The Benefits of Using a Complete Probability Distribution when Decision Making: An Example in Anticoagulant Drug Therapy

Michael Carney[1], Padraig Cunningham[1] and Stephen Byrne[2]

[1] Department of Computer Science, Trinity College Dublin, Ireland
[2] School of Pharmacy, University College Cork, Ireland

Abstract. In this paper we aim to show how probabilistic prediction of a continuous variable could be more beneficial to a medical practitioner than classification or numeric/point prediction of the same variable in many scenarios. We introduce a probability density forecasting model that produces accurate estimates and achieves statistically consistent predicted distributions. An empirical evaluation of this approach on the problem of warfarin dosage prediction is described and a comparison of results obtained from our probabilistic models with a number of classification techniques on this problem is also shown.

1 Introduction

Optimal decision making can be achieved through perfect information. Unfortunately, in most cases, complete information is not available to the decision maker. This is often the case in medicine, rational decisions must be made in an environment with imperfect information and hidden variables. Probability theory allows us to rationalise our decisions in these circumstances and determine an optimal course of action. Therefore, decision support systems that try to optimise decisions made by practitioners by determining relationships between patient's attributes and empirical evidence should present predictions in terms of probabilities.

In this paper we introduce a new class of model called the Pareto-Evolutionary Mixture Density Network or Pareto-EMDN. The Pareto-MDN forecasts the complete probability distribution of likely outcomes. As with simple point predictions it is important that probability distribution predictions are accurate. In addition to requiring that the probability distribution predictions are *sharp* or accurate, they also need to be well *calibrated* (Diebold et al., 1999). This means that the shape of the distribution should correspond to probability for the real phenomenon i.e. events that have a 33% probability should occur 1/3 of the time. By identifying these goals and optimising our model directly on both of them, we achieve better results than traditional models that only optimise under one objective. We demonstrate its usefulness by applying it to a medical decision problem.

The paper is laid out as follows. In Section 2 we outline the motivation of our work. Section 3 gives a brief review of probability density function estimation techniques. Section 4 explains the objectives of density forecasting and introduces the error functions that can be used to optimise these objectives. In Section 5 we describe the model that we use to achieve our defined objectives. An analysis of the described model on an artificial data set is shown in Section 6. Our case study is described in Section 7 and finally Section 8 concludes.

2 Regression Problems in Medical Decision Support

The supervised learning methodology from Machine Learning research is very useful in medical decision support because it facilitates the development of prediction systems from historic data. The notion of 'supervision' evokes the idea of a learner who associates class labels to training examples. These labeled examples can be used to train a prediction system that will be able to assign class labels to unlabeled examples. Such classification systems are common in medical decision support where the class might for instance be a diagnosis for a set of symptoms. In classification problems the outcome (dependent variable) is a class label, however there are also prediction problems where the outcome is a numeric value. These are termed regression problems and examples in medical decision support include:

- predicting Glycated Haemoglobin (HbA1C) levels for diabetes patients (Braatvedt et al., 1997).
- predicting life expectancy for children with cystic fibrosis (Aurora et al., 2000).
- predicting life expectancy for cancer patients (Vigano et al., 1999).
- predicting the life-span of artificial hip-joints (Young et al., 1998).
- predicting International Normalized Ratio (INR) values for patients on anticoagulant therapy (Ginsberg et al., 2001; Byrne, 2002)

A variety of techniques exist for building prediction models for regression problems such as these. In addition to the classical linear regression models from statistics (Weisberg, 2005), artificial neural networks are also well suited to problems of this type.

One problem with numeric predictions in applications such as those listed above is that precise numeric predictions produce estimates that give the user a false impression of perfect accuracy. For this reason *point* predictions are sometimes avoided by discretizing the outcome variable into a series of class labels, e.g. $\{< 18months, \geq 18months\}$. Issues of imprecision can also be addressed by augmenting the point prediction with a confidence interval or error bars.

A more serious problem exists in the situation where the point prediction is within some acceptable range but in reality there is a significant likelihood of a completely unacceptable outcome. Consider a situation where a prosthesis is predicted to last for 12 years but there is a 15% chance of it failing within 2 years. The solution presented in this paper addresses this problem by predicting a complete probability density function for the numeric output value (see example in Fig. 5). While this ability to describe the complete probability density function of the outcome offers more comprehensive information for the clinician it also raises some interesting issues. The first issue is to decide how to present this extra information to the user - some alternatives are considered in Section 7.2.

The second issue (from a research perspective) is how to evaluate the accuracy of these predictions. Evaluating the accuracy of a system that produces point predictions is straightforward; the accuracy can be quantified by measuring the root mean squared error of predictions and a good estimate of generalization accuracy can be produced using cross-validation. Working with a complete probability density function introduces another dimension in assessing the quality of of a prediction. Not only does the accuracy of the prediction need to be assessed (in Section 4 we propose a *sharpness* measure to achieve this), but the overall *shape* of the distribution must also be assessed. Intuitively, outcomes that are predicted to have a 10% probability of being over a threshold should be over that threshold one time in ten. In Section 4 we adopt a measure called *calibration* (Diebold et al., 1999) that captures this. So, while these density functions have great promise in the extra information that they offer the user, the models are difficult to build as there are two competing criteria that need to be optimised.

3 Review of Density Forecasting Techniques

The standard solution to regression problems is to apply a sum-of-squares error function to N training data pairs, $(\mathbf{x}_1, t_1), ..., (\mathbf{x}_n, t_n)$, to predict a conditional mean of a new unseen data vector, $\langle t_{n+1} | \mathbf{x}_{n+1} \rangle$. This form of *point prediction* is often inadequate in practice because there is no indication of the level of uncertainty in the model's predictions. The most rudimentary method of quantifying this uncertainty is to report a mean-squared-error (MSE). This is the average uncertainty in the model over the complete input training set. The combination of a point prediction with MSE makes the assumption that the uncertainty in the model is described by a Gaussian conditional density function with constant variance equal to the MSE.

