Calibrating Probability Density Forecasts with Multi-objective Search

Michael Carney and Pádraig Cunningham

Trinity College Dublin, Ireland, email: firstname.surname@cs.tcd.ie

Abstract. In this paper, we show that the optimization of density forecasting models for regression in machine learning can be formulated as a multi-objective problem. We describe the two objectives of *sharpness* and *calibration* and suggest suitable scoring metrics for both. We use the popular negative log-likelihood as a measure of sharpness and the probability integral transform as a measure of calibration. We show how optimization on negative log-likelihood alone often results in sub-optimal models. To solve this problem we introduce a multi-objective evolutionary optimization framework that can produce better density forecasts from a prediction users perspective. Our experiments show improvements over state-of-the-art approaches.

1 INTRODUCTION

Regression is a supervised learning problem where the fundamental task is to predict some *continuous* variable. There are innumerable examples of this type of prediction problem from the medical domain to financial and environmental modeling. Density forecasting is an important emerging subfield of regression that attempts to tackle the practical problem of uncertainty in predictions of a regression model. To achieve this, a density forecast estimates a complete probability density for the target variable, rather than just producing a single value point forecast. This is useful because prediction users are generally sensitive to the possible variance around a prediction. Furthermore, with a correctly specified density forecast they can make the *optimal* decision under uncertainty.

The most common approach to parameter estimation of regression models is maximum likelihood. Density forecasting is not an exception, using an adaptation of maximum likelihood called negative log-likelihood (NLL). The convenience of this approach is that the traditional non-linear optimization techniques such as *conjugate gradient* or *quasi newton* can be used with minimal adaptations. However, on closer examination of this approach, it can be shown that certain aspects of the problem of density forecasting are not addressed and so this technique can lead to poor, sub-optimal, and often misleading, models.

In this paper we suggest that the primary goals of any density forecasting model should be to maximize *sharpness* and *calibration* [1]. Sharpness refers to the variance of the prediction around the observation and calibration refers to the empirical validity of the probability estimates (see Section 2). In Section 3

we show that optimization on sharpness alone results in models that are often far from optimal. Therefore, we suggest that the optimization of density forecasting should be reformulated as a multi-objective search task where both sharpness and calibration can be taken into consideration. In Section 5 we outline a broad framework, based on a multi-objective evolutionary algorithm, that can be used to optimize most density forecasting models. Our framework can be combined with evolutionary computing techniques, such as, Evolutionary Strategies [2], multi-objective search [3] and evolutionary neural networks [4] to optimize density forecasting models e.g. Mixture Density Networks [5] and GARCH [6]. Section 6 compares results achieved from two different models optimized using our approach on a foreign exchange data set. Finally, Section 7 briefly concludes the paper.

2 GOALS OF DENSITY FORECASTING

We address the regression problem of estimating the parameters for a model given a set of training data $\{(\mathbf{x}_i, t_i)\}_{i=1}^{m}$, where the ith example is described by the pattern $\mathbf{x}_i \in \Re^n$ and the associated response $t_i \in \Re$. Point forecasting attempts to estimate, $\langle t_i|\mathbf{x}_i\rangle$, the conditional mean of the target variable given an input pattern. Density forecasting models attempt to estimate, $p(t_i|\mathbf{x}_i)$, the conditional probability density that the target is drawn from, a considerably more complex task. Rather than simply minimizing residual errors, a density forecast must maximize density at the target and also achieve empirical consistency in the probability estimates. These two goals of density forecasting are commonly called *sharpness* and *calibration*. The NLL addresses the first goal of maximizing density at the target by rewarding models based on the density of the prediction at the target.

$$NLL_i = -\log(p(t_i|\mathbf{x}_i)) \tag{1}$$

Calibration, the second goal, 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. This is a joint property of the target and the predictions. Unfortunately, the assessment of calibration is less straightforward than sharpness. It is dependent on the assumption that you are attempting to find the model that correctly describes the data generating process. However, this is a fair assumption as the correct model *weakly dominates* all other models[1]. In the case where the correct data generating process is described, the set of cumulative densities at the observations will be uniform. Therefore, to determine calibration you must carry out the following,

$$z_i = \int_{-\infty}^{t_i} p(u|\mathbf{x}_i)\,du \tag{2}$$

[1] Weakly dominant means the model is at least as good as, if not better than, any other possible model.

where z_i is the cumulative of the predicted density at the target t_i. This is known as the Probability Integral Transform (PIT) [7]. For a data set of length m, Diebold et al. [8] show that the z series should be $\{z_i\}_{i=1}^{m} \overset{iid}{\sim} U[0,1]$.

