Representing classifier confidence in the safety critical domain — an illustration from mortality prediction in trauma cases

Trevor C. Bailey, Richard M. Everson, Jonathan E. Fieldsend, Wojtek J. Krzanowski and Derek Partridge
University of Exeter, Exeter, EX4 4QF, UK.

Abstract

This work proposes a novel approach to assessing confidence measures for software classification systems in demanding applications such as those in the safety critical domain. Our focus is the Bayesian framework for developing a model-averaged probabilistic classifier implemented using Markov chain Monte Carlo (MCMC) and where appropriate its reversible jump variant (RJ-MCMC). Within this context we suggest a new technique, building on the reject region idea, to identify areas in feature space that are associated with "unsure" classification predictions. We term such areas "uncertainty envelopes" and they are defined in terms of the full characteristics of the posterior predictive density in different regions of the feature space. We argue this is more informative than use of a traditional reject region which considers only point estimates of predictive probabilities. Results from the method we propose are illustrated on synthetic data and also usefully applied to real life safety critical systems involving medical trauma data.

1 Introduction

A dominant philosophy in Computer Science has been correctness — algorithms and thus implementations of computational tasks should be correct, and, in its extreme manifestation, we should be able to prove formally that the computational system operates correctly. One aspect of this preoccupation involves the degree to which incorrect computational results may be believed to be correct, and this requires attention to be directed to developing meaningful estimates of the confidence which may be attached to individual results arising from a computational system.

This paper addresses this problem in the demanding context of critical system applications where an incorrect result, believed to be correct, can have far-reaching consequences. In critical systems, perhaps more than any other class of classification problem, the need to be confident (or conversely, to know when not to be confident) about the output of a system is of paramount importance. An incorrect output may lead to the death of a patient (in a medical diagnosis system) or the collision of aircraft (in a collision alert system). The life-threatening nature of the potential failure of some critical classification systems is a cause for great concern, and has been the impetus of recent work in the area, as well as strict government regulation.

A purely deterministic classifier, however well optimised to the classification task, is unable to provide measures of the confidence to be associated with its predictions beyond the crude assumption that all predictions will be subject to the same error rate as that experienced in the test set used in development of the classifier. For that reason, classifier models that provide a probabilistic output are more obviously attractive in a safety critical context, since they explicitly acknowledge that some examples presented to them can be classified with more certainty than others. This naturally leads to the use of a *reject region* as one way of assuring a required level of system performance. In this approach any example whose predicted probability of being in any particular state is not sufficiently high (beyond a pre-determined threshold) is marked as unclassified, or *UNSURE*, thereby alerting to the possible repercussions of misclassification. The accuracy rate of the classification system on the test set over a range of different rejection rates (i.e. different threshold values), may be compared by plotting these values in the so-called accuracy-rejection (A-R) plane and this plot provides an encapsulation of overall classifier operating characteristics with regard to confidence of predictions.

However, important additional considerations arise in the reject-region approach when one considers its use in conjunction with modern classification methodologies designed to reduce model mis-specification — an important objective in all safety critical classification systems. Such techniques explicitly acknowledge that the existence of a single 'true' classification model in the presence of noise is questionable, let alone its occurrence within the finite model search that is feasible to effect optimisation of a single classifier. Consequently, these methods draw on the decision-theoretic optimality of averaging over models [3]. One of the most popular approaches to facilitating such model averaging is to adopt the Bayesian paradigm for classification implemented via Markov Chain Monte Carlo (MCMC) and, where appropriate, its reversible jump variant (RJ-MCMC). Such an approach is potentially the best way of averaging models from within a specified family — not only does it ensure that each of the classifiers incorporated in the averaging constitutes a "plausible" choice (guided by the acceptance probabilities based on the data likelihood rather than through random choice), but also all classifiers chosen are then correctly probabilistically averaged in forming the overall classification prediction.

Work in the above area has demonstrated the improvement in classification accuracy that model averaging can lead to over that achieved from a single maximum a posteriori (MAP) model. However, the primary interest in this paper is more particularly in how the benefits of model averaging can also be exploited in refining the reject region idea so as to provide enhanced measures of classifier confidence to accompany the improvement in classification accuracy associated with model averaging methods.

In this paper we propose and evaluate the method of "uncertainty envelopes" (UE) for assessing confidence in classification systems in demanding contexts such as the safety critical domain. Such envelopes are extracted from the distribution of individual classification results on each example in the test set as delivered by a model averaging MCMC process. Like the reject region, the boundary of the UE is determined by a pre-set threshold and divides the output space (and hence, by inversion, the feature space) into *SURE* (i.e. high confidence) and *UNSURE* regions. However, unlike the conventional reject region, the UE allows for the fact that the classification probabilities generated by a probabilistic classifier are themselves sample estimates whose precision will vary over examples in the test set as reflected by the shape of their full posterior predictive distributions. In short, unlike the conventional reject region, the *SURE* region identified by our proposed UE technique includes only test examples whose "sureness" is assured.

This paper will proceed as follows. In Section 2 MCMC RJ-MCMC are introduced in the context of classifier averaging. The probabilistic k-nn model from [3] is also described as constituting a relatively straightforward application of this methodology which may be used to demonstrate the techniques proposed in later sections. In Section 3 the new UE method is introduced, and its use is illustrated and evaluated on a synthetic data set in the subsequent section. The penultimate section presents results from applying this method to real world applications in the critical systems domain using data sets concerned with medical trauma data and for some of which benchmark results are available. The paper concludes with a brief summary and discussion section.

