Working Paper Series

Identifying Stochastic Processes with Mixture Density Networks

Christian Schittenkopf
Georg Dorffner
Engelbert J. Dockner

Working Paper No. 11
May 1998

Working Paper Series

May 1998

SFB
'Adaptive Information Systems and Modelling in Economics and Management Science'

Vienna University of Economics
and Business Administration
Augasse 2–6, 1090 Wien, Austria

in cooperation with
University of Vienna
Vienna University of Technology

http://www.wu-wien.ac.at/am

This piece of research was supported by the Austrian Science Foundation (FWF) under grant SFB#010 ('Adaptive Information Systems and Modelling in Economics and Management Science').

Identifying Stochastic Processes with Mixture Density Networks

Christian Schittenkopf
Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Vienna, Austria
Email: chris@ai.univie.ac.at

Georg Dorffner
Dept. of Medical Cybernetics and Artificial Intelligence
University of Vienna
Freyung 6, A-1010 Vienna, Austria
Email: georg@ai.univie.ac.at

Engelbert J. Dockner
Dept. of Business Administration
University of Vienna
Brünner Straße 72, A-1210 Vienna, Austria
Email: dockner@finance2.bwl.univie.ac.at

Abstract

In this paper we investigate the use of mixture density networks (MDNs) for identifying complex stochastic processes. Regular multilayer perceptrons (MLPs), widely used in time series processing, assume a gaussian conditional noise distribution with constant variance, which is unrealistic in many applications, such as financial time series (which are known to be heteroskedastic). MDNs extend this concept to the modeling of time-varying probability density functions (pdfs) describing the noise as a mixture of gaussians, the parameters of which depend on the input. We apply this method to identifying the process underlying daily ATX (Austrian stock exchange index) data. The results indicate that MDNs modeling a non-gaussian conditional pdf tend to be significantly better than traditional linear methods of estimating variance (ARCH) and also better than merely assuming a conditional gaussian distribution.

1 Introduction

During the last decade a huge number of theoretical and practical results on neural networks has been acquired. Many publications deal with MLPs in the context of non-linear regression where one typically minimizes the mean squared error to fit a set of input vectors to a set of output vectors. The implicit assumption underlying this method is that the variance of the target (conditioned on the input) is constant, or more precisely, that the conditional pdf of the target (i.e. the noise) is *a single gaussian of constant variance*. In other words, the outputs of a MLP approximate the conditional expectation of the target (in dependence of the input) under this assumption (Bishop, 1995).

Recently, extensions of standard neural network estimation, so-called mixture density networks (MDNs, Bishop, 1994; Neuneier et al., 1994) have been proposed. MDNs are able to model conditional target distributions with non-constant variance or, more generally, arbitrary non-gaussian distributions. We apply MDNs to a real-world, economic time series (the Austrian stock exchange index ATX). Our aim is to identify the underlying stochastic process, while at the same time predicting the volatility of this time series. The volatility, i. e. the conditional variance, is an important economic quantity which has been extensively studied since the seminal works of Engle (1982) and Bollerslev (1986). Although we were able to train MDNs on this time series with a random initialization of the weights, the training procedure was much more stable when the MDNs were initialized to output the constant unconditional variance of the target using a trained MLP. When using a loss function based on the likelihood function used for training in a ten-fold cross-validation, our results show that using non-gaussian conditional pdfs tends to lead to significantly lower errors than using gaussian pdfs. They also tend to be better than traditional linear methods such as ARCH (Engle, 1982).

In Section 2 we describe the architecture and the training of our MDNs as a generalization of the simple case of a MLP, including a simple extension we propose. In Section 3 a first test on an artificial data set, as well as our experimental results on the ATX data are described in detail. We discuss the results in Section 4.

