Predicting Probability Distributions for Surf Height Using an Ensemble of Mixture Density Networks

Michael Carney MICHAEL.CARNEY@CS.TCD.IE
Pádraig Cunningham PADRAIG.CUNNINGHAM@CS.TCD.IE
Jim Dowling JIM.DOWLING@CS.TCD.IE
Ciaran Lee LEECN@NETSOC.TCD.IE
Department of Computer Science, Trinity College Dublin, Dublin 2, Ireland

Abstract

There is a range of potential applications of Machine Learning where it would be more useful to predict the probability distribution for a variable rather than simply the most likely value for that variable. In meteorology and in finance it is often important to know the probability of a variable falling within (or outside) different ranges. In this paper we consider the prediction of surf height with the objective of predicting if it will fall within a given 'surfable' range. Prediction problems such as this are considerably more difficult if the distribution of the phenomenon is significantly different from a normal distribution. This is the case with the surf data we have studied. To address this we use an ensemble of mixture density networks to predict the probability density function. Our evaluation shows that this is an effective solution. We also describe a web-based application that presents these predictions in a usable manner.

1. Introduction

There are many prediction problems where simple 'point' predictions are not adequate to guide action. In meteorology and in finance it is often important to know the probability of a variable falling within (or outside) different ranges. It might be that a 20% probability of flooding may be enough to place emergency services on alert. Or the best returns in financial trading might come from events that have a small but significant probability. Problems such as these have generated an interest in the area of probability density forecasting. Research in this area is complicated because the data may have a multi-modal generating distribution. In addition, in many scenarios, variance is input-dependent. Recently, research in the financial domain has suggested that the reason why non-linear models do not perform significantly better than linear models is because the non-linearity in the data is due to input-dependent higher order moments (Clements & Smith, 2000).

In this paper we consider the prediction of surf height with the objective of predicting if it will fall within a given 'surfable' range. To produce these ranges we have created a model that predicts the whole conditional probability density function for the target, from this we can then extract the probability of the estimated maximum surf height falling between the specified ranges from the density function. The surf data has a highly skewed unconditional distribution. We intend to show that our ensemble of probabilistic models can outperform standard neural networks on data that is conditionally skewed. We use Mixture Density Networks (MDNs) as our base model, this class of model has been shown to be successful for complex multi-valued functions where standard regression models fail.

The paper is laid out as follows. In Section 2 the details of the surf prediction problem are presented. In Section 3 a review of the issues associated with predicting distributions is presented and our method based on ensembles of Mixture Density Networks is described. Section 4 discusses the issues posed in evaluating density predictions and describes the metrics we use. These metrics are used to analyze the performance of the ensemble of MDNs on artificial data in Section 5. The Surf Prediction Engine is described in Section 6 and an evaluation of its performance is

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

presented in Section 7.

2. The Surf Prediction Problem

The popular sport of surfing requires suitable ocean and weather conditions. Of specific interest to surfers is the near-shore wave height at locations where surf breaks are accessible. The prediction of wave heights is aided by the US National Oceanic and Atmospheric Administration (NOAA) which provides open ocean wave forecasts online for specific locations in the Pacific and Atlantic oceans (Tolman, 1999). Any ocean wave can be characterised as a vector quantity, with the direction expressed in terms of degrees (clockwise, from Polar North) and the magnitude represented by two scalar quantities: the wave height (vertical distance from trough to crest) and its period (Komen et al., 1994). The period of a wave is directly proportional to its kinetic energy. NOAA generates forecasts of ocean waves using both the WAM model (Hasselmann et al., 1988), a model of the physical processes governing wave evolution, and real-time data from moored buoys. Seven-day forecasts are provided for the estimated height, period and direction of waves at the buoy locations. The relationship, however, between open ocean wave height and the subsequent surf height at near-shore surf breaks is complex. Physics allows us to model how waves can be refracted, reflected, diffracted and ultimately break when they come into contact with shallow water or land (Butt et al., 2004). However, mathematical models based on physics require precise parameters not only of the wave dynamics at buoys, but also of islands, shoreline features, the angle of exposure of the surf break and the profile of the sea floor around the surf break. Partial, informal information on these parameters is available in surf guides (Colas, 2004) (e.g. minimum and maximum height at which waves are surfable), but due to the lack of available information on the many other parameters, it is not currently possible to build general linear or non-linear models for predicting wave heights at the diverse surf breaks around the world.