2 Bayesian model averaging and k-nearest neighbours classification

As discussed in the introduction a number of benefits can be gained from averaging over a number of classifiers instead of choosing a single 'best' model [3]. Given the desirability of model-averaging to cope with uncertainty in model specification and parameter values, the question then arises as to how best to generate the models over which to average. Uniform sampling from the model space is expensive if the number of possible models is large and unless the posterior distribution of the models is uniform then a large number (vast majority) of models used will add little or nothing to the prediction (as the weight of a model should be determined by its posterior probability).

A more efficient approach is to use a Bayesian paradigm in conjunction with MCMC methods (see [8] for an extensive discussion of this problem). Such an approach ensures that models and associated parameter values are sampled in proportion to their posterior distribution. The probability process of MCMC for model averaging is outlined in Algorithm 1, where we include the reversible jump extension of MCMC (RJ-MCMC) introduced by Green [5], which permits jumping between models with parameter spaces of varying dimension.

1. Assign priors to the model parameters and generate initial model to start the chain.
2. Calculate likelihood of the model and then the model posterior (proportional to the likelihood multiplied by the prior for the parameters.
3. Adjust the parameters of the model to create a new model. Calculate the new model's posterior probability and add the new model onto the chain with a probability determined by the ratio of its posterior to that of the previous chain member. Otherwise use previous chain member as new chain member.
4. The period until the chain stabilises is known as the *burn-in* period and forms the prelude to sampling. Once this stability is achieved, collect model samples from the chain every nth member) to generate the final model-averaged predictive distribution.

Algorithm 1: RJ-MCMC

An arbitrary model with a corresponding parameter set is defined as an initial Markov chain state. By adjusting parameters of this model using Monte Carlo techniques the predictive posterior distribution can be realised. Where, in the classification context, the predictive posterior distribution for the class variable y_i of a datum $\mathbf{x}_i$ given training exemplars $\mathcal{D}$ and parameters θ is defined as:

$$p(y_i|\mathbf{x}_i, \mathcal{D}) = \int_\theta p(y_i|\mathbf{x}_i, \theta)p(\theta|\mathcal{D})\; d\theta$$

i.e. the predictive posterior distribution averages over the uncertainty in the parameter values as reflected by their posterior distribution. We also accept a change of state (add a new model to the chain) with a probability equivalent to the ratio of the posterior of the new model and the previous chain member. As model acceptance is proportional to posterior probability, no additional weighting of models is necessary when their predictions are collected to form the model-averaged predictive distribution. The burn-in period (step 4 of Algorithm 1) is determined by measuring the statistics of a number of parallel chains, and is concluded when these chain statistics converge [9].

The probabilistic k-nn model of Denison *et al.* [3] is a simple but powerful classification model which lends itself well to implementation within the MCMC model averaging framework described above. The model has two parameters: the k parameter of the traditional k-nn model (representing the number of nearest neighbours used to classify to one of the classes), and a second parameter β which controls the "strength of association" between neighbours. The conditional probability of the class variable y_i given the datum $\mathbf{x}_i$ and training exemplars $\mathcal{D}$ takes the form:

$$p\left(y_i|\mathbf{x}_i, \beta, k, \mathcal{D}\right) = \frac{\exp\left((\beta/k)\sum_{j \overset{k}{\sim} i} \delta_{y_i y_j}\right)}{\sum_{q=1}^{Q} \exp\left((\beta/k)\sum_{j \overset{k}{\sim} i} \delta_{q y_j}\right)}$$

where δ_{ab} is the Kronecker delta (takes the value one if $a = b$, otherwise is zero) and $\sum_{j \overset{k}{\sim} i}$ denotes the summation over the k nearest neighbours of the datum $\mathbf{x}_i$ in the training data.

In traditional k-nn the probability of being in a particular class is equivalent to the proportion of k nearest neighbours of that class. In the probabilistic k-nn this is no longer the case, the majority class of the k nearest neighbours still determines the assigned class, however the β term influences the exact probability assigned. The larger the value of β, the greater the separation of classes.

In standard k-nn (see for example [4]) the k nearest neighbours in the feature space to $\mathbf{x}$ are calculated amongst the training set of N points, where $1 \leq k \leq N$-1. The distance between points is typically measured as the Euclidean distance, although other metrics may also be employed [3]. In this study the tri-cube method is used, which is shown below:

$$p\left(y_i|\mathbf{x}_i, \beta, k, \mathcal{D}\right) = \frac{\exp\left((\beta/k)\sum_{j \overset{k}{\sim} i} u\left(\|x_i - x_j\|\right)\delta_{y_i y_j}\right)}{\sum_{q=1}^{Q} \exp\left((\beta/k)\sum_{j \overset{k}{\sim} i} u\left(\|x_i - x_j\|\right)\delta_{q y_j}\right)}$$

In the tri-cube method the extra weight function $u(\cdot)$ is a monotonically decreasing function of distance, which has the effect that the further a test point is from the training data, the lower the probability of the assigned class. This latter feature is particularly important for safety critical systems, which should be equivocal about classification of a new datum dissimilar or "far away" from those in the training data.

In the context of classifiers such as k-nn, which use the distance between data points to generate the classification, an important consideration is how different features should be weighted in the calculation of distance. Differential feature scaling may clearly improve performance if it is known that certain features are of more importance in relation to the classification task. However, in the absence of any strong *a priori* knowledge of an appropriate feature scaling, normalisation is usually encouraged and we adopt that approach in the applications we report in subsequent sections of this paper.

A more extensive definition and derivation of the probabilistic k-nn model can be found in [3].

