Prediction of Survival Probabilities with Bayesian Decision Trees

V. Schetinin, L. Jakaite

Computer Science Department and Technology, University of Bedfordshire, UK

W. Krzanowski

College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK

Abstract

Practitioners use Trauma and Injury Severity Score (TRISS) models for predicting the survival probability of an injured patient. The accuracy of TRISS predictions is acceptable for patients with up to three typical injuries, but unacceptable for patients with a larger number of injuries or with atypical injuries. Based on a regression model, the TRISS methodology does not provide the predictive density required for accurate assessment of risk. Moreover, the regression model is difficult to interpret. We therefore consider Bayesian inference for estimating the predictive distribution of survival. The inference is based on decision tree models which recursively split data along explanatory variables, and so practitioners can understand these models. We propose the Bayesian method for estimating the predictive density and show that it outperforms the TRISS method in terms of both goodness-of-fit and classification accuracy. The developed method has been made available for evaluation purposes as a stand-alone application.

Keywords: Bayesian prediction, survival probability, Markov chain Monte Carlo, classification tree, trauma care.
DOI: 10.1016/j.eswa.2013.04.009

1. Introduction

For patients alive on arrival at a hospital, the probability of survival is calculated by using a logistic regression model that employs the Trauma and Injury Severity Score (TRISS) system [5, 4, 27, 12, 28]. TRISS-based models

consider up to three most severe injuries which a patient can obtain in six regions of the body: head, face, chest, abdomen, extremities, and external (skin, subcutaneous tissue and burns). The model includes both continuous and categorical screening tests: the first type includes age, systolic blood pressure, and respiratory rate, while the second type includes the severity scores of injuries a patient can obtain in the six body regions, as well as Glasgow Coma Scale (GCS) and type of injury. The screening tests are performed on the patient's arrival by a trained scorer. The calculation of survival probabilities has been made available online as a TRISS Calculator [7].

The screening tests are used to form two aggregated predictors, Injury Severity Score (ISS), and Revised Trauma Score (RTS). The use of such aggregated predictors has revealed unexplained fluctuations of ISS over observed (or actual) probabilities of survival, and some researchers have raised a concern about its predictive ability [38, 4, 28, 1, 39].

The TRISS determines the probability of survival, Ps, using the following logistic regression model:

$$Ps = 1/(1 + e^{-b}), \tag{1}$$

where $b = b_0 + b_1 \times RTS + b_2 \times ISS + b_3 \times A$. Here b_0, b_1, b_2, and b_3 are the regression parameters, and A is is the dichotomised age: $A = 0$, if age < 55, and $A = 1$, otherwise. The regression parameters were defined for blunt and penetrating injuries in [5].

The analysis of actual survival against predicted probabilities is defined as *calibration*, and a visualized relationship between values of these probabilities is a *calibration curve*, see e.g. [22, 53]. An ideal calibration curve is a 45 degree line with zero intercept. In this light the calibration curve for TRISS models has been found to deviate significantly from the ideal [28, 44].

It has been shown that the accuracy of TRISS models can be improved by updating the model parameters on new data, see e.g. [8, 36]. This approach can be implemented efficiently when an appropriate model structure is known. However in practice such a model is difficult to identify. When goodness or fitness of a model can be measured in terms of its likelihood, the model can be fitted to data with the likelihood maximization method. In practice, this approach requires much effort to overcome the optimization problems while the desired improvement cannot be guaranteed [16, 23, 2, 37].

Despite the problems, the accuracy of TRISS predictions has been found acceptable when the types and severities of patients injuries are typical. How-

ever, for cases with four or more injuries as well as with some atypical combinations of injuries, the accuracy could be improved [8, 27, 28].

It is highly desirable for practitioners to estimate the uncertainty in predictions of survival. In general, estimates of uncertainty are required to minimize risks of mistaken decisions and, in particular, to estimate confidence intervals. These intervals can be accurately estimated when a predictive probability density is fully known, but the desired estimates cannot be provided within a concept such as TRISS which employs a maximum likelihood method to fit a logistic regression to data [43, 1].

The above problems motivated us to develop a Bayesian method for prediction of survival. In our research we used the US National Trauma Data Bank (NTDB) which is the major data source of records of injured patients admitted to hospitals and emergency units; these data are available for research from [12]. The NTDB data include information about a patients age, gender, type and regions of injuries along with some clinical and background information about a patients state. The NTDB also includes information about TRISS prediction and outcome of care, alive or died, for each patient. In our research we use well-known Decision Tree (DT) models which are induced from given data in such a way that features which make a distinguishable contribution to the model outcome are selected. These features make axis-parallel partitions, and as a result users find the DT models interpretable [6, 18].

We tested the proposed Bayesian method on a set of patients registered in the NTDB with multiple injuries. We expected that in such cases the proposed method would significantly outperform the TRISS method. However, in this research we did not generalize our method to the entire NTDB population including about 2 millions of patients because such a generalization would require much more intensive computational experiments. For evaluation of our method, we developed a Bayesian calculator of survival as a stand-alone application available from [51].

2. Related Work

In the related literature we found a number of machine learning and simulation methods which are competitive to conventional statistical methods. Simulation was principally via Markov chain Monte Carlo (MCMC) methods, while ML methods included artificial neural networks and support vector

machines which are well known for providing non-linear fitting of models to data [3, 56].

The goodness-of-fit or calibration of predictive models produced by these methods was measured in terms of least square error. As for conventional statistical methods, the performance was measured in term of the receiver operating characteristic curve (ROC), or more specifically, by the area under this curve (AUC), see e.g. [18, 31]. The performance was also assessed in terms of accuracy of classification or discrimination measured as the ability of a model to discriminate survived patients from ones that died [27].

2.1. Machine Learning Methods

The study by Sujin et al[55] compared the mortality prediction models built with different Machine Learning methods on a data set collected by the University of Kentucky Hospital. The authors of this study compared the artificial neural networks, support vector machines, DT, and conventional logistic regression models. The data set used for the experiments included 38,474 patients, information about which was represented by 41 variables. It was reported that only 15 out of these variables made a significant contribution to the prediction, including blood pressure, respiration rate, GCS, comorbidity, and blood serum composition. The best performance was achieved with the DT model, AUC = 0.892, whilst the AUC for the artificial neural networks, support vector machines, and logistic regression models were 0.874, 0.876, and 0.871, respectively. All the methods were implemented in the SPSS Clementine software [24].

In [52], an artificial neural network-based approach was described. This approach was compared with a logistic regression model on 13,164 patients whose physiological information was represented by 17 variables. Both the neural networks and the regression models were built to provide non-linear fits to the data, and their goodness-of-fit (or calibration) was evaluated in terms of least square error. It was reported that the resultant neural network model slightly outperformed the logistic regression in terms of AUC.

The authors of study [25] employed a probabilistic artificial neural network for predicting mortality in emergency room. The network with 10 input variables was trained with a genetic algorithm. The calibration was evaluated using Hosmer-Lemeshow statistic. The study conducted on a data set of 533 patients revealed that the performances of the neural network and logistic regression models were comparable.

In [40], such machine learning methods as Naive Bayesian classifier, DT, support vector machine and artificial neural network were used for predicting survival of burn patients. The authors reported that the prediction accuracies of these methods were comparable. All the methods were implemented in the Weka software [21]. The data were represented by 10 features, namely age, gender, and percentages of burn in the eight body regions. The DT was induced with the C4.5 algorithm [41].

The study by Clermont et al [11] compared an artificial neural network method with a logistic regression model for predicting mortality in the intensive care units. The study was carried out on 1,647 patient cases represented by 24 input features. The features included patient's age, the values of 16 physiologic variables including temperature, heart rate, blood pressure, respiratory rate, oxygenation, composition of the blood serum, GCS, as well as binary indicators for the absence or presence of chronic conditions. The goodness-of-fit was evaluated using Hosmer-Lemeshow statistics, and the comparisons were made in terms of classification accuracy as well as in terms of AUC value. It was shown that the both methods had similar performances.