Quantifying the uncertainty in each individual prediction is our goal. To model this uncertainty we could use confidence intervals, however, De Finetti (1974) argues that the only concept needed to express uncertainty is probability, and so all predictions over a future event should be made in terms of probability distributions. Consequently, our goal, and the goal when making any prediction with uncertainty, is to create a model of the process as a sequence of probability density functions. This means that for any given input value e.g. $\mathbf{x}_{n+1}$, our model should produce a conditional density function $p(t_{n+1} | \mathbf{x}_{n+1})$.

There exist a number of approaches to obtain estimates of the conditional density function. Fraser and Dimitriadis (1993) initially developed a Hidden Filter Hidden Markov Model using the EM algorithm to produce predictions of conditional density forecasts. Weigend and Nix (1994) suggest a neural network architecture that predicts the mean value and the local error bars (standard deviations) for given inputs, however, this approach assumes the data generating distribution is a Gaussian.

Neuneier et al. (1994) and Bishop (1995) developed a semi-parametric conditional density estimation network or mixture density network (MDN)[3]. This approach removes the need to assume a Gaussian and can model unknown distributional shapes. Husmeier (1999) developed a two hidden layer universal approximator called a random vector functional link (RVFL) which also can predict the whole conditional density function. Husmeier compares his RVFL model with an MDN and shows that both techniques produce results with equivalent accuracy. In this paper we will use an MDN as our base model to produce accurate calibrated predictions.

4 Density Forecasting Objectives: Sharpness and Calibration

Most techniques for evaluating predictions are based on the principle that the closer predictions are to the actual observation the better the predictor. When making a probabilistic prediction there is an added complexity. Direct comparison between the prediction and the actual observation is hindered because the prediction is a function representing the probable outcomes and the observed value is a single value. To resolve this problem we can define our goals and optimise our models to maximise them.

The density forecasting literature suggests that density forecasts should maximise sharpness subject to calibration (Gneiting et al., 2003). Sharpness is the amount of spread of the predicted probability around the observation. A sharp prediction is one that has small variance and high density at the observation. Calibration is the statistical consistency, or the empirical validity, of the models probability predictions. A well calibrated model will produce probability estimates that are consistent over a set of predictions. For example, a model that makes 20 predictions with an 80% probability of a particular outcome should result in 16 occurrences of that outcome.

Maximising sharpness and calibration is not a simple objective. Optimising sharpness solely can result in a poorly calibrated model and a well calibrated model does not necessarily produce useful predictions. Therefore a probabilistic model must trade-off between calibration and sharpness. However, if the predicted distributions are to be considered useful it is necessary for the predictions to be calibrated[4]. Therefore, we should think of calibration as a constraint for our probability distributions. Furthermore, we want our model to be predictive so we also want to maximise sharpness subject to our calibration constraint.

[3] Neuneier et al. use the name Conditional Density Estimation Network and Bishop uses the name Mixture Density Network, for consistency we will use only MDN from now on.

[4] A sharp but uncalibrated density forecasting model produces probability distributions that can not be used, thus it is no more useful than a point prediction model.

We have now defined our goals, in the next two sub-sections we describe both sharpness and calibration objective functions that will allow us to optimise our density forecasting models on these goals.

4.1 Sharpness Cost Function

To determine the level of sharpness in our predictions we use the negative log-likelihood, also known as the negative log predictive density, or ignorance score (Good, 1952). This is the measurement of sharpness that is most commonly used in the density forecasting literature. The negative log-likelihood penalizes relative to the predicted probability density at the actual observation. In Machine Learning, it is the mean value of the negative log-likelihood over all predictions that is used to determine the error in the predicted densities.

$$NL = \frac{1}{N}\sum_{i=1}^{N} -\log(p(t_i|\mathbf{x}_i)) \tag{1}$$

Where, $p(t_i|\mathbf{x}_i)$ is the predicted density given the input vector $\mathbf{x}$ for the observation t for case i, and N is the number of observations being evaluated.

The negative natural log is used to make this loss function a minimisation rather than maximisation score i.e. the error function is negatively oriented. Although the negative log-likelihood is a correct scoring rule in the sense that it penalizes both under confident and over confident predictions, its weakness is that as the error increases the negative log-likelihood error grows quadratically not linearly [5]. However, the datasets being studied here do not have sufficiently extreme outliers to be affected by this phenomenon.

4.2 Calibration Cost Function

Calibration is a joint property of the observations and the predictions. It has always been an *ex post* evaluation and appears never to have been specifically optimised during the training process until now. The standard means of evaluation requires the visual assessment of the Probability Integral Transform (PIT) histogram (Diebold et al., 1998). A well calibrated model will have a set of PIT values z, one for each observation, that are uniform between the interval [0,1], and independent and identically distributed, *i.i.d.*.

$$z = -\int_{-\infty}^{t} p(u)\mathrm{d}u \tag{2}$$

[5] The exponential nature of the loss function is the cause of its sensitivity to outliers. If extreme events exist in the data then this may affect the ability of this cost function to optimise. In cases where this happens, (Weigend & Shi., 2000) suggest using a trimmed mean to get around this issue. We suggest using an alternative sharpness error measure such as the continuous ranked probability score, (Matheson & Winkler, 1976).

Where, t, is the target variable, and $p(\cdot)$ is the probability density for target t.