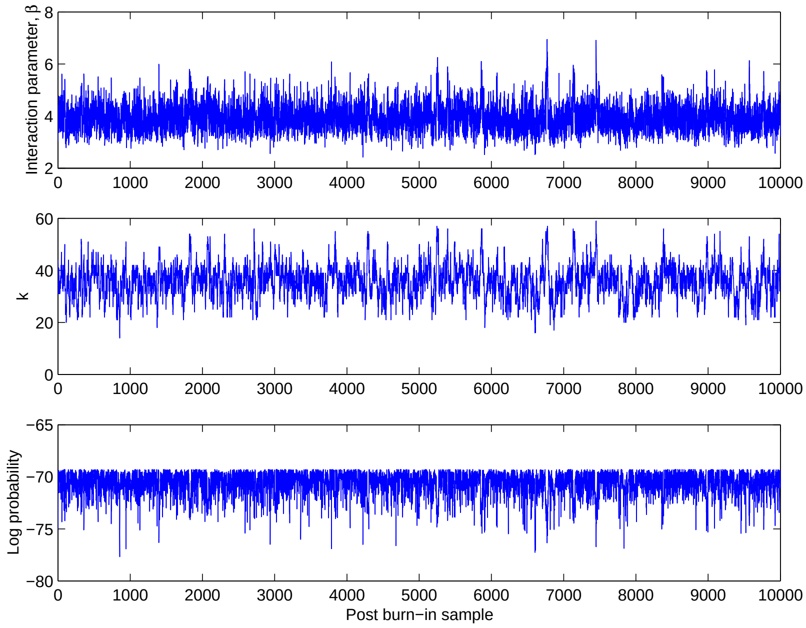

Figure 1. Example RJ-MCMC statistics for the probabilistic k-nn classifier

Examples of typical output obtained from this model are shown in Fig. 1, where every seventh Markov chain sample has been recorded after an initial burn-in of 10000, until 10000 samples have been generated. The top plot in Fig. 1 shows the value of β at each model sample, the middle plot the value of k and the lower plot the corresponding log posterior likelihood. The probabilistic k-nn forecast for any particular datum is subsequently calculated as the average of predictions for these 10000 individual models (providing a good approximation of the integration of the posterior predictive density).

3 Uncertainty envelopes

As mentioned in the introduction, one technique for handling the need for classification confidence when using a single probabilistic classifier is to use a reject region [4]. This region comprises a user defined space around the classification boundary within which the allocation of any datum is designated as *UNSURE* i.e. whose assigned class is viewed as highly uncertain.

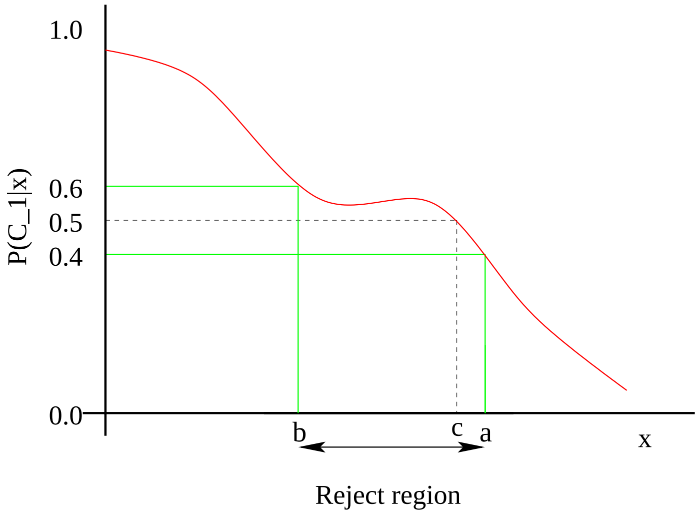

Figure 2. Illustration of reject region for a 1-D feature space and a two class classification problem.

An illustration of this idea is provided in Figure 2 where a 1-D feature space is shown for a two class problem with the corresponding posterior predictive probability $P(C_1 \,|\, x)$ indicated. In general, the Bayes decision boundary for a two class problem is defined by the surface over which $P(C_1 \,|\, \mathbf{x}) = P(C_2 \,|\, \mathbf{x})$, and in the case of the 1-D feature space considered here that corresponds to a feature value of $x = c$. The *UNSURE* region is here illustrated as lying between the classification probabilities [0.4,0.6], corresponding to x values lying in the range $[b, a]$. If $x < b$ a datum is classified as belonging to Class 1, if $x > a$ it is assigned to Class 2, otherwise it is *UNSURE*. In practice, the size of the reject region would usually be chosen with regard to the containment of misclassified points experienced on the training data.

MCMC generated model-averaged classifiers, such as those discussed in the previous section of this paper, not only potentially provide improvements to classifier accuracy through the reduction of model mis-specification, but we argue here that they also provide the opportunity to explore alternatives to the conventional reject region method described above when seeking to handle the need for assured classification confidence. For example, for such model-averaged classifiers the $P(C_1 \,|\, x)$ curve in Figure 2 would now be the average of the posterior predictive probabilities of a large number of classifiers (for instance probabilistic k-nn variants) selected through the Bayesian MCMC process. We actually have available full posterior predictive densities at each of the x values (or at least at those values represented in the test set). We should therefore be able to exploit more than just the mean of the posterior predictive density in determining an *UNSURE* region for classifications.

An important point to appreciate in this regard is that in general the shape and variance as well as the mean of posterior predictive distributions may vary dramatically in different parts of input feature space. To emphasise this point, empirical examples of the range of posterior predictive densities that may be encountered in practice are shown in Figure 3a relating to a synthetic set of 1000 test data points involving two features and two classes. Three points have been circled, one in Class 1, one close to the boundary between the two classes and one in Class 2. Histograms of the Class 1 posterior predictive distributions on these three points are indicated in Figure 3b as generated from post burn-in RJ-MCMC chain samples using a model-averaged k-nn classifier trained on 250 separate points. As can clearly be seen, none of the three histograms are symmetric, and their variance and shape differ widely.

