Making Density Forecasting Models Statistically Consistent

Michael Carney[1], Pádraig Cunningham[1] and Brian M. Lucey[2]

[1] Department of Computer Science, Trinity College Dublin, Ireland

[2] School of Business Studies, Trinity College Dublin, Ireland

Abstract. We propose a new approach to density forecast optimisation and apply it to Value-at-Risk estimation. All existing density forecasting models try to optimise the distribution of the returns based solely on the predicted density at the observation. In this paper we argue that probabilistic predictions should be optimised on more than just this *accuracy* score and suggest that the *statistical consistency* of the probability estimates should also be optimised during training. Statistical consistency refers to the property that if a predicted density function suggests P percent probability of occurrence, the event truly ought to have probability P of occurring. We describe a quality score that can rank probability density forecasts in terms of statistical consistency based on the probability integral transform (Diebold et al., 1998b). We then describe a framework that can optimise any density forecasting model in terms of any set of objective functions. The framework uses a multi-objective evolutionary algorithm to determine a set of trade-off solutions known as the *Pareto front* of optimal solutions. Using this framework we develop an algorithm for optimising density forecasting models and implement this algorithm for GARCH (Bollerslev, 1986) and GJR models (Glosten et al., 1993). We call these new models Pareto-GARCH and Pareto-GJR. To determine whether this approach of multi-objective optimisation of density forecasting models produces better results over the standard GARCH

and GJR optimisation techniques we compare the models produced empirically on a Value-at-Risk application. Our evaluation shows that our Pareto models produce superior results out-of-sample.

Keywords. Density Forecasting, Statistical Consistency, Calibration, Empirical Validity, Multi-objective Optimization/Optimisation, Probability Integral Transform, Value-at-Risk.

1 Introduction

A density forecast is an estimate of the probability distribution of the possible future values of a numeric variable. Density forecasting is of particular use in finance and has many applications e.g. estimating Value-at-Risk (Manganelli & Engle, 2001), options pricing (Christoffersen & Jacobs, 2004) and high-frequency trading (Diebold et al., 1998a). Further, the variance or volatility in financial data is characteristically predictive and can be accurately estimated making density forecasting possible and useful. How to produce, use and evaluate this type of forecast has become a substantial subfield of the literature in finance.

Conventional approaches to solving time series prediction problems optimise models based on the sum of squared deviations and predict a single value as a function of given inputs. The model thereby learns the conditional mean of the target given an input vector. When the underlying conditional distributions of the data are homoskedastic and Gaussian this is a suitable approach. However, if these distributions are in fact heteroskedastic, skewed or even multimodal, as is often the case in financial data, the conditional mean does not capture all the influences in the system. Probability density estimators can predict the shape of the conditional probability density functions of a series with non-constant variance and can even approximate arbitrary non-Gaussian conditional densities.

The richer information provided by a density forecasting model over a point prediction creates an issue when evaluating its predictions. Intuitively one would

postulate that having a complete probability density function introduces another dimension (other than density at the observation) that must be evaluated. However, the negative log-likelihood (NLL) or negative log predictive density (Good, 1952) is the most commonly used loss function in the density forecasting literature and it only considers density at the observation. The NLL is described as follows:

$$NLL_t = -log(p(y_t|\Omega_t)) \tag{1}$$

where, $p(y_t|\Omega_t)$ is the conditional density function of y_t given $\Omega_t = y_{t-1}, y_{t-2}, ..., y_1$ the information set.

The major weakness of this error function is that it evaluates density estimates based solely on the probability density at the observation and does not take the statistical consistency[3] of the forecast into consideration[4]. Statistical consistency refers to the property that if a predicted density function suggests P percent probability of occurrence, the event truly ought to have probability P of occurring.

It stands to reason that density forecasts should be both accurate and statistically consistent. "*Accuracy*" means the predicted densities produced have a high density (small variance) around the observation and "*statistically consistent*" means that the model has empirically valid probabilities in the long run e.g. predictions of 75% should be correct $\frac{3}{4}$ of the time. The NLL as an error function addresses the first objective, accuracy, however, it neglects assessment of statistical consistency. In most research in this area statistical consistency is

[3] Also known as calibration (Gneiting et al., 2003) or empirical validity (Seillier-Moiseiwitsch & Dawid, 1993). To maintain consistency throughout this paper we will use only statistical consistency.

[4] A further weakness of NLL is its sensitivity to outliers. This is due to the fact that a change of k in the NLL relates to a change of $\exp(k)$ in the observation values. (Weigend & Shi., 2000) suggests using a trimmed mean to get around this issue. This ameliorates the situation but does not solve the problem.

either evaluated ex-post or is ignored completely. The literature suggests few techniques to assess statistical consistency with the Probability Integral Transform (PIT)(Rosenblatt, 1952) being the only technique used with regularity.

In this study we have reformulated the optimisation of density forecasting models to be a multi-objective search task. We describe a new approach to optimising density forecasting models that uses the PIT and the NLL during training to find a set of *Pareto* optimal solutions. By taking statistical consistency into consideration during optimisation we aim to produce density forecasting models that perform better out-of-sample.

The paper is laid out as follows. In Section 2 we analyse the density forecasting optimisation problem and introduce a new density forecast quality score. In Section 3 we outline our framework for optimization of density forecasts and describe in detail a specific implementation of the framework. Section 4 is a case study on the performance of our optimisation technique for a Value-at-Risk estimation problem. We compare the models produced using our search technique against the standard constrained nonlinear optimisation technique. Section 5 concludes the paper.

2 Density Forecasting: Evaluation and Optimisation

The issues described in Section 1 stem from the fact that the prediction produced by a density forecasting model can rarely be compared to the true generating distribution in real world problems. Instead, only a single instance of the generating distribution, the actual outcome, is available to the forecaster to optimise and evaluate their model. Conventional diagnostics for evaluating point predictions such as the root-mean-squared error fail to assess probabilistic predictions. Furthermore, the ranking of different density forecasting models is difficult because a ranking depends on the loss function of the user (Diebold et al., 1998b).

For example, a user's loss function could be non-linear and/or asymmetric. In such cases the mean and variance of the forecast densities are not sufficient to rank predictive models. For example, a user with an asymmetric loss function would be particularly affected by the accuracy of a model's predictions of the skew in the conditional densities.

Diebold et al. (1998b) suggests that the problem of ranking density forecasts can be solved by assuming that the correct density is always preferred to an incorrect density forecast. Using the true density as a point of reference it is possible to rank densities relative to the true densities to determine the best models to use. Therefore, in the absence of a well defined loss function, the best model is the one that approximates the true density as well as possible. Diebold et al. (1998b) go on to suggest the probability integral transform (Rosenblatt, 1952) as a suitable means of evaluating density forecasts in this way. The main power of the PIT is its ability to assess the statistical consistency of the forecast densities.