The intuition behind this approach is, that 10% of the cumulative probabilities should be in the range 0-10% another 10% of cumulative probabilities should be in the range 10%-20%, etc.. A histogram of the z values is the standard form of presenting this information. A well calibrated model will return a histogram of equal bin heights[6]. A correct calibration error function must evaluate a model's z values based on their consistency with a U(0,1) distribution. Noceti et al. (2003) empirically tested a number of statistical methods for evaluating z values in this way and concluded that the most powerful statistic is the Anderson-Darling test[7]. Therefore, our calibration error function is,

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

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

This returns the Anderson-Darling test statistic. This is not a strictly proper scoring rule in the sense that you can not assess the calibration of a single prediction. Calibration can only be determined for a set of predictions. However, (Anderson & Darling, 1954) suggest that this statistic produces assessments of uniformity that are valid for sample sizes larger than 40. Therefore, this loss function should only be used when the dataset size is greater than 40.

5 Multi-objective Optimisation of Mixture Density Networks

Mixture Density Networks (MDN) refer to a special type of artificial neural network in which the target is represented as a probability distribution, or, more specifically, a conditional probability density function. MDNs were first introduced by Bishop (1995) and shown to successfully describe the conditional distribution for the multimodal inverse problem and Brownian process. Since then they have been successfully applied to both financial (**?**) and meteorological data sets (Cornford et al., 1999).

MDNs represent the conditional density function by a weighted mixture of Gaussians known as a Gaussian Mixture Model(GMM). GMMs are a flexible,

[6] The PIT histogram gives insight into the biases of the prediction technique. For example, a U shaped PIT histogram that has a large number of cumulative predictive densities at both extremes suggests that the model is producing predictions that are over-confident.

[7] Noceti et al. (2003) tested on the 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.

convenient, semi-parametric means of modeling unknown distributional shapes. The conditional density function is described in the form

$$p(t|\mathbf{x}) = \sum_{i=1}^{c} \alpha_i(\mathbf{x})\phi_i(t|\mathbf{x}) \tag{4}$$

where ϕ_i is a Gaussian as follows,

$$\phi_i(t|\mathbf{x}) = \frac{1}{(2\pi)^{q/2}\sigma_i(\mathbf{x})^c} \exp(-\frac{\|t - \mu_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})2}) \tag{5}$$

where c is the number of components in the mixture. The parameter α_i is the Gaussian weight or mixing coefficient and $\phi_i(t|\mathbf{x})$ represents the ith Gaussian component's contribution to the conditional density of the target vector t. For a full discussion on the Mixture Density Network architecture see (Bishop, 1995).

MDNs use the negative log-likelihood error function during optimisation. Due to the complexity of the local error curvature, straightforward gradient descent fails. To overcome this problem Bishop uses a Scaled Conjugate Gradients (SCG) (Moller, 1993) optimisation technique to dynamically adjust the learning rate during the training process to converge to a minimum. We use an Evolutionary Strategy to optimise our network in terms of sharpness and calibration.

5.1 Sharpness and Calibration Optimisation for MDN's

In the preceding sections we introduced the concept of sharpness and calibration in a density forecasting model, we also briefly introduced the MDN. Now we will describe how to train an MDN to optimise both sharpness and calibration.

Sharpness and calibration are conflicting objectives, optimising one compromises the other. Therefore, optimisation of both objectives is a multi-objective search problem. Unlike single objective search problems, multi-objective search problems have a set of global optima called the Pareto optimal set. Each member of the Pareto optimal set represents a trade-off solution to the multiple objectives. Progressing solutions towards the Pareto optimal set while maintaining a diverse set of trade-off solutions is the goal of a multi-objective search. The set of trade-off solutions is known as the non-dominated set, so called because no solution in the non-dominated set is worse in all objectives than (dominated by) another solution in the set. There are a number of comprehensive reviews of multi-objective optimisation techniques, for example (Deb, 2001; Van Veldhuizen & Lamont, 1998; Coello, 1999)

Fortunately, Evolutionary Artificial Neural Networks (EANN's) allow us to optimise an ANN using an evolutionary algorithm. EANN's have become increasingly popular over recent years. They have the ability to find good, global solutions to complex optimisation problems while simultaneously optimising the network architecture (Yao, 1999). They also provide a means of optimising an ANN on a non-differentiable objective function. More recently, a framework for a Pareto Evolutionary Neural Network (Pareto-ENN) (Fieldsend & Singh, 2005) has been developed to optimise ANN's under multiple objectives. We have developed an extension to this basic framework that optimises MDN's. Specifically, we use Fieldsend and Singh (2005), $\mathcal{M}_v$, optimisation framework. Our adaptation uses the two error functions described in Section 4 above, NL and zA^2.

To optimise a neural network using an Evolutionary Strategy (ES) you must be able to describe your network in terms of a decision vector that can be mutated by the ES. We represent our networks as a single vector of real values. To do this we simply carried out columnwise vectorisation of the weight and bias matrices of each network and concatenated the resulting vectors. We then created a mask for this vector so that mutations could be easily applied. The mutation strategy we applied included hidden node addition/deletion, weight addition/deletion and weight adjustment. This approach optimises the specified objectives as well as the network's architecture.

The initial training solution is determined through standard SCG training of an MDN. The resulting MDN characteristically will have excellent sharpness but poor calibration. This is because the objective function used is the negative log-likelihood which does not take calibration directly into account. From this initial sharp prediction that is near the Pareto optimal set our multi-objective optimisation should fan out from this point to create a diverse non-dominated set of good trade-off solutions. The initial population is determined by mutation of this initial solution. The ES used in the experiments in this paper is an ES$(1 + 1)$. At each epoch a decision vector is selected from the archive of non-dominated solutions using the Partition Quasi-Random Selection technique (Fieldsend et al., 2003). The selected decision vector is mutated and compared against the archive. If it is not dominated it is included in the archive of non-dominated solutions. This process is repeated for 25,000 epochs. For a full discussion on implementation of a Pareto-ENN see (Fieldsend & Singh, 2005).

