Bayesian Inductively Learned Modules for Safety Critical Systems

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

{J.E.Fieldsend, T.C.Bailey, R.M.Everson, W.J.Krzanowski, D.Partridge,
V.Schetinin}@exeter.ac.uk

Abstract

This work examines the use of Bayesian inductively learned classification methods in relation to safety critical systems. Central to our approach to critical software systems is the necessity to generate meaningful confidence measures not normally associated with predicted states of a system. This is achieved in this study by casting the problem in a Bayesian formulation, and is implemented using reversible jump Markov chain Monte Carlo (RJ-MCMC). We compare conventional and novel classification architectures, including generalised linear models, probabilistic k-nn and radial basis functions. Results from these methods are illustrated on real life critical systems, including medical trauma data. We develop a new technique, building on the reject region idea, to generate classification volume envelopes, in order to mark regions of classification uncertainty. We also report results on the trade-off between model complexity and the width of the posterior predictive probability. Finally, we discuss important emergent issues.

1 Introduction

In critical systems applications, perhaps more than any other class of classification problem, the need 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 work in the area [2, 6], as well as strict government regulation. One approach to dealing with the need for assurance of system performance, when using a single classifier, is the use of a *reject region*.

In this approach any example whose classification of being in a particular state is not sufficiently high (beyond a pre-determined threshold) is marked as unclassified/uncertain, thereby preventing the possible repercussions of misclassification. Judging how large this region should be to avoid excessive amounts of unclassified points is, of course, an area of interest.

In addition to this, however, is the general concern, when using a single classifier model, of model mis-specification. When selecting a single classifier for classification it is typically the one with the highest posterior probability that is chosen – as it is often assumed that it represents the true response/predictor relationship. However it soon becomes clear, even to the casual user of classification models, that the existence of a 'true' model in the presence of noise is questionable, let alone its occurrence within the finite model search that takes place in the selection of a classifier. Consequently recent research has drawn on the decision-theoretic optimality of averaging over models [3]. Done in a principled fashion, this averaging can be performed in proportion to the individual model posterior probabilities, giving the expectation of the posterior predictive distribution. One of the most popular approaches to facilitating this approach are Markov chain Monte Carlo (MCMC) methods. Work in this area has demonstrated the improvement in classification accuracy that this averaging can lead to, above that of a single maximum a posteriori (MAP) model.

In this study we shall demonstrate the benefit of averaging over models in the generation of more plausible reject regions – and their application in the critical systems domain.

This paper will proceed as follows. In Section 2 the synthetic and real world data sets used through-out this study are described. In Section 3 the MCMC process, and the reversible jump variant (RJ-MCMC) are introduced in the context of classifier averaging. Section 4 describes the probabilistic k-nn model from [3], an easy application of the RJ-MCMC process to classifier averaging which will be used to demonstrate the new methods derived in later sections of the study. In addition, an extension to the probabilistic k-nn involving the use of a dynamic *scaling matrix* is also described, which compensates for variation of discriminatory importance within the features in the k-nn classifier framework. In Section 5 the method of reject regions is described, and in Section 6 the new method of uncertainty envelopes is presented. Section 7 presents some preliminary results from this method using different data sets and classifiers of varying complexity, and Section 8 briefly discusses some conclusions.

2 Data

The two data sets that we use in this paper will now be described.
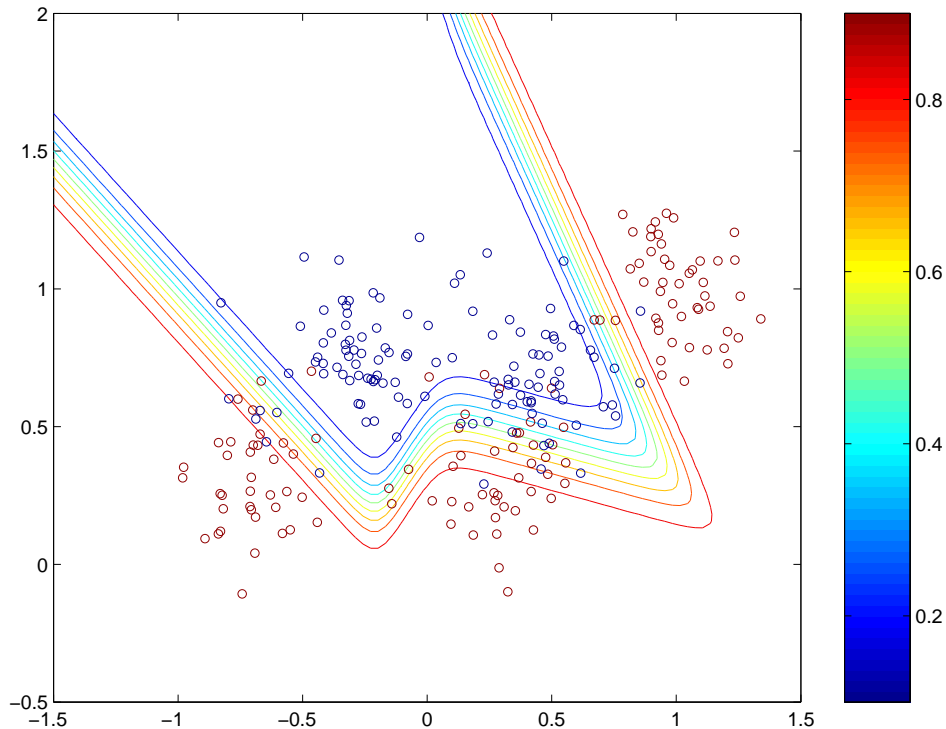
2.1 The synthetic data

Figure 1: Synthetic data, 250 test points with Bayes Rule decision boundary.