In [2], an artificial neural network was compared against a TRISS model on the UK Trauma data represented by 16 anatomical and physiological predictor variables. Both models provided the similar accuracy of classification, but the TRISS model showed better performance in terms of AUC, 0.941 versus 0.921. The neural network model provided a better calibration in terms of Hosmer-Lemeshow statistics, 58.3 versus 105.4. It was also found that the head injury, age, and chest injury made the most important contribution to the outcome, while respiration rate, heart rate, and systolic blood pressure were underestimated (their contribution was less important). The authors concluded that the TRISS model was adequate but not optimal.

A Bayesian belief network described in [54] was built to predict morbidity and outcomes in wounded patients. Bayesian belief networks are known as graphical models capable of explaining relationships between predictor variables; such models cannot be developed with conventional logistic regression methods. The study conducted on a data set of 22 patients revealed that the logistic regression outperformed the Bayesian belief networks in terms of AUC. The described Bayesian belief network was developed to estimate one of three types of outcomes: likelihood of infection, requirement for intensive care, and impaired wound healing. The relationships between clinical variables and outcomes were identified using DecisionQ FasterAnalytics Bayesian

modelling software[13]. It was found that the likelihood of infection could be estimated using serum albumin, injury severity score, and initial requirement for blood transfusion. The likelihood of intensive care admission was estimated using the blood transfusion requirement, physiological variables and serum biomarkers. Impaired wound healing was estimated using the indicator of intensive care admission, serum biomarkers and estimated hospital length of stay. The network consisted of 12 nodes, with two-to-three states per node.

2.2. Simulation

The authors of review [30] found that conventional modelling techniques used in critical care could be improved by simulation methods such as MCMC. They analysed the usefulness and limitations of applications of these methods presented in the related literature and concluded that simulation provided practitioners with additional information on risks.

In [33], conventional and Bayesian logistic random effects regression models were compared for predicting outcomes on a data set of 8,509 patients with Traumatic Brain Injury. The Bayesian method has been implemented in statistical packages such as WinBUGS [34], MLwiN [42], MCMCglmm [20], and SAS [46]. It was reported that both methods provided similar prediction accuracy. The results of the Bayesian method were critically dependant on the chosen priors as well as on the sampler's settings which can affect the convergence of the Markov Chain if given inappropriately.

3. Bayesian Predictions

When Bayesian inference is employed for predictions, it is typically assumed that there exist a number of models which can appropriately approximate the relationship between predictor variables and output variable (or outcome) observed in given data. Given models with parameters, we can fit them to the data. It is most often the case that none of the models describes the true relationship between input and output variables. However, we assume that averaging over the models could result in more accurate approximation to the true relationship. The most efficient averaging over models is achieved within the Bayesian methodology. However, when Bayesian methods are applied to real data, a number of problems are raised. One specific problem we address in the paper is associated with Bayesian averaging over hierarchical models, such as DTs.

When DT models are collected in an ensemble, for interpretation purpose a single DT model with Maximum a Posterior likelihood can be selected as described in [10]. However our technique proposed in [47] for interpretation of an ensemble of DT models has been capable of finding a single DT model providing a better accuracy of predicting survival probabilities.

The methodology of Bayesian averaging over DT models has been made computationally feasible with the MCMC method [9, 14]. This method aims to explore a posterior density of model parameters by making random walk proposals. The desired density is approximated by drawing samples from areas with high posterior density of the model parameters (so-called areas of interest).

In the case of DT models, a model parameter space is often of variable size, and the MCMC method is extended to Reversible Jump (RJ) [19]. The desired approximation is then achieved when the RJ MCMC algorithm can explore all areas of the posterior density in a model parameter space of a variable size. However, a posterior density function is often multimodal and the detailed exploration of the areas of interest cannot be achieved in a reasonable time, see e.g. [43]. This affects the accuracy of approximation as the Bayesian model averaging tends to act more as model selection [17, 14].

In practice, when DT models (and hence a model parameter space) are large, results of Bayesian averaging are critically dependent on prior information as shown in [14, 43]. When prior information is available, the averaging is mostly done over areas of high posterior, and the estimates of the desired predictive density are likely to be accurate. However, when prior information is absent, the areas of possible interest cannot be specified and hence detailed exploration may not be possible in a reasonable time [14, 45]. As a result, the MCMC sampler cannot explore all possible areas of interest proportionally to the posterior density of the parameters.

One possible reason for the above disproportion is that the RJ MCMC algorithm tends to simulate samples from an oversized model parameter space [9, 14]. In our previous work we attempted to reduce factors that cause the RJ MCMC to sample from overgrown DT models and proposed a sweeping strategy [48]. In the experiments on the benchmark problems, we described in [49], this strategy has been shown more efficient than the restarting [9] and restricting [14] MCMC strategies. In these experiments the proposed strategy also outperformed the conventional random forest [15].

In our previous research, we also observed that when prior information on predictors was absent, the posterior density cannot be explored in detail,

and some DT models were disproportionally sampled [26]. When in the post-analysis phase we evaluated the frequencies of using predictor variables in the DT models, we found that some predictors were employed rarely. We assumed that these predictors made a weak contribution to the outcome. When we removed DT models which explored such weak predictors from the ensemble, we observed a decrease in entropy of the model mix. The fact of decreasing entropy has been proven as an indicator of improving estimates of the predictive density, see e.g. [32].

The above analysis motivated us to extend the methodology of Bayesian averaging over DT models for predicting survival probabilities. We attempt to improve the RJ MCMC method used for implementation of the Bayesian methodology. We are also interested in exploring the importance of the predictive variables within the proposed method, and expect that the posterior information on predictors can be useful to optimize existing procedures of scoring injury severities.

The methodology of Bayesian averaging over DT models is well developed and described in the literature (see e.g. [9, 14]). The details of the Bayesian method and the proposed MCMC strategy are described in the Appendix.

In the following sections we explore the proposed strategy on a data set of patients from the NTDB. We attempt to improve the accuracy of estimates of predictive density.

4. Trauma Data

We used data on patients registered in the NTDB who were alive on arrival at hospitals, and whose survival probabilities have been calculated with the TRISS method. Table 1 shows the screening tests denoted as variables $x_1, \ldots, x_{17}$ which were used for the TRISS predictions. Variables x_1 (age), x_4 (blood pressure), and x_5 (respiration rate) are continuous, and the others are categorical. The predicted output is the discharge status, $y = \{0, 1\}$.

Figure 1 shows that the TRISS predictions and the observed survival probabilities progressively decrease with the number of injuries ranging from 1 to 20.

The actual survival of patients with one injury was 0.98 while for patients with 20 injuries it was 0.71. The proportions of these patients were 0.17 and 0.0002, respectively. We see that the predicted values are below the actual frequencies and that the difference between their values, or prediction error, tends to increase with the number of injuries obtained by a patient.

Table 1: **Screening Tests**

Test	*Name*	*Range*
x_1	Age	0-99
x_2	Gender	female, male
x_3	Injury type	penetrating, blunt, burn
x_4	Blood pressure	0-299
x_5	Respiration rate	0-59
x_6	GCS Eye	1-5
x_7	GCS Verbal	1-5
x_8	GCS Motor	1-6
x_9	Head severity	0-6
x_{10}	Face severity	0-6
x_{11}	Neck severity	0-6
x_{12}	Thorax severity	0-6
x_{13}	Spine severity	0-6
x_{14}	Abdomen severity	0-6
x_{15}	Upper extremity severity	0-6
x_{16}	Lower extremity severity	0-6
x_{17}	External severity	0-6
y	Discharge status	alive, dead

The majority of patients were registered with 1 to 3 injuries and the TRISS predictions for this largest group of patients are close to the observed survival. The proportions of patients with a larger number of injuries are smaller and so the TRISS model does not fit these data well. In our research we attempt to improve the accuracy of predictions for such groups of patients. In particular, we target a group which includes all the 14,840 patients registered in the NTDB with 11 to 15 injuries. This set does not include about 2% of patient's records we found with one or more missing values as we did not attempt to fill the absent values in this research.