6 Analysis of the Pareto-EMDN on Artificial Data

In this section we will demonstrate, using a synthetic dataset, our proposed calibrated density estimating model. The advantage of using a synthetic dataset is that we can develop a data generating process where we know the conditional distribution of the data at every point on the function. For the purpose of this experiment we chose an artificial data source that had both higher moment influences and bimodality. The data takes the form of a random phase shift sinusoid. The function we use is adapted from Lo and Bassu (2002) and is fully described by,

$$f(x) = sin(x + c) + \epsilon \tag{6}$$

Where x is in the interval $[0,\, 2\pi]$, c is equal to 0 with probability 0.75 and 2 with probability 0.25; and

$$\epsilon \sim \frac{(W(\lambda, k) - median)}{5} \tag{7}$$

Where W is the Weibull distribution, with $\lambda = 2$ and $\mathrm{k} = 2$ and *median* is the median of the distribution. The median offsets the errors on the sinusoids and the denominator scales the errors. Datasets can be created by drawing values at random from the interval $[0, 2\pi]$ and evaluating the corresponding function values. We generated 2000 data pairs for training, 1000 for validation and 2000 for testing for each experiment run. Figure 1. is a sample plot of a training set generated from Equation (6).

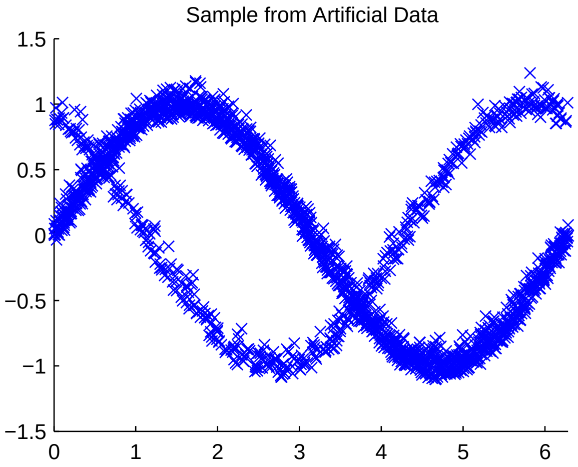

Fig. 1. Sample plot of artificial data set.

6.1 Experiment Details

For each experiment run, a single MDN was trained using a Scaled Conjugate Gradient (SCG) optimisation algorithm for 5000 epochs. The epoch with the lowest negative log-likelihood error on the validation set determined the weights for the network. This is the base network that we use for benchmarking our calibrated model against. It is denoted as, Standard MDN, in figures and tables. In this experiment the MDNs topology has 5 hidden nodes and 12 outputs, which equates to a 4 Gaussian mixture model conditional density estimate.

Before commencing training, the archive and validation archive of the multiobjective search are populated by carrying out 200 mutations on the initial solution determined by the Standard MDN. The parameters for the optimisation process are summarised in Table 1. Training was 25000 epochs. The experiment was repeated 20 times using different initialization vectors, different random seeds and different training, validation and test data sets drawn from the generating function.

Table 1. The parameters of the Pareto-EMDN used on the artificial data sets.

Initial weights	$N(0,1)$
Probability of weight perturbation	0.2
Perturbation	$N(0,1)$ x 0.1
Probability of node addition/deletion	0.02
Probability of weight flip	0.02

6.2 Experiment Results

The artificial data set allows us to visually compare our Pareto-EMDN predicted density functions against the real conditional distribution of the data. Figure 2. is a surface plot of the predicted conditional densities from a member of the validation archive set compared with the real conditional distribution of the data.

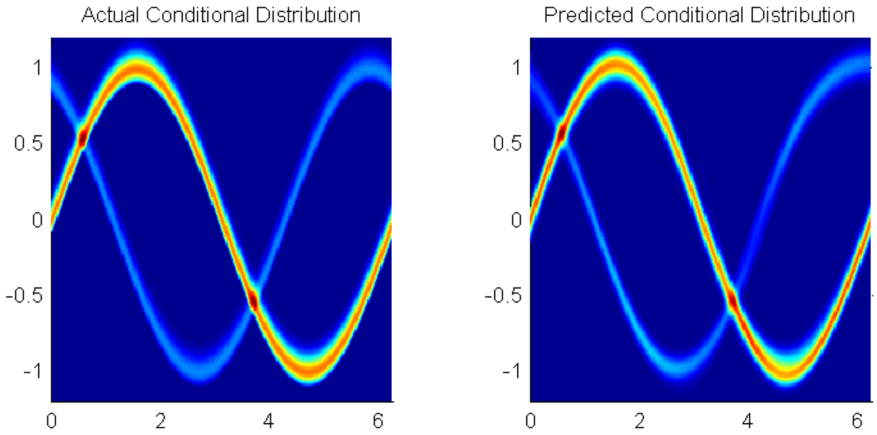

Fig. 2. The plot on the left is the actual conditional distribution of the data and the plot on the right is the predicted distribution. Simple visual comparison shows that the predicted densities accurately capture the dynamics of the data.

The expectation for our Pareto-EMDN models is that training will result in a set of models ranging from accurate and not well calibrated, to less accurate but well calibrated. Figure 3. is a plot of the archive of models produced during training on one run of the experiment. The characteristic shape of the non-dominated set can be clearly seen and the high degree of sharpness of the initial Standard MDN solution is present.