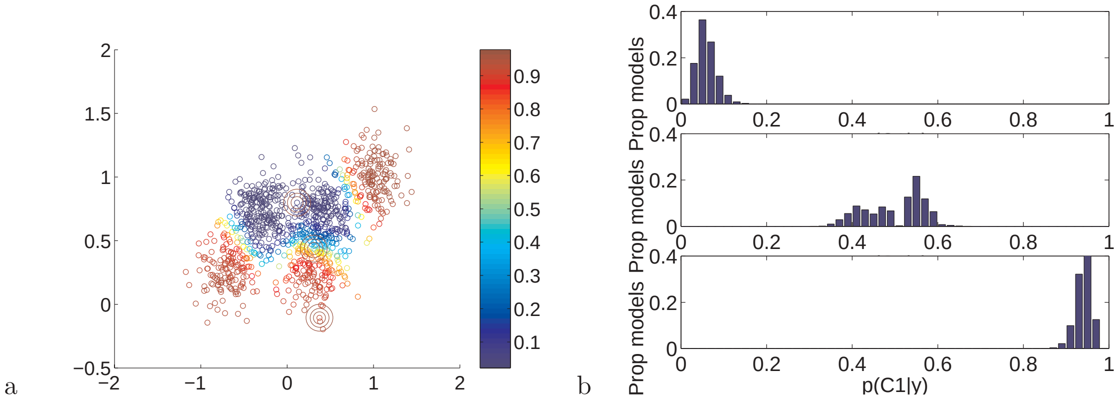

Figure 3. Selected histograms of posterior predictive distributions. (a) shows the synthetic test data. Three points are circled, one in Class 1, one close to the boundary between the two classes and one in Class 2. (b) shows histograms of the Class 1 predictions on these three test points.

Given the illustration above, one would for example be wary of making any assumptions about the shape of posterior predictive distributions about the mean $P(C_1 \,|\, x)$ curve in Figure 2. That would militate against sole use of any simple summary measures of such distributions in refining the definition of a reject region for *UNSURE* classifications (e.g. standard deviations such as used in the different context of forecast assessment from Bayesian trained neural network non-linear regression by MacKay [7] and Bishop [1]).

In light of the above, we now introduce the idea of an “uncertainty envelope” (UE) as a more informative method for identifying *UNSURE* classifications in the model-averaging framework. The UE we propose is based on the proportion of sampled models in the MCMC chain that classify a test point in a class other than the overall class predicted by the model averaging. In addition to mean posterior predictive classification, we have the additional information of the full posterior predictive distribution for any input datum in the test set. What these values represent are a set of plausible (i.e. parameter optimised) classifications for that point, from models that might well have been selected were we using a single classifier. Using this knowledge we can determine to what extent a point was classified by one or more of the sampled models as being in a class other than the actual class attributed to it by averaging (i.e. we quantify the degree to which sampled models disagree with the overall class assigned to a test point by effectively integrating the tail area of the posterior predictive distribution lying outside of its mean class). For any pre-determined threshold value, the result can be realised by wrapping an envelope in feature space around those points where the proportion of different classifiers sampled in the RJ-MCMC chain which assign to an alternative class to the mean prediction exceeds the threshold.

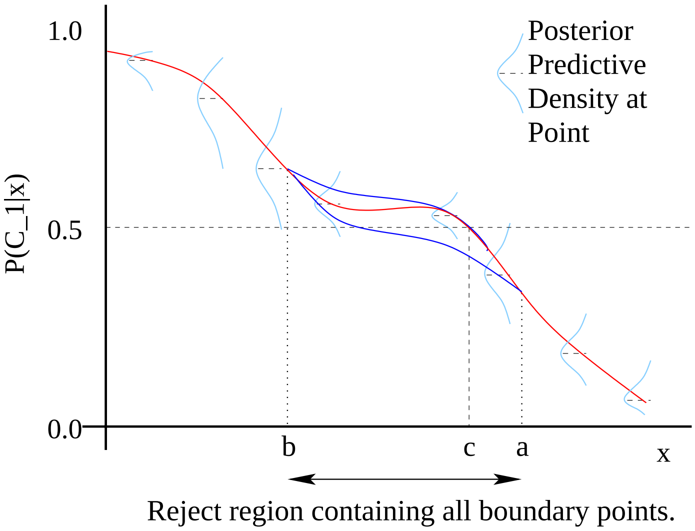

Figure 4. Illustration of uncertainty envelope for a 1-D feature space and a two class classification problem.

To illustrate the idea for a 1-D feature space, Figure 4 indicates the average posterior predictive curve, $P(C_1 \,|\, x)$, generated from MCMC model-averaging and also outlines full posterior predictive distributions about this curve at various points. The Bayes decision boundary based on the mean posterior predictive distribution is indicated at $x = c$. The feature range $[b, a]$ represents the UE, or *UNSURE* region, where all test points were not unanimously classified by all sampled models in the MCMC chain. If we were averaging over 5000 MCMC samples points outside this envelope would represent a situation where all 5000 models assigned points with that datum or feature value to an identical class.

The zero threshold used to define the UE in the above illustration is extreme. In practice the method can be used to define an UE for any pre-determined threshold value (for instance the 1 in 1000 level (.001), 1 in 100 (.01) level etc.). Classification results corresponding to varying thresholds can be used in much the same way as the rejection region is traditionally used in conjunction with an accuracy-rejection (A-R) plot. For present purposes the *UNSURE* rates at different threshold values (i.e. the proportions of test points in the UE) correspond to the rejection rates, while “accuracy” is reflected by the validity of the classifications of test points in the *SURE* category at different threshold values (i.e. those not in the UE). We may use either the *SURE INCORRECT* rate (aiming to minimise) or the *SURE CORRECT* rate (aiming to maximise). We have generally chosen the former, so considering the curve produced when *SURE INCORRECT* is plotted against *UNSURE* for a range of threshold values. If the *SURE INCORRECT* values are general lower than the *UNSURE* values at all thresholds for a classifier and the *UNSURE* rates are acceptable, then we would tend to favour that as producing reasonably confident predictions. We consider such plots in our numerical investigations below.