The PIT score is defined as:

$$z_t = \int_{-\infty}^{y_t} p_t(u|\Omega_t)\,du \tag{2}$$

For a series of length m the probability of those events occurring in their predicted densities should result in a random sample as would appear using the true generating densities. Diebold et al. (1998b) shows that this random sample will be $U(0,1)$ and $i.i.d.$ for the true generating density and any correctly specified density forecasting model.

$$\{z_t\}_{t=1}^{m} \overset{iid}{\sim} U(0,1) \tag{3}$$

A test for statistical consistency relates directly to a test for whether the z values for predictions follow a uniform distribution. Therefore, a number of possible goodness-of-fit tests can be carried out to determine statistical consistency. Diebold et al. (1998b) argues that using the PIT values in this way is uninformative because "*Such tests [...] are not likely to be of much value in practical applications, because they are not constructive; that is, when rejection occurs, the tests generally provide no guidance as to why.*". However, if you are only interested in ranking the density forecasting models in terms of their statistical consistency the reason "why" the model is not statistically consistent is not important - this is the case during model optimisation. When the PIT is being used to evaluate a model it makes sense to take a more detailed look at the series of z values. At the evaluation stage it is suggested that visual assessment of the z series is useful. A histogram of the series is generally used because of the ease of verification of the requirement that z is uniform over the unit interval (see Figure 9 for example). Also, assessment of the *iid* requirement can be achieved through plotting of the autocorrelation in the series.

At this point we should contextualise the problem again by referring back to our initial postulation that density forecasts should be both accurate and statistically consistent. It stands to reason that given two statistically consistent models, any rational user would prefer the system that produces the more specific forecasts. The PIT does not evaluate accuracy, so it must be used in conjunction with a score that can evaluate on this dimension, such as the NLL. Furthermore, a uniform z series is a necessary but not sufficient criterion for determining that the model is reliable (Hamill, 2000). It is possible that an incorrect density model could have a uniform z series. In this case the accuracy score (NLL) should highlight the existence of an incorrectly specified model. However, it is important to be aware of this point.

2.1 Interpreting PIT Histograms

To further understand the PIT approach to determining statistical consistency this section describes the effect of bias and variance[5] on the PIT histogram. To do this we simplify the problem; all data points in our test series are random samples from an $N(0,1)$ distribution. The true conditional density at every point is therefore an $N(0,1)$ density. Knowing the true density means we can artificially simulate bias and variance in the predicted densities by making all predicted densities $N(\mu,\sigma)$. Bias is simulated by varying the μ of the density and variance is simulated by varying the σ of the density. The μ values tested were -1.5, -1, -0.5, 0.25, 0, 0.25, 0.5, 1, and 1.5. The σ values tested were 0.5, 0.75, 1.0, 1.5, and 2.0.

Figure 1. shows the resulting PIT histograms for each μ and σ pair. 10,000 points randomly sampled from the data generating distribution $N(0,1)$ were used to determine each PIT histogram. The PIT histogram in the middle of the grid is the only correctly specified histogram because the predicted density is $N(0,1)$, the same as the distribution used to generate the data. Bias and variance affect the PIT histogram in different ways. Too narrow a variance forms a "U" shaped PIT, this can be thought of as over-confident density estimates. Too wide a variance creates a hump in the middle of the PIT, this can be thought of as under-confidence. Bias causes a sloping effect and in the extreme case it creates a "J" or "L" shaped PIT histogram. Knowing this, a practitioner can make a fast examination of the quality of the predicted densities and diagnose common model problems e.g. over-confidence.

[5] In this experiment, bias refers to the incorrect specification of the mean in the predicted density and variance refers to the incorrect specification of the variance or standard deviation of the predicted density.

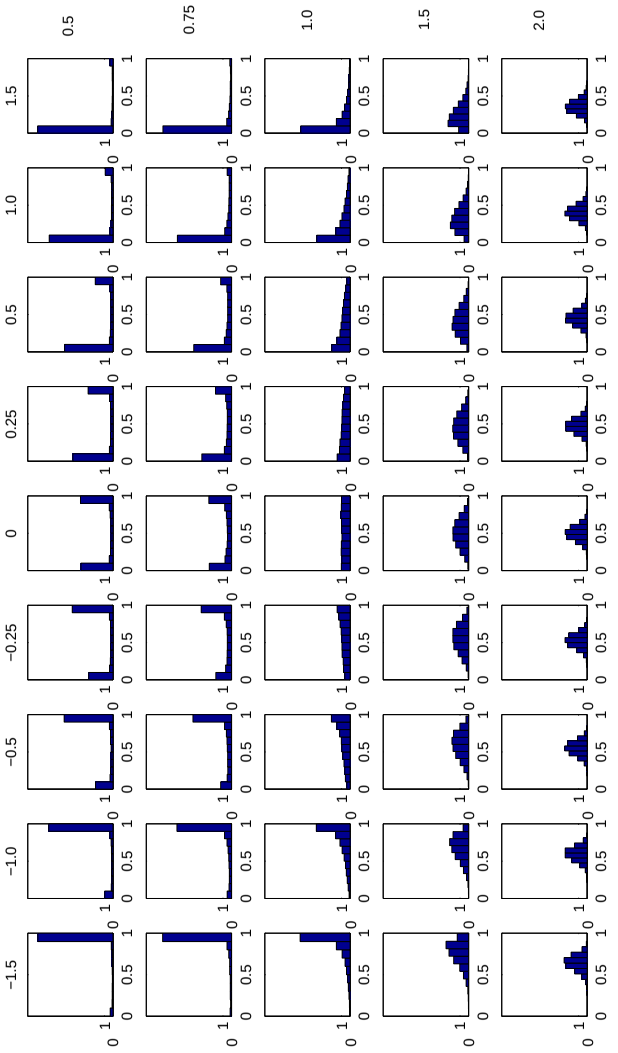


Fig. 1. PIT histograms where the underlying data distribution is sampled from $N(0,1)$ and the density models predict a $N(\mu,\sigma)$ distribution, where the μ for each plot is shown at the top of each column of PIT histograms and the σ for each histogram is plotted on the right hand side of each row. Each PIT histogram is determined from a sample of 10,000 points and each bin interval is 10%.

2.2 The PIT Objective Function

Assuming the PIT histogram is a satisfactory means of ranking models' statistical consistency, we should try to represent the PIT as an objective function so that we can actively search for models that are statistically consistent. We know from equation (3) that the z values will follow a uniform distribution if the predicted densities take the correct form. Therefore, an objective function should quantify how $U(0,1)$ and *i.i.d.* the z values from each model are and then the models can be ranked based on this. There is an extensive literature on goodness-of-fit tests for uniformity and there are several candidate approaches to solving this problem. However, Noceti et al. (2003) empirically tested a number of these statistical methods for evaluating z values in this way and concluded that the most powerful statistic, amongst those tested, was the Anderson-Darling test[6] (Anderson & Darling, 1954).

The Anderson-Darling test is known as a quadratic test because it is based upon a weighted square of the vertical distance between the empirical stepwise density function and target cumulative density function. It differs from the well known Kolmogorov-Smirnov test, which finds the maximum vertical distance between the empirical and target density. Similar approaches, such as the Cramer-von Mises statistic and the Watson statistic, use different weighting schemes. The Anderson-Darling is defined as follows,

$$A^2 = -m - \frac{1}{m}\sum_{j=1}^{m}(2j-1)[\log(z_i) + \log(1 - z_{m-j})] \tag{4}$$

Where, m, is the number of z values, and the z values are sorted in ascending order.

[6] Noceti et al. (2003) tested on Kolmogorov-Smirnov, Kuiper, Cramer-von Mises, Watson and Anderson Darling. Independently, we tested the Chi-square and Shapiro-Wilks tests for uniformity. The Anderson-Darling test performed better than these two other metrics also.