Our approach is intended to sit in between the low resolution open ocean forecast and the very high resolution of a physics model. We use an ensemble of well calibrated mixture density networks to obtain accurate forecasts of the probable maximum surf-zone wave heights given data buoy readings. The ensemble is trained on data that combines the environmental readings from the off-shore data buoy with local, expert, visual recordings of the surf-zone maximum wave heights.

We have compiled a new database that combines surf zone observations with off-shore buoy readings. The data used in this study comes from two primary sources. The off-shore environmental data needed to make the near-shore surf predictions comes from a moored buoy[1] off the south west of Ireland. The observations of the near-shore wave heights are recorded by a surf expert[2]. The observations are coupled with the buoy data by calculating the time delay between the buoy reading and the experts observation using the wave period and buoy to shore distance. The database contains 260 vectors.

3. Predicting Distributions

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 ar-

[1]The National Center for Environmental Prediction (NCEP) maintains a number of buoys in the eastern Atlantic, our data comes from buoy 62081. See http://www.ncep.noaa.gov/

[2]Surf observations are published daily by the Lahinch Surf Shop. See http://www.lahinchsurfshop.com/

chitecture 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 ensemble of MDNs to make stable, accurate predictions of conditional density functions.

There are two important points that restrict predicting the true generating distribution. Firstly, to define the true conditional density function accurately infinite parameters may be required. Secondly, each prediction is a *function*, not a value; however, this function is estimated from observations alone. The aim of a probabilistic model is to simulate the real conditional density function as accurately as possible.

3.1. Ensembles of Mixture Density Networks

Mixture Density Networks (MDN) refer to a special type of ANN 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 (Schittenkopf et al., 2000) 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, convenient, semi-parametric means of modeling unknown distributional shapes. The conditional density function is described in the form

$$p(t|\mathbf{x}) = \sum_{i=1}^{C} \alpha_i(\mathbf{x})\phi_i(t|\mathbf{x}) \qquad (1)$$

[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.

where ϕ_i is a Gaussian as follows,

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

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).

Due to the complexity of the local error curvature, straightforward gradient descent fails to optimise MDNs. To overcome this problem the optimisation technique requires dynamic adjustment of the learning rate during the training process to efficiently converge to a global minimum. We use Scaled Conjugate Gradients (SCG)(Moller, 1993) for optimisation of our Mixture Density Networks. SCG uses the second derivative Hessian matrix to find the most direct route to the minimum.

With any neural network optimisation there is an inherent instability. This is due to the random initialisation of the network's weight vectors. One approach to reducing the instability in a neural network is to train several neural networks and return an aggregate prediction. This is called the ensemble technique. There are two primary requirements for any ensemble approach, *diversity* in the individual networks, and the ability to *aggregate* your results. Two approaches to achieving these requirements are Bagging (Breiman, 1996) and Boosting (Freund & Schapire, 1996). We have implemented both and outline our implementations below.

3.2. Bagging MDNs

Bagging is an abbreviation for bootstrap aggregation. It uses the bootstrap, a statistical re-sampling technique, to generate multiple, diverse, training sets for networks of an ensemble. A bagged neural network ensemble generates training data for each ensemble member by sampling from the bootstrap empirical distribution. A training set is created for each neural network in the ensemble and the networks are trained on this data.