4 Illustration of Uncertainty envelopes with synthetic data

To aid the visualisation and validation of the UE technique introduced in the previous section, a two dimensional synthetic data set was generated with known (stochastic) properties. This data set comprises a mixture of five bivariate Gaussians, with two of these Gaussians contributing to one class and three Gaussians to the other. Formally:

$$p\left(x|C_j\right) = \sum_{m=1}^{M} P_{jm} p\left(x|\theta_{jm}\right)$$

where $p\left(x|\theta_{jm}\right)$ are Gaussians with common covariance matrices $0.03I$. The mixing weights, P_{jm}, and means, μ_{jm}, were as follows:

Class 1.	$\mu_{11} = (1.0, 1.0)'$	$P_{11} = 0.16$
	$\mu_{12} = (0.7, 0.3)'$	$P_{12} = 0.17$
	$\mu_{13} = (0.3, 0.3)'$	$P_{13} = 0.17$
Class 2.	$\mu_{21} = (-0.3, 0.7)'$	$P_{21} = 0.25$
	$\mu_{22} = (0.4, 0.7)'$	$P_{22} = 0.25$

250 data points generated from this process are shown in Figure 5, and form the training data set used by models subsequently in this section. This synthetic set is almost identical to the 4-Gaussian model used by Ripley in [10], apart from the addition of a distribution centred in the upper right portion of the feature space, which causes the theoretical Bayes decision boundary of the process to “flip-back” on itself - creating an interesting ‘W-shaped’ boundary. Test data for this problem consists of 1000 points, and the theoretical Bayes error rate is 9.3% (due to the heavy overlapping of classes).

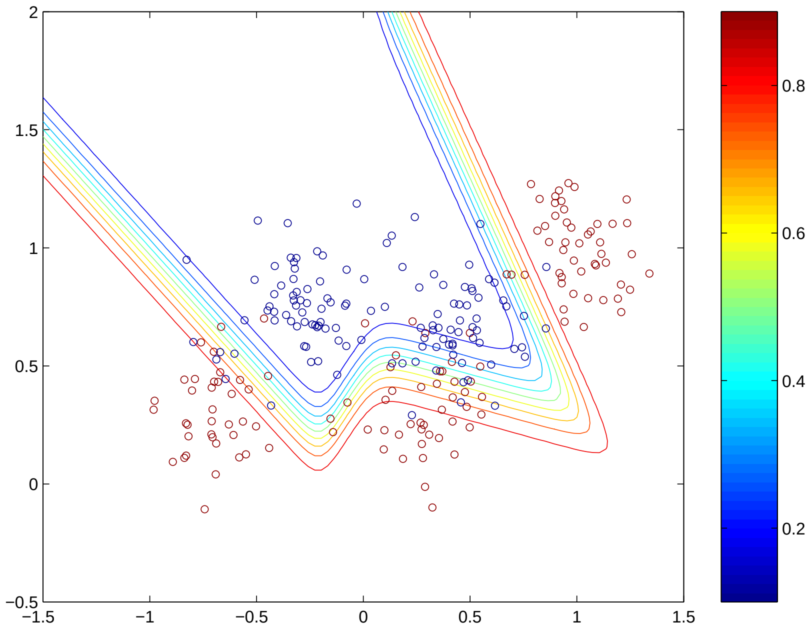

Figure 5. Synthetic data, 250 test points with theoretical Bayes decision boundary.

Model averaging using the probabilistic k-nn classifier as described earlier was applied to this synthetic data set. The MCMC was run using a burn-in period of 10000 chain samples, after which every seventh sample was collected until a total of 5000 chain samples were obtained. Based on mean posterior predictive probabilities the model-averaged probabilistic k-nn classifier performed with a testing error level of 9.8% (the single maximum *a posteriori* (MAP) model performed with a 9.9% error rate).

Figure 6 shows the decision contours generated by the probabilistic k-nn method in the feature space of the synthetic data. The probabilistic k-nn' contours reflect the Bayes decision boundary (dashed line) and the classification probabilities of points away from those in the training data approach 0.5 which was the object behind our preference for use of the tri-cube distance measure as discussed earlier.

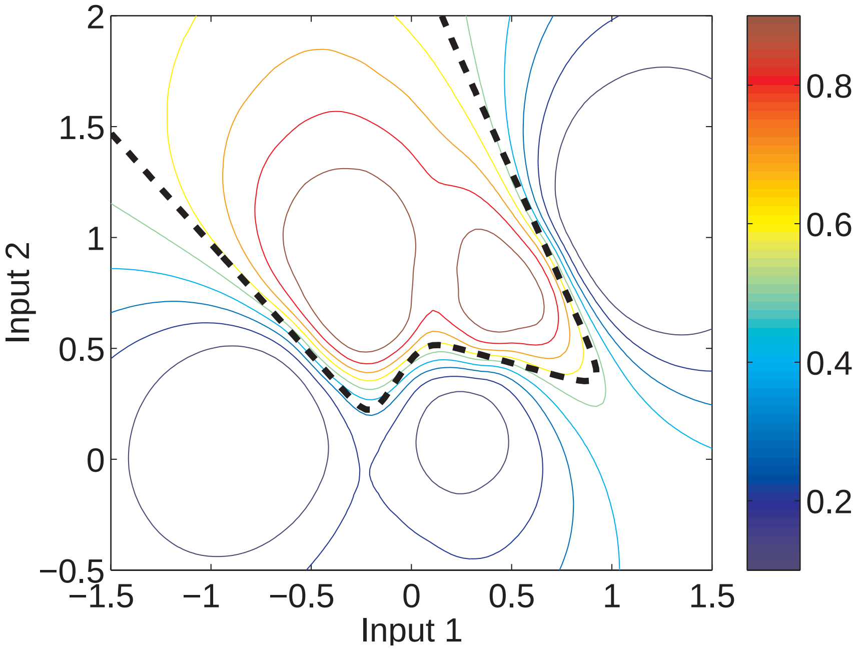

Figure 6. Decision contours for synthetic data using probabilistic k-nn. The dashed black line shows the Bayes rule decision boundary.