A^2 is not a strictly proper scoring rule in the sense that you can not assess the statistical consistency of a single prediction - it can only be determined for a set of predictions. However, Anderson and Darling (1954) suggest that this statistic produces assessments of uniformity that are valid for sample sizes larger than 40. Therefore, this quality score should be used when the dataset size is greater than 40. Naturally, larger sample sizes give more accurate scores.

Empirically Testing the Relationship between A^2 and Statistical Consistency From the value A^2 (Equation 4) you can determine rejection or acceptance of the null hypothesis, however, because there is a direct relationship between uniformity and the A^2 value it is reasonable to use the equation above as an error score. To demonstrate the direct relationship between statistical consistency, and the A^2 error function we performed the following experiment.

If you think of statistical consistency as the correct specification of a probability forecast it is possible to assess the performance of a model on specific quantiles over a large test set by determining the quantile accuracy. Intuitively, over a large set of predictions, the number of target values within the 5th percentile of the predicted densities should be 5%. This value is called the $HitRate$ (Engle & Manganelli, 1999).

$$HitRate = \left| \left(\frac{\sum_{i=1}^{N} 1\left\{ \Phi_i^{-1}(q) > t_i \right\}}{N} \right) - q \right| \tag{5}$$

Where, q, is the specified percentile, Φ^{-1} is the inverse cumulative of the predicted density and N is the size of the large set.

We created 100 samples of 6,000 observations using a GARCH(1,1) process with parameters (0.3, 0.05, 0.9). Then we estimated the GARCH parameters using the standard GARCH optimisation technique on the first 3,000 observations of each data set. For each sample, we calculated both the A^2 and the $HitRate$

of the 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99, and 0.999 percentiles[7] for the remaining 3,000 observations. The $HitRate$s for all the percentiles were averaged to determine an interval statistical consistency score for the density forecasting model. Figure 2. plots the normalized average $HitRate$ score and the normalized A^2 score for each model in ascending $HitRate$ order. It is clear that there is a strong correlation between the scoring mechanisms, this verifies the legitimacy of the A^2 error function as a means of ranking models in terms of statistical consistency.

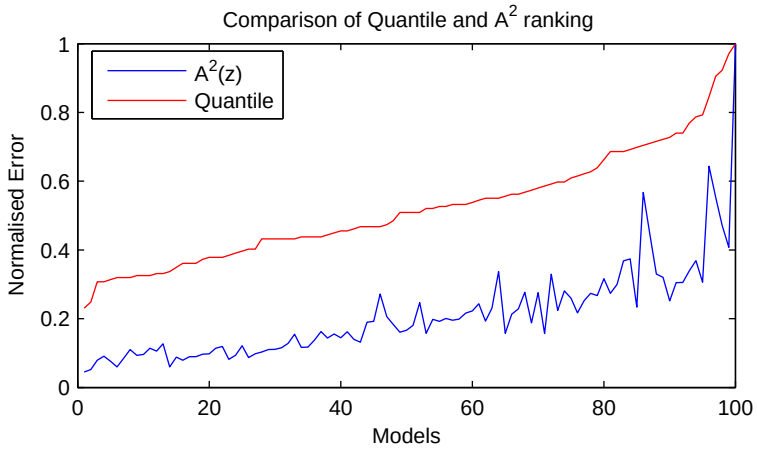


Fig. 2. Comparison of quantile ranking and A^2 ranking on simulated data. The correlation coefficient is 0.874. The quantiles used were 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99 and 0.999.

3 A Framework for Multi-objective Optimisation of Density Forecasting Models

In the preceding sections we introduced the concepts of accuracy and statistical consistency to describe the desirable objectives of a good density forecasting

[7] We chose these percentiles to focus on the tails of the distribution because our application, Value-at-Risk, is most interested in this part of the distribution.

model. Implicitly we have outlined a multi-objective optimisation problem with accuracy and statistical consistency as the competing quality criteria. There is an inherent tension between these two objectives, a negative log-likelihood optimisation attempts to find the parameter estimates that give the highest probability of generating a probability density function with *maximum* density at the observed sample. In contrast, the Anderson-Darling objective attempts to find the parameter estimates that give the highest probability of generating a probability density function with the *correct* density at the observed sample.

A weighted combination of the two error scores is a possible approach, however, this is far from optimal because choosing a good objective function weighting is not trivial. Das and Dennis (1997) outline the drawbacks of this approach. Instead, we suggest density forecasting models should be trained using an *a posteriori* multi-objective search approach (Van Veldhuizen & Lamont, 1998). *A posteriori* techniques optimise on all objectives simultaneously finding a *Pareto set* of candidate solutions. The user then selects the result that best suits their goals from this set.

In this section we outline the principles behind multi-objective optimisation and show how evolutionary algorithms can be used to solve this type of problem. Finally, we outline a framework for optimisation of density forecasting models on multiple criteria.

3.1 Principles of Multi-objective Optimisation

The concept of the *Pareto optimum* was formulated by Vilfredo Pareto in the 19th century (Pareto, 1896), and constitutes by itself the origin of research in multi-objective optimization. The set of solutions of a multi-objective optimization problem consists of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another - these vectors are known as *Pareto optimal*. Mathematically, the concept of

Pareto optimality is as follows: Assume, without loss of generality, a minimisation problem and consider two decision vectors $\mathbf{a}, \mathbf{b} \in \Omega$. Then, $\mathbf{a}$ is said to dominate $\mathbf{b}$ (also written as $\mathbf{a} \succ \mathbf{b}$) iff:

$$\forall i \in \{1, .., k\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \wedge$$
$$\exists j \in \{1, .., k\} : f_j(\mathbf{a}) < f_j(\mathbf{b})$$

where k is the number of objective functions and $f_i(\mathbf{x})$ returns the fitness for decision vector $\mathbf{x}$ on the i^{th} objective function. All decision vectors which are not dominated by any other decision vector of a given set are called *non-dominated* regarding the set. The decision vectors that are non-dominated within the entire search space are denoted as *Pareto optimal* and constitute the so called Pareto-optimal set or Pareto-optimal front (Zitzler & Thiele, 1999). The goal of multi-objective optimisation is to find a diverse Pareto-optimal set. Figure 3 depicts a sample objective space for a density forecasting model given our formulation of the problem.

3.2 Multi-objective Evolutionary Algorithms

Rosenberg (1967) was the first to apply an evolutionary algorithm (EA) to solve a multi-criteria search problem. An EA is a generic term used to indicate any population-based metaheuristic optimization algorithm that uses mechanisms inspired by biological evolution, such as reproduction, mutation and recombination. Candidate solutions represented as a vector of decision variables play the role of individuals in a population, and the objective function determines the environment within which the population exists. Evolution of the population then takes place after the repeated application of mutations and/or reproduction. There are a number of different evolutionary algorithms available, the most

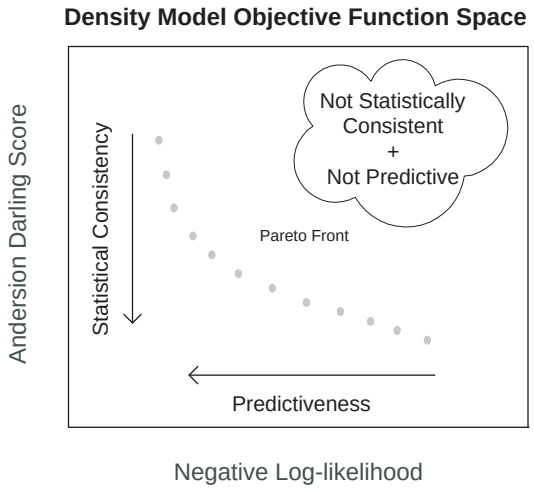


Fig. 3. This is a sample representation of a Pareto-optimal front for a multi-objective density forecasting model. Because both our objective functions are negatively oriented the convexity of the Pareto optimal front is in the direction of the origin. Each grey dot represents a density model in the objective function space. Representing the models in this space makes it easy for a user to select the one that best suits their goals.

common types are genetic algorithms (GA), evolutionary strategy (ES), and evolutionary programming (Back, 1996).