To aid the visualisation and validation of the new techniques introduced in later sections, 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

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

where the kernel densities $p\left(x|\theta_{jm}\right)$ are Gaussians, which all have common covariance matrice $0.03I$. The mixing weights and kernel centres are as follows:

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

250 data points generated from this process are shown in figure 1, and form the training data set used by models in later sections. This synthetic set is almost identical to the 4-Gaussian model used by Ripley in [10], apart from the addition of a distribution in the upper right portion of space, which causes the Bayes Rule 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 Bayes error rate is 9.3% (due to the heavy overlapping of classes).

2.2   The trauma data

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. This data is maintained in the "trauma" data set, including classification on whether the patient survived or died of their injuries. The second data set used in this study is composed of a balanced subset of this trauma dataset, including an equal number of died and survived patients (316 in total). 210 data points were used for training and 106 for testing. The critical software system in this case is therefore used for predicting, using the 16 variables, whether a patient will live or die.

3   MCMC processes and the reversible jump

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 coping with some of the uncertainty in model parameters, the question 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 parameters is large - and unless the posterior distribution of the models is uniform then a large number (vast majority) of models generated will add little or nothing to the prediction (as the weight of a model is dependent on its posterior probability). A more efficient approach is to use MCMC methods.[1] Such an approach ensures that models are sampled in proportion to their posterior distribution. An arbitrary model with a set of parameters can be defined as an initial Markov chain state. By then adjusting parameters of this model using Monte Carlo techniques, and accepting a change of state (adding the new model to the chain) with a probability equivalent to the ratio of posteriors of the new model and the previous chain member, a search over the predictive posterior density can be realised. In addition, as model acceptance is proportional to its posterior, no additional weighting of models is necessary before their summation in forming the model average. The probability process of MCMC for model averaging is therefore given in Alg. 1. Here we use the reversible jump extension of MCMC (RJ-MCMC) introduced by Green [5], which permits jumping between models of varying dimension.

The burn-in period of an MCMC process (part 4 of Algorithm 1) is usually either a pre-determined number of chain samples, or is determined by measuring the statistics of a number of parallel chains, and is concluded when these chain statistics converge [9].

---

[1] An extensive discussion of this problem can be found in [8].

Algorithm 1 RJ-MCMC

1. Assign priors to the model parameters and generate initial model.
2. Calculate likelihood of the model, and then calculate the model posterior (proportional to the likelihood multiplied by the parameter prior).
3. Adjust the parameters of the new 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. Continue process until the chain stabilises (the *burn-in* period).
5. Now collect model samples from the chain (every nth member). The final prediction is generated by summing across the collected models

4 The probabilistic k-nn model

The probabilistic k-nn model of Denison *et al.* [3] is a simple but powerful model which has two parameters: the k parameter of the traditional k-nn model, and a second parameter β which controls the 'strength of association' between neighbours. The conditional probability of the class variable y given the datum $\mathbf{x}$ takes the form:

$$p\left(y_i|\mathbf{x}, \beta, k, \mathcal{D}\right) = \frac{\exp\left((\beta/k)\sum_{j\sim i}^{k}\delta_{y_iy_j}\right)}{\sum_{q=1}^{Q}\exp\left((\beta/k)\sum_{j\sim i}^{k}\delta_{qy_j}\right)} \tag{2}$$

where k is the number of nearest neighbours used in the classification of the datum $\mathbf{x}$ to one of the classes. δ_{ab} is the Kronecker delta (takes the value one if $a = b$, otherwise is zero). $\sum_{j\sim i}^{k}$ denotes the summation over the k nearest neighbours of the datum $\mathbf{x}_i$ (from the training data). 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 $1 \le k \le N$-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:

$$p\left(y_i|\mathbf{x}, \beta, k, \mathcal{D}\right) = \frac{\exp\left((\beta/k)\sum_{j\sim i}^{k} u\left(\|x_i - x_j\|\right)\delta_{y_iy_j}\right)}{\sum_{q=1}^{Q}\exp\left((\beta/k)\sum_{j\sim i}^{k} u\left(\|x_i - x_j\|\right)\delta_{qy_j}\right)} \tag{3}$$

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 the tri-cube method the extra weight function $u(d)$ is a monotonically decreasing function of the distance d, which has the effect that the further a test point is from the training data, the lower the probability of the assigned class. A more extensive definition and derivation of the model can be found in [3].

Examples of the typical statistics from this model on the synthetic data set introduced in Section 2 are shown in Fig. 2, where every seventh Markov chain sample is recorded after an initial burn-in of 10000, until 10000 samples have been generated. The probabilistic k-nn forecast for any particular datum is subsequently calculated as the average of these 10000 individual models (providing a good approximation of the integration of the posterior density). The top plot in Fig. 2 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.