Figure 7 shows the UE in the feature space of the synthetic data. The thin black lines indicate the UE defined by a 0 threshold, whereas the coloured area corresponds to a 0.5 threshold. The UE can be seen to generally map itself well to the Bayes decision boundary.

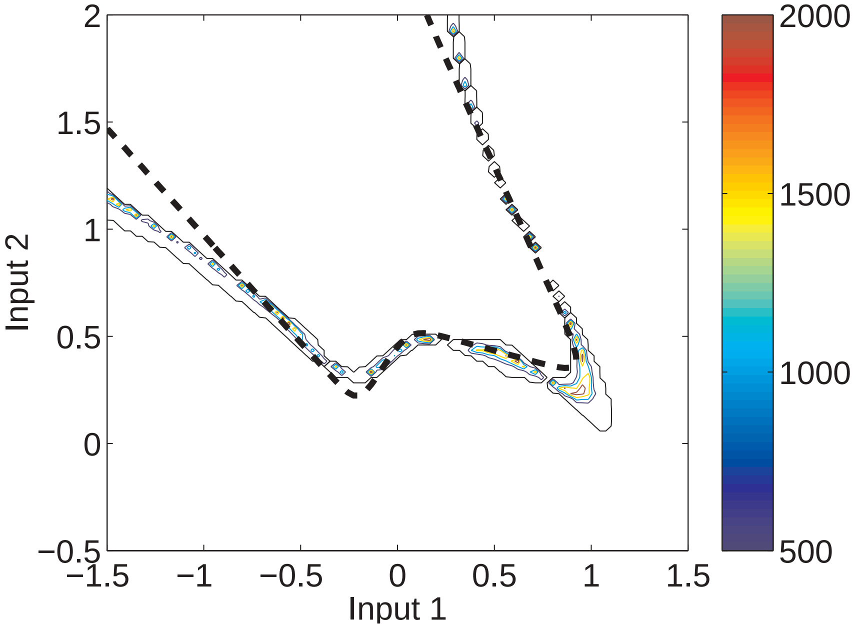

Figure 7. Uncertainty envelopes generated for synthetic data using probabilistic *k*-nn at 0 and 0.5 thresholds. The dashed black line shows the Bayes rule decision boundary.

Figure 8 shows the realised *SURE INCORRECT* versus *UNSURE* plot for the test synthetic data. The x-axis is the proportion of points contained within the UE and the y-axis is the proportion of incorrectly classified points lying outside the envelope. As we move left to right on the figure, the UE or *UNSURE* region is being gradually expanded as thresholds gradually decrease from 0.5 to zero. In this case the relationship is good — a 1% increase in *UNSURE* rate leading to approximately a 1% decrease in *SURE INCORRECT* classifications. The miminal threshold value of zero (i.e. the maximal possible sized UE) corresponds to an *UNSURE* rate of around 10% and a *SURE INCORRECT* rate of approx 7%. In one sense this represents a 'confidence bound' for this classifier. At that threshold level no evidence of classification uncertainty from any of the 5000 MCMC samples has been observed for test points in the *SURE* region. There are all 'surely' *SURE* and therefore there is no basis (other than pure random selection) upon which to choose any particular sub group of them to be flagged as *UNSURE* in an attempt to reduce the *SURE INCORRECT* rate further.

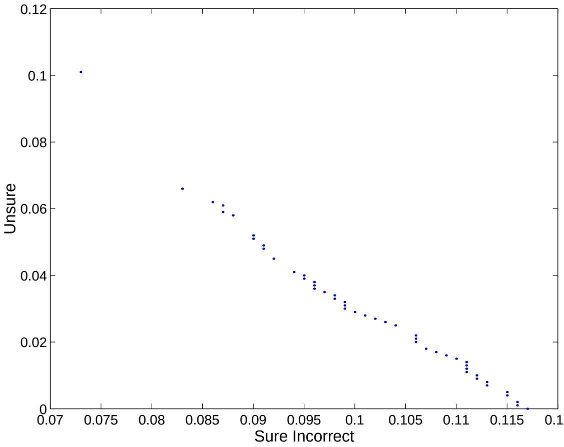

Figure 8. *SURE INCORRECT* versus *UNSURE* rates for the synthetic test data using probabilistic *k*-nn

By increasing the amount of training data used to model the synthetic process, we may observe some further interesting properties of the UE. Figure 9 shows the average and extreme *SURE CORRECT* classification rates obtained over five runs with different training sets, versus the logarithm of the training set size. The *UNSURE INCORRECT* rates are also plotted (one minus the *SURE INCORRECT* rate). The difference between these two rates is the *UNSURE* rate. All values shown are for a 0.01 threshold level. As the volume of data is seen to increase, the proportion of test data points marked as *UNSURE* is seen to decrease, and the *SURE INCORRECT/CORRECT* rates converge to the Bayes rates. Obviously this property depends upon the underlying classifier (through the Bayesian averaging framework) to be of sufficient complexity for it to be capable of modelling the Bayes rate classification boundary.