We know $z_i \in [0,1]$ because it is a value from a cumulative density. Therefore, a test for calibration relates directly to a test for whether the z series is a $U[0,1]$. A useful method for discerning the calibration of a model is to plot a histogram of the z series (e.g. Figure 1), or alternatively, plot the z series as an empirical stepwise distribution and compare against the cumulative uniform distribution. However, both these techniques require visual assessment, it would be more desirable to have a means of *ranking* a set of models in terms of their uniformity. Fortunately, this is a common problem and relates to testing the goodness-of-fit of a sample of data to a specific distribution. Noceti et al. [9] compared a number of goodness-of-fit tests and concluded that the Anderson-Darling (A^2) [10] test for uniformity was the most robust among the most common tests for uniformity. The A^2 test is negatively oriented returning a 0 in the case here a model is perfectly calibrated to the data. The formula for A^2 is,

$$A^2 = -m - \frac{1}{m}\sum_{j=1}^{m}(2j-1)[\log(z_i) + \log(1 - z_{m-j})] \tag{3}$$

Where, m, is the number of z values, and the z values are sorted in ascending order. We can now rank the calibration of a set of models based on their A^2 score on a test set. It is important to note at this point that this ranking score should be used in conjunction with a sharpness metric because it gives no indication of the predictive ability of a model. For example, a model that simply predicts the unconditional density of the set of target variables will produce a calibrated model.

3 PROBLEMS WHEN OPTIMIZING WITH NEGATIVE LOG LIKELIHOOD

The central message in this paper is that the calibration of a density forecasting model can be as important as the sharpness of the model in many applications. If the model predicts that a variable has a 10% chance of exceeding a particular threshold, then the observation should exceed that threshold roughly one time in ten. We show in the evaluations in section 6 that models that are optimized on sharpness only are likely to produce predictions that are over confident at the observation and do not allocate enough probability to rare events. Thus the real probability of rare events is underestimated.

The negative log-likelihood (NLL) is used as a measure of sharpness in many density forecasting techniques, including [5,6,11]. Optimizing a model using NLL is an attempt to find the set of model parameters that maximise the probability density at the observation. This approach produces accurate estimates of the conditional density function in situations where the underlying generating distribution is known or can be approximated well by the distribution assumed

by the model. In circumstances where the data generating process is very complex, or the incorrect assumptions are made about the underlying distribution, or over-fitting is an issue, the NLL performs poorly, particularly in terms of calibration. To demonstrate, we present two scenarios where the NLL produces poorly calibrated predictions.

For the first example we have constructed a simple synthetic data set based on the sine function with added noise drawn from a *Student-t* distribution with 3 degrees of freedom. Inputs for the data are uniformly drawn from the interval [0,5]. The target values are generated according to,

$$t(x) = sin(x) + \epsilon \tag{4}$$

where ϵ is the noise. We trained a Mixture Density Network (MDN) [5] with 5 hidden units and 1 Gaussian output for 2,000 epochs on 1,000 input/output pairs of the sine wave data, (see Section 5.1 for more information on MDNs). Figure 1 depicts the resulting model's PIT histogram. The histogram shows that the model has produced a set of forecasts where the observations occur too often in the area around the first standard deviation of the predicted distribution. We call this an under-confident model i.e. it has too broad a variance around the conditional mean. This effect is often seen when the generating distribution has fat tails. It can be attributed to the fact that the NLL applies error penalties on an exponential scale i.e. a change in density of Δ results in a change in error of $\exp(\Delta)$. In this case the extreme outliers that are more common in the *Student-t* distribution than the Gaussian predicted by the MDN have a disproportionately large effect on the NLL causing the resulting prediction to have too large a variance about the mean of the distribution.