In the context of classifiers such as k-nn, which use the distance between data points to generate the classification, it is difficult to know how different features should be weighted in the calculation of distance. Normalisation is usually
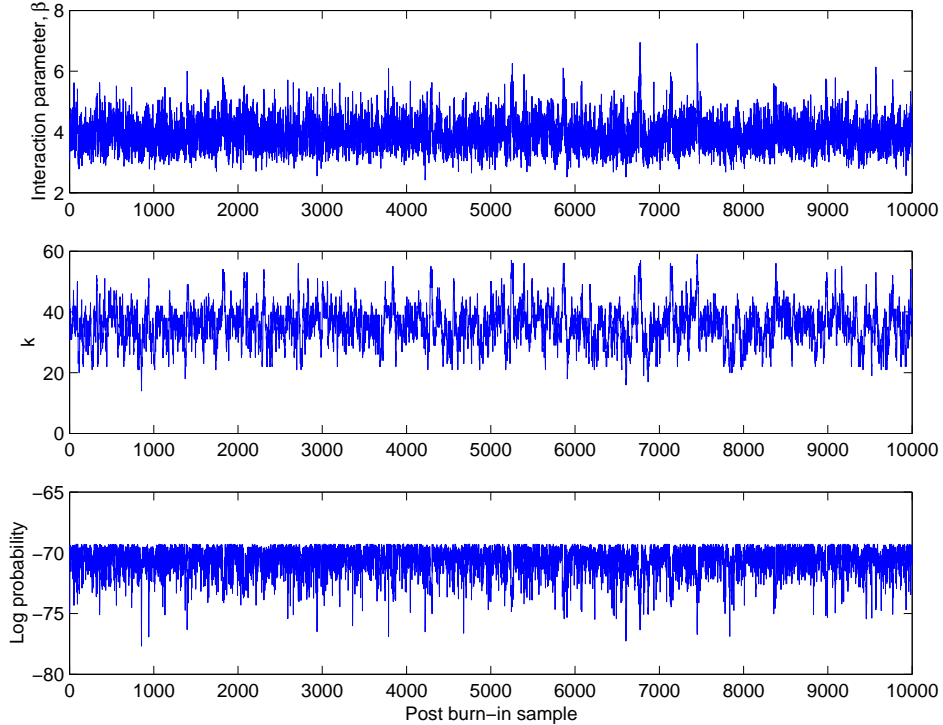
Figure 2: Example k-nn statistics

encouraged, but having features scaled differently may improve performance if the features themselves are of varying importance in relation to the classification task. In this study we introduce an extension to the standard probabilistic k-nn model in order to address this problem within the general RJ-MCMC framework. This model uses a scaling matrix with (initially) an additional p variable parameters (where p is the number of features used). The input features are multiplied with a p by p matrix whose diagonal elements, α_i, sum to p (or alternatively one) and whose off diagonal elements in the basic model are fixed at zero.

$$M = \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix} \tag{4}$$

The squared distance between two points $\mathbf{u}$ and $\mathbf{v}$ is thus given by $(\mathbf{u} - \mathbf{v})'M(\mathbf{u} - \mathbf{v})$.

Fig. 3 shows synthetic data with an elliptical class contained in a circular one. Using standard probabilistic k-nn a misclassification rate of 4.93% is achieved. The extended version can stretch the y-axis and compress the x-axis of the 2-D feature space: and consequently achieves a misclassification of 1.05% (training data contained 250 points and test data 1000 points).

By using a scaling matrix there is no need for *a priori* knowledge of appropriate feature scaling, however it does make the probabilistic k-nn more computationally expensive, as distances need to be re-calculated at each model sample (in the traditional approach, there are finite number of distances, which can be stored).
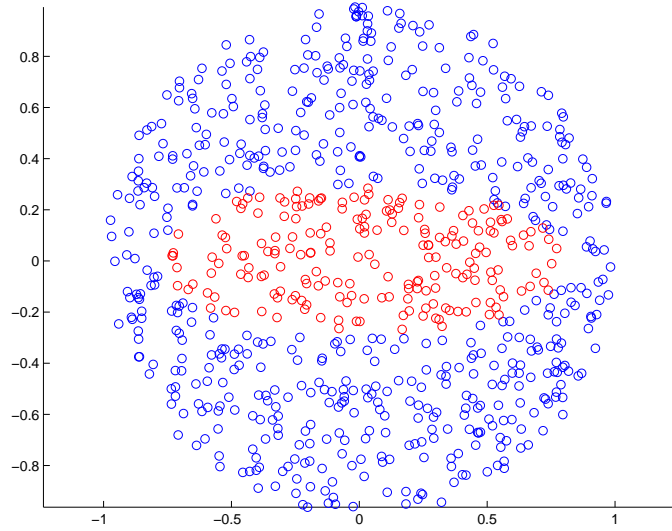
5

Figure 3: Ellipse problem (test data).

5 Representing uncertainty

5.1 Reject regions

As mentioned in the introduction, one technique for representing uncertainty is that of the reject region [4]. When a single classifier is used, this region is typically a user defined space around the classification boundary, where the allocation of any datum is designated as unknown - or whose assigned class is viewed as highly uncertain.
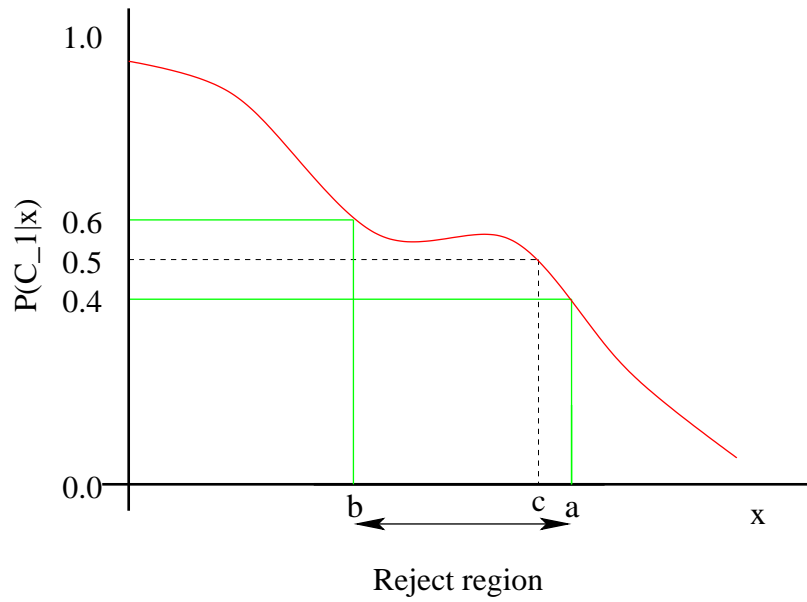

Figure 4: Reject region.