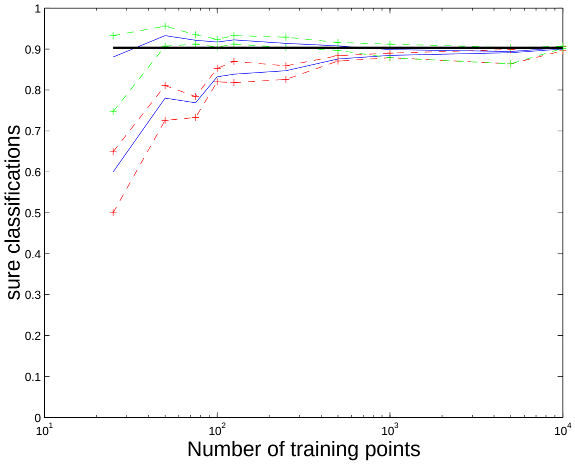

Figure 9. Mean and extreme *SURE CORRECT* and *(1-SURE INCORRECT)* rates versus training set size for 5 different training set sizes. The Bayes error rate is also shown

Figure 10 shows the relation between the (log) *UNSURE* rate and the (log) training set size for the same set of runs. The nature of the plot indicates there may be a power-log relationship between the proportion of points classified as unsure, and the amount of training data available in the UE procedure. This suggests the possibility of being able to estimate the amount of training data required to generate an *UNSURE* rate of zero and therefore access the best classification possible for the classifier family.

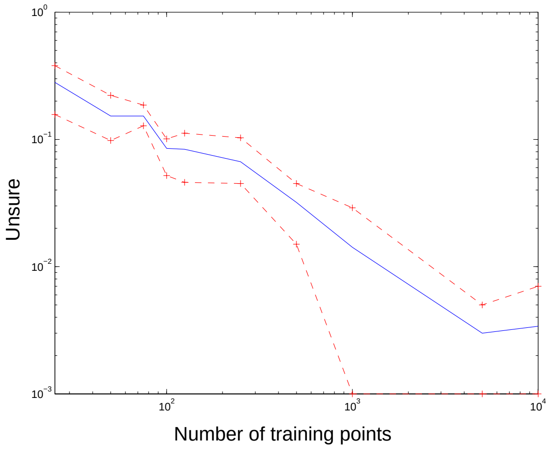

Figure 10. Log *UNSURE* versus log training set size for 5 different training set sizes.

Data Set	#F	train/test	NHS model	pknn	sure correct	sure incorr
HEMS Trauma (balanced)	16	158/158	-	18.98	73.43	16.46
TARNB Trauma, ISS$\geq$ 16	4	1091/1091	13.57	12.83	81.85	8.89

Table 1. Summary results for medical trauma data sets, using established classifier (where available) and model averaged probabilistic knn model.

5 Use of Uncertainty envelopes in Medical classification

The Royal London Hospital operates the only helicopter emergency medical service (HEMS) in London. Upon arrival in the hospital 16 physiological and anatomical variables together with a standardised description of the injury are collected. These data are subsequently collated with outcome classification, i.e. whether the patient survived or died of their injuries. In this study we use a subset of this HEMS trauma data set, balanced so as to include equal numbers of patients in both outcome categories (158 who died, 158 who survived). A model averaged probabilistic k-nn classifier was developed to predict whether a patient will live or die using the 16 input features. Of the 316 cases, 158 were used for training the classifier and 158 for testing.

The second set of medical information we use also relates to trauma cases in the form of the Trauma Audit and Research Network data set B (TARNB). This contains variables measured on NHS accident and emergency patients including classification as to whether the patient survived or died of their injuries. Again the classification of primary interest is whether a patient will live or die. The current classifier used by the NHS uses only four of the collected variables (Age, injury type, ISS (injury severity score) and RTS (revised trauma score), which we also use here. A very low proportion of individuals died in this data set (only 5% of the 18391 complete entries), so we concern ourselves here with only the severely injured patients – those with an ISS value exceeding 16. Over a quarter of the 2182 cases in this category died. 1091 of the cases were used for training a model averaged probabilistic k-nn classifier using the four input features, 1091 cases were used for testing.

Summary results for both the above data sets are shown in Table 1. For the HEMS trauma data the model averaged k-nn classifier performed with an error rate of 19% on the test data. There is no existing established classifier for comparison, but it is interesting to note that the single MAP model performed with an error rate of 22%.

Given the dimensionality of the feature space in this case, it is not possible to easily visualise the corresponding UE, but Table 1 includes *SURE CORRECT* and *SURE INCORRECT* rates for the probabilistic k-nn classifier based on a .01 threshold for the UE definition. More generally, Figure 11 shows the shows the realised *SURE INCORRECT* versus *UNSURE* plot for this classifier generated by considering UEs defined using posterior predictive probability thresholds ranging from 0.5 to zero. The picture is not encouraging — increasing the size of the UE generally fails to deliver much gain in terms of lowering the *SURE INCORRECT* rate. Even at the zero threshold limit limit the *SURE INCORRECT* rate is still around 16%.

