Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model

Benjamin Letham
MIT

Cynthia Rudin
MIT

Tyler H. McCormick
University of Washington

David Madigan
Columbia University

Technical Report no. 609
Department of Statistics
University of Washington
August 2013

Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model

Benjamin Letham, Operations Research Center, Massachusetts Institute of Technology. bletham@mit.edu

Cynthia Rudin, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. rudin@mit.edu

Tyler H. McCormick, Department of Statistics, Department of Sociology, Center for Statistics and the Social Sciences, University of Washington. tylermc@u.washington.edu

David Madigan, Department of Statistics, Columbia University. madigan@stat.columbia.edu

Abstract

We aim to produce predictive models that are not only accurate, but are also interpretable to human experts. Our models are decision lists, which consist of a series of *if...then...* statements (for example, *if high blood pressure, then stroke*) that discretize a high-dimensional, multivariate feature space into a series of simple, readily interpretable decision statements. We introduce a generative model called Bayesian Rule Lists that yields a posterior distribution over possible decision lists. It employs a novel prior structure to encourage sparsity. Our experiments show that Bayesian Rule Lists has predictive accuracy on par with the current top algorithms for prediction in machine learning. Our method is motivated by recent developments in personalized medicine, and can be used to produce highly accurate and interpretable medical scoring systems. We demonstrate this by producing an alternative to the $CHADS_2$ score, actively used in clinical practice for estimating the risk of stroke in patients that have atrial fibrillation. Our model is as interpretable as $CHADS_2$, but more accurate.

1 Introduction

Our goal is to build predictive models that are highly accurate, yet are highly interpretable. These predictive models will be in the form of sparse *decision lists*, which consist of a series of *if... then...* statements where the *if* statements define a partition of a set of features and the *then* statements correspond to the predicted outcome of interest. Because of this form, a decision list model naturally provides a reason for each prediction that it makes. Figure 1 presents an example decision list that we created using the Titanic dataset available in R. This dataset provides details about each passenger on the Titanic, including whether the passenger was an adult or child, male or female, and their class (1st, 2nd, 3rd, or crew). The goal is to predict whether the passenger survived based on his or her features. The list provides an explanation for each prediction that is made. For example, we predict that a passenger is less likely to survive than not *because* he or she was in the 3rd class. The list in

if male **and** adult **then** *survival probability* 21% (19% - 23%)
else if 3rd class **then** *survival probability* 44% (38% - 51%)
else if 1st class **then** *survival probability* 96% (92% - 99%)
else *survival probability* 88% (82% - 94%)

Figure 1: Decision list for Titanic. In parentheses is the 95% credible interval for the survival probability.

Fig. 1 is one accurate and interpretable decision list for predicting survival on the Titanic, possibly one of many such lists. Our goal is to learn these lists from data.

Our model, called Bayesian Rule Lists (BRL), produces a posterior distribution over permutations of *if... then...* rules, starting from a large, pre-mined set of possible rules. The decision lists with high posterior probability tend to be both accurate and interpretable, where the interpretability comes from a hierarchical prior over permutations of rules. The prior favors concise decision lists that have a small number of total rules, where the rules have few terms in the left-hand side.

BRL provides a new type of balance between accuracy, interpretability and computation. Consider the challenge of constructing a predictive model that discretizes the input space in the same way as decision trees [Breiman et al., 1984, Quinlan, 1993], decision lists [Rivest, 1987] or associative classifiers [Liu et al., 1998]. Greedy construction methods like classification and regression trees (CART) or C5.0 are not particularly computationally demanding, but in practice the greediness heavily affects the quality of the solution, both in terms of accuracy and interpretability. At the same time, optimizing a decision tree over the full space of all possible splits is not a tractable problem. BRL strikes a balance between these extremes, in that its solutions are not constructed in a greedy way, yet it can solve problems at the scale required to have an impact in real problems in science or society, including modern healthcare.

A major source of BRL's practical feasibility is the fact that it uses pre-mined rules, which reduces the model space to that of permutations of rules as opposed to all possible sets of splits. The complexity of the problem then depends on the number of pre-mined rules rather than on the full space of feature combinations. As long as the pre-mined set of rules is sufficiently expressive, an accurate decision list can be found, and in fact the smaller model space might improve generalization [through the lens of statistical learning theory, Vapnik, 1995]. An additional advantage to using pre-mined rules is that each rule is independently both interpretable and informative about the data.

BRL's prior structure encourages decision lists that are sparse. Sparse decision lists serve not only the purpose of producing a more interpretable model, but also reduce computation, as most of the sampling iterations take place within a small set of permutations corresponding to the sparse decision lists. In practice, BRL is able to compute predictive models with accuracy comparable to state-of-the-art machine learning methods, yet maintain the same level of interpretability as medical scoring systems.