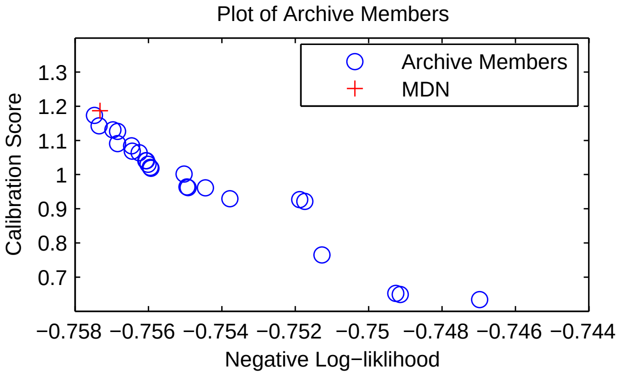

Fig. 3. This is a plot of the objective space after training a Pareto-EMDN on a single run of the artificial data experiment. The initial MDN produced a single very sharp but poorly calibrated model. Using this as an initial solution the Pareto-EMDN creates a set of non-dominated or trade-off solutions varying in calibration and sharpness. The user can then pick the model that best suits their objectives.

As can be seen in Figure 3. our Pareto-EMDN produces a number of candidate models that describe different trade-offs on objectives in decision space. It is intended that the decision maker selects the model that optimises their objectives in the way that best suits their needs. Therefore, to evaluate the effectiveness of our model we can suppose that there are two different decision makers (User A and User B). User A wishes to maximise the accuracy/sharpness of their predictions. The other, User B, wishes to maximise the calibration of his predictions. Table 2. summarises the performance of the selected models over 20 runs of the experiment, assuming the users choose the model that optimised their goals the most on the validation set. We compare results against the Standard MDN.

Table 2. The results show the average error and standard deviation over 20 runs of our model on the artificial data. User A and User B both improve their respective objectives. User B obtains a statistically significant improvement over the standard MDN in the calibration metric. The errors and the standard deviation of the errors are presented in each case.

	NL	zA^2
Standard MDN	-0.736 ±0.057	2.942 ±1.598
User A (Pareto-EMDN)	**-0.737** ±0.056	2.432 ±1.110
User B (Pareto-EMDN)	-0.734 ±0.057	**1.391** ±1.057

This straightforward evaluation shows that there is an improvement in the calibration score. However, it is necessary to assess whether an improved calibration score results in improved calibration of predictions. To do this we carried out 2 tests of calibration that use the fact that we are using an artificial data set, and can determine the true conditional distribution, to evaluate calibration.

6.3 Analysis of the Calibration Score

To determine whether our zA^2 objective function affects the actual calibration of the Pareto-EMDN predictions we developed two tests. The first examines each predictions divergence from the conditional density directly. We used the Kullback-Leibler divergence to calculate the average predicted density divergence from the actual conditional density. The second evaluation is an empirical test on the set of percentiles of the predicted distributions. This test we call the percentile empirical hit rate and is described in detail below.

Kullback-Leibler Results The Kullback-Leibler divergence (1951) is used to determine the divergence of an estimated distribution p from the true distribution p^{true}. If p and p^{true} are identical the Kullback-Leibler divergence is zero. The formula is,

$$K(p, p^{true}) = \int_{-\infty}^{\infty} \mathrm{d}y p(y) \log \left(\frac{p(y)}{p^{true}(y)} \right) \tag{8}$$

However, the classic Kullback-Leibler formula, in this form, is asymmetric[8]. We can correct for this asymmetry by adjusting the formula to the following

$$KL(p, p^{true}) = \left| K(p, p^{true}) - K(p^{true}, p) \right| \tag{9}$$

Figure 4. has a plot of the normalised average KL divergence against the normalised zA^2 for the set of non-dominated models produced after 1 run of training. The plot shows a strong correlation between the zA^2 and the KL divergence. This is compelling evidence towards confirming our conjecture that it is possible to optimise an MDNs calibration.

Empirical Validity - Percentile Test A second test for calibration that we devised is a percentile evaluation that empirically tests the consistency of the predictions. Intuitively, over a large set of predictions, the number of target values within the 5th percentile of the predicted distributions should be 5%. We call this value the hit rate.

[8] $K(p, p^{true}) \neq K(p^{true}, p)$. Although the K divergence is often called a distance metric this is in fact a misnomer. The asymmetric nature of the K means that the 'distance' from p^{true} to p is not the same as the 'distance' from p to p^{true}. This fact means that the K is not a true distance metric.

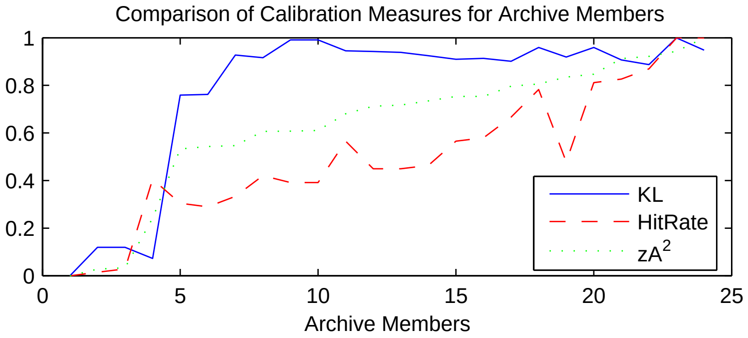

Fig. 4. A comparison of measures of calibration. The correlation coefficient between the KL divergence and the Calibration score is 0.9 in this example. Similarly, the correlation coefficient between the Hit Rate and the Calibration score is also 0.9.

$$HitRate = \left| \left(\frac{\sum_{i=1}^{N} 1 \left\{ \Phi_i^{-1}(q) > t_i \right\}}{N} \right) - q \right| \tag{10}$$

Where, q, is the specified percentile and Φ^{-1} is the inverse cumulative distribution.