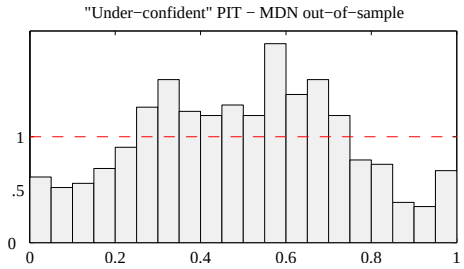

Fig. 1. Sample under-confident model. Dashed line represents the desired distribution shape.

A second example of when the NLL has difficulty producing good probability estimates is when it "over-fits" the variance of it's predictions to the training data. Not unlike traditional over-fitting of point estimates to training data, over-fitting of density forecasts is a common and unwanted side-effect of over training a density forecasting model to the available training data. This results in forecasts that are "over-confident", applying a very narrow density around

the predicted mean. This is harder to diagnose than under-confident predictions because in this case it is generally not apparent from the training data that this problem has occurred. Instead the in-sample PIT histogram produces a uniform z series. Figure 2 shows a sample over-confident prediction. This example is taken from [12], a GARCH type model (see section 5.2) is trained on IBM daily closing prices. The model performs well in-sample, however, out-of-sample it is clear that the model is over-confident about it's density forecasts resulting in a U-shaped PIT histogram.

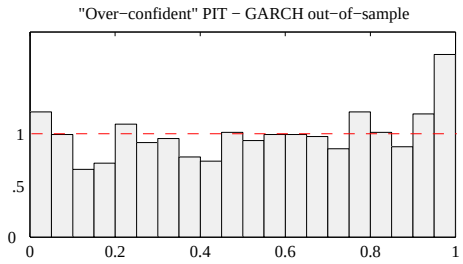

Fig. 2. Sample over-confident model.

Here we have highlighted two of the most common problems that can be diagnosed through of evaluation of the PIT, other problems, including bias in the mean prediction, can also be identified through this technique.

4 ANALYSIS ON THE RELATIONSHIP BETWEEN NLL AND A^2

In the previous sections we describe the two goals of density forecasting, here we briefly analyse their relationship in terms of the parameter space of a model. We have constructed an experiment using a very simple density forecasting model from the econometrics literature called GARCH [6]. The GARCH model has two parameters of importance commonly called the ARCH and GARCH terms that relate to weights applied to the residual and variance for the preceding time-step (see section 5.2 for more information on GARCH). In this experiment we use a synthetic data set so that we can specify the other parameters of the GARCH model correctly *a priori*. Since we have restricted the model to only two free parameters, an error function will be a surface above a 2-dimensional parameter space. Figure 3 shows plots of the error function surfaces in terms of the two parameters (ARCH and GARCH) of the model around the NLL global minimum. It is clear from the surface plots that they are completely different functions and the minima of the two error functions are located in different regions of the parameter space. This confirms that NLL and A^2 are conflicting objectives.

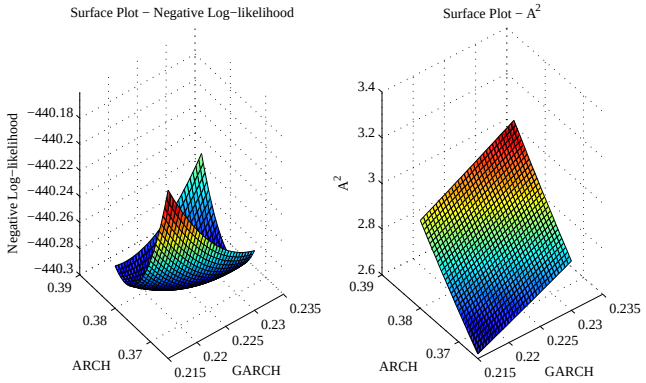

Fig. 3. Comparison of the NLL and A^2 error surfaces in the region of the NLL minimum. This is an error function for a GARCH(1,1) model trained on data from 1,000 observations simulated from an EGARCH(1,1) model that assumes a *Student-t* distribution [13].