An illustration of this idea is provided in Fig. 4. Here a 1-D feature space is shown for a two class problem, with the posterior probability of a single model drawn in red. The decision boundary is marked, which corresponds to a feature value of $x = c$. If $x < c$ it is classified as belonging to Class 1, otherwise it is assigned to Class 2. The reject region is shown as lying between the classification values [0.4,0.6], corresponding to x values lying in the range $[a, b]$. The extremes of the reject region may by chosen *ad hoc* or to reflect performance on the training data (with regards to the containment of misclassified points). However it is obvious that this is subject to the same problems that beset the fitting of a *single* model to a problem as discussed earlier. Since the development of Bayesian approaches to complex model training, and the integration of many models, other approaches have been developed to encapsulate uncertainty.

5.2   Error bars

In the case of non-linear regression, the MCMC approaches discussed previously have led to the generation of meaningful error bars. For example MacKay [7] and Bishop [1] describe a method for Bayesian training of neural network regressors. Here the forecast is again through the integration of the posterior density, estimated through averaging of MCMC samples. Error bars are then placed over these samples by calculating the standard deviations of the chain member forecasts at each point.[2] Transferring this approach to the classification domain may seem initially to be the most obvious progression, given the identical use of MCMC methods. Fig. 5 illustrates why this is not necessarily the case.
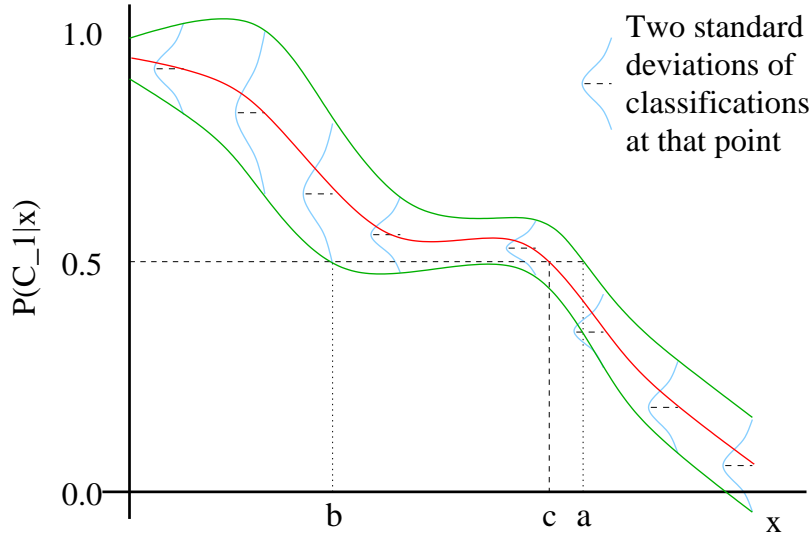

Figure 5: Classification error bars.

Like Fig. 4, Fig. 5 illustrates a 1-D feature problem, however the red decision boundary now shown represents the average of a large number of classifiers selected through a Bayesian MCMC process (for instance probabilistic k-nn). The green lines illustrate the error bars generated by two standard deviations of the posterior predictive densities at each data point. One of the principal drawbacks of this type of approach in the classification domain is immediately apparent, in that these bars may extend beyond the 1.0 and 0.0 classification boundaries (any interpretation of which is nonsensical). This is because the predictive posterior densities may be heavily skewed toward the classifier limits - meaning the standard deviations may still be large. The assumption of symmetry in the generation of error bars may be additionally harmful as it may misrepresent where the uncertain area of feature space actually lies. An empirical example of the types of posterior predictive densities that are encountered in practice are shown in Fig. 6.

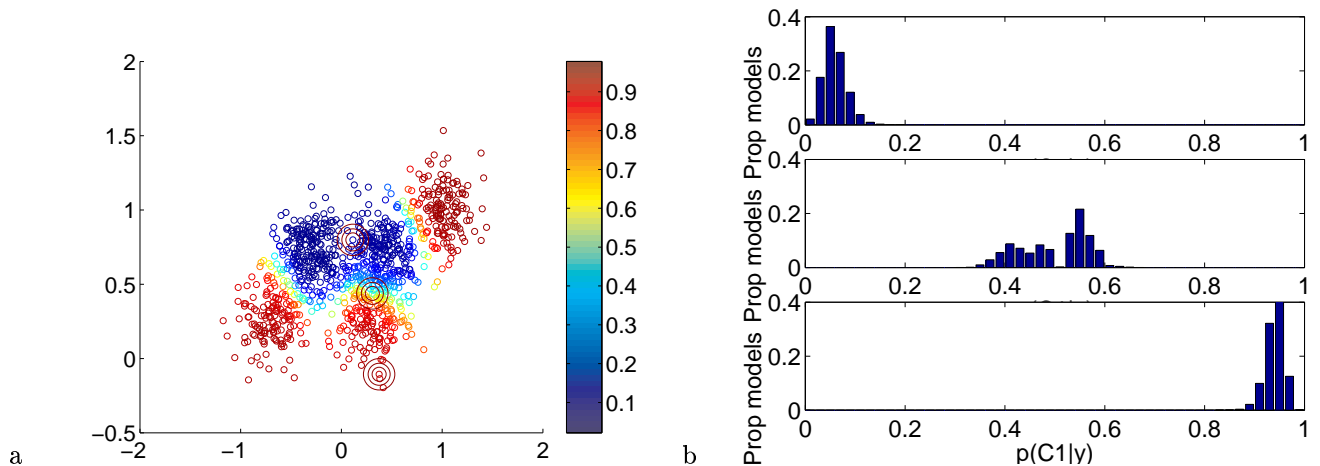[2] The error bars being two standard deviations to each side.