We tested every 5th percentile between 5% and 95%. We measured the absolute difference between the percentile being tested and the hit rate. Figure 4. shows the relationship between the hit rate and the zA^2 score.

This evaluation empirically demonstrates that our zA^2 error function is a valid means of improving the calibration of our models.

7 A Case-Study in Anticoagulant Drug Therapy

Warfarin is the only oral anticoagulant licensed for use in the Republic of Ireland. The indication for its use in many patients is to reduce the embolic risk associated with diseases such as atrial fibrillation, left ventricular dysfunction, deep vein thrombosis and mechanical aortic valve replacement (Byrne, 2002). While a patient is receiving therapy with an oral anticoagulant, careful monitoring and adjustment of their dosage is necessary so that the length of time it takes for their blood to clot is maintained within a predefined range, known as the International Normalised Ratio (INR) range (Sachdev et al., 1999). Giving a patient too low a dose increases their risk of developing a clot while giving a patient too high a dosage exposes them to potential bleeding complications (Scottish Intercollegiate Guidelines Network, 1999). The current method of determining the dosage of warfarin required by a particular patient is on a trial and error basis. The reason for this approach is that every patient will respond to warfarin differently as each patient's pharmacological response is affected by many exogenous factors other than the dosage prescribed, some of which are listed in Table 3.

Our data was collected from patients undergoing anticoagulant therapy. The variable that we are trying to predict is the patient's observed INR value. The Target INR value for all patients in our data set is 2.5 (Target INR range of 2.0-3.0). If the patient's INR measurement drops below 1.5, they are said to be at unacceptable risk of clotting and if a patient's INR rises above 4.5 they are exposed to increased risk of bleeding. Fifteen patient attributes were recorded for this problem. Table 3 is a summary of these attributes.

Table 3. The fifteen attributes compiled for each case in our input set.

Age
Weight
Height
Marital Status
Prescribed Medication
Gender
Smoking
Alcohol
Compliance
Previous warfarin dosage
Measured INR
Duration of therapy
Target INR value
Hours since last warfarin dosage
Any adverse events

7.1 Predicting Density Functions for Anticoagulant Drug Therapy

In this section we will show how our Pareto-EMDN model can be used to aid a physician when prescribing warfarin. The physician's objective is to prescribe the dosage that maximises the likelihood of the patient's INR value being 2.5 at their next visit. An example of how our model can help a physician achieve this goal follows. The target INR value is based on the patient's disease state and is specified by the prescribing physician. Guidelines regarding specific ranges are endorsed by the British Society for Haematology (1998).

Given a patient's attributes (see Table 3), a physician can use our system to estimate the affect a warfarin dosage will have on a patient. In our example, the physician wishes to estimate the affect a 1mg, 2.5mg or a 4mg dosage will have on his patient's INR. Figure 5 shows the set of predicted INR distributions for a specific patient if prescribed these warfarin dosages. The probability distributions clearly show that a prescription of 2.5mg will greatly reduce the patient's risk of either clotting or bleeding. We suggest that the probability distributions should be presented with a table of the risks of the patient's INR moving outside their therapeutic index, (see Table 4).

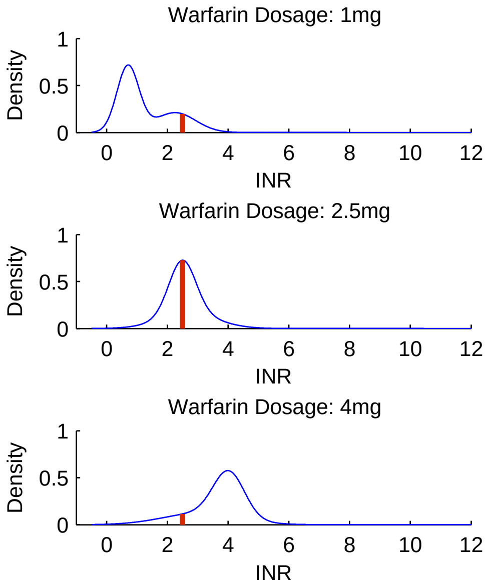

Fig. 5. This figure shows the predicted INR distributions for a patient under different warfarin dosages. The target INR for the patient is 2.5 (red line). Three scenarios are presented; the predicted INR distribution if the patient is prescribed (top) 1mg warfarin (middle) 2.5mg warfarin (bottom) 4mg warfarin.

Table 4. This is a table of estimated risk of the patient exceeding specific INR values determined from the predicted distributions shown in Figure 5. This should be used in conjunction with the distributions and not as a substitute.

Dosage	INR ≤ 1.5	$1.5 <$ INR < 4.5	INR ≥ 4.5
1 mg	68.1%	31.8%	0.1%
2.5 mg	4.6%	94.4%	1.0%
4 mg	3.8%	80.6%	15.6%

7.2 Evaluation

In our artificial data example above we demonstrated empirically the effectiveness of our zA^2 cost function at optimising the calibration of our density forecasting model. Table 5. shows comparative results on both the NL and zA^2 cost functions for three model types. Performance of the Pareto-EMDN is benchmarked against the Standard MDN over 10-fold cross-validation. Not surprisingly, User A achieves the best sharpness score and User B achieves the best calibration score on average.

Table 5. Results after 10-fold cross-validation on the warfarin data set. User A selected members from the non-dominated set that maximised sharpness. User B selected members that maximised calibration. The results demostrate the Pareto-EMDN's ability to optimise density forecasts under multiple objectives.