Sporadically MDNs do not converge to a good solution due to a very poor random starting position. To reduce the effect of these weak solutions on the ensemble prediction we integrate the member solutions using a form of balancing (Heskes, 1997). The training sets in a bagged ensemble are generated by sampling with replacement from the original training set T. The prob-

ability that the individual training vector, taken from T, will not be part of a bootstrap re-sampled training set, $T^{bootstrap}$, is $(1-1/N)^N \approx 0.368$ where N is the number of training vectors in T. These omitted training vectors make a new out-of-bootstrap (oob) test set, T^{obb} for the ensemble member trained with $T^{bootstrap}$. For each ensemble member we calculate the error, E^{oob}, over the vectors in its T^{oob}. We use these error values to weight each member's contribution to the ensemble prediction. Assuming there are M ensemble members,

$$p(t|\mathbf{x}) = \sum_{i=1}^{M} \sum_{j=1}^{C} \frac{E_i^{oob} \alpha_{i,j}}{\sum_{k=1}^{M} E_k^{oob}} (\mathbf{x}) \phi_{i,j}(t|\mathbf{x}) \qquad (3)$$

where E is the Continuous Ranked Probability Score error function and is described below. The new aggregated GMM produced by the ensemble still obeys the properties of a probability distribution.

3.3. Boosting MDNs

We have implemented *AdaBoost.R2* (Drucker, 1997) for MDNs. The means of obtaining diverse training sets is distinctly different in boosting to bagging. In boosting, training of ensemble members is carried out sequentially. The vectors that have been 'difficult' for ensemble members to estimate i.e. obtain large errors, are more likely to appear in the training sets of subsequent ensemble members. This restricts ensemble members from being trained in parallel, however, it has the advantage of reducing the bias of the resulting ensemble.

To combine the ensemble members a weighted median approach is used, the better performing members are assigned a heavier weighting. Integration is achieved in a similar way to the balancing method above; however, instead of using out-of-bootstrap vectors alone the whole training set is used to calculate the weightings.

4. Evaluating Probability Forecasts

A significant problem with probabilistic forecasting models is determining a means of evaluating the results. If we assign some probability to the actual observation, can the prediction be wrong? To determine whether a probabilistic model performs well you must first decide on your goals. The objective when making a density forecasting model is to, 1) create probability density functions that accurately predict the region in which the target lies and, 2) provide well calibrated probability estimates (Gneiting & Raftery, 2004). The first criterion refers to the spread of the predictive density about the target. The second criterion, calibration, is a measure of the statistical consistency between the distributional forecasts and the observations. For example, a well calibrated model over 100 predictions for 100 events giving a particular outcome a 10% probability would result in that outcome occurring 10 times.

In this paper we use three error scores. Firstly, we use the average negative log predictive density (NLPD) (Good, 1952). This is the average value of the negative of the logarithm of the predictive density $p(...)$ at each observation t.

$$NLPD = \frac{1}{N} \sum_{i=1}^{N} -log(p(y_i|\mathbf{x}_i)) \qquad (4)$$

The NLPD is a means of evaluating the amount of probability that the network assigns the target. It penalises predictions that are either under or over confident. This metric is generally used in conjunction with a standard distance metric to determine how far the target is from the median or mean of the distribution. We use the Mean Absolute Error (MAE) to calculate this error. The MAE is the average absolute error of predictions. It is a useful indication of the inaccuracy of the model because it is in the resulting error value is in the same units as the target values. We will use the median of the predictive densities as point predictions for our MDNs.

The third error score we use is the most comprehensive, the Continuous Ranked Probability Score or the CRPS. The CRPS (Gneiting & Raftery, 2004), is a generalisation of the Brier Score. The advantage of this error score is that it takes the whole distribution into consideration when measuring the error. The CRPS score uses the cumulative probability density function to determine the error.

$$crps(p,t) = -\int_{-\infty}^{\infty} (p(y|\mathbf{x}) - H(y,t))^2 dy \qquad (5)$$

where $H(x,t)$ is the Heaviside function i.e.

$$H(x,t) = \mathbf{1}\{x \geq t\} \qquad (6)$$

This function has a value of 1 if x is greater than t, otherwise it has a value of 0.