7

Figure 6: Prediction histograms. (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 predictions on the three test points.

Fig. 6a shows the synthetic test data (1000 points, as opposed to 250 in training). 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 predictions on the three points are given in the Fig. 6b in terms of Class one (predictions generated from post burn-in chain members of the probabilistic k-nn Markov chain). As can clearly be seen, none of the histograms over the three points are symmetric. In light of this, the new approach of uncertainty envelopes is now introduced.

6 Uncertainty envelopes

The reject region approach is *ad hoc* in the selection of its region boundaries. It is difficult to say anything meaningful in terms of why a boundary region of [0.4,0.6] is selected, rather than [0.41,0.59] for instance. The error bar approach to uncertainty has a more solid interpretation, however, the underlying assumption of symmetric predictive posterior distributions can often be misleading. Here we introduce the *uncertainty envelope* method as a more reasoned method for generating uncertainty values to be associated with predictions, based on the proportion of likely models which classify a point in a class other than the overall class predicted by the model averaging. After selecting the different models for averaging, and generating the average classification, we still have the additional information of the predictive posterior densities at any input datum. What these represent are a set of *plausible* classifications for that point, from models that might well have been selected were we using a single classifier. Using this knowledge we can now look at uncertainty from a standpoint of not simply forecast deviation, but also what types of forecasts were made at that point. We can for instance see whether a point was classified by one or more of the models as a class other than the actual class attributed to it by averaging (meaning that some of our models disagree with the assigned class). This can be visualised by wrapping an envelope in feature space around those points that have been assigned more than one class by the different classifiers.

Fig. 7 illustrates the classification boundary from averaging the different chain members, and also the predictive densities as various points. The area bounded by the dark blue line contains points that lie on the classification boundaries of chain members, meaning the envelope that contains all points that have been classified by different chain members as different classes lies on the range $[a, b]$. If we were averaging over 5000 plausible models, points outside this envelope would represent a situation where all 5000 models assigned that point to an identical class. This new method can be used to mark new data points whose assigned classification we are uncertain of, at different credibility interval levels (for instance the 1 in 5000 level, 1 in 1000 level, 1 in 100 level etc.).
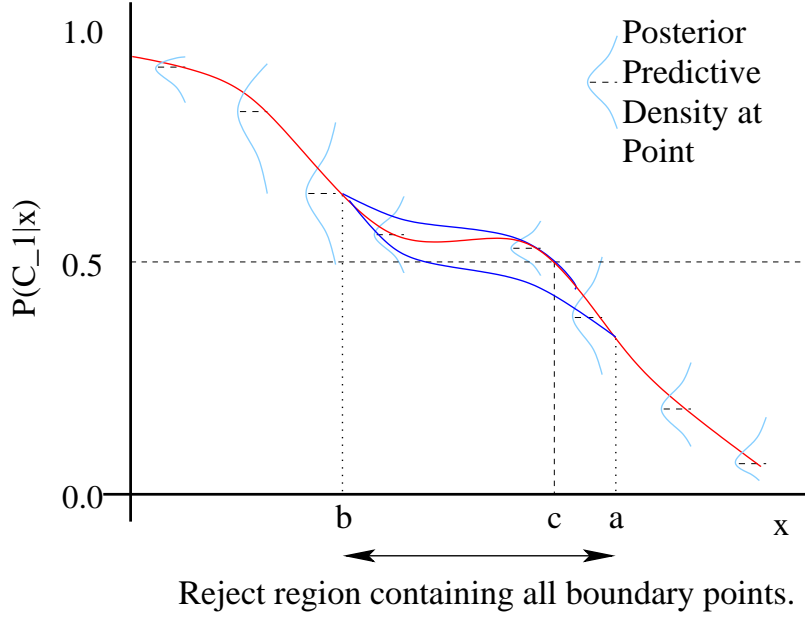
8

Figure 7: Uncertainty envelopes.

Results illustrating this new approach will now be shown both for the probabilistic k-nn on synthetic data and real world critical systems data.

7 Experiments & Results

Both probabilistic k-nn were run using a burn-in period of 10000 chain samples. After burn-in every seventh sample was collected until a total of 5000 chain samples were selected, and averaged.

7.1 Synthetic data results

For the synthetic data, the two models, probabilistic k-nn and scaling matrix probabilistic k-nn, performed with testing error levels of 9.8% and 10.2% respectively (with the single maximum *a posteriori (*MAP) model of each chain performing with and 9.9% and 10.6% error rates each).

Figs. 8a and 8b show the decision contours generated by the two probabilistic k-nn methods in the feature space of the synthetic data. The probabilistic k-nns' contours show a close mapping to the Bayes rule boundary and the classification of points away from the training data approaching 0.5 due to the tri-cube distance measure. This feature is particularly important for safety critical systems, which should be equivocal about classification far from training data.