2 Architecture and Training

The concept of MDNs (Bishop, 1994; Neuneier et al., 1994) has turned out to be very appropriate to model conditional pdfs in the areas of nonlinear inverse problems (Bishop, 1994), volatility forecasting (Ormoneit and Neuneier, 1995) and time series analysis (Schittenkopf and Deco, 1997). Thereby the main idea is to use MLPs to predict the parameters of the conditional pdf of the next value x_t in dependence of the past values $x_{t-1}, \ldots, x_{t-m}$. In general, these parameters are the priors, the centers and the widths of a weighted sum of gaussian pdfs. This representation is completely general since gaussian mixture models can approximate any pdf to, in principle, arbitrary accuracy (McLachlan and Basford, 1988) just as MLPs can approximate any smooth, non-linear function to arbitrary accuracy (Hornik et al., 1989). In this paper the conditional pdf is modelled by

$$\rho(x_t|x_{t-1}, \ldots, x_{t-m}) = \sum_{i=1}^{n} \alpha_{i,t} g(\mu_{i,t}, \sigma_{i,t}^2), \tag{1}$$

$$g(\mu_{i,t}, \sigma_{i,t}^2) = \frac{1}{\sqrt{2\pi\sigma_{i,t}^2}} \exp\left(-\frac{(x_t - \mu_{i,t})^2}{2\sigma_{i,t}^2}\right) \tag{2}$$

where the parameters $\alpha_{i,t}$, $\mu_{i,t}$ and $\sigma_{i,t}^2$ are estimated by

$$\alpha_{i,t} = s(\tilde{\alpha}_{i,t}) = \frac{\exp(\tilde{\alpha}_{i,t})}{\sum_{j=1}^{n} \exp(\tilde{\alpha}_{j,t})}, \tag{3}$$

$$\tilde{\alpha}_{i,t} = \mathrm{MLP}_{1,i}(x_{t-1}, \ldots, x_{t-m}), \tag{4}$$

$$\mu_{i,t} = \mathrm{MLP}_{2,i}(x_{t-1}, \ldots, x_{t-m}), \quad (5)$$
$$\sigma^2_{i,t} = (\mathrm{MLP}_{3,i}(x_{t-1}, \ldots, x_{t-m}))^2. \quad (6)$$

The softmax function $s(\tilde{\alpha}_{i,t})$ ensures that the weights $\alpha_{i,t}$ are positive and that they sum up to one, which makes the right-hand side of Eq. (1) a pdf. The quadratic output function in Eq. (6) guarantees positive variances. As a result each MLP receives the same m-dimensional input $x_{t-1}, \ldots, x_{t-m}$ and produces a different, n-dimensional output where n equals the number of gaussian components. This is an extension of the standard MDN (Bishop, 1994) in that it uses separate MLPs to estimate the three sets of parameters, which appears more appropriate for stochastic processes.

All MLPs used are standard instantiations, i. e.

$$\mathrm{MLP}_i(x_{t-1}, \ldots, x_{t-m}) = \sum_{j=1}^{h} v_{ij} \tanh(\sum_{k=1}^{m} w_{jk} x_{t-k} + c_j) + b_i. \quad (7)$$

i is the index of the output neurons, h denotes the number of hidden neurons, w_{jk} and v_{ij} the weights of the first and second layer and c_j and b_i the biases of the first and second layer.

The values of the parameters of the MDNs were obtained by minimizing the negative logarithm of the likelihood function (Bishop, 1994) with a scaled gradient and a conjugate gradient algorithm:

$$\mathcal{L} = -\frac{1}{N} \log \prod_{t=m+1}^{m+N} \rho(x_t | x_{t-1}, \ldots, x_{t-m}). \quad (8)$$

If we assume for a moment that our MDN has only one component ($n = 1$), this function reduces to

$$\mathcal{L} = \frac{1}{N} \sum_{t=m+1}^{m+N} \left(\frac{1}{2} \log(2\pi\sigma_t^2) + \frac{(x_t - \mu_t)^2}{2\sigma_t^2} \right). \quad (9)$$

One can easily see that the standard MLP error function is a special case of Eq. (9) ($\sigma_t = const.$). To test the performance of the network on independent validation sets, the same function applied to the test data can be used as a loss or generalized error function.

3 Experimental Results

To evaluate whether our MDN is, in principle, able to identify a complex stochastic process with non-gaussian conditional target distributions, we first tested the network on an artificial data set with a known first order process, given by

$$\rho(x_t | x_{t-1}) = 0.2g(\mu_t + 0.01, \sigma_t^2) + 0.8g(\mu_t - 0.1, \sigma_t^2), \quad (10)$$
$$\mu_t = 3x_{t-1}(1 - x_{t-1}), \quad (11)$$
$$\sigma_t^2 = (0.05(x_{t-1}^2 + 0.1))^2. \quad (12)$$

This system is characterized by the following facts: Given the last value x_{t-1} the next value x_t is drawn from a bimodal distribution which is a weighted sum

of two gaussians. The conditional means are identical (except for a constant) and a non-linear function of the last value (the well-known logistic map for the parameter value (3) at the period 1 - period 2 bifurcation point). The conditional variance is the same for both gaussians and also a non-linear function of x_{t-1}. This complex data set is depicted on the left-hand side of Figure 1. Due to the priors in Eq. (10) about 80% of the data points belong to the "lower cloud" of points and about 20% to the upper one. Most points are clustered around 0.6. There can also be outliers, such as the one at $x_{t-1} \approx 0.3$.