A secondary motivation for the use of an EA optimisation technique in this particular research is that the A^2 quality score is non-derivative, thus it could not be optimised using a gradient approach. However, EAs can optimise objective functions that are non-derivative. For further information on the theory of multi-objective evolutionary algorithms (MOEAs) see (Deb, 2001; Coello, 1999; Van Veldhuizen & Lamont, 1998).

3.3 The Framework

Given that we have changed the specifications for the optimisation of density forecasting, a well defined approach to implementing multi-objective optimisation of density forecasting models is needed. Further, it is likely that different

objective functions may substitute the NLL and A^2 error functions when user objectives change, therefore the generality of the framework is important.

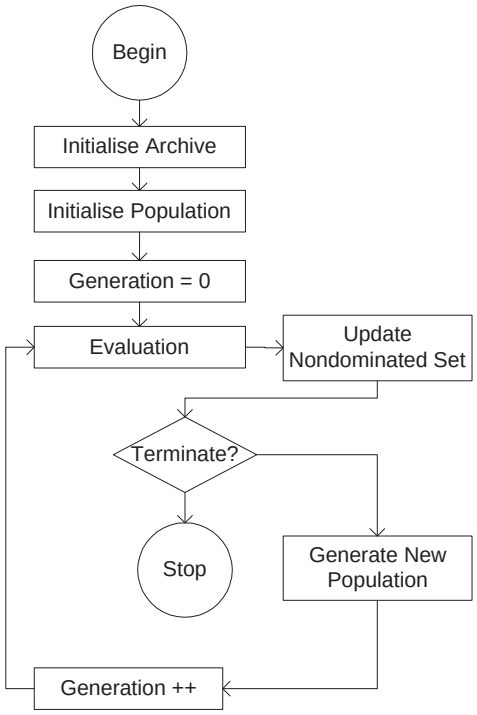


Fig. 4. Flow diagram of the general multi-objective evolutionary algorithm density model framework.

Figure 4. is a flow diagram of the framework we propose[8]. The EA maintains a set of predicted Pareto-optimal models during the training process. The population contains a set of models. Any density forecasting model that can be represented as a vector of values can be optimised using this framework. Moreover, it is generalised so that any evolutionary algorithm can be used to optimise these models. In this study, a new method for optimising density mod-

[8] This framework was inspired by the Pareto Evolutionary Neural Network introduced by (Fieldsend & Singh, 2005).

els is derived from the general framework introduced above called the Pareto GARCH.

Pareto GARCH Model There is a large family of GARCH models available to the practitioner for empirical volatility prediction. Here we present the Pareto GARCH model as a new means of optimising these models to obtain statistically consistent volatility predictions. Following the work of Engle (1982) and Bollerslev (1986), a voluminous econometric literature has developed on volatility estimation and forecasting. Our multi-objective optimisation approach can optimise any GARCH model that can be represented as a set of parameters $\mathbf{x} = (x_1, ..., x_n)$. Our MOEA then minimises the error for the two objective functions. The following two examples show how a GARCH(P,Q) and a GJR(P,Q) model can be parameterised.

GARCH(P,Q): This model can be described as follows, (Bollerslev, 1986);

$$\sigma_t^2 = \kappa + \sum_{i=1}^{P} \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^{Q} \beta_j \epsilon_{t-j}^2 \tag{6}$$

The parameters of interest in the equation for the GARCH(P,Q) model above are κ, a constant, the weightings α applied to the P previous forecasts $\sigma^2_{t-\{1...P\}}$ and the β weightings for the last Q periods' squared disturbances $\epsilon^2_{t-\{1...P\}}$. This type of model is commonly accompanied by a constant mean model[9].

$$y_t = C + \epsilon_t \tag{7}$$

This equation contains the final parameter C for our decision vector. The vector representation of a GARCH(P,Q), therefore, takes the form
$\mathbf{x} = (C, \kappa, \alpha_1, ..., \alpha_P, \beta_1, ..., \beta_Q)$.

[9] This can be thought of as an ARMAX(0,0,0) model.

GJR(P,Q): This model can be described as follows, (Glosten et al., 1993);

$$\sigma_t^2 = \kappa + \sum_{i=1}^{P} \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^{Q} \beta_j \epsilon_{t-j}^2 + \sum_{j=1}^{Q} L_j S_{t-j}^* \epsilon_{t-j}^2 \tag{8}$$

Where,

$$S_{t-j}^* = \begin{cases} 1 & \epsilon_{t-j} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

This model has one additional parameter, L the leverage term, all other parameters are the same as the GARCH(P,Q) model. The leverage term captures the characteristic asymmetry that is seen in many return series. Again, the model can be accompanied by a constant mean model. Therefore, the vector representation of the GJR(P,Q) model takes the form $\mathbf{x} = (C, \kappa, \alpha_1, ..., \alpha_P, \beta_1, ..., \beta_Q, L_1, ..., L_Q)$.

ARMA(R,M)/GJR(P,Q): It is possible to apply this framework to much more complex models. For example, in our case study (Section 4) we train an ARMA(R,M)/GJR(P,Q) model using our framework. To do this we simply include the additional R autoregressive coefficients (ϕ) and M moving-average coefficients (θ) to our parameter vector.
$\mathbf{x} = (C, \phi_1, ..., \phi_R, \theta_1, ..., \theta_M, \kappa, \alpha_1, ..., \alpha_P, \beta_1, ..., \beta_Q, L_1, ..., L_Q)$.

Pareto GARCH Algorithm The EA used for optimising the GARCH models is based on an (1+1)-Evolutionary Strategy (ES) process where a single individual is selected from the archive at each iteration, evolved and inserted in the archive if it is non-dominated. This technique requires each individual in the population to have a mutation strength vector. The mutation strength vector means the ES is capable of learning to adapt to problems where each variable has an unequal contribution to the objective function. This mutation strength

vector or strategy vector is combined with the decision variables of the model using a logarithmic update rule.

$$\mathbf{s}_{t+1,i} = \mathbf{s}_{t,i} \exp(\tau' N(0,1) + \tau N_i(0,1))$$
$$\mathbf{x}_{t+1,i} = \mathbf{x}_{t,i} + \mathbf{s}_{t+1,i} N_i(0,1) \tag{9}$$

where, $\tau' \propto (2n)^{-1/2}$, $\tau \propto (2n^{1/2})^{-1/2}$, n is the length of the decision vector and $N(0,1)$ is a random number generated from a standard normal distribution[10]. This is known as a non-isotropic self-adaptive ES (Back, 1996). The selection method used is the partition quasi-random selection (PQRS) method (Fieldsend et al., 2003). During training we discard all parameter vectors that violate the constraint set for the specified model[11].

Algorithm 1.

Inputs:

D, the data to be modeled. T, the maximum number of generations. L, size of the initial random population of solutions. M, the model type.

