Measuring DAX Market Risk: A Neural Network Volatility Mixture Approach

Kai Bartlmae, Folke A. Rauscher
DaimlerChrysler AG, Research and Technology – FT3/KL,
P. O. Box 2360, D-89013 Ulm, Germany
E–mail: {kai.bartlmae, folke.a.rauscher}@daimlerchrysler.com

February 2, 2000

Abstract

In this paper we propose a framework for estimation and quality control of conditional neural network volatility models for market risk management. In a first step, we derive a conditional volatility model based on gaussian mixture densities, that can be used with linear or neural regression models (extendable even to rule systems or decision trees). In a second step, we introduce performance measures, that measure two different properties of the models' volatility forecasts important to risk-management. The proposed framework is being tested on daily DAX (German stock index) data. Results show, that the neural network volatility mixture approach outperforms GARCH models.

1 Introduction

One of the fundamental assumptions used today by risk measurement systems is that the underlying returns on financial processes are distributed according to a conditional normal distribution (i.e. JP Morgan's RiskMetrics). This assumption has been a focus point of today's research on risk measurement systems because the distribution of many observed financial return series have tails that look heavier than the tails that are implied by the conditional normality assumption. In those cases the models may underestimate the risk as if the return would follow a true conditional normal distribution. It is therefore an important issue to be able to calibrate models in the way that they account for the possibility of such large returns. Here the method of neural volatility mixture models based on gaussian mixtures is used. Because of its straightforward extension of the standard conditional normal models and its modeling power, gaussian mixtures models can be used to model any arbitrary density.

In a first step we derive the neural volatility mixture model. The framework incorporates topics from three disciplines: A component-oriented conditional mixture density approach known from statistics, a neural network approach from AI modeling the mixture components, and an input feature representation known from econometrics. In a second step, two groups of models for forecasting one-day DAX volatility are presented and compared: GARCH and the here proposed neural network volatility models.

2 A Mixture Model Framework

Mixtures of normal distributions are used by statisticians in many academic disciplines. Here the use of these mixture models in finance and risk-management is further investigated. Especially in the case of market risk management mixture distributions appear in Zangari [Zangari 1996], Venkatamaran [Venkatamaran 1997], and Locarek-Junge/Prinzler [Locarek-Junge, Prinzler 1998]. They all concluded, that the use of these models proved fruitful in modeling financial returns. While Zangari and Venkatamaran used unconditional mixture models, Locarek-Junge/Prinzler used one neural network (called Mixture Density Network) to model the density conditionally.

Here a different model is proposed, that estimates each of the mixtures parameters separately: The model is conditional and parts of it are based in AI. It is component-orientated and can be used with a

variety of different modeling approaches from statistics and AI. Prior knowledge and prior expectations for each mixture components parameter can be used to tailor the method used to model the parameter. Further the specification on density estimation for a return time series is given.

Here we are interested in a problem of density estimation and consider the conditional distribution $p(y|\mathbf{x})$. In other words, we are interested in the distribution of a target y given the values of a feature vector **x**. One suitable approach for modeling complex conditional densities is a mixture representation. The idea is to use M simple conditional component densities with known properties building or approximating more complex densities. Using this approach, one is interested in modeling the observed data as a sample from a mixing density:

$$p(y|\mathbf{x}) = \sum_{j=1}^{M} g_j(\mathbf{x}) p(y|\mathbf{x}, j)$$

where $g_j(\mathbf{x})$ is a conditional mixing proportion and can be regarded as prior probability (conditioned on **x**) on the target y having been generated from the jth component. The $p(y|\mathbf{x}, j)$ are the component densities, generally taken from a simple parametric family. Here Gaussian component densities have been chosen. The probability of generating a value y by component j is given by:[1]

$$p(y|\mathbf{x}, j) = \frac{1}{\sqrt{2\pi\sigma_j^2(\mathbf{x})}} exp(\frac{-(y - \mu_j(\mathbf{x}))^2}{2\sigma_j^2(\mathbf{x})})$$

We propose a framework that takes the various conditional parameters of the mixture model, namely the mixing proportion $g_j(\mathbf{x})$, the means $\mu_j(\mathbf{x})$ and the variances $\sigma_j^2(\mathbf{x})$ and bases each of them on general, continuos and differentiable functions $f_j^s(\mathbf{x}, \Theta_j^s)$ in **x**, $s \in \{\mu, \sigma, g\}$ with parameter vector Θ_j^s.[2] The jth mixture components parameters are based on the three functions $f_j^\mu(\mathbf{x})$, $f_j^\sigma(\mathbf{x})$ and $f_j^g(\mathbf{x})$. Here any function or approximator $f_j^s(\mathbf{x})$ that can be estimated using gradient information of the later derived cost-function can be used, i.e. linear, non-linear models or neural networks.