	NL	zA^2
Standard MDN	1.106 ±0.151	2.577 ±3.772
User A (Pareto-EMDN)	**0.945** ±0.114	1.300 ±0.709
User B (Pareto-EMDN)	1.001 ±0.094	**1.279** ±0.970

The unconditional distribution of INR values that we have compiled is shown in Figure 6. Mac Namee et al. (2002) showed the effect of bias in this dataset and suggested a technique to meliorate results using an ensemble of point predictors. By maximising calibration, we correct the bias in the predicted distributions, a corollary to this is that the tails of the predicted distributions on a calibrated model should be statistically consistent. As the tail predictions are of more interest, this is an important improvement to the predictions. To show that our calibration score improves the tail errors we repeated the empirical validity test that we carried out on the artificial data. However, this time we assessed the validity of the tail percentiles. In Table 6. we compare the results of this test. The calibrated models selected by User B perform the best. This improvement is statistically significant against both User A and the Standard MDN in a two-tail t-test ($p = 0.05$).

Comparative Evaluation The final evaluation that we have carried out compares the classification accuracy of a number of other prediction approaches to this problem. We have converted the dataset into a classification problem by labeling cases that have an INR of less than or equal to 1.5 as **clotting**, those with an INR greater than or equal to 4.5 as **bleeding** and patient's with an INR between these two values as **normal**. This results in 835 cases of normal, 124 of clotting and 47 of bleeding. Table 7. below compares the classification accuracy for each class over 10-fold cross-validation for each of the following classifiers.

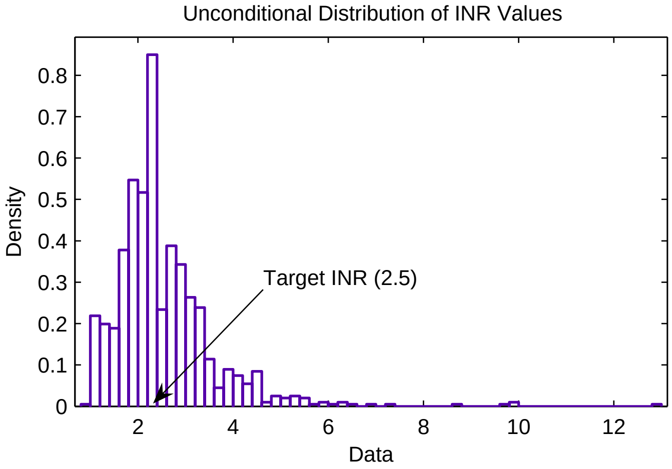

Fig. 6. A histogram of INR values of the 1006 cases in our data set. Every case in this dataset has been assigned a target INR value of 2.5. It is clear that there is a skew in the data with the majority of cases centered around the mode, 2.5. There is a fat, right tail with values stretching out as far as 13. The left tail, however, only extends as far as 0.5. Physicians are interested in cases that are on the tails of the distribution, as these values represent patient's that are outside their therapeutic index and are at risk of bleeding or clotting.

Table 6. Results after 10-fold cross-validation on the warfarin data set. User A selected members from the non-dominated set that maximised sharpness. User B selected members that maximised calibration. Results show the average Hit Rate error over the percentiles 0.01, 0.05, 0.1, 0.9, 0.95, 0.99.

	HitRate
Standard MDN	0.0834 ±0.030
User A (Pareto-EMDN)	0.0782 ±0.032
User B (Pareto-EMDN)	**0.0722** ±0.030

- C4.5 - decision tree algorithm (Quinlan, 1993).
- Logistic Regression - multinomial logistic regression model with a ridge estimator (Witten & Frank, 2005).
- Naive Bayes - simple probabilistic classifier (Witten & Frank, 2005).
- kNN - k nearest neighbour classifier (Doyle et al., 2005).

We convert the predicted density function produced by our models into classifications by determining the predictive density for each class range and classifying based on a probability threshold. We based our thresholds on the percentage contribution of each class to the unconditional distribution.

Table 7. Results after 10-fold cross-validation on the warfarin classification dataset for 7 different classifiers. % represent class accuracies.

Classifier	**clotting**	**normal**	**bleeding**
C4.5	95.2%	29.0%	12.8%
Logistic Regression	97.4%	27.4%	23.4%
Nave Bayes	92.2%	10.5%	17.0%
kNN (k=1)	91.3%	21.8%	10.6%
kNN (k=3)	97.0%	9.7%	6.4%
Standard MDN	76.9%	71.0%	72.3%
User A (Pareto-EMDN)	76.6%	77.4%	74.5%
User B (Pareto-EMDN)	79.6%	78.2%	76.6%

The specificity and sensitivity[9] of the classifiers are shown in Table 8. To calculate the specificity and sensitivity of the classifiers we reduced the number of classes from 3 to 2. This was achieved by combining **bleeding** and **clotting** into a new **warning** class and leaving the **normal** class as it was. This approach probably over simplifies the problem, but the results are useful to demonstrate the effect of skewed data on classifiers in standard medical evaluation terms.

Table 7. and Table 8. clearly show that the skew in the data creates a bias that causes very poor classification accuracy for the minority classes in the non-MDN classifiers, but extremely high prediction accuracy in the majority class (**normal**). The Pareto-EMDN shows a consistent accuracy across each class suggesting that it has overcome this bias problem. This consistency allows the practitioner to be more confident when using this system in practice. Combining this with a well calibrated probability distribution estimate of the INR further enhances the insight the user gets from the predictive model. By clearly displaying the uncertainty in the predictions, density forecasting mitigates some of the concerns of using artificial neural networks in the medical domain and provides an attractive alternative to the more traditional prediction techniques.