5 MULTI-OBJECTIVE OPTIMIZATION FRAMEWORK

In the preceding sections we showed how our quality scoring metrics for calibration and sharpness (A^2 and NLL) are conflicting, i.e. they do not converge to the same point in parameter space. Therefore, implicitly, we have described a multi-objective optimization problem. There are a number of ways to solve a multi-objective search problem, however, the preferable approach is to use an *a posteriori* multi-objective evolutionary algorithm (MOEA). In the context of MOEA's, *a posteriori* means that the optimization process maintains an archive of optimal trade-off (non-dominated) solutions known as the Pareto front [14] throughout training and the user selects the model that best optimizes their goals from the resulting Pareto front of solutions [3].

Figure 4 shows the MOEA for optimization of density forecasting models. This is a general framework that can be applied to almost any density forecasting model that can be represented as a set of parameters. It is similar to most other evolutionary algorithms. To implement the MOEA the modeler must;

1. Determine a vector representation for the parameters of the density forecasting model.
2. Be able to calculate the A^2 and NLL score for the model's predicted densities.
3. Decide on a mutation and selection strategy for the evolutionary algorithm.

In the following subsections we will briefly describe the implementation of this algorithm for two particular density forecasting models.

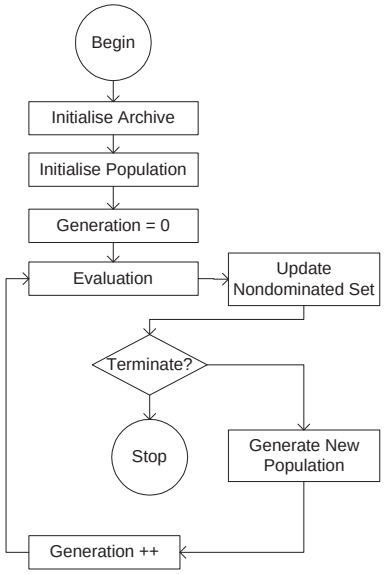

Fig. 4. Multi-objective density forecasting algorithm.

5.1 Pareto Mixture Density Network

In Section 3 we introduced the MDN, a particular class of neural network adapted to produce density forecasts. MDNs represent the conditional density function by a weighted mixture of Gaussians known as a Gaussian Mixture Model(GMM). GMMs are a flexible, convenient, semi-parametric means of modeling unknown distributional shapes. The conditional density function is described in the form $p(t|\mathbf{x}) = \sum_{i=1}^{C} \alpha_i(\phi_i(t|\mathbf{x}))$ where ϕ_i is the ith of C Gaussian components, and α_i is the weighting for that component. MDNs are generally optimized using a conjugate gradient technique, however, like most neural networks they can also be optimized using an evolutionary algorithm. Recently, Fieldsend et. al introduced the Pareto Evolutionary Neural Network (PENN) for multi-objective optimization of neural networks [15]. We have adapted this to fit our framework by using an MDN as the underlying model and setting the objective functions to the NLL and A^2 scores. To generate new individuals we select a model from the non-dominated set using Partition Quasi Random Selection (PQRS) [16] and mutate the selected individuals through network weight addition, deletion and adaptation. This results in objective function and network architecture optimization. For a full description of the Pareto-MDN see [17].

5.2 Pareto GARCH

Generalized Autoregressive Conditional Heteroscedasticity (GARCH) models are commonly used in finance to estimate the conditional variances of a time

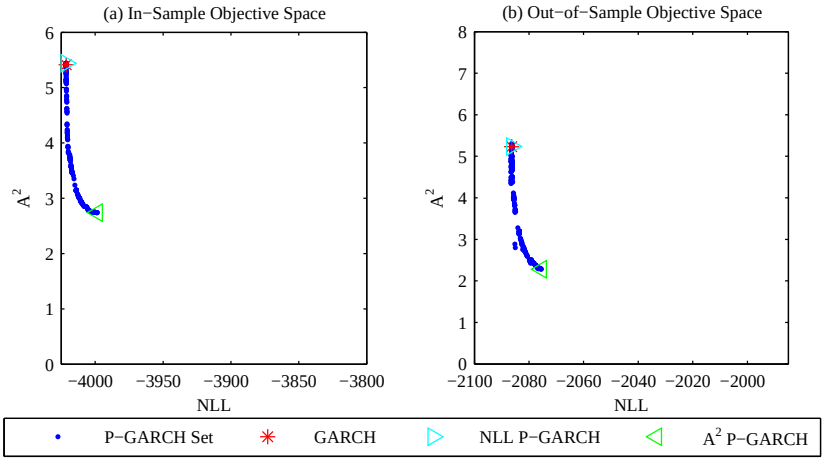

Fig. 5. In-sample and out-of-sample objective spaces after 10,000 iterations of Pareto-GARCH