These functions are defined as follows, enforcing certain constraints for the values of $g_j(\mathbf{x})$ and $\sigma_j^2(\mathbf{x})$:

$$\mu_j(\mathbf{x}, \Theta_j^\mu) = f_j^\mu(\mathbf{x}, \Theta_j^\mu)$$

In order to satisfy the constraint $\sum_{j=1}^{M} g_j(\mathbf{x}) = 1$ and $g_j(\mathbf{x}) > 0$, the actual value of $g_j(\mathbf{x})$ is calculated

$$g_j(\mathbf{x}, \Theta_j^g) = \frac{exp(f_j^g(\mathbf{x}, \Theta_j^g))}{\sum_{i=1}^{M} exp(f_i^g(\mathbf{x}, \Theta_i^g))}$$

Further to enforce positive values for σ^2, the actual value is calculated

$$\sigma_j^2(\mathbf{x}, \Theta_j^\sigma) = exp(f_j^\sigma(\mathbf{x}, \Theta_j^\sigma))$$

For a vector **x**, this mixture model framework gives a general approach for modeling an arbitrary conditional density function $p(y|\mathbf{x})$. See figure 1 for an system-overview of the proposed framework.

[1] Here we assume independence between each component of the distribution.

[2] From hereon called $sub\text{-}models$. We omit the Θ_j^s when they are not explicitly needed.

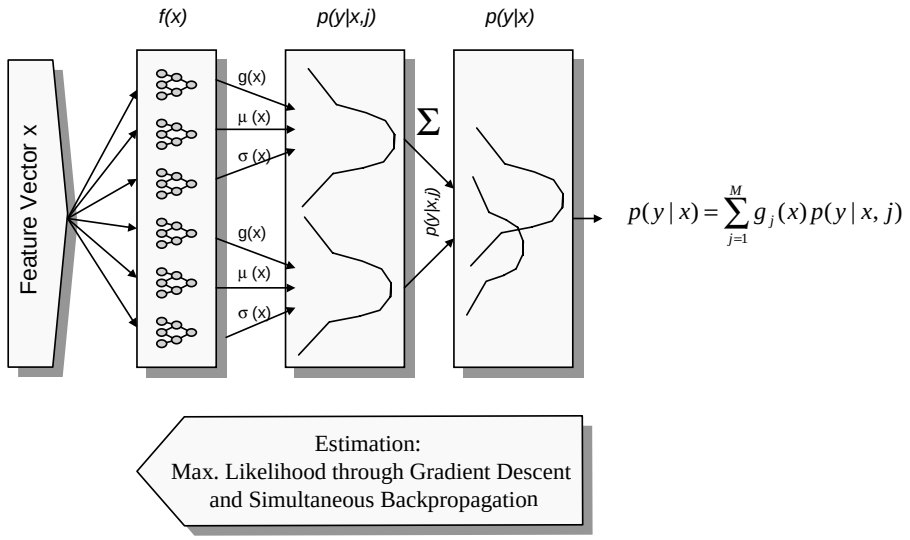

Figure 1: System representation of the proposed framework. The sub-models are used to parameterize a gaussian mixture model. The sub-models are estimated through gradient descent maximizing the mixture models likelihood on the in-sample data.