This is a *strictly proper scoring rule*, i.e. the forecaster will maximise their result for the forecast F if they use the forecast F, rather than any $G \neq F$. This is our preferred metric for scoring error because of its sensitivity to distance. We use the average *crps* to evaluate the results of a complete input set.

$$CRPS = \frac{1}{N}\sum_{i=1}^{N} crps(p_i, y_i) \quad (7)$$

5. Analysis of Ensembles of MDNs on Artificial Data

To demonstrate the MDNs ability to produce estimates of the higher moment conditional density effects we created a simple synthetic data set. Inputs for the data are uniformly drawn from the interval [0,2]. The target values are generated according to

$$t(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{x}) + \epsilon(x). \quad (8)$$

where $\epsilon(x)$ represents random sample noise drawn from the following distribution,

$$SN(x) = \frac{2}{0.2}\phi(\frac{y+0.1}{0.2})\Phi(\lambda\frac{y+0.1}{0.2}) \quad (9)$$

SN is the skew normal function (Azzalini, 1985) that produces skewed distributions. $\phi(x)$ is the probability density function and $\Phi(x)$ is the cumulative density function. λ is the shape or skew parameter, a positive value results in a positive skew and vice versa. When λ is 0 the distribution is symmetric.

We generated 400 data pairs for training and 400 for testing. The test set is illustrated in Figure 1. For our experiment we trained five different types of networks; two ensembled point predictors, one trained with boosting (boostNN), the other bagging (bagNN); two ensembled MDNs, again one boosted (boostMDN) and one bagged (bagMDN), and a single MDN. The single MDN and the MDN ensemble members have a 2 Gaussian GMM output. We carried out tests for three different values of λ (3 - slight skew, 6 - significant skew, 9 - extreme skew). The variance is constant so that we can measure the effects of the skew.

Table 1 shows the average result over 10 runs for the three different experiments. The ensembles of MDNs provide the best predictions over all datasets. The single MDNs instability problem is also observed as the MDN performs poorly on the slightly skewed data but comparatively better on the more skewed data.

Table 1. Experiment results for synthetic data

Slight Skew	**CRPS**	**NLPD**	**MAE**
boostMDN	0.070	-0.660	0.099
bagMDN	**0.070**	**-0.668**	**0.099**
boostNN	0.071	-0.633	0.100
bagNN	0.071	-0.633	0.101
MDN	0.077	-0.603	0.107
Significant	**CRPS**	**NLPD**	**MAE**
boostMDN	0.071	-0.678	0.102
bagMDN	**0.071**	**-0.704**	**0.102**
boostNN	0.072	-0.619	0.105
bagNN	0.073	-0.619	0.104
MDN	0.075	-0.668	0.107
Extreme	**CRPS**	**NLPD**	**MAE**
boostMDN	0.067	-0.787	0.094
bagMDN	**0.067**	**-0.790**	**0.094**
boostNN	0.068	-0.677	0.098
bagNN	0.069	-0.664	0.098
MDN	0.070	-0.737	0.095

6. The Surf Prediction Engine

The surf prediction data is an example of a data set which has an underlying noise dynamic that shows evidence of higher moment influence. To demonstrate this we carried out the following experiment. We used 10-fold cross-validation with a standard bagged ensemble of neural networks and calculated the skew of the out-of-sample errors over all folds. The skew of this distribution is 2.25 for the surf data[4]. This is a significant skew compared to a normal distribution.

Standard noise caused by observation errors would naturally assume a Gaussian shape. If we assume that ensembles of standard neural networks are a good means of obtaining well generalised predictions and we observe that the residuals of the results from the ensemble of neural networks have significant higher moment influences, we can conclude that these higher moment influences are present as an underlying dynamic in the data. This is the case in the analysis on artificial data presented in Figure 1. The skew of the residuals is almost identical to the skew of the added noise. Now that we have determined the surf data is affected by skewness we can continue to develop a prediction engine that takes this information into consideration.