series [6]. There is a large family of GARCH type models. In our experiments we use the simplest and most popular model GARCH(1,1). Here the predicted distribution is a Gaussian, the mean is presumed to be constant and the conditional variance for the next time step is predicted as a weighted sum of the previous time-steps residual, predicted variance and the unconditional variance of the series. This very simple model can successfully capture the serial dependence in financial data. The GARCH(1,1) can be represented as a vector of 4 parameters. This vector representation is used to encode an individual in our evolutionary algorithm. We use an Evolutionary Strategy (ES) for optimization because it has a number of advantageous characteristics [2]. Besides being able to optimize a non-differentiable objective function (e.g. A^2 score), ES is attractive because it can solve complex, high dimensional, multimodal, real valued problems. However, most other evolutionary algorithms could be used instead. For a full description of the Pareto-GARCH model see [12].

6 CASE STUDY: FINANCIAL DATA

The financial domain has many applications where density forecasts are of use, not least, the popular area of risk management, which is effectively dedicated to making accurate density forecasts. Other applications include, derivatives pricing, where the range of possible future values of the underlying financial instrument has the most influence on the price, and high frequency trading, where strategies such as statistical arbitrage depend on well calibrated trading models to ensure profits.

In this case study, we analyse the performance of both an MDN and GARCH model on the notoriously difficult domain of foreign exchange data. The data is

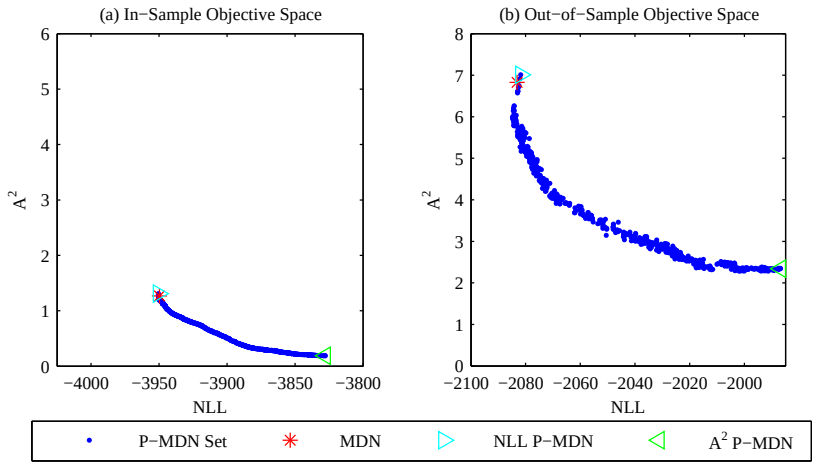

Fig. 6. In-sample and out-of-sample objective spaces after 10,000 iterations of Pareto-MDN

comprised of 1,501 examples of daily price observations for the DeutscheMark/British Pound foreign exchange rate, from April 1985 to January 1992[2], a particularly volatile period for these two currencies. We transform the daily prices into a log returns series by $r_i = \log(\frac{p_{i+1}}{p_i})$, where p_i is the price at interval i. The return series was separated into a training set of the first 1,000 observations and a test set comprised of the last 500 observations.

Using a non-linear optimization technique (*quasi-newton*) we train a GARCH(1,1) model minimizing the NLL[3]. This solution is used as the initial model for our ES. We can presume that this model represents a near global optimum solution in terms of the NLL and should be present in the Pareto front. The aim of the next step in training, the multi-objective search, is to start from this point on the Pareto front and search to find as diverse a Pareto front as possible. This process should provide new solutions that improve on calibration.