The most important conclusion from this test was that using the initialization proposed by Bishop (1994) did not lead to satisfactory results, due to local minima in the likelihood function. In addition, the MDNs tended to drift into areas where the likelihood function (due to the unconstrained widths of the gaussians) tended toward infinity. Therefore, we initialized the MDNs by the resulting weight matrix from a simple MLP, trained on the conditional expectation under the assumption of gaussian noise of constant variance.

In the case of a MDN outputting a mixture of two gaussians, initialization can simply be done by transfering the weight matrix from the MLP to the corresponding MLP_2 of the MDN. In addition, the weight matrix of MLP_3 predicting the widths of the gaussians (see Eq. (6)) can be initialized to output the constant standard deviation of the target (by transfering the weight matrix from a MLP trained on this constant). In order to provide an appropriate seed for subsequent training of the MDN, the weight matrices of MLP_2 and MLP_3 of the MDN should be perturbed by gaussian noise of small variance.

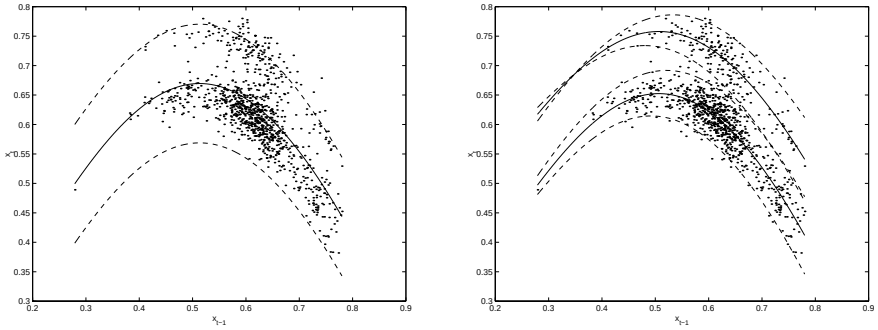

Figure 1: (Left) The data set (1000 points) generated by Eqs. (10)-(12) together with the conditional mean learned by a MLP (solid) and the resulting 95% confidence interval (dashed). (Right) The conditional means learned by a MDN (solid) and the learned 95% confidence intervals (dashed).

The results of the MLP used for initialization are depicted on the left-hand side of Figure 1. On the right-hand side of Figure 1 we show the training result for a MDN with one input (x_{t-1}), five hidden units in each MLP and two gaussians as outputs. All results are depicted with 95% confidence intervals. The bimodal conditional pdf with identical means (except for a shift) and increasing variances (and therefore increasing confidence intervals) is clearly visible. Figure 2 gives more details. The priors and the conditional means were learned with very high accuracy for both gaussians. The conditional variance of the lower

gaussian (with prior 0.8) is very close to the true conditional variance σ_t^2 specified by Eq. (12). For the other gaussian the character of σ_t^2, i. e. increasing with increasing x_{t-1}, was also clearly detected. We see that the conditional variance is close to zero for $x_{t-1} < 0.4$ because there is not a single data point in this region. In summary, the true structure of this complex process was revealed by the MDN.

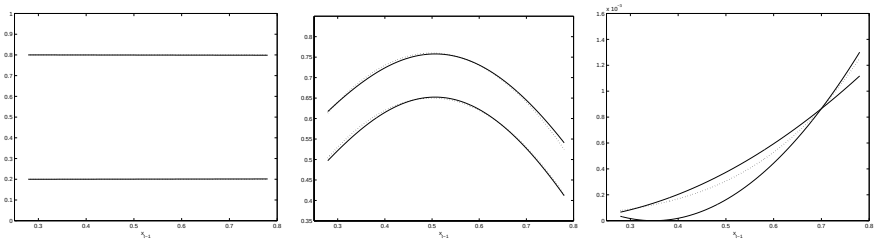

Figure 2: The true parameters (dotted) specified by Eqs. (10)-(12) and the parameters estimated by the MDN (solid): (Left) Priors. (Middle) Conditional means. (Right) Conditional variances.