Outputs:

A non-dominated set of models that are an estimate of the true Pareto front defined by the data generation process.

1. **Initialisation:** Determine $\mathbf{x}$, a vector representation for the specified model type. Generate a random population of l of these vectors. For each of the $\mathbf{x}$ vectors in the population generate a random mutation strength vector, $\mathbf{s}$. Create the empty frontal set F_0. Evaluate each member of the population on the two criteria to find the initial set of non-dominated

[10] We have found that replacing the normally distributed random numbers with random draws from a Cauchy distribution reduces convergence time.

[11] In future research we intend to amend this algorithm to include evaluation on a validation set to prevent the models over-fitting.

solutions. Update F_0 with the non-dominated solutions. Initialise generation counter to $t = 0$.

2. **Selection:** Use selection method to select a model from F_t, $\mathbf{x}_t$. This single model parameter vector, $\mathbf{x}_t$, is the population for iteration t.
3. **Mutation:** Mutate $\mathbf{x}_t$ using the mutation strength vector $\mathbf{s}_t$ to produce $\mathbf{x}_t^*$ and $\mathbf{s}_t^*$.
4. **Evaluation**: Evaluate $\mathbf{x}_t^*$ with respect to the error measures (NLL and A^2) and D. If $\mathbf{x}_t^*$ is not dominated by any member in F_t go to 5, otherwise delete $\mathbf{x}_t^*$ and go to 6.
5. **Archive Update:** a) Insert $\mathbf{x}_t^*$ into F_t. b) Remove individuals in F_t that are dominated by $\mathbf{x}_t^*$.
6. **Loop:** Iterate counter, $t = t + 1$. If $t = T$ then go to 7, else go to 2.
7. **Terminate:** Terminate the algorithm and save members of F_T for hold-out (or alternative) evaluation.

4 Case Study: Statistically Consistent GARCH for VaR Estimation

In this section we analyse the effectiveness of our optimisation technique for GARCH models. The experiments are carried out in the context of a risk management application, specifically estimating Value-at-Risk.

4.1 Value-at-Risk

Value at Risk, or VaR, is a commonly used statistic for measuring potential risk of economic losses in financial markets. It is a method of risk assessment using probabilistic measures. See Duffie and Pan (1997), Jorion (2001) for a survey and overview of VaR. VaR is a metric, a way of interpretation, rather than a measure. In essence, it is a measure of expected loss, measured typically in dollar

terms, on a given portfolio of assets in a given time frame. The operationalising of this measure requires then an estimate of the likelihood of a given percentage change in the returns on which the loss is to be measured. For example, an $\alpha = 5\%$ 1-day VaR of \$5 million can tell us that on one out of 20 days, we could expect to realize a loss of at least \$5 million. Alternatively, the maximum loss we would reasonably expect from our model on 19 out of 20 days is \$5million. In other words, VaR is defined as the maximum loss over a given time horizon at a given confidence level. Mathematically, let P_t be the price of a financial asset on day t. A k-period VaR on day t is defined by

$$P(P_{t-k} - P_t \leq VaR(t, k, \alpha)) = 1 - \alpha \tag{10}$$

Clearly, to evaluate this, we need two elements: an estimate of the distribution of the returns, and from this an estimate of the percentiles of the return. If $Q\alpha$ is the α percentile of the returns, then

$$VaR(t, k, \alpha) = (1 - e^{Q\alpha})P_{t-k} \tag{11}$$

A popular parameterisation of the VaR method is the Riskmetrics model (Riskmetrics, 1996). Other implementations are of course possible: extreme value theories, high frequency realised volatility and conditional moments of GARCH are popular alternatives, as are recent developments in cupola modeling. See (Danielsson & de Vries, 1997), (Beltratti & Morana, 1999), (Ho et al., 2000), (Wong & So, 2003), (Chen & Fan, 2005) and (Fernandez, 2005) for discussion of these models. In this paper we concentrate on models that are characterized by GARCH family distributions. Giot and Laurent (2004) and Lee and Saltoglu (2001) show that GARCH type models are an adequate estimator of VaR parameters. In our case study we apply our optimisation technique to three different

ARCH type models and compare the resulting models performance against a standard[12] ARCH optimisation technique.

4.2 Sample Pareto-GARCH Result

The nature of *a posteriori* optimisation dictates that the search process results in a large set of diverse models that represent the Pareto front of trade off solutions. The decision of which is the best model for the given task is then left to the practitioner. The presumption is that the practitioner has the capacity to discern which model is optimal for their needs. Figure 5 is a sample Pareto front obtained after optimisation on the S&P500 data (see Section 4.4).

4.3 Experiments

To implement our methodology on real data, the researcher needs to construct the historical series of portfolio returns and select a density forecasting type. We took a sample of 5145 daily prices from Datascope for S&P 500, General Motors and IBM. We computed the daily returns as the difference of the log of the prices. The samples range from January 1 1986 to September 20 2005. This includes the crash of 1987. We use the first 3745 samples to train the model and the last 2000 for out of sample testing.

We use a GARCH(1,1) model with constant mean for the S&P 500 series, a GJR(1,1) model with constant mean for the GM series, and an ARMA(1,1)/GJR(1,1) model for the IBM series. The motivation for using a GJR model for the two equity series is because the difference in effect of positive and negative innovations to returns on the conditinoal volatility can be captured using this model type. We estimated the 0.1%, 1%, 5% and 10% one day VaR for each experiment. The estimated quantiles relative to the series are plotted in

[12] We use the standard constrained nonlinear optimisation implementation for GARCH training. See Mathworks Matlab^{TM} Garch Toolbox for more information.

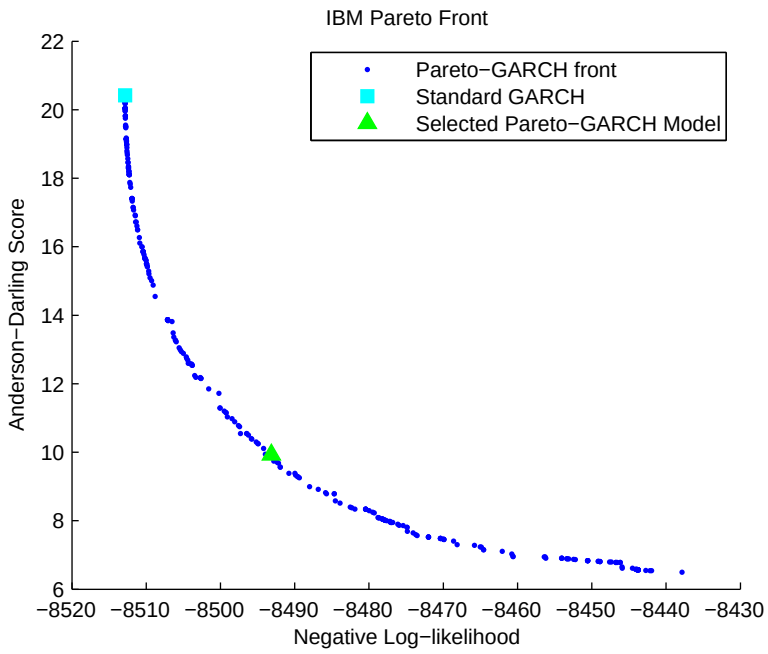


Fig. 5. Sample Pareto front determined after training of Pareto-GARCH model. The characteristic arc of non-dominated solutions is clearly visible. As would be expected the standard model trained using only the NLL objective function performs very well on the NLL objective, however, it has the poorest statistical consistency.