Figs. 9a and 9b show the credible envelopes generated by the three methods in the feature space of the synthetic data. The standard probabilistic k-nn's envelope can be seen to tightly map itself around the Bayes rule boundary, however it is less accurate in its assessment in areas beyond the training data. The scaling matrix probabilistic k-nn's envelope in comparison spreads further away from the Bayes rule decision boundary where the feature space is devoid of training data and contains the Bayes rule boundary within the feature space it marks as uncertain (at different levels). The thin black lines on the extremes of the envelope indicate the 1/5000 credible boundary, whereas the areas circled by deep red show where the 2500/5000 credible boundaries are.
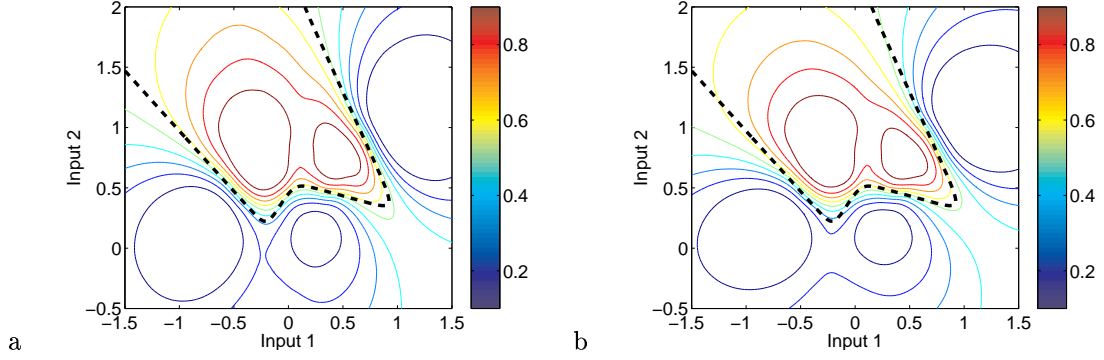
Figure 8: Decision contours, synthetic data; (a) standard probabilistic k-nn and (b) feature scaling probabilistic k-nn. The dashed black line shows the Bayes rule decision boundary.
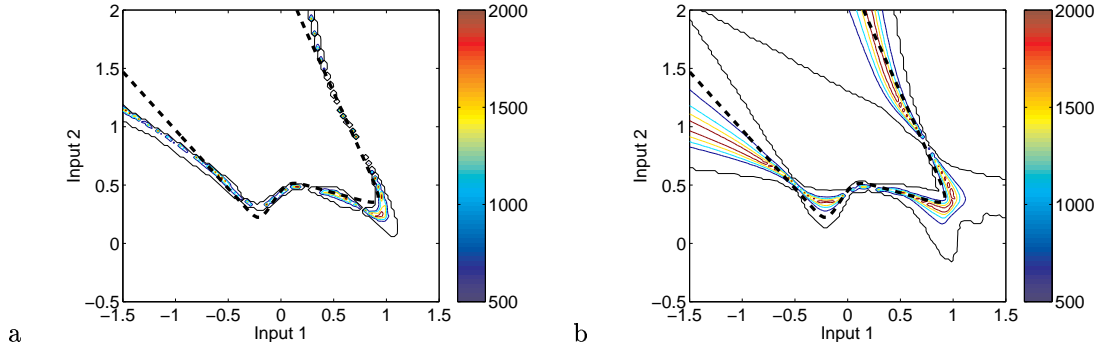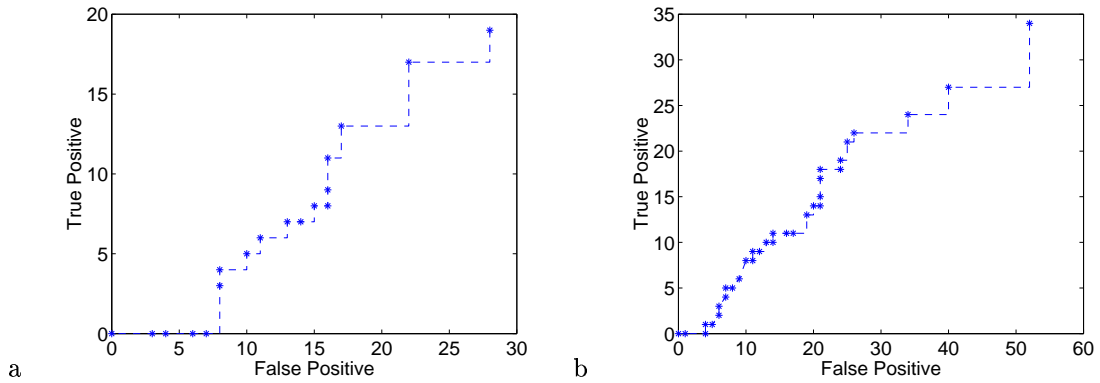

Figure 9: Decision envelopes, synthetic data; (a) standard probabilistic k-nn and (b) feature scaling probabilistic k-nn. The dashed black line shows the Bayes rule decision boundary.

Figure 10: ROC, synthetic data results; (a) probabilistic k-nn and (b) feature scaling probabilistic k-nn.

Figs. 10a and 10b show the realised receiver operating characteristic (ROC) curves on the test data. The x-axis is the number of misclassified points contained within the envelope and the y-axis is the number of correctly classified points contained within the envelope. As we move left to right on the figures, the envelope reject region is being

gradually expanded, from the 2500/5000 credible level to the 1/5000 credible level. The feature scaling probabilistic k-nn model envelope can be seen to contain a higher proportion of its mis-classified points than that of the standard probabilistic k-nn method.