Figure 2 shows the calibration curve of the TRISS model for this group of patients. We see that the observed probabilities are significantly higher than the predicted values. The difference is largest for patients with predicted survival between 0.1 to 0.5.

We can evaluate quantitatively the goodness-of-fit (or calibration) of the

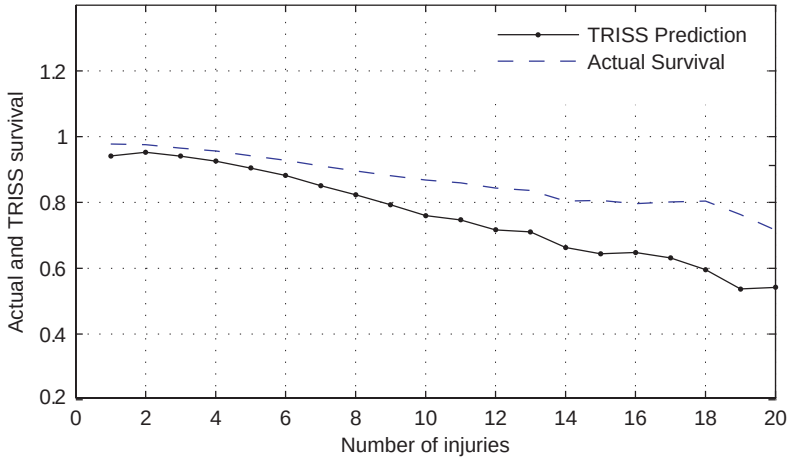


Figure 1: **Observed and predicted survival probabilities for patients with different numbers of injuries.** The TRISS predictions and the observed survival probabilities progressively decrease with the number of injuries ranging from 1 to 20.

TRISS model for this group of patients by using the Hosmer-Lemeshow (HL) statistic used in the related literature [27, 4, 53]. When we split the data into 30 equally sized subsets, its value was 3680.5.

5. Experiments

We used the above data to test the proposed MCMC simulation strategy. The experiments were run within 3-fold cross-validation. We compared the TRISS and Bayesian predictions in terms of classification accuracy, and Hosmer-Lemeshow statistic.

The results were obtained using settings which allowed us to achieve a stationary distribution of the Markov chain and an efficient acceptance rate, while DT models were of a reasonable size. These settings are described in the Appendix.

Figure 3 shows the calibration curve for the proposed Bayesian model. We can see that the Bayesian predictions are much closer to the observed survival rate than the TRISS predictions shown in Figure 2. The value of the Hosmer-Lemeshow statistic was significantly reduced from 3680.5 to 93.4.

We compared the classification accuracies of the Bayesian and TRISS

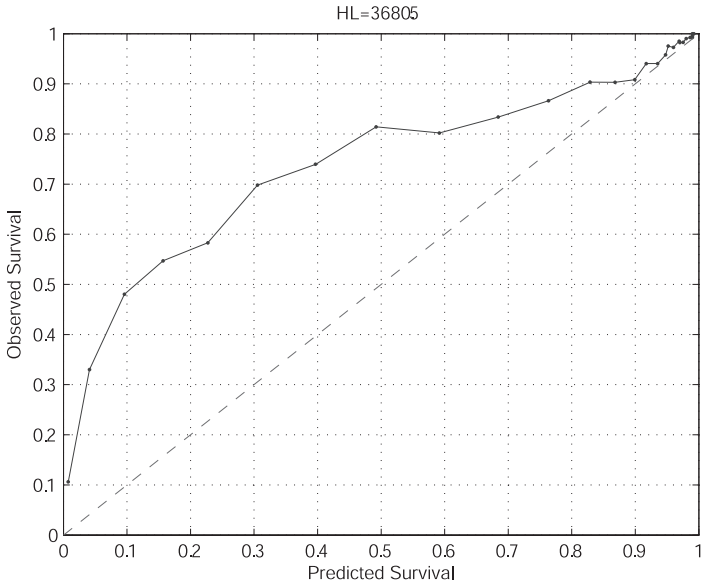


Figure 2: **Calibration curve for TRISS model for patients with 11-15 injuries.** The observed probabilities of survival are significantly higher than the predicted values. The difference is largest for patients with predicted survival between 0.1 to 0.5.

methods. Table 2 shows the classification accuracy (AC), true positive (TP), false negative (FN), true negative (TN), false positive (FP), sensitivity (SE), and specificity (SP) which were calculated by assigning the outcome alive if a patient's survival prediction is higher than 0.5. We can observe that the accuracy of the Bayesian method is higher by 6%. Having a significantly higher FN rate (0.129 versus 0.024), the TRISS model provides the better TN rate (0.121 versus 0.078). Using the standard bootstrap method, we found that the p-values were less than 0.001 for all the statistical tests.

Table 2 shows that the accuracy of the proposed Bayesian method is higher than that of the TRISS method by approx 6%. This part of classification error is reducible, and this error appears when a decision threshold t used for classification differs from the optimal and thus becomes biased. Setting threshold t below the optimal value decreases the FP and increases the FN rates and, vice versa, setting the t above the optimal value increases the false positive and decreases the false negative rates. Therefore a value of t can be set so as to find a compromise between the SE and SP of a diag-

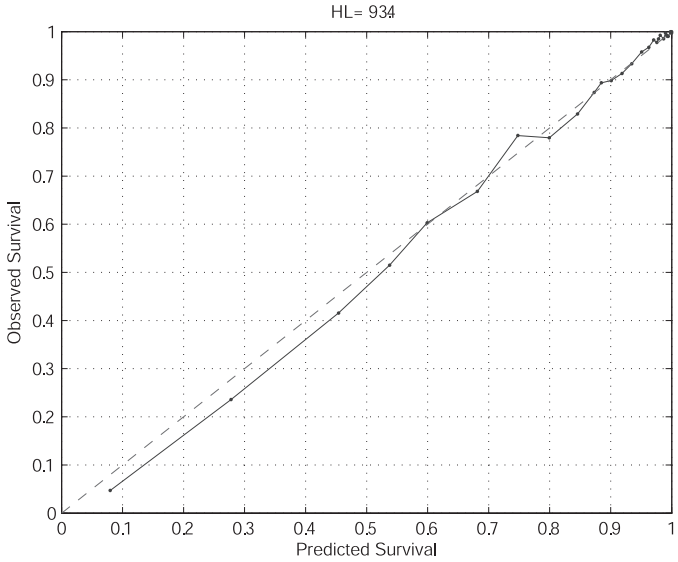


Figure 3: **Calibration curve for the proposed Bayesian model.** The Bayesian predictions are much closer to the observed survival rate than the TRISS predictions.

nostic method, calculated as SE=TP/(TP+FN) and SP = TN/(TN+FP). The sensitivity, SE, shows how the diagnostic method is sensitive to positive results, and the specificity, SP, shows how the method specifies negative results. In our case, positive results are associated with the conditions of a "died" patient, and negative with those of an "alive" patient.

The sensitivity of the Bayesian method shown in Table 2 for the threshold $t = 0.5$ is 0.4868, that is less than the 0.7588 provided by the TRISS method. To achieve a similar sensitivity rate, we can increase the threshold t. For example, we can set $t = 0.74$ to increase the sensitivity to 0.7496, accepting a small decrease in the accuracy from 0.8936 to 0.8671, as shown in the third line of Table 2. In this case the differences between the values of confusion matrices for the Bayesian and TRISS methods remain significant with p-value < 0.001.