In the actual experiment we applied MDNs to a real-world time series for prediction. The time series $\{x_t\}$ consisted of 2567 daily values of the Austrian stock exchange index ATX from 20 January 1986 to 16 June 1996. The data were preprocessed using the transformation $r_t = 100(\log x_{t+1} - \log x_t)$ and analyzed for temporal correlations. We found that for $\{r_t\}$ there is a significant autocorrelation only for lag one (0.34) whereas the autocorrelation function of $\{r_t^2\}$ shows a significant structure for all lags. In this paper we will choose the time series $\{r_t^2\}$ as a measure of volatility of the series $\{x_t\}$. Our goal is to model and predict the volatility of the ATX by MDNs and to compare their performance to the performance of the classical linear ARCH (Engle, 1982) and GARCH (Bollerslev, 1986) models where the conditional pdf is a gaussian with time-varying mean μ_t and time-varying variance σ_t^2, i. e. $\rho(x_t|x_{t-1}) = g(\mu_t, \sigma_t^2)$.

In order to measure the performance of the models reliably we used the concept of cross validation. More precisely, the time series was divided into ten subsequent intervals of equal length: $I_1 = (r_1, \ldots, r_{200}), \ldots, I_{10} = (r_{1800}, \ldots, r_{2000})$. The rest of the data $T = (r_{2001}, \ldots, r_{2566})$ was used as an independent test set (see the left-hand side of Figure 3). Each model was trained on nine of these ten intervals and the normalized loss function $\mathcal{L}_j$ (see Eq. (9)) on the missing interval I_j was calculated ($1 \leq j \leq 10$). Additionally, each model was trained on the whole training data set $I = (r_1, \ldots, r_{2000})$ and evaluated on the test set T. We fitted an ARCH(1) and a GARCH(1,1) model with $\mu_t = a x_{t-1}$ (due to the results of the correlation analysis). The conditional variance σ_t^2 of these models is given by $\sigma_t^2 = \alpha_0 + \alpha_1 (x_{t-1} - \mu_{t-1})^2$ and $\sigma_t^2 = \alpha_0 + \alpha_1 (x_{t-1} - \mu_{t-1})^2 + \beta_1 \sigma_{t-1}^2$, respectively. We also trained a MDN with one input (x_{t-1}), five hidden units (for each MLP) and one gaussian (MDN(1-5-1)) and a MDN with one input, five hidden units and two gaussians (MDN(1-5-2)) on the data sets mentioned above. The initialization procedure described above was applied to initialize the weights of MLP_3 (standard deviation of the target).

Our training results are summarized on the right-hand side of Figure 3 and in Table 1. For each model there are eleven marks. The jth mark (from left to right) indicates the error $\mathcal{L}_j$ on the test set I_j, $1 \leq j \leq 10$. The last mark gives the performance of the models (trained on I) on the independent test set T.

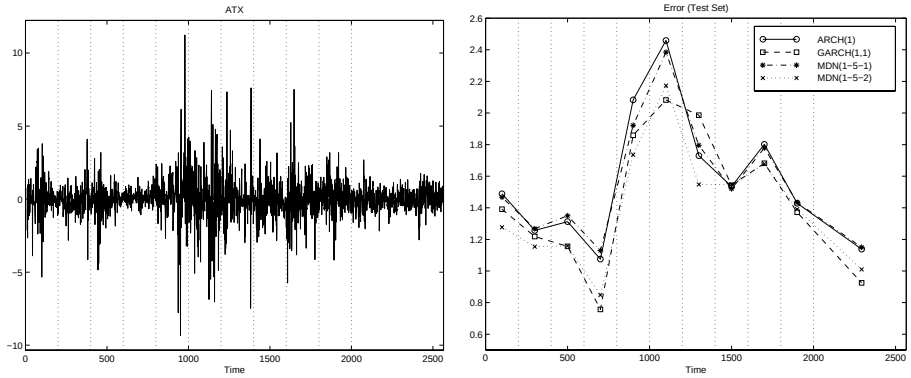

Figure 3: Left: The time series $\{r_t\}$ of the transformed Austrian stock exchange index ATX and its partition into training and test sets (indicated by dotted lines). Right: The normalized loss function on the test sets for the ARCH(1) models (circles), the GARCH(1,1) models (boxes), the MDNs(1-5-1) (stars) and the MDNs(1-5-2) (x-marks).