7.2   Trauma data

For the trauma data the models performed with testing error levels of 18.9% and 19.8% respectively (with the single MAP model of each chain performing with and 21.7% and 19.8% error rates each), the ROC curves generated by gradually expanding the envelope region from the 2500/5000 credible level to the 1/5000 credible level are shown in Figs. 11a and 11b.
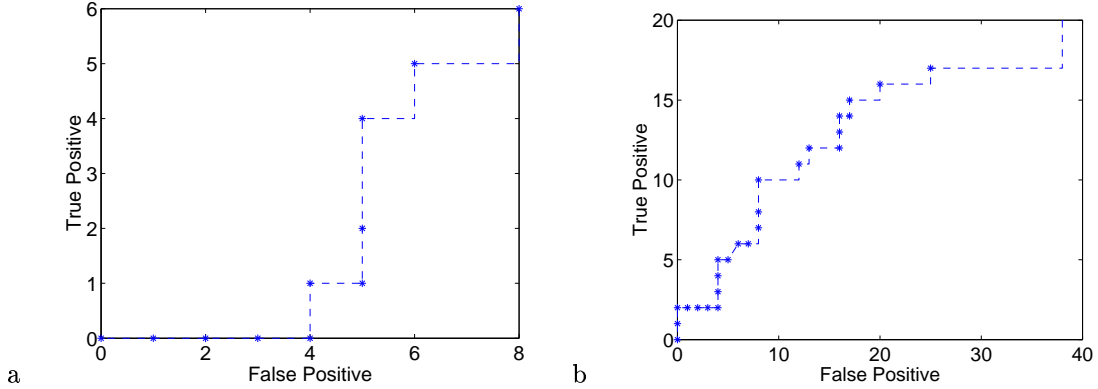

Figure 11: ROC, balanced trauma data results; (a) standard probabilistic k-nn and (b) feature scaling probabilistic k-nn.

8   Discussion

A new approach to generating confidences in classification models has been presented - based upon MCMC methods and averaging over a large number of plausible models. In situations where some forecast error is always going to occur - this approach identifies predictions in which little confidence can be placed. This confidence is measured in terms of the posterior predictive density at that point with different credible intervals leading to different ROC operating points.

The envelope approach introduced here is applicable to a range of classifier types. As two further examples we now provide results of its application to a simple classifier, a generalised linear model (GLM), and a more complex classifier, a radial basis function (RBF) network. Both the GLM and the RBF were implemented using the auxiliary variable method [3]; geometric priors over the number of features (GLM) or kernels (RBF) were used.

The GLM model takes the form

$$y = \varphi(\mathbf{x}) \tag{5}$$

where:

$$\varphi(\mathbf{x}) = \mathrm{erf}\left(\sum_{i=1}^{|\mathbf{x}|} f_i \beta_i x_i\right) \tag{6}$$

β_i denotes the weight assigned to each input feature x_i and f_i takes a value of either 0 or 1 (denoting whether a feature is used by the model). erf denotes the error function. The MCMC process therefore adjusts the βs of the GLM, and the reversible-jump is over the features used.

The RBF model takes the form

$$y = \text{erf}\left(\beta_0 + \sum_{j=1}^{J} \beta_j \psi(\mathbf{x}_j; \boldsymbol{\mu}_j)\right) \tag{7}$$

where:

$$\psi(\mathbf{x}; \boldsymbol{\mu}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\left(\mathbf{x} - \boldsymbol{\mu}\right)' \Sigma^{-1} \left(\mathbf{x} - \boldsymbol{\mu}\right)/2\right) \tag{8}$$

are the kernel or basis functions, located at $\boldsymbol{\mu}$. Here we used basis functions with a fixed variance, $\sigma^2 = 0.1$. The RJ-MCMC process integrates over the basis function location $\boldsymbol{\mu}_j$, kernel weights $\boldsymbol{\beta_j}$ and intercept β_0.

For the synthetic data, the GLM and the RBF, performed with testing error levels of 36.3% and 12.6% respectively. The fact that the GLM performed so poorly is not particularly surprising as we know the actual Bayes rule decision boundary is far from linear.
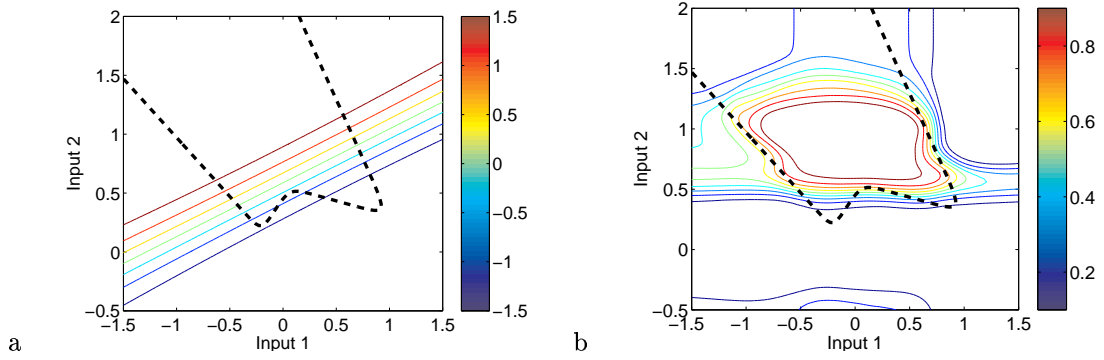

Figure 12: Decision contours, synthetic data; (a) GLM and (b) RBF. The dashed black line shows the Bayes rule decision boundary.