We also tested the restarting MCMC strategy of sampling DT models [9, 14] on the same set of patients. The Hosmer-Lemeshow statistic was 17% higher on average than that for the proposed method. The classification accuracy was not significantly lower, however the uncertainty of estimates

Table 2: **Performances of the TRISS and Bayesian Model (BM) with thresholds t**

Model	*AC*	*TP*	*FN*	*TN*	*FP*	*SE*	*SP*
TRISS	0.832	0.121	0.038	0.708	0.129	0.759	0.846
BM, $t = 0.50$	**0.894**	0.078	0.082	0.816	0.024	0.487	0.971
BM, $t = 0.74$	0.867	0.120	0.040	0.747	0.093	0.750	0.890

of the predictive density in terms of entropy of model mixing was significantly higher than that for the proposed method. This allows us to conclude that the proposed MCMC simulation strategy is capable of providing better conditions for Bayesian averaging over DT models.

6. Importance of Screening Tests

The ensemble of DT models collected during MCMC simulation allows us to estimate the contribution of the predictive variables (screening tests) to the outcome. The importance of the variables can be estimated in terms of frequencies of using them in the ensemble. These frequencies (or posterior probabilities) are shown in Table 3.

We can see that the most important contribution is made by the variables x_4 (Blood pressure), x_9 (Head severity), and x_{15} (Upper extremity severity). By contrast, the variables x_5 (Respiration rate), x_{11} (Neck severity), and x_{17} (External severity) are least important, and therefore their contribution can be insignificant for predicting the survival of patients in the target group.

7. Bayesian Calculator of Survival Probabilities

For evaluating the proposed Bayesian method, we developed a Calculator for predicting survival and tested it on the target data set described in Section 4. The Calculator allows the user to compare the Bayesian and TRISS predictions for a random patient drawn from the data set. The comparison is made in terms of prediction accuracy as described in Section 5. The user can also input new values of the screening tests to make predictions.

It is important that the Calculator allows the user to estimate the predictive probability density in order to assess the confidence intervals, which are associated with risk of making mistaken decisions. These estimates are

Table 3: **Importance of Screening Tests**

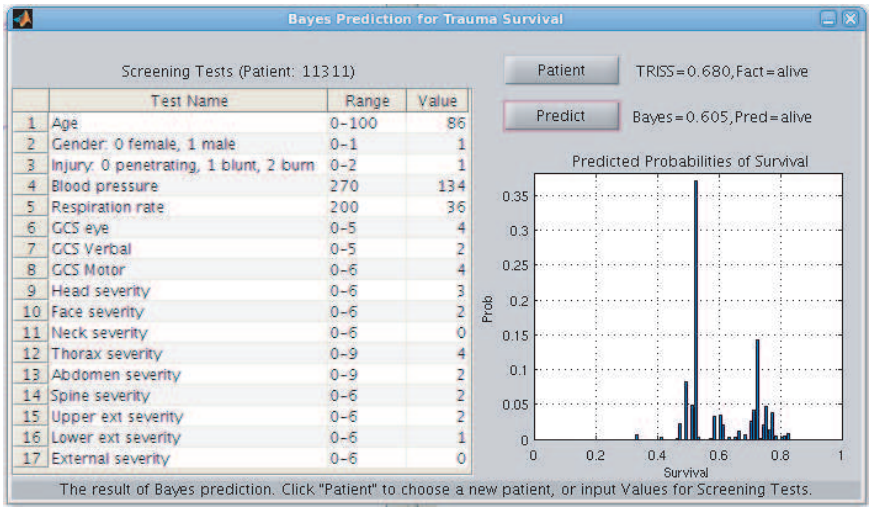
Test	*Name*	*Importance*
x_4	Blood pressure	0.157
x_9	Head severity	0.122
x_{15}	Upper extremity severity	0.115
x_1	Age	0.107
x_{13}	Spine severity	0.093
x_{12}	Thorax severity	0.081
x_{16}	Lower extremity severity	0.064
x_2	Gender	0.053
x_{10}	Face severity	0.052
x_8	GCS Motor	0.050
x_{14}	Abdomen severity	0.045
x_6	GCS Eye	0.018
x_7	GCS Verbal	0.014
x_3	Injury type	0.014
x_5	Respiration rate	0.006
x_{11}	Neck severity	0.004
x_{17}	External severity	0.003

made individually for each patient, whilst the TRISS method is unable to provide such estimates.

Figure 4 presents a screenshot of the calculator interface. The first column in the table Screening Tests shows the 17 screening tests that are described in Table 1. The second column shows the ranges of these tests. The third column displays values which the user can input or edit within the specified ranges.

The graph Predicted Probabilities of Survival displays the probabilities of survival for a patient with the given screening tests. Each of the predicted probabilities can be interpreted as a hypothesis which is tested on the data set in the context of Bayesian inference. The bars on the graph show the observed probabilities of these hypotheses. The estimates of the predictive density shown in the graph provide all the information required to calculate the confidence intervals.

Consider the example shown in Figure 4 for patient No 11311 with TRISS



	Test Name	Range	Value
1	Age	0-100	86
2	Gender: 0 female, 1 male	0-1	1
3	Injury: 0 penetrating, 1 blunt, 2 burn	0-2	1
4	Blood pressure	270	134
5	Respiration rate	200	36
6	GCS eye	0-5	4
7	GCS Verbal	0-5	2
8	GCS Motor	0-6	4
9	Head severity	0-6	3
10	Face severity	0-6	2
11	Neck severity	0-6	0
12	Thorax severity	0-9	4
13	Abdomen severity	0-9	2
14	Spine severity	0-6	2
15	Upper ext severity	0-6	2
16	Lower ext severity	0-6	1
17	External severity	0-6	0

Figure 4: **Bayesian Calculator Screenshot.** The table Screening Tests shows the 17 screening tests and the values of these tests which the user can input or edit within the specified ranges. The graph Predicted Probabilities of Survival shows the probabilities of survival for a patient. The estimates of the predictive density shown in the graph provide the information required to calculate the confidence intervals.

survival probability 0.680 and outcome alive. For this patient, the Calculator predicts a survival probability of 0.605. As this value exceeds 0.5, the predicted outcome is alive. The locations and heights of the bars shown in the graph present the estimates of predictive density. All the bars on the left from the 0.5 mark on the x-axis represent low probabilities of survival associated with the outcome died whereas all the bars on the right represent high probabilities of survival. Observing these probabilities, the user can analyse the risk for this patient.

In this example, the sum over the first bars (on the left from 0.5) is smaller than the sum over the bars on the right. The substantial proportion of the former bars warns the user about a high death risk attached to this prediction. These bars make the predicted probability distribution wider and the uncertainty interval larger.

In addition to a predicted probability of survival, the user can analyse the confidence interval calculated for a given patient. The lengths of the intervals can be associated with the difficulty of treatment for patients – the larger the interval, the more difficult is the case. We also observed some bars with unex-

pectedly low values of the observed probabilities, see the bar located around 0.51 in the graph, which is a case of a multimodal distribution. The user can conclude that patients with such predictions require special attention, and an investigation of similar cases could provide additional information about possible causes of the uncertainty.

The Calculator can be downloaded from a web page [51] to be installed and run on a Windows XP 32-bit or Linux 64-bit machine The Bayesian risk assessments are computationally expensive, and so a high performance machine with a 64-bit processor and 4 GB memory is recommended. The Calculator can be used for evaluation purposes and allows users to observe both the predictive distribution and the calculated confidence intervals.

8. Discussion

In the related publications reviewed in Section 2, the main Machine Learning methods (Artificial Neural Networks, Support Vector Machines, and Decision Trees) have not been found to outperform the TRISS method in terms of accuracy of predicting survival on the trauma data. In the Introduction we noted that the Machine Learning methods cannot provide an assessment of the predictive posterior distribution as proposed in our work. None of these methods is currently used in emergency care. It is important to note also that the use of Machine Learning methods (in particular Support Vector Machines) requires careful and accurate setting of parameters (such as kernels) as well as of a proper model structure. The use of Artificial Neural Networks requires using a proper back-propagation algorithm and its learning parameters. Finding proper settings typically requires the analysis of results of experiments run with different parameters of a machine Learning method and a model structure. However when such experiments are done, one might question whether the settings were explored within only a limited range of possible values. This is what we called the problem of likelihood maximization.