Figures 8, 10 and 12. The results of the estimates are reported in Tables 1, 2, and 3. In each table, we report the percentage of times the VaR is exceeded both in and out-of-sample. Figures 7, 9 and 11 show the probability integral transform histogram and the autocorrelations of the PIT values for each model in each experiment. The multi-objective algorithm was given an initial population of 1,000 models and was trained for 5,000 epochs. The parameters for each model in the experiments are also included.

Our results show that our Pareto-GARCH(1,1) model gives comparable results to the standard GARCH(1,1) model, because of the restrictiveness in this model the Pareto-GARCH does not achieve a considerable uplift in performance.

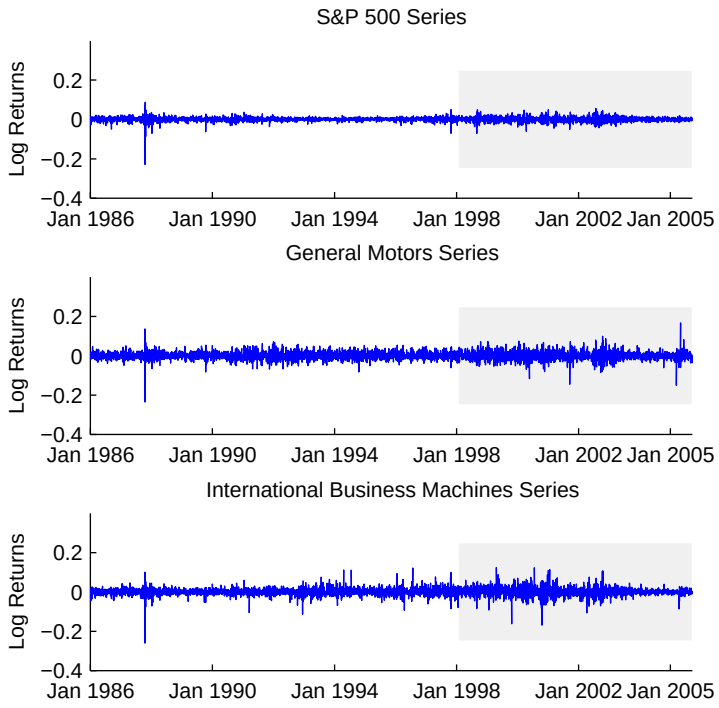


Fig. 6. Return series for the data used in the empirical experiments. The highlighted area represents the out-of-sample data.

However, the GJR(1,1) models trained on the equity data show that the model optimised using our approach have a considerable uplift in performance especially on the out-of-sample data.

To fully appreciate the performance of the Pareto GARCH models, recall that the samples over which the models are estimated include the crash of October 1987 and that the length of the out-of-sample period is 2000 trading days. This is roughly 8 calender years. It is likely that financial institutions will re-estimate their models on a more frequent basis (yearly, monthly, weekly) and that the performance of our model will improve with this procedure.

The issue of model selection is a critical one. Ideally, the best model should be statistically consistent without too much of a degradation in accuracy performance. A model with these features would very likely have a good performance out-of-sample, which is what a practitioner is interested in. In these experiments we chose the model from the Pareto front that was mid way between the two objective extremes.

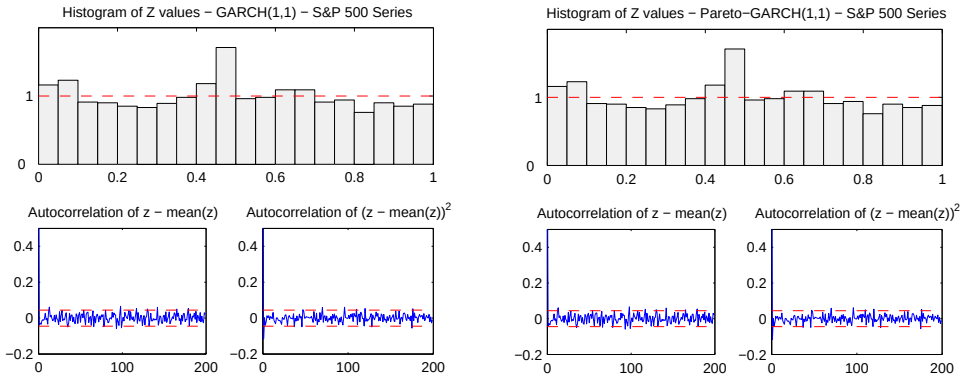
4.4 S&P 500 and GARCH(1,1)

In this experiment we compare a standard GARCH(1,1) model against a Pareto-GARCH(1,1) model. Table 3 compares the quantile accuracy. The GARCH and Pareto-GARCH model perform comparatively well, with the Pareto-GARCH achieving only a marginal uplift in performance out-of-sample. This result can be expected for a model that makes such restrictive assumptions on the data.

Table 1. S&P 500 quantile estimates. Comparison of quantile accuracies. Percentages represent the percentage of times the quantile was actually exceeded.

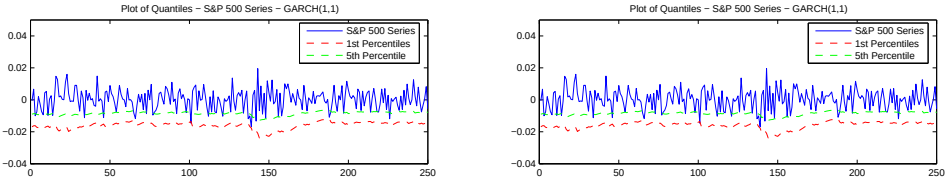
Quantiles	**0.10%**	**1.00%**	**5.00%**	**10.00%**
	In Sample			
GARCH(1,1)	0.60%	1.88%	4.74%	8.39%
Pareto-GARCH(1,1)	0.60%	1.88%	4.74%	8.39%
	Out of Sample			
GARCH(1,1)	0.40%	1.50%	5.80%	12.00%
Pareto-GARCH(1,1)	0.40%	1.50%	5.80%	11.95%

Figure 7 contains two plots comparing the Probability Integral Transform histogram of a standard GARCH(1,1) model against the selected model from the Pareto set of GJR models on the out-of-sample S&P500 set. Figure 8 shows the actual predicted quantiles relative to the series on the last 250 days in the out-of-sample data.



(a) **(b)**

Fig. 7. Comparison of the PIT histograms obtained on the out-of-sample S&P500 data. (a) is the standard GARCH(1,1) model with an A^2 score of 5.4008 on the out-of-sample data. (b) is the Pareto-GARCH(1,1) model, it obtained an A^2 score of 5.3927. The histograms show that the Pareto-GARCH(1,1) model has comparable statistical consistency to the standard GARCH(1,1) model.



(a) **(b)**

Fig. 8. Comparison of the predicted quantiles determined from the GARCH(1,1) models.

The parameter vectors for the two models in this experiment are as follows.

$$\text{GARCH(1,1)} = (0.00064948, 1.6102e-006, 0.89784, 0.089808)$$

$$\text{Pareto-GARCH(1,1)} = (0.00065048, 1.607e-006, 0.89784, 0.089808)$$

4.5 General Motors and GJR(1,1)

In this experiment we compare a standard GJR(1,1) model against a Pareto-GJR(1,1) model. Table 2 compares the quantile accuracy. Although, the standard GJR(1,1) performs better at some probability values in-sample, out-of-sample the Pareto-GJR(1,1) performs better on all quantiles.

