new methodology to identify hidden units, consisting of random permutation of units to increase MCMC efficiency and identification a posteriori. To deal with NN model complexity, we performed Bayesian model selection based on model evidence.

The application of the proposed methodology to financial market data showed its capability to find nonlinear structure in volatility behavior. We obtain strong support for nonlinearity in the volatility process modeled by MLP with two hidden units. Adding one more hidden unit leads to overparameterization and is punished by showing scant model evidence.

Future work will deal with extensions of empirical study of linear versus nonlinear volatility models in two directions: under different assumptions about the conditional distribution of returns and for different financial data.

Acknowledgments

This work was funded by the Austrian Science Fund under grant SFB#010: Adaptive Information Systems and Modeling in Economics and Management Science. The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Education, Science and Culture and by the Austrian Federal Ministry for Transport, Innovation and Technology.

References

Andrieu, C., de Freitas, N., & Doucet, A. (2000). Robust full Bayesian methods for neural networks. In S. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems, 12* (pp. 379–385). Cambridge, MA: MIT Press.

Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Bollerslev, T. (1986). A generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*, 307–327.

Carlin, B., & Louis, T. (1996). *Bayes and empirical Bayes methods for data analysis*. London: Chapman & Hall.

Chen, M., Shao, Q., & Ibrahim, J. (2000). *Monte Carlo methods in Bayesian computation*. New York: Springer.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, *90*(432), 1313–1321.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *American Statistician*, *49*, 327–335.

Chib, S., & Greenberg, E. (1996). Markov chain Monte Carlo simulation methods in econometrics. *Econometric Theory*, *12*, 409–431.

Chib, S., & Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, *96*(453), 270–281.

Darrat, A. F., & Zhong, M. (2000). On testing the random walk hypothesis: A model-comparison approach. *Financial Review*, *35*(3), 105–124.

Donaldson, R., & Kamstra, M. (1997). An artificial neural network-GARCH model for international stock return volatility. *Journal of Empirical Finance*, *4*(1), 17–46.

Dunis, C., & Huang, X. (2001). *Forecasting and trading currency volatility: An application of recurrent neural regression and model combination* (Tech. rep.). Liverpool: Liverpool Business School.

Dunis, C. L., & Jalilov, J. (2002). Neural network regression and alternative forecasting techniques for predicting financial variables. *Neural Network World*, *12*, 113–139.

Frühwirth-Schnatter, S. (1995). Bayesian model discrimination and Bayes factor for linear gaussian state space models. *Journal of the Royal Statistical Society, series B*, 57, 237–246.

Frühwirth-Schnatter, S. (2001). MCMC estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association*, *96*, 194–209.

Frühwirth-Schnatter, S. (2004). Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques. *Econometrics Journal*, *7*, 143–167.

Gelfand, A., & Dey, D. (1994). Bayesian model choice: Asymptotic and exact calculations. *Journal of the Royal Statistical Society, Ser. B*, *56*, 501–514.

Geweke, J. (1989). Exact predictive densities for linear models with ARCH disturbances. *Journal of Econometrics*, *40*, 63–86.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In J. Bernardo, J. Berger, A. Dawid, & A. Smith (Eds.), *Bayesian statistics* (vol. 4, pp. 169–193). New York: Oxford University Press.

Geweke, J. (1999). Using simulation methods for Bayesian econometric models: Inference, development and communication. *Econometric Reviews*, *18*, 1–126.

Gilks, W., Richardson, S., & Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*. London: Chapman & Hall.

Green, P. (1995). Reversible jump MCMC computation and Bayesian model determination. *Biometrika*, *82*, 711–732.

Hills, S., & Smith, A. (1992). Parameterization issues in Bayesian inference. In J. Bernardo, J. Berger, A. Dawid, & A. Smith (Eds.), *Bayesian statistics* (vol. 4, pp. 227–246). New York: Oxford University Press.

Holmes, C., & Mallick, B. (1998). Bayesian radial basis functions of variable dimension. *Neural Computation*, *10*, 1217–1233.

Kaufmann, S., & Frühwirth-Schnatter, S. (2002). Bayesian analysis of switching ARCH models. *Journal of Time Series Analysis*, *23*(4), 425–458.

Lampinen, J., & Vehtari, A. (2001). Bayesian approach for neural networks—review and case studies. *Neural Networks*, *14*(3), 257–274.

Lee, H. (1999). *Model selection and model averaging for neural networks*. Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

Locarek-Junge, H., & Prinzler, R. (1998). Estimating value-at-risk using neural networks. In H.-U. Buhl, & C. Weinhardt (Eds.), *Informationsysteme in der Finanzwirtschaft*. Berlin: Springer.

MacKay, D. (1992). A practical Bayesian framework for backprop networks. *Neural Computation*, *4*, 448–472.

Marrs, A. (1998). An application of reversible-jump MCMC to multivariate spherical gaussian mixtures. In M. Jordan, M. Kearns, & S. Solla (Eds.), *Advances in neural information processing systems* (*10*, pp. 577–583). Cambridge, MA: MIT Press.

Menchero, A., Diez, R. M., & Insua, D. R. (2005). Bayesian analysis of nonlinear autoregression models based on neural networks. *Neural Computation*, *17*, 453–485.

Meng, X., & Wong, W. (1996). Simulating ratios of normalizing constants via a simple identity. *Statistical Sinica*, *6*, 831–860.

Miazhynskaia, T., & Dorffner, G. (2006). A comparison of Bayesian model selection based on MCMC with an application to GARCH-type models. *Statistical Papers*, *47*, 525–549.

Müller, P., & Insua, D. R. (1998). Issues in Bayesian analysis of neural network models. *Neural Computation*, *10*, 571–592.

Nakatsuma, T. (2000). Bayesian analysis of ARMA-GARCH models: A Markov chain sampling approach. *Journal of Econometrics*, *95*, 57–69.

Neal, R. (1996). *Bayesian learning for neural networks*. New York: Springer-Verlag.

Newton, M., & Raftery, A. (1994). Approximate Bayesian inference by the weighted likelihood bootstrap. *Journal of Royal Statistical Society, Ser. B*, *56*, 1–48.

Raftery, A., & Lewis, S. (1992). How many iterations in the Gibbs sampler. In J. Bernardo, J. Berger, A. Dawid, & A. Smith (Eds.), *Bayesian statistics* (vol. 4, pp. 763–773). New York: Oxford University Press.

Schittenkopf, C., Dorffner, G., & Dockner, E. J. (2000). Forecasting time-dependent conditional densities: A seminonparametric neural network approach. *Journal of Forecasting*, *19*, 355–374.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Ser. B*, *62*, 795–809.

Vehtari, A., & Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, *14*(10), 2439–2468.

Yao, J., & Tan, C. (2001). Guidelines for financial forecasting with neural networks. In *Proceedings of International Conference on Neural Information Processing* (pp. 757–761). Shanghai: Fudan University Press.

Received April 25, 2005; accepted April 13, 2007.