2.1 Estimation

In order to estimate the parameters of each sub-model, we use the dataset $D_x = (\mathbf{x}_1, ..., \mathbf{x}_T)$, the targets $D_y = (y_1, ..., y_T)$ and obtain the full likelihood by taking the product over the likelihood of the individual samples (assuming that the training data is being drawn independently from the mixture distribution):

$$L = \prod_{t=1}^{T} p(y_t|\mathbf{x}_t) \tag{1}$$

$$= \prod_{t=1}^{T} \sum_{j=1}^{M} g_j(\mathbf{x}_t) p(y_t|\mathbf{x}_t, j) \tag{2}$$

$$= \prod_{t=1}^{T} \sum_{j=1}^{M} g_j(\mathbf{x}_t) \frac{1}{\sqrt{2\pi\sigma_j^2(\mathbf{x}_t)}} \exp\left(\frac{-(y_t - \mu_j(\mathbf{x}_t))^2}{2\sigma_j^2(\mathbf{x}_t)}\right) \tag{3}$$

For this equation, the suggestion is to take an estimator for each Θ_j^s, that minimizes the negative log-likelihood. We define an error function to be the negative logarithm of the likelihood function:

$$E = -\ln L \tag{4}$$

$$= -\ln\left(\prod_{t=1}^{T} \sum_{j=1}^{M} g_j(\mathbf{x}_t) \frac{1}{\sqrt{2\pi\sigma_j^2(\mathbf{x}_t)}} \exp\left(\frac{-(y_t - \mu_j(\mathbf{x}_t))^2}{2\sigma_j^2(\mathbf{x}_t)}\right)\right) \tag{5}$$

In order to minimize E we apply an advanced gradient descent algorithm which uses information about the partial derivatives $\frac{\partial E}{\partial f_j^s}$ of the error function with respect to the parameters of the mixture model and each sub-model.[3] Using the chain-rule, for each parameter in Θ_j^s of function f_j^s, the gradient according to the error-function E can be calculated and the parameters be adjusted accordingly. This is possible since we chose the functions to be differentiable with respect to each parameter in Θ_j^s.

E can be minimized with many numerical-search algorithms. Here we use the optimization-algorithm RPROP [Riedmiller, Braun 1993]. The estimation takes place[4] in the following two steps: First, for each parameter in the whole framework, the error-gradient for all samples is being calculated. Second, all parameters are updated according to the RPROP-algorithm. The estimation ends, when in the following update-step almost no improvement in the likelihood can be achieved.

[3] See Bishop for the correct formulation of the gradients.

[4] After a random initialization of the parameters.

3 Specification for Financial Return Time-Series

A further enhancement of this approach is based on its intended use: Modeling the density of a (daily) return time-series $D_y = (r_1, ..., r_T)$.

Inspired by simple GARCH models, we are basically interested in modeling volatility. Here the component-oriented approach comes into place: We chose appropriate functions $f_j^s(\mathbf{x})$ which in our opinion fit the sub-model's needed expressiveness best. Here prior expectation and knowledge sets in. Since volatility is our major concern, we select all $f_j^\sigma(\mathbf{x})$ from the function-classes of neural networks and/or simpler linear models. On the other hand, we want to avoid changing values in $g_j(\mathbf{x})$ and $\mu_j(\mathbf{x})$, if good mixture models can be found with this restriction.[5] Therefore, in a first step we select all $f_j^\mu(\mathbf{x})$ and $f_j^g(\mathbf{x})$ to be estimated constants.

Further we are using an input vector representation $\mathbf{x}$ that is based on simple GARCH-models and is used by major market risk management systems. So far we only model $f_j^\sigma(\mathbf{x})$ conditionally on $\mathbf{x}$. Let $\sigma_{j,t}^2$ be the value of $\sigma_j^2(\mathbf{x}_t)$ at time step t. The specification of the components variance $\sigma_{j,t}^2$ is then modeled as a function of past values:

$$\sigma_{j,t}^2 = \sigma_j^2(\mathbf{x}_t)$$

with $\mathbf{x} = (\sigma_{j,t-1}^2, r_{t-1})$.

For the simpler functions, we decided to estimate a constant parameter for now, leading to the specification of $f_j^\mu(\mathbf{x})$ and $f_j^g(\mathbf{x})$:[6]

$$f_j^\mu(\mathbf{x}) = c_j^\mu$$
$$f_j^g(\mathbf{x}) = c_j^g$$

This specification can now be changed in a very flexible way by choosing other functions for $f_j^s(\mathbf{x})$, by using another input vector $\mathbf{x}$ or changing the number of mixing components.

3.1 An autoregressive artificial neural network volatility model

Artificial neural networks(ANN) are a class of non-linear regression models, that can flexibly be used as parametric, semi- and non-parametric methods. Their primary advantage over more conventional techniques lies in their ability to model complex, possibly non-linear processes without assuming prior knowledge about the data-generating process. This makes them particular suited for financial applications. In this area such methods showed much success and are therefore often used.