The boosted MDN is the base model for our surf prediction engine (see Figure 2). It is trained on the database described in Section 2 and produces conditional density forecasts for future events. Once a conditional density forecast has been made, this information can be used by the surfer. The most useful means of extracting the required information from the model is for the surfer to specify the wave heights that they can surf i.e. their 'surfable' range and the times in which they would like to surf. The engine will then determine from the predicted conditional density function the probability of those conditions. In this sense the

[4] A standard normal distribution has a skew of 0

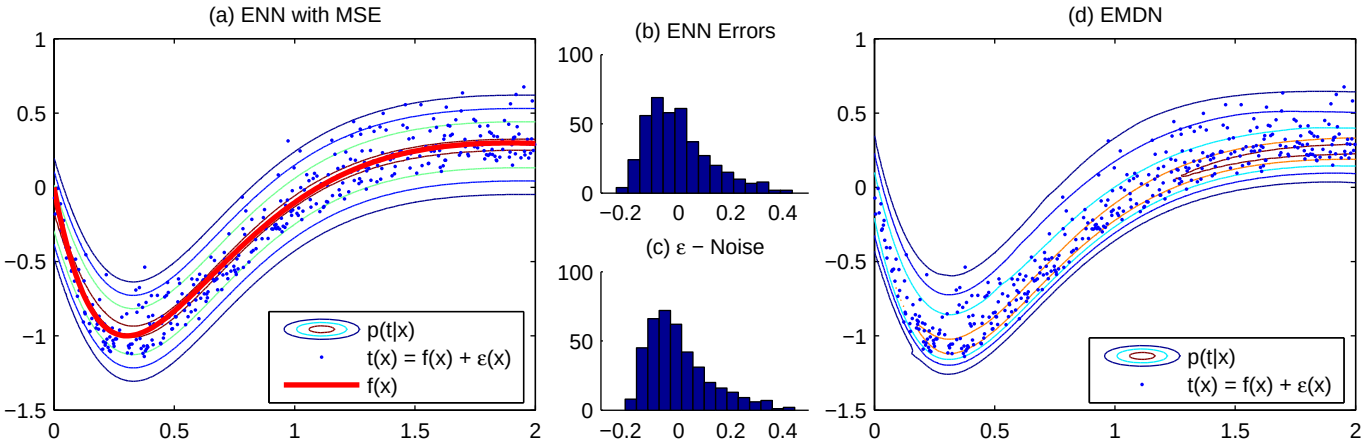

Figure 1. (a) shows a contour map of the bagged ensemble of neural networks (bagNN). f(x) is the data-generating function without added noise. The bounds of the distribution are symmetric and over estimate the uncertainty on the lower values. (b) is a histogram of the errors from the bagNN point predictions, it strongly resembles the histogram of the added noise (c). (d) shows the result from the Ensemble of MDNs. It can clearly be seen the skewed noise has been correctly described.

model is giving a prediction that is tailored to the user. By providing this auxiliary information the decision maker can make a more informed choice.

7. Performance Evaluation

Figure 3 (a) shows 10 probability distributions derived from the predicted conditional density functions of the boosted ensemble of MDNs. This type of prediction gives the user a number of useful pieces of information. Firstly, between zero and four feet there is little uncertainty for these predictions. However, the model is much more uncertain about predictions of days when the actual surf height is higher. The tenth prediction is for the largest target and the probability distribution shows a large amount of uncertainty (broad distribution). Figure 3 (b) plots the position of the median and mode values of the predicted distributions. The mode is the highest point on the distribution and represents the most typical outcome. The median is the point that divides the distribution in half. The plot shows that the model's predicted median and mode are a good point estimate of the observed values.

3 (b) confirms that the ensemble of MDNs is a useful predictor in the classic point estimate sense. However, we need to evaluate the accuracy in terms of the 'surfable range' predictions as presented to the user in Figure 2. Intuitively, an acceptable model from a users perspective is one that is correct on average 83% of the time on predictions that have an 83% confidence. For our model this is the case, as shown in Table 2. In