Table 2. General Motors quantile estimates. Comparison of quantile accuracies. Percentages represent the percentage of times the quantile was actually exceeded.

Quantiles	**0.10%**	**1.00%**	**5.00%**	**10.00%**
	In Sample			
GJR(1,1)	0.60%	1.84%	5.47%	9.92%
Pareto-GJR(1,1)	0.41%	1.37%	4.61%	8.04%
	Out of Sample			
GJR(1,1)	1.20%	2.65%	7.25%	12.8%
Pareto-GJR(1,1)	0.85%	1.90%	5.10%	9.70%

Figure 9 contains two plots comparing the Probability Integral Transform histogram of a standard GJR(1,1) model against the selected model from the Pareto set of GJR models on the out-of-sample GJR set. Figure 10 shows the actual predicted quantiles relative to the series on the last 250 days in the out-of-sample data.

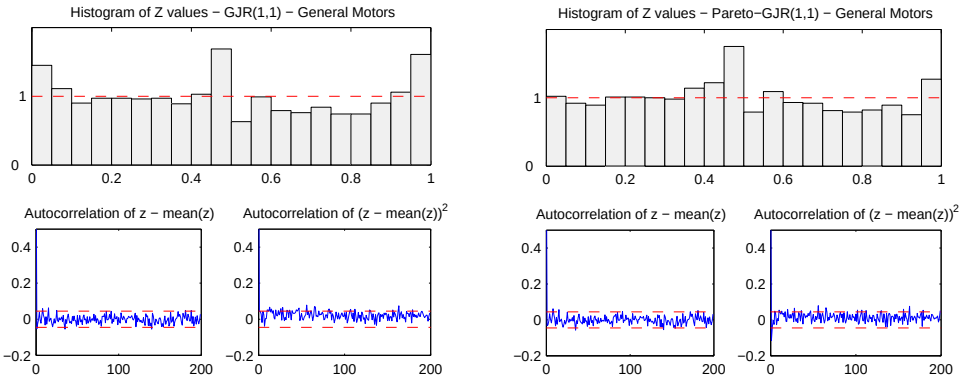
The parameter vectors for the two models in this experiment are as follows.

$$\text{GJR(1,1)} = (0.00014403, 8.9246e-005, 0.39414, 0.41152, -0.12861)$$

$$\text{Pareto-GJR(1,1)} = (5.1845e-005, 2.4905e-005, 0.82722, 0.053715, 0.080627)$$

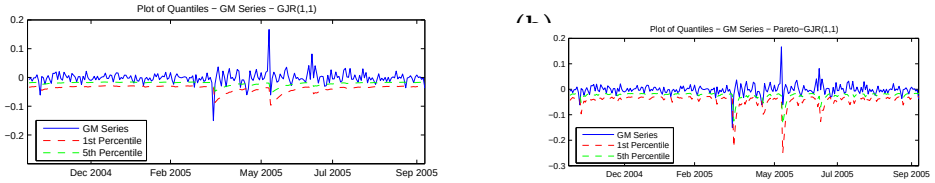
4.6 International Business Machines and GJR(1,1)

In this experiment we model the conditional mean and variance of the series. We compare an ARMA(1,1)/GJR(1,1) model against a Pareto-ARMA(1,1)/GJR(1,1) model. Table 3 compares the quantile accuracy. Although, the standard GJR(1,1)



(a) **(b)**

Fig. 9. Comparison of the PIT histograms obtained on the out-of-sample General Motors data. (a) is the standard GJR(1,1) model with an A^2 score of 14.236 on the out-of-sample data. (b) is the Pareto-GJR(1,1) model, it obtained an A^2 score of 6.557. The histograms clearly show that the Pareto-GJR(1,1) model (b) has better statistical consistency than the standard GJR(1,1) model (a).



(a) **(b)**

Fig. 10. Comparison of the predicted quantiles determined from the GJR(1,1) models. (b) clearly follows the series with better accuracy.

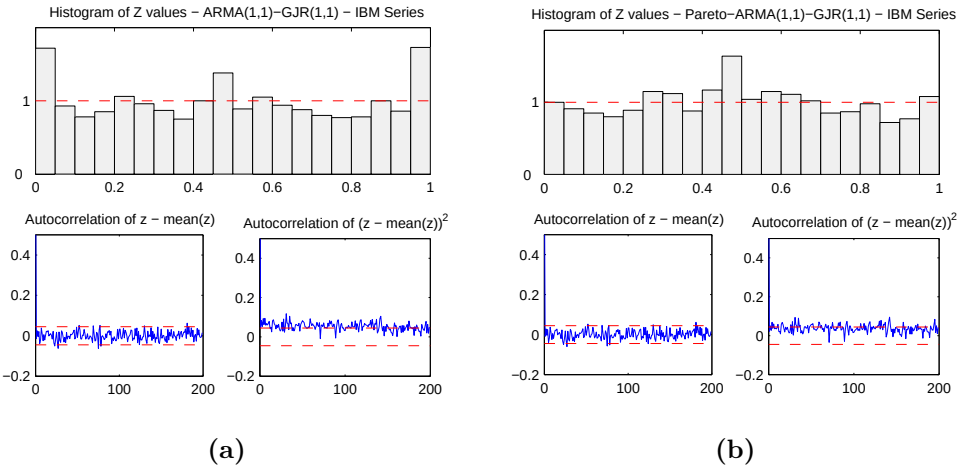
performs better at some probability values in-sample, out-of-sample the Pareto-GJR(1,1) performs better on all quantiles.

Figure 11 contains two plots comparing the Probability Integral Transform histogram of a standard ARMA(1,1)/GJR(1,1) model against the selected model from the Pareto set of ARMA(1,1)/GJR(1,1) models on the out-of-sample GJR

Table 3. International Business Machines quantile estimates. Comparison of quantile accuracies. Percentages represent the percentage of times the quantile was actually exceeded.

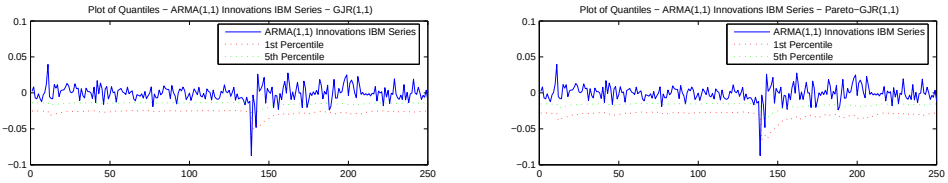
Quantiles	**0.10%**	**1.00%**	**5.00%**	**10.00%**
	In Sample			
ARMA(1,1)/GJR(1,1)	1.18%	2.80%	6.71%	11.03%
Pareto-ARMA(1,1)/GJR(1,1)	0.57%	1.59%	4.20%	7.76%
	Out of Sample			
ARMA(1,1)/GJR(1,1)	1.80%	3.70%	8.60%	13.25%
Pareto-ARMA(1,1)/GJR(1,1)	1.00%	2.00%	5.00%	9.55%

set. Figure 12 shows the actual predicted quantiles relative to the series on the last 250 days in the out-of-sample data.