Table 1: The mean value and the standard deviation of the normalized loss function over ten runs (the ten test sets I_j) and the mean value (over the ten models) on the independent test set T.

MODEL	**MEAN**	**STD.**	**MEAN (T)**
ARCH(1)	1.6171	0.4147	1.139
GARCH(1,1)	1.5046	0.4105	0.920
MDN(1-5-1)	1.6051	0.3705	1.154
MDN(1-5-2)	1.4511	0.3718	1.017

4 Discussion

A two-way anova revealed that the difference between the MDN(1-5-2) and the MDN(1-5-1), on one hand, as well as the ARCH(1) model, on the other, tend to be statistically significant ($p < 0.002$)[1]. Therefore, one can conclude that assuming a non-gaussian conditional pdf tends to significantly improve the

[1] There are several reasons why applying an anova to these results bears some risk. First of all, a gaussian distribution of the values of the loss function would have to be guaranteed. Visual inspection revealed that this is only approximately the case. Secondly, one must apply the assumption that lower bias is better than lower variance, if models from different model classes are compared. Thirdly, performing several pairwise anovas for comparing several methods would require a correction of the F-values to account for multiplicity effects. Therefore

identification of the underlying process. Furthermore, non-linear neural network models tend to be superior to traditional linear models.

Taking a closer look at the widely used GARCH(1,1) model (which was not significantly worse than the MDNs) reveals that using the previous estimation of σ_t^2 amounts to a kind of recurrent connection in the model, viewed in neural network terms. Therefore, a fair comparison would require a recurrent extension of the MDN, which is currently under investigation.

Figure 3 also reveals a large variance in the results due to an obvious change in structure at time $t \approx 950$. Despite this non-stationarity, the models perform reasonably well on average.

5 Conclusion

We have presented experimental results from applying mixture density networks to the identification of complex stochastic processes. The results point toward the viability of such an approach, indicating that assuming non-gaussian conditional target (noise) distributions can lead to more intricate identification of the underlying process. They also point to advantages of neural networks as non-linear estimators. Future research will investigate recurrent extensions of MDNs (in the realm of the well-known GARCH models), as well as the evaluation of different trading strategies (e.g. for option pricing) to validate whether the improved identification of the process can be used to extract more information about the underlying behavior.

Acknowledgements

The implementation of the MDNs is based on the NETLAB software written by I. Nabney and C. Bishop (`http://neural-server.aston.ac.uk/`). This work was supported by the Austrian Science Fund (FWF) within the research project "Adaptive Information Systems and Modelling in Economics and Management Science" (SFB 010). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Transport. We thank F. Leisch and A. Weingessel for valuable discussions and comments.

References

Bishop, C.M. (1994) Mixture density networks. *Neural Computing Research Group Report: NCRG/94/004.* Birmingham: Aston University.

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

Bollerslev, T. (1986) A generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics* **31**:307-327.

Engle, R.F. (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of U.K. inflation. *Econometrica* **50**:987-1008.

Hornik, K., Stinchcombe, M. & White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **2**(5):359-366.

the conclusions have to be handled with care and be viewed as the best estimation possible for the time being.

McLachlan, G.J. & Basford, K.E. (1988) *Mixture models: inference and applications to clustering.* New York: Marcel Dekker.
Neuneier, R., Finnoff, W., Hergert, F. & Ormoneit, D. (1994) Estimation of conditional densities: a comparison of neural network approaches. In M. Marinaro and P. G. Morasso (eds.), *ICANN 94 - Proceedings of the International Conference on Artificial Neural Networks*, pp. 689-692. Berlin: Springer.
Ormoneit, D. & Neuneier, R. (1995) Reliable neural network predictions in the presence of outliers and non-constant variances. In A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (eds.), *Proceedings of the Third International Conference on Neural Networks in the Capital Markets, London, England.*
Schittenkopf, C. & Deco, G. (1997) Testing nonlinear Markovian hypotheses in dynamical systems. *Physica D* **104**(1):61-74.