The neural models used here are called feed-forward single hidden-layer neural networks. They consist of three layers: The input layer, which is connected to the output layer through the hidden layer. The transfer function in the hidden layer is *tanh*, whereas for the output unit a linear transfer function is applied.
In this case a mixture model with up to three gaussians (and therefore three neural models) are estimated. In each case the variance is modeled as a function of past values of the dependent variables (see above).

4 GARCH

Autoregressive Conditional Heterskedasticity (ARCH) models are designed to forecast and model conditional variances. The variance of the dependent variable is modeled as a function of past values of the dependent variable (added by independent, or exogenous variables).

ARCH models were introduced by Engle [Engle 1982] and generalized as GARCH by Bollerslev [Bollerslev 1986]. These models are widely used in various branches of econometrics, i.e. in financial time series analysis. In an ARCH model, two distinct specifications are considered: One for the

[5] Zangari and Venkatamaran enforce this restriction further by selecting specific values through a Bayesian prior.
[6] If one is more interested in forecasting the mean, a neural net for $\mu_j(\mathbf{x})$ would be more appropriate.

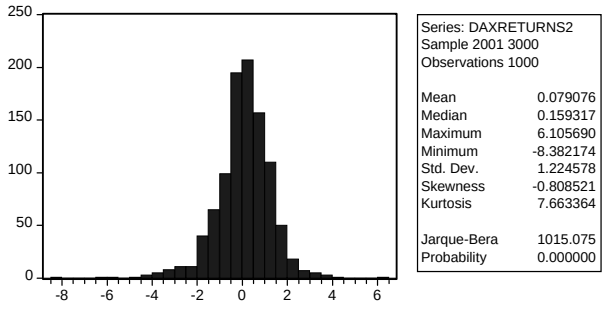

Figure 2: Histogram and statistics of the out-of-sample DAX-data

conditional mean and one for the conditional variance. These models are estimated in a Maximum Likelihood approach.

In the case of J.P Morgan's RiskMetrics, the used models are basically non-stationary GARCH(1,1) models, whose coefficients are preset by the RiskMetrics approach. Dave and Stahl [Dave, Stahl 1997] observed, that RiskMetrics models perform in tail-measures equal or worse than GARCH(1,1) models. Therefore, here only GARCH(1,1) is investigated.

5 Forecast Evaluation

We are interested in the performance of the proposed models with respect to market risk measurement. Hence, we evaluate Value-at-Risk like measures to asses the performance of the models i.e. by counting the number of loss-exceedances over the calculated Value-at-Risk with respect to a given percentile level (see Duffie and Pan [Duffie, Pan 1997] for an overview on Value-at-Risk). While Zangari and Venkatamaran only calculate the performance of their models against one or two of these percentile levels, we use a more sophisticated approach. Here we want to enhance this measure to give a better overview of the models performance: We calculate the exceedances of percentile not only for one or two confidence levels, but for all levels. Further we use a measure *Mean log likelihood against volatility percentile* introduced by Dave and Stahl [Dave, Stahl 1997]. This measure calculates the performance of the model for movements exceeding a specific absolute size.

5.1 Observed/predicted exceedance ratio against confidence level

This measure counts the events of a loss of assets that exceed the predicted loss by the model given a confidence level c over a time-interval (one day). This count is normalized by the expected count of exceedances. The measure relates very closely to backtesting VaR exceedances at a confidence level c. Here we backtest not only to one but all confidence levels c in order to get a better overview. If a model performs adequate, this measure should be around one. A measure of more than one stems from a model, that exceeded more often than expected and is therefore worse from a risk managers standpoint than one that has a ratio below that threshold value of one.

5.2 Mean log likelihood against volatility percentile

The second measure calculates the mean log-likelihood over all events that exceed a certain volatility percentile, speaking a return $|r_t|$. For a all events $|r_t|$ that exceed a volatility percentile c the average log-likelihood is being calculated: All models are being compared to the same events rather than compared to a certain confidence level. This measure tests the models predictions to the tail of the empirical unconditional distribution and has the advantage to measure the performance of the models in (for the risk management important) extreme events. Here a higher value of the measure stands for a better performing model. Again, all volatility percentiles are observed.