[9] Specificity is the proportion of people that have an unsafe INR value and the classifier has predicted this. Sensitivity is the proportion of people that have a safe INR value and the classifier has predicted they are safe.

Table 8. Specificity and sensitivity of classifiers on the warfarin problem when converted into a binary classification task.

Classifier	**Specificity**	**Sensitivity**
C4.5	24.9%	95.0%
Logistic Regression	26.5%	97.2%
Nave Bayes	13.0%	91.2%
kNN (k=1)	19.5%	90.5%
kNN (k=3)	8.9%	96.7%
Standard MDN	76.7%	75.8%
User A (Pareto-EMDN)	82.4%	75.6%
User B (Pareto-EMDN)	83.1%	78.6%

8 Conclusions

We propose a new approach to density forecasting. Existing methods optimise on sharpness alone and don't explicitly address calibration. Thus they can have poor performance in identifying situations when rare events have a significant chance of occurring. Our density models are optimised directly to achieve sharpness and calibration. To do this we introduce a new class of models called the Pareto-Evolutionary Mixture Density Network or Pareto-EMDN. An experiment on an artificial data set shows that our new objective function can optimise calibration during training. An application of this model in the medical domain shows that the model can be especially useful when the response variable has a skewed distribution, overcoming biases that adversely affect traditional estimation techniques. The type of application described in Section 7.1 shows that this extra information can be very useful in medical decision support.

Bibliography

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

Aurora, P., Wade, A., Whitmore, P., & Whitehead, B. (2000). A model for predicting life expectancy of children with cystic fibrosis. *European Respiratory Journal*, *16*, 1056–1060.

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

Braatvedt, G., Drury, P., & Cundy, T. (1997). Assessing glycemic control in diabetes: Relationships between fructosamine and HbA1c. *New Zealand Medical Journal*, *110*, 459–62.

British Society for Haematology (1998). Haemostasis and thrombosis task force. guidelines on oral anticoagulation: Third edition. British Journal of Haematology.

Byrne, T. S. (2002). *Warfarin therapy in the republic of ireland: The potential role of neural networks in optimising therapy.* Doctoral dissertation, Trinity College Dublin.

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

Cornford, D., Nabney, I. T., & Bishop, C. M. (1999). Neural network-based wind vector retrieval from satellite scatterometer data. *Neural Computing and Applications.*

De Finetti, B. (1974). *Theory of probability, volume 1.* Wiley.

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

Diebold, F., Hahn, J., & Tay, A. (1999). Multivariate density forecast evaluation and calibration in financial risk management: High-frequency returns on foreign exchange. *Review of Economics and Statistics*, *81*, 661–673.

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

Doyle, D., Loughrey, J., Nugent, C., Coyle, L., & Cunningham, P. (2005). Fionn: A framework for developing CBR systems. *Expert Update*, *8*, 11–14.

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.

Fraser, A. M., & Dimitriadis, A. (1993). Forecasting probability densities by using hidden markov models with mixed states. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison Wesley.

Ginsberg, J., Crowther, M., White, R., & Ortel, T. (2001). Anticoagulant therapy. *Hematology (Americal Society of Hematology Education Program)* (pp. 339–57).

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).*

Husmeier, D. (1999). *Neural networks for conditional probability estimation: Forecasting beyond point predictions.* Springer.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*, 76–86.

Lo, J., & Bassu, D. (2002). Robust approximation of uncertain functions where adaptation is impossible. *Proceedings of IEEE International Joint Conference in Neural Networks, Hawaii* (pp. 1889–1894).

Mac Namee, B., Cunningham, P., Byrne, S., & Corrigan, O. (2002). The problem of bias in training data in regression problems in medical decision support. *Artificial Intelligence in Medicine*, *24*, 51–70.

Matheson, J., & Winkler, R. (1976). Scoring rules for continuous probability distributions. *Management Science*, *22*, 1087–1095.

Moller, M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* (pp. 525–533).

Neuneier, R., Hergert, F., Finnoff, W., & Ormoneit, D. (1994). Estimation of conditional densities: A comparison of neural network approaches. *In Proceedings of ICANN'94* (pp. 689–692). Springer.

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

Quinlan, J. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann Publishers Inc.

Sachdev, G., Ohlrogge, K., & Johnson, C. (1999). Review of the fifth american college of chest physicians consensus conference on antithrombotic therapy; outpatient management of adults. *Am J Health-Syst Pharm*, *56*, 1505–1514.

Schittenkopf, C., Dorffner, G., & Dockner, E. J. (2000). Forecasting time-dependent conditional densities: A semi-nonparametric neural network approach. *Journal of Forecasting.* Chichester.

Scottish Intercollegiate Guidelines Network (1999). Pregnancy and the puerperium. in: Antithrombotic therapy. SIGN Publication No. 35. Edinburgh: SIGN.

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

Vigano, A., Dorgan, M., Bruera, E., & Suarez-Almazor, M. (1999). The relative accuracy of the clinical estimation of the duration of life for patients with end of life cancer. *Cancer*, *86*, 170–176.

Weigend, A. S., & Nix, D. (1994). Predictions with confidence intervals (local error bars). *In Proceedings of the Int. Conf. Neural Info. Processing (ICONIP'94)* (pp. 847–852). Seoul, Korea.

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

Weisberg, S. (2005). *Applied linear regression, 3rd edition*. Wiley.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques, 2nd edition*. Morgan Kaufmann.

Yao, X. (1999). Evolutionary artificial neural networks. *Proceedings of the IEEE* (pp. 1423–1447).

Young, N., Cheah, D., Waddell, J., & Wright, J. (1998). Patient characteristics that affect the outcome of total hip arthroplasty: a review. *Canadian Journal of Surgery*, *41*, 188–95.