In particular, we found that a single classification tree model [6], constructed with the Matlab Statistics Toolbox [35], can provide only the frequentist estimation of survival, although its discrimination performance was comparable with that of the proposed method.

As we discussed in the Introduction, when a model structure is defined properly we can use the likelihood maximization method to fit the model to a given data set. Optimal results will be guaranteed if the model likelihood

function is unimodal. However, often we do not know whether our model structure is given properly or whether its likelihood function is unimodal. A more practical scenario is when we have a few models, each of which provides a suitable goodness-of-fit in a particular area of the model parameter space. The use of Bayesian methodology in such cases allows us to average outcomes of the models according to their likelihood values and given prior information. The Bayesian average has been shown improving results even if there is no prior information on models and the so-called uniform (or flat) prior is used, see e.g. [14]. The improvement is, however, achievable if the models do not suffer from the overfitting problem as discussed in [17]. Therefore the average provides a practical way to combine prior information (if available) with a set of models which the user could consider competitive to a single model believed to be of a suitable structure. In this context, the strength of the Bayesian method is in averaging over possible models regardless of how much we know about a proper model structure and its parameters.

When a single DT model is built, its likelihood is estimated for a new split made by an assigned feature. If the likelihood is increased, the new split is included in the model, and the assigned feature is assumed making a distinguishable contribution to the classification. Otherwise, the new split is unlikely included in the model as the splitting feature is unable to make a significant contribution. Therefore, a resultant DT model includes only features that make a significant contribution. The importance of these features cannot be evaluated within a single DT. However, the MCMC method allows us to generate DT models of different configurations, and so we can calculate the frequencies of using the features in these models; these frequencies allow us to estimate the desired feature importance.

9. Conclusion

We analysed conventional TRISS-based regression models for predicting survival probabilities of injured patients. In the related literature we found firstly that the prediction accuracy of the TRISS method can be improved and secondly that such attempts have been made by employing the ML and simulation methods. However, we have not found evidences that any of these attempts significantly improved the TRISS method.

Based on a regression model, the TRISS method cannot provide the estimates of predictive probability density that is required to evaluate confidence intervals. The ISS, which is used as an aggregated predictor for TRISS, is ob-

served with unexplainable fluctuations and so may be misleading. Moreover practitioners find that the TRISS models are difficult to interpret. This motivated us to explore Bayesian model averaging over DT models for predicting survival.

In this paper, we analysed the implementation of the Bayesian methodology with RJ MCMC simulation and found that in the burn-in simulation phase DT models tend to grow excessively. The existing MCMC strategies were unable to manage the excessive growth efficiently in terms of diversity of model mixing. Moreover these strategies require additional settings which have to be adjusted experimentally.

To provide better conditions for detailed exploration of the posterior density during the simulation, we proposed a sweeping strategy. This strategy was tested on a large set of patients registered in the NTDB with multiple injuries. The results showed that Bayesian model averaging significantly outperforms the TRISS-based model in terms of goodness-of-fit and classification accuracy. Moreover, the proposed strategy has slightly outperformed the conventional MCMC strategy.

The use of DT models for the Bayesian averaging allowed us to estimate the contribution of each screening test to the outcome. This is a reasonable argument for practitioners to use the DT models discussed in Section 3 in the context of the transparency of models and their ability to select important explanatory variables.

The above results allow us to conclude that the proposed method is capable of improving the accuracy of predictions for survival of a patient with multiple injuries. The desired confidence intervals can be accurately estimated for each patient. Information about the importance of screening tests could be useful for cost analysis and for further improvement of the prediction accuracy.

References

[1] T. Bailey, R. Everson, J. Fieldsend, W. Krzanowski, D. Partridge, V. Schetinin, Representing classifier confidence in the safety critical domain an illustration from mortality prediction in trauma cases, Neural Computing and Applications 16 (2007) 1–10.

[2] D. Becalick, T. Coats, Comparison of artificial intelligence techniques with UK TRISS for estimating probability of survival after trauma, Journal of Trauma 51 (2001) 123–133.

[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

[4] O. Bouamra, A. Wrotchford, S. Hollis, A. Vail, M. Woodford, F. Lecky, A new approach to outcome prediction in trauma: A comparison with the TRISS model, Journal of Trauma 61 (2006) 701–710.

[5] C.R. Boyd, M.A. Tolson, W.S. Copes, Evaluating trauma care: The TRISS method, Journal of Trauma 27 (1984) 370–378.

[6] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Chapman and Hall, 1984.

[7] K. Brohi, TRISS - Overview and desktop calculator, http://www.trauma.org/index.php/main/article/387/, 2012.

[8] M. Chawda, F. Hildebrand, H. Pape, P. Giannoudis, Predicting outcome after multiple trauma: which scoring system?, Injury 35 (2004) 347–358.

[9] H. Chipman, E. George, R. McCullock, Bayesian CART model search, Journal of American Statistics 93 (1998) 935–960.

[10] H.A. Chipman, E.I. George, R.E. McCulloch, Bayesian treed models, Mach. Learn. 48 (2002) 299–320.

[11] G. Clermont, D.C. Angus, S.M. DiRusso, M. Griffin, W.T. Linde-Zwirble, Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic regression models, Critical Care Medicine 29 (2001) 291–296.

[12] Committee on Trauma. American College of Surgeons, NTDB Version 7.2, http://www.facs.org/trauma/ntdb/ntdbapp.html, 2007.

[13] DecisionQ Corporation, Turning data into decisions, http://www.decisionq.com, 2010.

[14] D. Denison, C. Holmes, B. Mallick, A. Smith, Bayesian Methods for Nonlinear Classification and Regression, Wiley, 2002.

[15] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Mach. Learn. 40 (2000) 139–157.

[16] S. DiRusso, T. Sullivan, C. Holly, S. Cuff, J. Savino, An artificial neural network as a model for prediction of survival in trauma patients: Validation for a regional trauma area, Journal of Trauma 49 (2000) 220–223.

[17] P. Domingos, Bayesian averaging of classifiers and the overfitting problem, in: The 17th International Conference on Machine Learning, Morgan Kaufmann Publishers, 2000, pp. 223–230.

[18] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2001.

[19] P.J. Green, Reversible jump Markov chain Monte Carlo and Bayesian model determination, Biometrika 82 (1995) 711–732.

[20] J.D. Hadfield, MCMC methods for multi-response generalized linear mixed models: The MCMCglmm R package, Journal of Statistical Software 33 (2010) 1–22.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explorations Newsletter 11 (2009) 10–18.

[22] J. Hilden, J.D. Habbema, B. Bjerregaard, The measurement of performance in probabilistic diagnosis, Methods of Information in Medicine 17 (1978) 227–237.

[23] A. Hunter, L. Kennedy, J. Henry, I. Ferguson, Application of neural networks and sensitivity analysis to improved prediction of trauma survival, Computer Methods and Programs in Biomedicine 62 (2000) 11–19.

[24] IBM Corporation, SPSS software, http://www-01.ibm.com/software/analytics/spss/, 2013.

[25] F. Jaimes, J. Farbiarz, D. Alvarez, C. Martnez, Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room, Critical Care 9 (2005) 150–156.

[26] L. Jakaite, V. Schetinin, Feature selection for Bayesian evaluation of trauma death risk, in: The 14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics, Springer, 2008, pp. 123–126.

[27] P. Kilgo, J. Meredith, T. Osler, Incorporating recent advances to make the TRISS approach universally available, Journal of Trauma 60 (2006) 1002–1009.

[28] P. Kilgo, J. Meredith, T. Osler, Injury severity scoring and outcomes research, in: D.V. Feliciano, K.L. Mattox, E.E. Moore (Eds.), Trauma (6th ed), New York, McGraw-Hill, 2008, pp. 223–230.

[29] T. Koshy, Catalan Numbers with Applications, Oxford University Press, 2008.

[30] J.E. Kreke, A.J. Schaefer, M.S. Roberts, Simulation and critical care modeling, Current Opinion in Critical Care 10 (2004) 395–398.