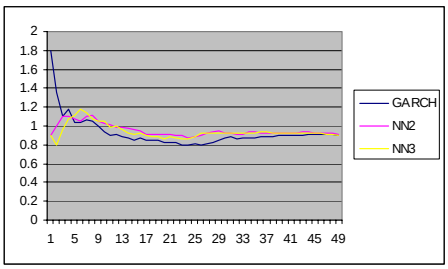

Figure 3: Long DAX portfolio: observed/predicted exceedance ratio against confidence level

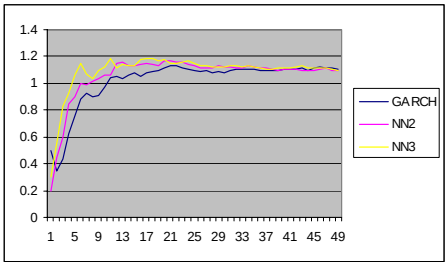

Figure 4: Short DAX portfolio: observed/predicted exceedance ratio against confidence level

6 Application and Results

The proposed framework was being tested on daily DAX (German stock index) data. The data used in this section were 3001 daily closing values of the DAX from 27.3.1987 to 28.9.1998. The returns were calculated according to:

$$r_t = 100(\log(S_{t-1}) - \log(S_t))$$

The data has been divided into two sets, an in-sample part with 2000 and an out-of-sample part with 1001 returns. An overview of the out-of-sample data can be seen in figure 2 with additional statistics. It is visualized, that the normal-assumption can be clearly rejected (The Jarque-Bera statistic tests whether a series is normally distributed).

Now three models were estimated, GARCH(1,1) models as a benchmark and two models with two (here called *NN2*) and three (here called *NN3*) mixtures. These models used neural nets with eight hidden units. The results of the models will be compared in the next section. The measures were evaluated on a long and a short DAX portfolio.

6.1 Observed/predicted exceedance ratio against confidence level

In figure 3 and 4 the results of the long and short DAX portfolio can be seen. The *Observed/Predicted exceendance ratio against confidence level*-measure shows for the long DAX-portfolio that our proposed approach leads to a better performance of the volatility-models. This becomes especially clear for the lower percentile levels. The lower percentile levels are more interesting, as here one can find the usually used 1%, 2,5% and 5% percentiles (see also figure 7). It can be seen, that the exceedances of the GARCH-model has up to twice as much exceedances as excepted. On the other hand the mixture models show a good performance having a ratio around one with no large difference between the models. Investigating the short DAX portfolio, all models show a comparable performance. While all models have a ratio below one, they all have an intrinsic value for risk management (avoidance of exceedances). Comparing the models for the used 1% and 5% percentiles in figure 7, it can be seen, that for the portfolios the mixture approach performs more conservative then the GARCH approach for the important very small levels of c.

6.2 Mean log likelihood against volatility percentile

If one observes figure 5 and 6, it shows that in the case of large DAX-moves, the models of the proposed framework perform better in the case of a long DAX portfolio than the GARCH model and showing

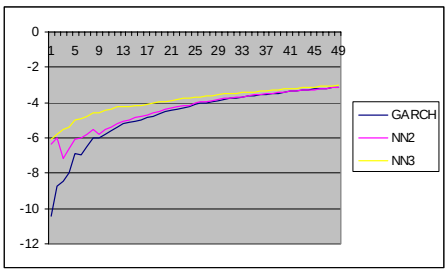

Figure 5: Long DAX Portfolio: Mean log likelihood against volatility percentile

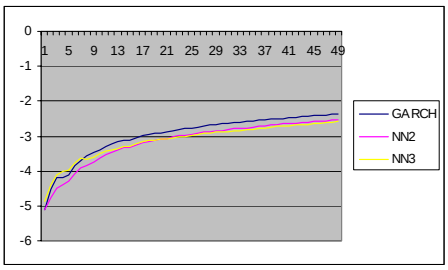

Figure 6: Short DAX portfolio: Mean log likelihood against volatility percentile

a higher mean log-likelihood. For the short portfolio a comparable performance of all models can be observed. In figure 8 the results for the volatility percentiles 1% and 5% can be noticed, showing an advantage for the neural volatility models.