(a) **(b)**

Fig. 11. Comparison of the PIT histograms obtained on the out-of-sample International Business Machines data. (a) is the ARMA(1,1)/GJR(1,1) model with an A^2 score of 21.150 on the out-of-sample data. (b) is the Pareto-ARMA(1,1)/GJR(1,1) model, it obtained an A^2 score of 6.5716. The histograms clearly show that the Pareto-ARMA(1,1)/GJR(1,1) model (b) has better statistical consistency than the standard ARMA(1,1)/GJR(1,1) model (a).



(a) **(b)**

Fig. 12. Comparison of the predicted quantiles determined from the ARMA(1,1)/GJR(1,1) models.

The parameter vectors for the two models in this experiment are as follows.

$$\begin{aligned}\text{ARMA(1,1)/GJR(1,1)} &= (0.00013223, 0.44335, -0.46756, ... \\ &\quad 0.00001640, 0.84315, 0.042892, 0.13576) \\ \text{Pareto-ARMA(1,1)/GJR(1,1)} &= (0.000245142, 0.44334, -0.46755, ... \\ &\quad 0.00001816, 0.84753, 0.082851, 0.13826)\end{aligned}$$

In summary, the accuracy of Value-at-Risk estimates is of concern to both banks and their regulators. The results outlined in this case study show that more accurate daily VaR estimates can be predicted by taking statistical consistency as well as negative log-likelihood into consideration at the optimisation stage.

5 Conclusions

In this paper we have argued that predictions that produce a complete probability density function should not only be accurate, they should also be statistically consistent. That is, values that are predicted to have a $P\%$ chance of occurring should occur roughly $P\%$ of the time. Statistical models such as GARCH or GJR that are trained only to be accurate will not necessarily be very statistically consistent in the tails of the distribution. In fact, the evaluation presented

here shows that these models are inclined to be over-confident producing density predictions that are too narrow. This is particularly an issue in applications such as the estimation of VaR where the ability to accurately predict the tails of the distribution is important.

We have introduced a multi-criteria optimisation technique based on evolutionary search that can build GARCH and GJR models that are more statistically consistent than conventional GARCH and GJR models. We have shown in particular that the GJR models have more statistically consistent hit-rates in the tails of the predicted distribution on data describing financial returns.

One way in which this multi-criterial optimisation process might be improved for this VaR task would be to focus on the statistical consistency in the tails of the distribution. The current process optimises on the PIT score which measures statistical consistency across the whole distribution. In future work we will look at modifications to the PIT score that would give more weight to statistical consistency in the tails and less to the centre of the distribution.

Acknowledgments This research was supported by Science Foundation Ireland under grant No. S.F.I.-02IN.1I111. Data was provided by the Institute for International Integration Studies at Trinity College Dublin, Ireland.

Bibliography

Anderson, T., & Darling, D. (1954). A test of goodness of fit. *Journal of the American Statistical Association*, *19*, 765–769.

Back, T. (1996). *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. Oxford Univ. Press.

Beltratti, A., & Morana, C. (1999). Computing value at risk with high frequency data. *Journal of Empirical Finance*, *6*, 431–455.

Bollerslev, T. (1986). Generalized autoregressive conditional hetroskedasticity. *Journal of Econometrics*, *31*, 307–327.

Chen, X., & Fan, Y. (2005). Estimation of copula-based semiparametric time series models. *Journal of Econometrics*, *In Press, Corrected Proof, Available online*, 1.

Christoffersen, P., & Jacobs, K. (2004). Which GARCH model for option valuation? *Management Science*, *50*, 1204–1221.

Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, *1*, 129–156.

Danielsson, J., & de Vries, C. (1997). *Value-at-risk and extreme returns* (Technical Report). London School of Economics and Institute of Economic Studies.

Das, I., & Dennis, J. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Struct. Optimisation*, *14*, 63–69.

Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Wiley.

Diebold, F., Hahn, J., & Tsay, A. (1998a). *Real-time multivariate density forecast evaluation and calibration: Monitoring the risk of high-frequency returns on*

foreign exchange (Technical Report). National Bureau of Economic Research.

Diebold, F. X., Gunther, T. A., & Tay, A. S. (1998b). Evaluating density forecasts with applications to financial risk management. *International Economic Review*, *39*, 863–83.

Duffie, D., & Pan, J. (1997). An overview of value at risk. *The Journal of Derivatives*, *5*, 7–49.

Engle, R. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of uk inflation. *Econometrica*, *50*, 987–1008.

Engle, R., & Manganelli, S. (1999). *Caviar: Conditional value at risk by quantile regression* (Technical Report). MP Page - NBER, Working Paper.

Fernandez, V. (2005). Risk management under extreme events. *International Review of Financial Analysis*, *14*, 113–148.

Fieldsend, J., Everson, R. M., & Singh, S. (2003). Using unconstrained elite archives for multiobjective optimization. *IEEE Trans. Evolutionary Computation*, *7*, 305–323.

Fieldsend, J., & Singh, S. (2005). Pareto evolutionary neural networks. *IEEE Trans. Neural Networks*, *16*, 338–354.

Giot, P., & Laurent, S. (2004). Modeling daily value-at-risk using realized volatility and arch type models. *Empirical Finance*, *11*, 379–398.

Glosten, L., Jagannathan, R., & Runkle, D. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance*, *48*, 1779–1801.

Gneiting, T., Raftery, A. E., Balabdaoui, F., & Westveld, A. (2003). Verifying probabilistic forecasts: Calibration and sharpness. *Proceedings Workshop on Ensemble Forecasting, Val-Morin, Quebec.*

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological).*

Hamill, T. M. (2000). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, *129*, 550–561.

Ho, L., Burridge, P., Cadle, J., & Theobald, M. (2000). Value-at-risk: Applying the extreme value approach to asian markets in the recent financial turmoil. *Pacific-Basin Finance Journal*, *8*, 249–275.

Jorion, P. (2001). *Value at risk: The new benchmark for managing financial risk.* McGraw-Hill.

Lee, T., & Saltoglu, B. (2001). Assessing the risk forecasts for japanese stock market. *Japan and the World Economy*, *14*, 63–85.

Manganelli, S., & Engle, R. (2001). *Value at risk models in finance* (Technical Report). European Central Bank.

Noceti, P., Smith, J., & Hodges, S. (2003). An evaluation of tests of distributional forecasts. *Journal of Forecasting*, *22*, 447–455.

Pareto, V. (1896). *Cours d'economie politique.* Lausanne.

Riskmetrics (1996). *Riskmetrics technical document* (Technical Report). Morgan Guaranty Trust Company of New York.

Rosenberg, R. (1967). *Simulation of genetic populations with biomedical properties.* Doctoral dissertation, University of Michigan, Ann Harbor, Michigan.

Rosenblatt (1952). Remarks on a multivariate transformation. *Annals of Mathematics and Statistics*, *23*, 470–472.

Seillier-Moiseiwitsch, F., & Dawid, A. (1993). On testing the validity of sequential probability forecasts. *Journal of the American Statistical Association*, *88*, 355–359.

Van Veldhuizen, D. A., & Lamont, G. M. (1998). *Multiobjective evolutionary algorithm research: A history and analysis* (Technical Report). Air Force Institute of Technology.

Weigend, A. S., & Shi., S. (2000). Predicting daily probability distributions of S&P500 returns. *Journal of Forecasting*, *19*, 375–392.

Wong, C., & So, M. (2003). On conditional moments of garch models, with applications to multiple period value at risk estimation. *Statistica Sinica*, *13*, 1015–1044.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*, 257–271.