[31] W.J. Krzanowski, D.J. Hand, ROC Curves for Continuous Data, Chapman & Hall/CRC, 1st edition, 2009.

[32] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley and Sons, Inc., 2004.

[33] B. Li, H.F. Lingsma, E.W. Steyerberg, E. Lesaffre, Logistic random effects regression models: a comparison of statistical packages for binary and ordinal outcomes, Medical Research Methodology 23 (2011) 11–77.

[34] D. Lunn, A. Thomas, N. Best, D. Spiegelhalter, WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility, Statistics and Computing 10 (2000) 325–337.

[35] MathWorks, Inc, Matlab Statistics Toolbox, version 2012a, http://www.mathworks.co.uk/products/statistics/, 2012.

[36] F. Millham, W. LaMorte, Factors associated with mortality in trauma: re-evaluation of the TRISS method using the National Trauma Data Bank, Journal of Trauma 56 (2004) 1090–1096.

[37] J.S. Oakland, Statistical Process Control (5th edition), Butterworth-Heinemann, 2002.

[38] T. Osler, R. F.B., G. Badger, M. Healey, D. Vane, S. Shackford, A simple mathematical modification of TRISS markedly improves calibration, Journal of Trauma 53 (2002) 630–634.

[39] T. Osler, L. Glance, J. Buzas, D. Mukamel, J. Wagner, A. Dick, A trauma mortality prediction model based on the anatomic injury scale, Annals of Surgery 247 (2008) 1041–1048.

[40] B.M. Patil, R.C. Joshi, D. Toshniwal, S. Biradar, A new approach: role of data mining in prediction of survival of burn patients, Journal of Medical Systems 35 (2011) 1531–1542.

[41] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.

[42] J. Rasbash, C. Charlton, W.J. Browne, M. Healy, B. Cameron, MLwiN. Version 2.02. Centre for multilevel modelling, University of Bristol, http://www.bristol.ac.uk/cmm/software/mlwin, 2005.

[43] C. Robert, G. Casella, Monte Carlo Statistical Methods, Springer Texts in Statistics, Springer, 2004.

[44] F. Rogers, T. Osler, M. Krasne, A. Rogers, E. Bradburn, J. Lee, D. Wu, N. McWilliams, M. Horst, Has TRISS become an anachronism? A comparison of mortality between the National Trauma Data Bank and major trauma outcome study databases, Journal of Trauma and Acute Care Surgery 73 (2012) 326–331.

[45] S. Rogers, M. Girolami, A First Course in Machine Learning, Chapman & Hall/CRC, 1st edition, 2011.

[46] SAS/STAT, User guide 9.2. Introduction to Bayesian procedures. Centre for multilevel modelling, SAS Institute Inc., 2009.

[47] V. Schetinin, J.E. Fieldsend, D. Partridge, T.J. Coats, W.J. Krzanowski, R.M. Everson, T.C. Bailey, A. Hernandez, Confident interpretation of Bayesian decision tree ensembles for clinical applications, IEEE Transactions on Information Technology in Biomedicine 11 (2007) 312–319.

[48] V. Schetinin, J.E. Fieldsend, D. Partridge, W.J. Krzanowski, R.M. Everson, T.C. Bailey, The Bayesian decision tree technique with a sweeping strategy, in: The International Conference on Advances in Intelligent Systems, IEEE Computer Society, 2004.

[49] V. Schetinin, J.E. Fieldsend, D. Partridge, W.J. Krzanowski, R.M. Everson, T.C. Bailey, A. Hernandez, Comparison of the Bayesian and randomized decision tree ensembles within an uncertainty envelope technique, Journal of Mathematical Modelling and Algorithms 5 (2006) 397–416.

[50] V. Schetinin, L. Jakaite, Classification of newborn EEG maturity with Bayesian averaging over decision trees, Expert Systems with Applications 39 (2012) 9340–9347.

[51] V. Schetinin, L. Jakaite, W. Krzanowski, Bayesian calculator, Standalone application for Linux and Windows, http://www.traumacalc.org/bc, 2012.

[52] A. Silva, P. Cortez, M.F. Santos, L. Gomes, J. Neves, Mortality assessment in intensive care units via adverse events using artificial neural networks, Artificial Intelligence in Medicine 36 (2006) 223–234.

[53] E. Steyerberg, A. Vickers, N. Cook, T. Gerds, M. Gonen, N. Obuchowski, M. Pencina, M. Kattan, Assessing the performance of prediction models: A framework for traditional and novel measures, Epidemiology 21 (2010) 128–138.

[54] A. Stojadinovic, J. Eberhardt, T.S. Brown, J.S. Hawksworth, F. Gage, D.K. Tadaki, J.A. Forsberg, T.A. Davis, B.K. Potter, J.R. Dunne, E.A. Elster, Development of a Bayesian model to estimate health care outcomes in the severely wounded, Journal of Multidisciplinary Healthcare 16 (2010) 125–135.

[55] K. Sujin, K. Woojae, W.P. Rae, A comparison of intensive care unit mortality prediction models through the use of data mining techniques, Healthcare Information Research 17 (2011) 232–243.

[56] V. Vapnik, Statistical Learning Theory, Wiley, 1998.

Appendix. Bayesian Averaging over Decision Trees

MCMC implementation of the Bayesian Method

When averaging is made over DT models, the Bayesian formalism can be outlined as follows [14]. First we specify the parameters Θ of a DT model we aim to induce from the labelled data $\mathbf{D}$ represented by an m-dimensional input vector x. The outcome of a DT model is $y = 1, \ldots, C$, where $C \geq 2$ is the number of categories to one of which a DT model assigns a given input x.

Given models $M_1, \ldots, M_L$ with parameters $\Theta_1, \ldots, \Theta_L$, we can write the desired predictive distribution as an integral over the extended parameter vector $\Theta = (\Theta_1, \ldots, \Theta_L)$:

$$p(y|x, \mathbf{D}) = \int_{\Theta} p(y|x, \Theta) p(\Theta|\mathbf{D}) \mathrm{d}\Theta = \sum_{i=1}^{L} p(y|x, \Theta_i) p(\Theta_i|M_i, \mathbf{D}) p(M_i), \quad (2)$$

were $p(M_i)$ is the prior distribution of model M_i, $p(\Theta_i|M_i, \mathbf{D})$ is the posterior density of Θ_i given model M_i and data $\mathbf{D}$, and $p(y|x, \Theta_i)$ is the posterior predictive density given the parameters Θ_i.

The above integral is analytically tractable only in trivial cases when the distribution $p(\Theta|\mathbf{D})$ is known. In practice, we can estimate this distribution by drawing N random samples $\Theta^{(1)}, \ldots, \Theta^{(N)}$ from the posterior distribution $p(\Theta|\mathbf{D})$, and then we can write:

$$p(y|x, \mathbf{D}) \approx \sum_{i=1}^{N} p(y|x, \Theta^{(i)}, \mathbf{D}) p(\Theta^{(i)}|\mathbf{D}) = \frac{1}{N} \sum_{i=1}^{N} p(y|x, \Theta^{(i)}, \mathbf{D}). \quad (3)$$

The above approximation is achieved with the MCMC method of simulation or stochastic integration. The accurate approximation is achieved when a Markov chain becomes a random sequence with a stationary probability distribution. Then according to Eq. (3), we can draw the random samples and calculate the desired predictive density.

Figure 5 shows an example of a DT model consisting of two splitting nodes, s_1 and s_2, and three terminal nodes $t_1, \ldots, t_3$. The first node, s_1, called the root, splits the entire data into two disjoint subsets so that data samples from one subset fall into node s_2 via the left branch, and samples from the other subset fall into the terminal node t_2 via the right branch. The node s_2 further partitions the data samples which fall into the terminals t_2 or

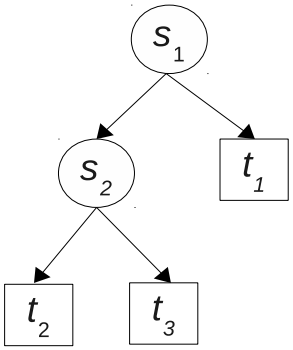


Figure 5: **An example of DT model.** The DT consists of two splitting nodes s_1, s_2 and three terminal nodes $t_1, \ldots, t_3$.

t_3 via the left and right branches. Finally one of the terminal nodes assigns the given input to one of the given classes.