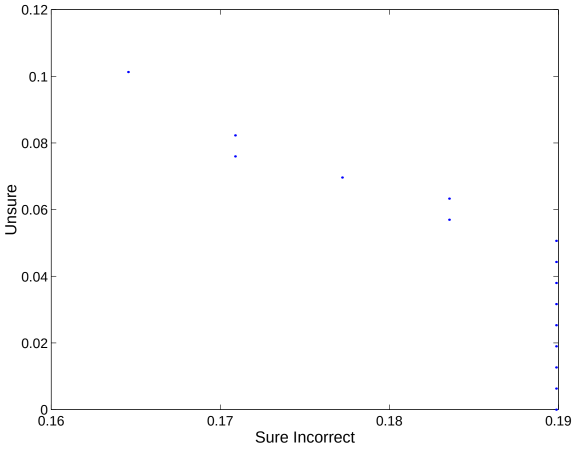

Figure 11. *SURE INCORRECT* versus *UNSURE* rates for balanced HEMS trauma data using model averaged probabilistic *k*-nn classifier

The summary results for the TARNB data set in Table 1 show the probabilistic k-nn classifier performed with an error rate of 13% on the test data which is broadly comparable to the the 14% error rate achieved by the benchmark NHS classifier. No analysis of confidence is of course routinely available for the existing NHS classifier, but Table 1 includes *SURE CORRECT* and *SURE INCORRECT* rates for the probabilistic k-nn classifier based on a .01 threshold for the UE definition. It is interesting to note that by segmenting feature space into *SURE* and *UNSURE* regions by using the UE techniques suggested in this paper, the effective misclassification rate of the classifier can be reduced to just 9% if one is prepared to accept around 10% of classifications being *UNSURE*. Operating on such a .01 threshold may seem stringent and Figure 12 shows the realised *SURE INCORRECT* versus *UNSURE* plot for this classifier generated by considering UEs defined using posterior predictive probability thresholds ranging from 0.5 to zero. The relationship is good for this classfying showing a steady trade off between the *UNSURE* rate and the *SURE INCORRECT* rate until the zero threshold limit is reached.

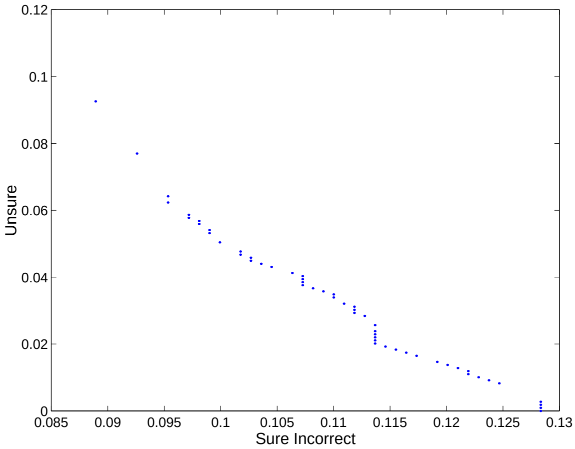

Figure 12. *SURE INCORRECT* versus *UNSURE* rates for TARNB, ISS$\geq$ 16 trauma data using model averaged probabilistic *k*-nn classifier

Conclusion

A new approach in terms of “uncertainty envelopes” has been presented to generating confidences (*SURE* or *UNSURE*) in relation to predictions from model average probabilistic classifiers generated by MCMC. In situations where some classification error is always going to occur this approach identifies predictions in which little confidence can be placed and we have argued that it does so in a way which is more informative than use of a traditional reject region. Lack of confidence is measured in terms of the full characteristics of the posterior predictive density in that region of the feature space at which the prediction is being made. By adjustment of a pre-determined threshold the confidence assessments can be tuned to different classifier operating requirements using a technique which broadly equates with an accuracy-rejection plot.

The "uncertainty envelope" approach introduced here has been validated on synthetic data and usefully applied to real life data from the safety-critical domain involving mortality prediction from medical trauma data. The method was illustrated throughout using a model-averaged probabilistic k-nn classifier, but it would equally be applicable to a range of classifier types such as a simple generalised linear model (GLM), or a more complex classifier, such as a radial basis function (RBF) network. Both of the latter can be implemented within the RJ-MCMC framework using the auxiliary variable method [3] and with geometric priors over the number of features (GLM) or kernels (RBF) employed.

In this initial study we have presented a method for the marking of classified points in feature space as 'uncertain' through analysing the predictive posterior distributions of plausible classifiers. This in turn has led to the marking of meaningful reject regions in feature space, based upon credible intervals. It has also however highlighted the problem of using a single family of classifiers, even within a RJ-MCMC model-averaging framework. In some instances evaluated here the UE for the k-nn model even at the 0 threshold level contains a lower proportion of mis-classified points than would be ultimately be desirable. In our future work we hope to investigate simultaneous averaging over different plausible families of models as well as within these families, as a potential method to tackle this problem.

Acknowledgment

The authors would like to thank .... for their helpful comments. The authors would also like to thank Timothy Coats for the TARNB data. This work is supported by EPSRC grant GR/R24357/01. The probabilistic k-nn code, provided online by C. Holmes (`http://www.stats.ma.ic.ac.uk/~ccholmes/Book_code/book_code.html`) is used as the basis of the standard probabilistic k-nn model.

References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1998.

[2] M. Bouissou, F. Martin, and A. Ourghanlian. Assesment of saftey critical systme including software: a Bayesian belief network for evidence sources. In *Proceedings of the Reliability and Maintainability Symposium*, Washington DC, January 1999.

[3] D.G.T. Denison, C.C. Holmes, B.K. Mallick, and A.F.M. Smith. *Bayesian Methods for Nonlinear Classification and Regression.* Wiley, 2002.

[4] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Morgan Kaufmann, 1990.

[5] P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[6] B. Little and L. Strignini. Validation of ultra-high dependability for software-based systems. *Communications of the ACM*, 36(11):69–80, 1993.

[7] D.J.C. MacKay. Bayesian Interpolation. *Neural Computation*, 4(3):415–447, 1992.

[8] D.J.C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, NATO Science Series, pages 175–204. Kluwer Academic Press, 1998.

[9] I.T. Nabney. *NETLAB: Algorithms for Pattern Recognition.* Springer, 2002.

[10] B.D. Ripley. Neural Networkis and Related Methods for Classification. *Journal of the Royal Statistical Society B*, 56(3):409–456, 1994.