We carried out 10,000 iterations of the ES algorithm resulting in a set of 316 non-dominated individuals. Figure 5 shows the the in-sample and out-of-sample objective spaces for each of the models in the Pareto set. Each point on the objective space represents a model. We have highlighted some models of interest, the *GARCH* model represents the initial solution trained using the standard optimization procedure. *NLL P-GARCH* is the model that has the best NLL score and A^2 *P-GARCH* is the model with best A^2 score from the Pareto front.

[2] This data is included with the *Mathworks*^{TM} *Matlab*^{TM} *Garch Toolbox*.

[3] There is no standard implementation of the GARCH optimization algorithm, however, the error function, NLL, is the same in all cases. Therefore, there is usually negligible difference between the models that are produced by different implementations.

For comparison, we also trained an MDN model on the same data. The MDN was given 6 hidden units and outputs were represented as a 2 component GMM. Again, an initial model was trained on the data using a standard optimization technique, in this case we use a Scaled Conjugate Gradient method [18]. We use this model as our initial starting position in parameter space for our evolutionary algorithm. The resulting objective spaces, on both training and test data, after 10,000 epochs of training are shown in Figure 6. The Pareto front has 736 individuals.

We have determined the Spearman rank correlation coefficients between the in-sample and out-of-sample models on each objective function. Both for the GARCH Pareto set and for the MDN Pareto set the model rankings are strongly correlated suggesting little over-fitting of the models to the data on either objective function. Table 1 shows the correlations.

	NLL	A^2
P-GARCH	0.9967	0.9943
P-MDN	0.8589	0.9848

Table 1. Spearman rank correlation coefficients between the in and out-of-sample objective function values for each model.

The in-sample and out-of-sample MDN objective spaces (Figure 6) are scaled so that they can be compared easily with the GARCH objective spaces (Figure 5). Also, as in the GARCH objective space, the MDN models with best NLL and A^2 scores are highlighted. The objective space figures suggest that the MDN produces a far better calibrated set of solutions on the in-sample data. This is confirmed in Figures 7 and 8 where it is shown that the training sets of A^2 *P-GARCH* and A^2 *P-MDN* models have almost perfect calibration. This shows us that the multi-objective optimization has achieved a degree of success on the in-sample data at least. Fortunately, this success is also present in the test data where the same effect can be seen. Interestingly, however, the simpler GARCH models outperform the MDN models considerably in the out-of-sample data. Again, from the PIT histograms it can be confirmed that the GARCH models, although suffering slightly from under-confidence due to the assumption of normality, produce superior models on both the NLL and A^2 scores. Analysis of the dominance of solutions shows that in-sample the MDN and GARCH models produce solutions that do not dominate each other. However, out-of-sample there are 70 dominant solutions out of the possible 1,052 and 69 of these solutions are GARCH models.

In summary, this strategy allows the user to identify models that score well on both sharpness and calibration. There are many domains such as finance or weather forecasting where calibration is almost as important as sharpness. If rare events are significant then it is important that the model assigns the correct probability to them. This approach allows the prediction users to select

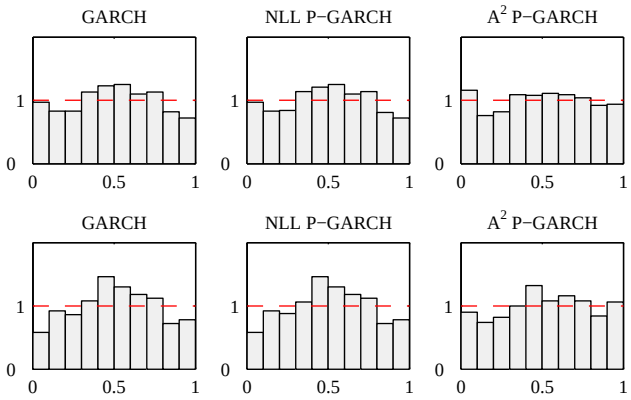

Fig. 7. GARCH pit histograms for the training data (top row) and test data (bottom row).