Figs. 12a and 12b show the decision contours generated by the two methods in the feature space of the synthetic data, the GLM models contours being close to linear (by averaging across models there is a slight curve apparent). The RBF model has a more complex decision boundary, although the prior on low kernel numbers means that it does not produce one quite as nonlinear as the probabilistic k-nn models discussed earlier.
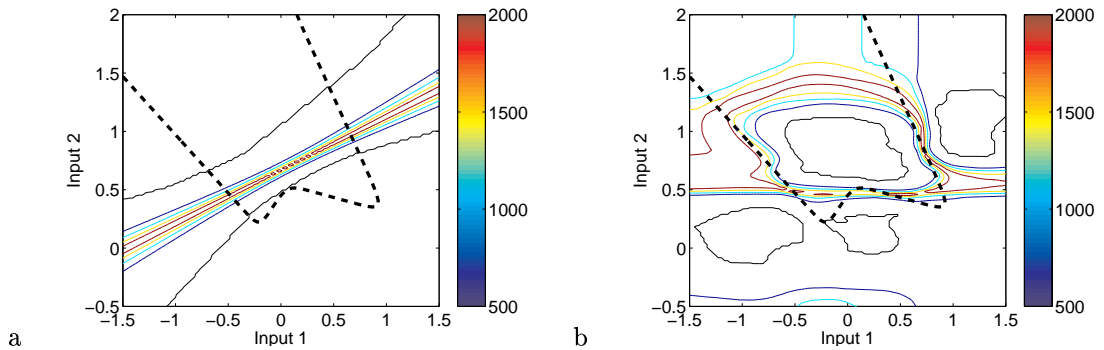

Figure 13: Decision envelopes, synthetic data; (a) GLM and (b) RBF. The dashed black line shows the Bayes rule decision boundary.

Figs. 13a and 13b show the credible envelopes generated by the two methods in the feature space of the synthetic data. The GLM produced a very simple envelope, separating the decision space into three sections, with roughly a

third of feature space classified as Class 1 at the 1/5000 credibility level, a third of feature space classified as Class 2 at the 1/5000 credibility level and the rest lying within the 1/5000 envelope. The RBF envelopes, in contrast, enclose four distinct regions that are confidently classified (at the 1/5000 level). This is of interest as, although the RBF model tends to classify areas where there have been no training points with fairly high probabilities (for instance the upper left portion of space), these areas are also marked with high uncertainty levels (for instance 20% of the models disagree with the class assigned the upper left portion of the input space).
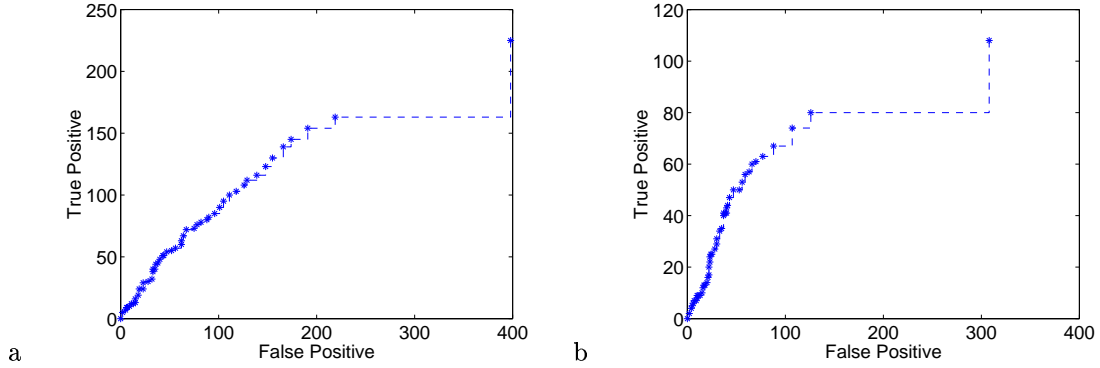

Figure 14: ROC, synthetic data results; (a) GLM and (b) RBF.

Figs. 14a and 14b show the realised ROC curves on the test data. The x-axis is the number of misclassified points contained within the envelope and the y-axis is the number of correctly classified points contained within the envelope. As we move left to right on the figures, the envelope reject region is being gradually expanded, from the 2500/5000 credible level to the 1/5000 credible level, Fig. 14b therefore shows that at the 1/5000 credible level virtually all the data points mis-classified by the RBF model average lie inside the RBF model envelope.

For the trauma data the GLM and RBF performed with testing error levels of 15.1%, 16.0% respectively. The ROC curves generated by gradually expanding the envelope region from the 2500/5000 credible level to the 1/5000 credible level are shown in Figs. 15a and 15b.
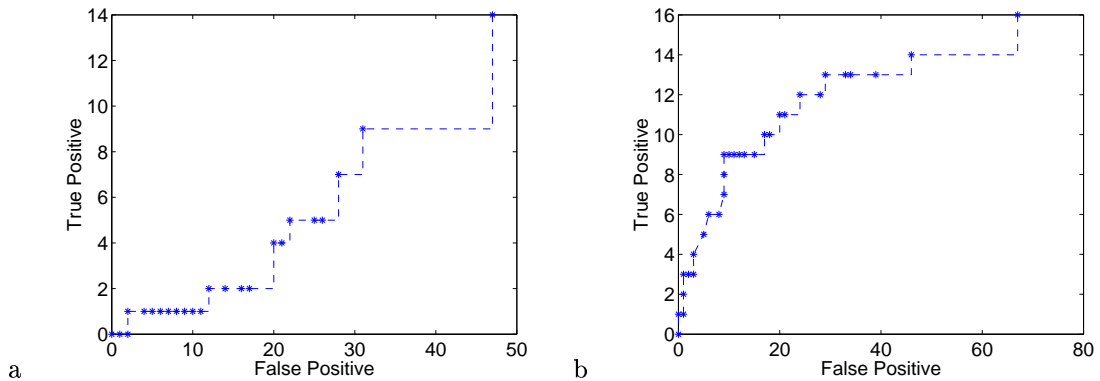

Figure 15: ROC, balanced trauma data results; (a) GLM and (b) RBF.

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 *type* of classifier, even within a RJ-MCMC framework. For some classifiers evaluated here (notably

the k-nn models) the envelopes at the $1/5000$ credibility level contain only a small proportion of those points that were mis-classified. In our future work we hope to investigate the averaging over different plausible *models* as well as different plausible model parameters, as a potential method to tackle this problem.

Acknowledgment

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.