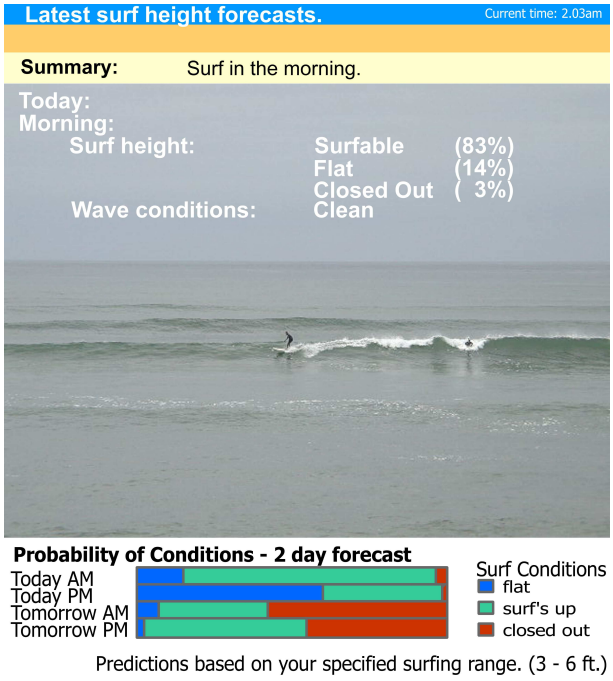

Figure 2. This is a demonstration of the surf prediction engine forecasting waves in the range 3 - 6ft. with 83% probability. Detailed predictions for the morning are given at the top with an image of the most similar day shown. A two day summary chart is given at the bottom that shows the probability of the different classes. Predictions are provided over a seven day horizon.

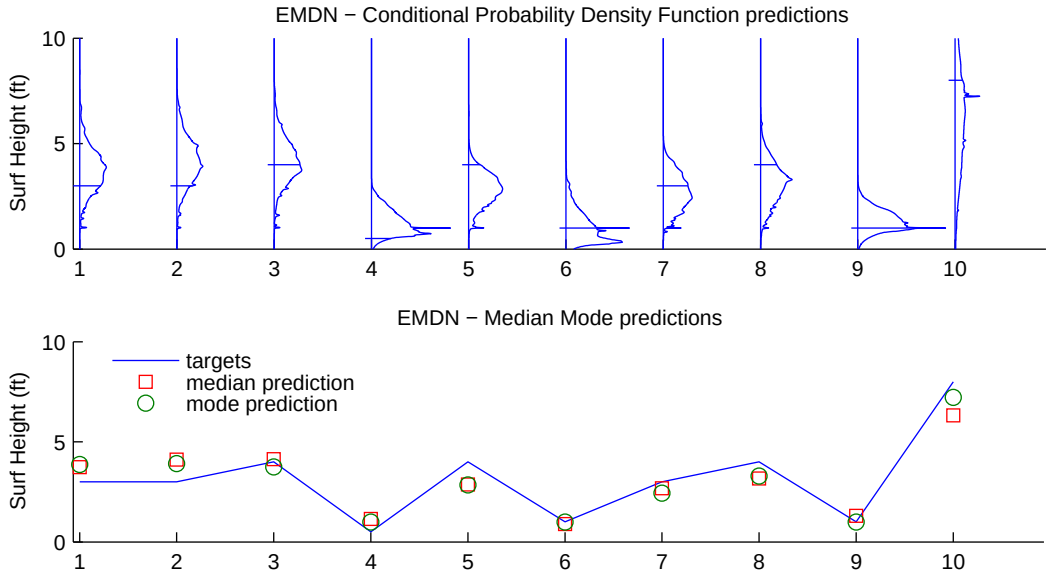

Figure 3. (a) 10 out-of-sample predictions from a boostMDN on the surf data. The horizontal lines through each distribution represent the target observations. (b) shows the median and mode predictions relative to the target observations of the predicted density functions.

Table 2. boostMDN results for prediction of surfable range 3 - 6 ft. The accuracy column represents the percentage of correct predictions given the threshold. The Classifications column represents the number of instances where the probability for a class exceeded the threshold.