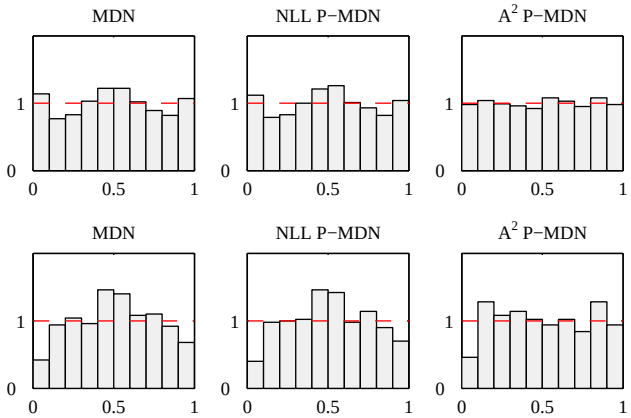

Fig. 8. MDN pit histograms for the training data (top row) and test data (bottom row).

the model that gives them the best trade-off solution from the Pareto set of solutions.

7 CONCLUSIONS

In this paper we have outlined the two goals of density forecasting. We show that taking only one of these goals into consideration during training of a density forecasting model results in poor performance. To solve this problem we introduce a new technique for density forecasting optimization that uses a multi-objective search algorithm to find the best solution. Our framework can be applied to

most likelihood based density forecasting models. An attractive advantage of this approach is that the underlying model is not augmented in any way so the model can be interpreted in the normal manner. Our experiments have shown that this optimization approach can find models that are better calibrated than those found through negative log-likelihood.

References

1. Gneiting T, Raftery AE, Balabdaoui F, Westveld A. Verifying probabilistic forecasts: Calibration and sharpness. In: Proceedings Workshop on Ensemble Forecasting, Val-Morin, Quebec; 2003. .
2. Rechenberg I. Evolutionsstrategie '94. Tech. rep.. frommannholzboog. Stuttgart; 1994.
3. Deb Kalyanmoy. Multi-Objective Optimisation using Evolutionary Algorithms. Wiley; 2001.
4. Yao Xin. Evolutionary artificial neural networks. In: Proceedings of the IEEE. vol. 87; 1999. p. 1423–1447.
5. Bishop CM. Mixture density networks. Tech. rep.. Neural Computing Research Group, Aston University, Birmingham; 1994.
6. Bollerslev T. Generalized autoregressive conditional hetroskedasticity. Journal of Econometrics 1986;31:307–327.
7. Rosenblatt. Remarks on a multivariate transformation. Annals of Mathematics and Statistics 1952;23:470–472.
8. Diebold FrancisX, Gunther ToddA, Tay AnthonyS. Evaluating density forecasts with applications to financial risk management. International Economic Review 1998;39(4):863–83.
9. Noceti Pablo, Smith Jeremy, Hodges Stewart. An evaluation of tests of distributional forecasts. Journal of Forecasting 2003;22(6-7):447–455.
10. Anderson TW, Darling DA. A test of goodness of fit. Journal of the American Statistical Association 1954;19:765–769.
11. Husmeier Dirk. Neural Networks for Conditional Probability Estimation: Forecasting Beyond Point Predictions. Springer; 1999.
12. Carney Michael, Cunningham Pádraig, Lucey BrianM. Making density forecasting models statistically consistent. Tech. rep.. Department of Computer Science, Trinity College Dublin; 2006.
13. Nelson DB. Conditional heteroskedasticity in asset returns: A new approach. Econometrica 1991;59:347–370.
14. Pareto Vilfredo. Cours D'Economie Politique. Lausanne; 1896.
15. Fieldsend Jonathan, Singh Sameer. Pareto evolutionary neural networks. IEEE Trans Neural Networks 2005;16(2):338–354.
16. Fieldsend Jonathan, Everson RichardM, Singh Sameer. Using unconstrained elite archives for multiobjective optimization. IEEE Trans Evolutionary Computation 2003;7(3):305–323.
17. Carney Michael, Cunningham Pádraig, Byrne Stephen. The benefits of predicting density functions over point predictions in medical decision support: An example in anticoagulant drug therapy. Tech. rep.. Trinty College Dublin; 2005.
18. Moller M. A scaled conjugate gradient algorithm for fast supervised learning. In: Neural Networks. vol. 6; 1993. p. 525–533.