In general, a binary DT with k terminals consists of $(k-1)$ splitting nodes, $s_i, i = 1, \ldots, (k-1)$. The node s_i, has parameters including: the node position in the DT model, $s_i^p, p = 1, \ldots, (k-1)$, an input variable $s_i^v, v = 1, \ldots, m$, and a threshold s_i^q, $q \in (min(x_v), max(x_v))$. The node s_i tests the vth variable against the threshold q and assigns the input x to the left branch if $x_v < q$, or to the right one otherwise. A terminal node t_i assigns the input x to class c with a probability $P_i^c, i = 1, \ldots, k$.

Consequently, a DT model is described by a vector of parameters, Θ, consisting of two parts. The first part includes the following parameters of nodes s_i: positions s_i^p, variables s_i^v, and thresholds s_i^q, $i = 1, \ldots, (k-1)$. The second part includes the probabilities $P_i^c, c = 1, \ldots, C$ for each terminal node $i, i = 1, \ldots, k$.

DT models whose nodes split data into two disjoint subsets are called binary. The number of possible configurations of binary DTs with k terminal nodes, S_k, is determined by the Catalan number, see e.g. [29].

The number S_k grows exponentially with k and becomes very large for DTs with relatively small k. For example, for $k = 25, S_k$ becomes a number

to the power of 12.

In practice, to explain data we need to induce DT models of a reasonable size; the size of a DT model is defined by the number of its terminal nodes, k. Oversized DT models are difficult to interpret, and moreover they are prone to overfit data.

The size of DT models is dependent on the number of data points, p_{min}, allowed to be in terminal nodes – setting a smaller p_{min} increases the size, while setting a greater p_{min} decreases the size. In most cases, prior information on the size of DT models is unavailable, and a suitable p_{min} has to be found empirically.

In practice, the size of DT models is unknown or can be given within a range. In such cases, areas of interest (high posterior density of parameters Θ), which have to be explored in Eq. 3, are of variable size, and MCMC has to be extended to Reversible Jump (RJ) proposed in [19].

Prior information about input variables, such as importance of variables $x_1, \ldots, x_m$, is also often unknown. In such cases, we can assign a variable v for the the node s_i to be drawn randomly from the uniform discrete distribution, $v \sim U(1, m)$. Similarly, a threshold q can be drawn from the uniform discrete distribution, $q \sim U(min(x_v), max(x_v))$.

It has been shown that the above priors are sufficient in order to build and explore DT models of different configurations within the RJ MCMC method [9, 14]. For binary DT models, the number of possible configurations, S_k, is defined by Eq. **??**. From this equation, we see that the larger the k, the larger is the number S_k. So we expect that MCMC algorithm will explore possible DT configurations of size k with probabilities proportional to S_k.

RJ MCMC for Averaging over DT Models

The RJ MCMC method has been implemented for Bayesian averaging over DT models of variable size [9, 14, 50]. To explore DT models it has been proposed to use the *birth*, *death*, *change-split*, and *change-rule* moves made with Metropolis-Hastings (MH) sampler.

The first two, birth and death, moves were proposed to reversibly change the number of nodes in a DT model (or the dimensionality of the model parameter vector Θ). The third and fourth moves, change-split and change-rule, were aimed at changing the parameters Θ within a current dimensionality. The change-split move replaces a variable v in a chosen DT node s_i, while the change-rule move modifies a threshold q in node s_i.

The change-split moves are aimed at making large changes in the model parameters in order to potentially increase the chance of sampling from areas of interest. Such moves are intended to disrupt a long sequence of the posterior samples drawn from a local area of interest.

In contrast, the change-rule moves are aimed at making small changes in the parameters to let MCMC explore a surrounding area in detail. These moves are made more frequently than the others.

The MH sampler starts with a DT consisting of one splitting node whose parameter Θ is assigned within the predefined priors. Making the above moves, the sampler attempts to grow the DT model to a reasonable size by fitting its parameters Θ to the data. The fitness or likelihood of DT models is gradually increased and then becomes oscillatory around some value. This phase, named the *burn-in*, has to be preset sufficiently long in order to achieve a stationary distribution of the Markov chain. When the Markov chain becomes stationary, the samples of the posterior distribution are collected to approximate the desired predictive distribution – this phase is called *post burn-in*.

The above moves are made with the given proposal probabilities. Their values are dependent on the complexity of a classification problem – more complex problems require larger DT models. To grow such models, the proposal probabilities for the death and birth moves are set to larger values. In general, there is no guidance for setting proper parameters of the MH sampler, and their values have to be found empirically [9, 14, 50].

The proposed change is accepted according to the MH rule, see e.g. [14]:

$$\alpha = \min\left(1, \frac{p(\mathbf{D}|\Theta^p)p(\Theta^p)}{p(\mathbf{D}|\Theta)p(\Theta)}\frac{q(\Theta|\Theta^p)}{q(\Theta^p|\Theta)}\right), \tag{4}$$

where $p(\mathbf{D}|\Theta^p)$ and $p(\mathbf{D}|\Theta)$ are the likelihoods of DT models with the proposed and current parameters Θ^p and Θ, respectively; $p(\Theta^p)$ and $p(\Theta)$ are the prior distributions of the parameters; $q(\Theta^p|\Theta)$ is the conditional distributions of moving from the current parameter Θ to a proposed parameter Θ^p (so-called transition distribution); and $q(\Theta|\Theta^p)$ is the density of the reverse transition.

When the birth or death move changes a dimensionality of a DT model, the acceptance rule needs to count a proposal ratio, R. This ratio is dependent on the number of possible configurations of DT models, S_k, and so we need to count R to keep the Markov chain reversible during the MCMC

simulation. According to [14], the reversibility is kept when the following condition is met:

$$p(\Theta^p|\mathbf{D})q(\Theta|\Theta^p) = p(\Theta|\mathbf{D})q(\Theta^p|\Theta). \tag{5}$$

The above density $p(\Theta|\mathbf{D})$ is

$$p(\Theta|\mathbf{D}) = \left[\prod_i^{k-1} \frac{1}{N(s_i^v)}\frac{1}{m}\right]\frac{k}{S_k}\frac{1}{K}, \tag{6}$$

where $N(s_i^v)$ is the total number of possible splitting rules for variable s_i^v; K is the maximal number of terminals allowed for DT models induced from data.

The transition distributions in Eq. 5 can be written as follows:

$$q(\Theta^p|\Theta) = \frac{b_k}{k}\frac{1}{N(s_i^v)}\frac{1}{m}, \tag{7}$$

$$q(\Theta|\Theta^p) = \frac{d_{k+1}}{D_Q}, \tag{8}$$

where d_k and b_k are the proposal probabilities of death and birth moves, respectively, and D_Q is the number of splitting nodes whose both branches are terminals.

For the birth move, the Θ^p is a $(k+1)$-dimensional vector, and therefore the reversibility is kept when R_b is

$$R_b = \frac{p(\Theta^p|\mathbf{D})q(\Theta|\Theta^p)}{p(\Theta|\mathbf{D})q(\Theta^p|\Theta)}, \tag{9}$$

The above definitions Eq. 7 and Eq. 8 allow us to rewrite the ratio R_b as follows:

$$R_b = \frac{d_{k+1}}{b_k}\frac{k}{D_{Q+1}}\frac{S_k}{S_{k+1}}. \tag{10}$$

Taking into account that $D_Q < k$ and $S_k < S_{k+1}$, we see that the ratio R_b ranges between 0 and 1 ,

Similarly, we can write the ratio R_d for the death move:

$$R_d = \frac{b_k}{d_{k-1}}\frac{D_Q}{k-1}\frac{S_k}{S_{k-1}}. \tag{11}$$

For the above D_Q, the ratio $R_b : R_b \geq 1$.