6.3 Conclusion

The results calculated in the performance evaluation strongly suggest that there is support for the validity for the proposed volatility modeling framework in the case of risk management. Figure 8 shows, that extreme events are better forecasted by the neural volatility models. Also considering the daily exceedance, the performance is in favor for the proposed models, which show less exceedances for the very small levels of c (figure 7).

7 Summary and Further Work

In this paper, we introduced a framework for neural network volatility modeling and applied it successfully to market risk management. In a first step we derived a framework for estimating a neural network volatility model based on a mixture of gaussians. This framework extends the ideas of Zangari and Venkatamaran as well as Bishop and Locarek-Junge/Prinzler. Further this framework has been applied to a task in market risk management, measuring the risk of a DAX portfolio.

The comparison showed that the proposed framework performs in a risk-managers sense better than the often used simple GARCH-models. The new models showed to be more conservative and underestimated risk less often than the GARCH models for very small levels. A comparison of the average likelihood of the models in extreme events further supported these findings.

Portfolio	GARCH	NN2	NN3
1% Long	1.8	0.9	0.9
5% Long	1.04	1.08	1.12
1% Short	0.5	0.2	0.3
5% Short	0.76	0.9	1.06

Figure 7: Observed/predicted exceedance ratio against 1% and 5% level

Portfolio	GARCH	NN2	NN3
1% Long	-10.43	-6.35	-6.08
5% Long	-6.91	-6.08	-5.01
1% Short	-5.12	-5.10	-4.88
5% Short	-4.12	-4.29	-3.98

Figure 8: Mean log likelihood against 1% and 5% volatility percentile

Still further research has to be conducted on the proposed framework like investigating, how more complex mixture models perform on these tasks as well as how these models perform, if the whole density forecast is considered. Of further interest is the incorporation of these models into the known portfolio-approaches considering correlations. In addition, selecting the right architecture of neural networks is still to be investigated in the area of statistical selection theory.

References

[Bishop 1994] Bishop,W.: *Mixture Density Networks*, Technical Report NCRG/94/004, Neural Computing Research Group, Aston University, Birmingham, February 1994

[Bollerslev 1986] Bollerslev, T.: *Generalized Autoregressive Conditional Heteroskedasticity*, Journal of Econometrics 31:307-327, 1986

[Dave, Stahl 1997] Dave, R., Stahl., G.: *On the accuracy of VaR estimates based on the Variance-Covariance approach*, Working Paper, Olsen and Associates, Zurich, Switzerland

[Duffie, Pan 1997] Duffie, D., Pan, J.: *An overview on value at risk*, The Journal of Derivatives, 4, pp. 7-49

[Engle 1982] Engle, R.: *Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation*, Econometrica 50:987-1008, 1982

[Hamilton 1991] Hamilton, J.: *A quasi-bayesian approach to estimating parameters for mixtures of normal distributions*, Journal of Business and Economic Statistics, Vol 9, No.1 1991 pp.27-39

[Jordan, Bishop 1996] Jordan, M., Bishop, C.: *Neural Networks*, in CRD Handbook of Computer Science, Tucker, A. (ed.), CRC Press, Boca Raton, 1996

[Locarek-Junge, Prinzler 1998] Locarek-Junge, H., Prinzler, R.: *Estimating Value-at-Risk Using Neural Networks*, In G. Nakhaeizadeh, E. Steurer (Ed.), Application of Machine Learning and Data Mining in Finance, ECML'98 Workshop Notes (24. April 1998), Chemnitz

[Riedmiller, Braun 1993] Riedmiller, M., Braun, H.: *A direct adaptiv method for faster backpropagation learning: The RPROP algorithm*, Proceedings of the IEEE International Conference on neural networks, 1993

[Venkatamaran 1997] Venkatamaran, S.: *Value at risk for a mixture of normal distributions: The use of quasi-Bayesian estimation techniques*, Economic Perspectives (Federal Reserve Bank of Chicago), 1997 (March/April), pp. 3-13

[Weigend, Mangeas, Srivastava 1995] Weigend, A., Mangeas, M., Srivastava, A. : *Non-linear gated experts for time series. Discovering regimes and avoid overfitting*, International Journal of Neural Systems, Vol. 6, No. 4, 1995, pp. 373-399

[Zangari 1996] Zangari, P.: *An improved methodology for measuring VaR*, in RiskMetrics Monitor 2, 1996