Threshold	Accuracy	Classifications
Max	81.3%	260
70%	87.8%	193
80%	93.2%	158
90%	96.0%	130
95%	98.1%	110
99%	100.0%	72

Table 3. Results of experiments with surf data

	CRPS	NLPD	MAE
boostMDN	**0.493**	1.186	0.701
	0.161	0.299	0.226
bagMDN	0.504	**1.170**	0.718
	0.184	0.340	0.262
boostNN	0.519	1.534	0.704
	0.188	0.601	0.245
bagNN	0.521	1.483	**0.700**
	0.127	0.451	0.147
MDN	0.518	1.477	0.736
	0.169	0.540	0.244

this scenario the surfer has specified a 3ft - 6ft surfable range and predictions have been produced for 260 test days[5]. We can see from the table that the error is well correlated with the prediction confidence. The results show that the boosted ensemble of MDNs has a slightly over-cautious bias, but in general is well calibrated and consistent. The number of correct predictions with respect to incorrect predictions obeys the thresholds assigned i.e. when the threshold is set to 95% then the model is correct $\geq 95\%$ of the time.

Finally, we evaluate the ensemble of MDNs predictions on the surf data against our benchmark models, the ensembles of NNs and a single MDN. These results are shown in Table 3 with the best score on each metric highlighted in bold. These figures are based on a 10-fold cross validation and the variances over the 10-folds are also shown underneath each error score. The boostMDN and bagMDN perform best on the probabilistic error measures; however, they also perform comparably to the ensembles of NNs on the MAE score.

[5]We use 10-fold cross validation to obtain test results over the complete data set

8. Conclusions

There are a range of applications for Machine Learning where a simple point prediction is not adequate to guide decision making, for instance in meteorology and finance. In this paper we present such an application. Surf prediction involves forecasting whether conditions on a surfing beach will be suitable for an individual given their acceptable range of wave heights. We use an ensemble of MDNs to predict the probability distribution for the maximum wave height. The application uses this distribution to estimate the probability that the surf height will be within the range desired by the user.

We have demonstrated that MDNs can be successfully combined in an ensemble to produce stable results. MDNs optimised using SCG generally produce stable results; however, sporadically the network will not converge to a useful minimum and the MDN will not produce a good result. We overcome this by combining several MDNs into an ensemble. Of the two ensemble techniques we evaluate (bagging and boosting) it appears that boosting produces slightly better results. In our future work we intend to experiment with other boosting techniques to further improve conditional density prediction.

References

Azzalini, A. (1985). A class of distributions which includes the normal ones. *J. Statist.* (pp. 171–178).

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

Breiman, L. (1996). Bagging predictors. *Machine Learning* (pp. 123–140).

Butt, T., Russell, P., & Grigg, R. (2004). *Surf science: An introduction to waves for surfing*. Alison Hodge.

Clements, M. P., & Smith, J. (2000). Evaluating the forecast densities of linear and non-linear models: Applications to output growth and unemployment. *Journal of Forecasting*. Chichester.

Colas, A. (2004). *The world stormrider guide*. Low Pressure Publishing.

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.

Drucker, H. (1997). Improving regressors using boosting techniques. *ICML* (pp. 107–115).

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.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *ICML* (pp. 148–156).

Gneiting, T., & Raftery, A. (2004). *Strictly proper scoring rules, prediction, and estimation* (Technical Report 463). University of Washington.

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

Hasselmann, S., Hasselmann, K., Bauer, E., Janssen, P., Komen, G., Bertotti, L., Lionello, P., Guillaume, A., Cardone, V., Greenwood, J., Reistad, M., Zambresky, L., & Ewing, J. (1988). The wam model-a third generation ocean wave prediction model. *Journal of Physical Oceanography* (pp. 1775–1810).

Heskes, T. (1997). Balancing between bagging and bumping. *Advances in Neural Information Processing Systems* (pp. 466–472).

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

Komen, G., Cavaleri, L., Donelan, M., Hasselmann, K., Hasselmann, S., & Janssen, P. (1994). *Dynamics and modelling of ocean waves*. Cambridge University Press.

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. *ICANN'94*. Springer.

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

Tolman, H. (1999). User manual and system documentation of wavewatch-iii version 1.18. *NOAA, Technical Note 166* (p. 110).

Weigend, A. S., & Nix, D. (1994). Predictions with confidence intervals (local error bars). *ICONIP'94*.