We can see that the ratios R_b and R_d take different values when a model parameter space is of variable dimensionality. This allows the MH sampler to keep the desired reversibility and explore a parameter space proportionally to the numbers of configurations S_k.

Problems of Sampling DT models

DT models are multilevel hierarchical structures, as shown in Figure 5. Nodes located at a lower hierarchical level are strongly dependent on the predecessor nodes located at upper level. In such hierarchical structures, changes proposed by the MH sampler can significantly redistribute data points falling into DT terminal nodes. The change made in a node close to the DT root is most influential on the distribution. The changes in terminal nodes can be so significant that the likelihood of the DT model is decreased – the closer the node is to the root, the more significant is the change in distribution of data points. In most cases such proposals are rejected. In contrast, a change proposed in a node close to DT terminals is most likely to be accepted as such a change will insignificantly redistribute data samples in the DT terminals. As a result, the MH sampler will only explore limited configurations of DT models [9, 14].

Another problem occurs when the MH algorithm aims to sample large DT models. When a DT model is small and consists of a small number of terminal nodes, the number of data samples falling into the nodes is expected to be much larger than the given minimal number of points, p_{min}. However, when a DT has grown large, the number of data points is decreased so that further partitions become unavailable. This means that birth moves cannot be made until a death move merges two terminal nodes into one node. As a result the MH algorithm will sample a series of DT models with similar distributions of data samples over terminal nodes. Such series affect the diversity of samples from the posterior distribution and, therefore, the accuracy of approximation of the predictive distribution [14, 48].

Another negative effect is that unavailable moves degrade the given proposal probabilities of birth and change moves. When a move is unavailable, the MH algorithm will repeat the current sample, which reduces the diversity of model mixing [14].

In most cases, the number p_{min} is found from experiments – complex problems typically require a small p_{min} to allow growth of large DT mod-

els. However, an inappropriately small p_{min} leads to excessive growth of DT models.

Growing a DT model, the MH algorithm makes birth moves and almost each birth move increases the likelihood of the model. The MH algorithm accepts these moves and the DT model grows rapidly. The growth of the model continues while the number of data samples in its terminal nodes exceeds p_{min} and the likelihood of a proposed model remains acceptable. During this period, the dimensionality of the DT model increases rapidly, and the sampler cannot explore the posterior within each dimensionality in detail. It is unlikely that samples will be drawn from areas of highest posterior density [9, 14].

The growth of DT models is typically monitored, and the modeller can reduce excessive growth by increasing p_{min} as well as by setting a smaller value of the proposal probability for the birth moves.

To mitigate the negative effect of fast growing DT models, Chipman et al [9] have proposed a restarting strategy. This strategy allows a DT model to grow within a limited period in multiple runs. The average over all models grown in these runs produces a better approximation accuracy when the duration of the growth period and the number of the runs are properly set.

A similar idea of restricting the growth of DT models has been proposed by Denison et al [14]. The growth is restricted within a given interval to allow the MH sampler to explore a model parameter space in detail. Both strategies require additional settings for the MH sampler, which have to be found experimentally.

Sweeping Strategy

As an alternative to the restricting strategies, the RJ MCMC method could be modified so as to reduce the number of replications of samples from the posterior density. In our previous work [48], we proposed a sweeping strategy aimed at reducing the number of unavailable moves.

For making a change-split move, the sweeping strategy assigns a new variable $x_v, v \sim U(1, m)$, and a threshold q:

$$q \sim U(a, b), \tag{12}$$

where $U(a, b)$ is a uniform distribution on the interval between $a = \min(x_{v,j})$ and $b = \max(x_{v,j})$ defined by N_p data points falling into the chosen node, where $j = 1, \ldots, N_p$.

For making change-rule moves, a new threshold q' is drawn from a restricted Gaussian distribution:

$$q' \sim N'(q, \sigma^2, a, b), \tag{13}$$

with mean q and given proposal variance σ^2 on the interval (a, b).

The proposed move can be made so that one or more terminal nodes in a DT model will contain fewer data points than p_{min}. If this happens in terminal nodes with a common parent node, these terminals are recombined into one terminal node, and the MH sampler counts such a move as a death move. If however there are two or more such terminals with different parents, the algorithm will assign the proposal unavailable in order to keep the reversibility of the Markov chain.

Similarly to a change move, a birth move assigns a new splitting node with parameters drawn from the given prior. A new splitting variable x_v is drawn from a uniform distribution, $v \sim U(1, m)$, and a new threshold q is assigned as described by Eq. 12.

In our experiments, we observed that a MH sampler using the above prior on change moves proposes fewer unavailable moves and, therefore, the sampler accepts fewer replications of a current parameter vector Θ. Taking this into account, we hypothesise that a reduced number of the replications collected during the MCMC simulation will improve the diversity of model mixing.

In support of this hypothesis, in our previous experiments [49] on the benchmark problems, we observed that the MH sampler using the above prior significantly reduced the dimensionality of parameter vector Θ as well as the uncertainty in estimates of predictive density. The above strategy, named *sweeping* in [48], is applied to the Markov chain in both burn-in and post burn-in phases.

Let a MH sampler make the birth, death, and change moves as described in Section 9. Then we can describe the main steps of the sweeping strategy as follow.

- Birth move:
    1. Select a random terminal node $i \sim U(1, k)$ and count the number of data samples, p, in this node
    2. If $p > 2p_{min}$ then assign a variable, $v \sim U(1, m)$ and threshold given by Eq. 12 to a new splitting node

3. Count the numbers of data samples, p_1 and p_2, split by the new node
4. If $(p_1 \geq p_{min})\&(p_2 \geq p_{min})$ let the MH sampler check the acceptance of the proposal

- Change move (change-split or change-rule):
  1. Select a random splitting node $i \sim U(1, k-1)$ and read its variable v and threshold q
  2. For change-split assign a new variable, $v' \sim U(1, m)$
  3. For change-rule assign a new threshold q' defined by Eq. 13
  4. Apply the proposed change to the DT
  5. Count the number of terminal nodes, n_0, with $p_i < p_{min}$
  6. If $n_0 == 1$, then apply the death move
  7. If $n_0 > 1$, then assign the proposal unavailable and draw a new sample
  8. Let MH sampler check acceptance of the proposal

Here, $n_0 = \sum_i^k I(p_i \leq p_{min})$, where $I(\cdot) = \begin{cases} 1 & \text{if } p_i \leq p_{min}, \\ 0 & \text{otherwise.} \end{cases}$

Setting for the Metropolis-Hastings Sampler

The settings include the following parameters of the sampler:

1. the proposal probabilities for birth, death, change-split, and change-rule moves, Pr
2. the proposal distribution, a Gaussian distribution with the zero mean and standard deviation s
3. the numbers of burn-in and post burn-in samples, nb and np, respectively
4. the sampling rate of the Markov chain, sr
5. the minimal number of data points allowed in terminal nodes, p_{min}

The parameters Pr, s, and nb were the most significant factors that impact on the convergence of the Markov chain, and so we tested a number of variants of these parameters to achieve an acceptable convergence. The sampling rate sr was used to elevate the independence of samples drawn from the Markov chain during the post burn-in phase. At the second stage, the

settings s and p_{min} have been refined to let the MCMC algorithm efficiently sample the posterior distribution of Θ keeping the size of DT models reasonably small; the efficient sampling is achieved when the acceptance rate ranges between 0.25 and 0.5. The use of such a two-stage technique allowed us to reduce the number of possible combinations of the settings to a realistic number not exceeding 10.

The best results were obtained with the following settings. The probabilities for the birth, death, change-split and change-rule moves were $Pr = (0.2, 0.2, 0.1, 0.5)$, respectively. The proposal distribution was a Gaussian with $s = 1.0$. The numbers of samples were $nb = 20,000$ and $np = 5,000$, and the number of data points was $p_{min} = 10$.