The motivation for our work lies in developing interpretable patient-level predictive models using massive observational medical data. To this end, we use BRL to construct an alternative to the $CHADS_2$ score of Gage et al. [2001]. $CHADS_2$ is widely-used in medical

practice to predict stroke in patients with atrial fibrillation. A patient's $CHADS_2$ score is computed by assigning one "point" each for the presence of congestive heart failure (C), hypertension (H), age 75 years or older (A), and diabetes mellitus (D), and by assigning 2 points for history of stroke, transient ischemic attack, or thromoembolism (S_2). The $CHADS_2$ score considers only 5 factors, whereas the updated CHA_2DS_2-VASc score [Lip et al., 2010b] includes three additional risk factors: vascular disease (V), age 65 to 74 years old (A), and female gender (Sc). Higher scores correspond to increased risk. In the study defining the $CHADS_2$ score [Gage et al., 2001], the scores was calibrated with stroke risks using a database of 1,733 Medicare beneficiaries followed for, on average, about a year.

Our alternative to the $CHADS_2$ was constructed using 12,586 patients and 4,148 factors. Because we are using statistical learning, we are able to consider significantly more features; this constitutes over 5000 times the amount of data used for the original $CHADS_2$ study. In our experiments we compared the stroke prediction performance of BRL to $CHADS_2$ and CHA_2DS_2-VASc, as well as to a collection of state-of-the-art machine learning algorithms: C5.0 [Quinlan, 1993], CART [Breiman et al., 1984], ℓ_1-regularized logistic regression, support vector machines [Vapnik, 1995], random forests [Breiman, 2001], and Bayesian CART [Dension et al., 1998, Chipman et al., 1998]. The balance of accuracy and interpretability obtained by BRL is not easy to obtain through other means: None of the machine learning methods we tried could obtain both the same level of accuracy and the same level of interpretability.

2 Bayesian Rule Lists

The setting for BRL is multi-class classification, where the set of possible labels is $1, \ldots, L$. In the case of predicting stroke risk, there are two labels: stroke or no stroke. The training data are pairs $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ are the features of observation i, and y_i are the labels, $y_i \in \{1, \ldots, L\}$. We let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$.

In Sections 2.1 and 2.2 we provide the association rule concepts and notation upon which the method is built. Section 2.3 introduces BRL by outlining the generative model. Sections 2.4 and 2.5 provide detailed descriptions of the prior and likelihood, and then Sections 2.6 and 2.7 describe sampling and posterior predictive distributions.

2.1 Bayesian association rules and Bayesian decision lists

An association rule $a \rightarrow b$ is an implication with an antecedent a and a consequent b. For the purposes of classification, the antecedent is an assertion about the feature vector x_i that is either true or false, for example, "$x_{i,1} = 1$ and $x_{i,2} = 0$." This antecedent contains two conditions, which we call the cardinality of the antecedent. The consequent b would typically be a predicted label y. A Bayesian association rule has a multinomial distribution over labels as its consequent rather than a single label:

$$a \rightarrow y \sim \text{Multinomial}(\theta).$$

The multinomial probability is then given a prior, leading to a *prior consequent distribution*:

$$\theta|\alpha \sim \text{Dirichlet}(\alpha)$$

Given observations $(\mathbf{x}, \mathbf{y})$ classified by this rule, we let $N_{\cdot,l}$ be the number of observations with label $y_i = l$, and $N = (N_{\cdot,1}, \ldots, N_{\cdot,L})$. We then obtain a *posterior consequent distri-*

bution:

$$\theta|\mathbf{x}, \mathbf{y}, \alpha \sim \text{Dirichlet}(\alpha + N).$$

The core of a Bayesian decision list is an ordered antecedent list $d = (a_1, \ldots, a_m)$. Let $N_{j,l}$ be the number of observations x_i that satisfy a_j but not any of $a_1, \ldots, a_{j-1}$, and that have label $y_i = l$. This is the number of observations to be classified by antecedent a_j that have label l. Let $N_{0,l}$ be the number of observations that do not satisfy any of $a_1, \ldots, a_m$, and that have label l. Let $\mathbf{N}_j = (N_{j,1}, \ldots, N_{j,L})$ and $\mathbf{N} = (\mathbf{N}_0, \ldots, \mathbf{N}_m)$.

A Bayesian decision list $D = (d, \alpha, \mathbf{N})$ is an ordered list of antecedents together with their posterior consequent distributions, which are obtained by excluding data that have satisfied an earlier antecedent in the list:

if a_1 **then** $y \sim \text{Multinomial}(\theta_1)$, $\theta_1 \sim \text{Dirichlet}(\alpha + \mathbf{N}_1)$
else if a_2 **then** $y \sim \text{Multinomial}(\theta_2)$, $\theta_2 \sim \text{Dirichlet}(\alpha + \mathbf{N_2})$
$\vdots$
else if a_m **then** $y \sim \text{Multinomial}(\theta_m)$, $\theta_m \sim \text{Dirichlet}(\alpha + \mathbf{N}_m)$
else $y \sim \text{Multinomial}(\theta_0)$, $\theta_0 \sim \text{Dirichlet}(\alpha + \mathbf{N}_0)$.

Any observations that do not satisfy any of the antecedents in d are classified using the parameter θ_0, which we call the default rule parameter.

2.2 Antecedent mining

We are interested in forming Bayesian decision lists whose antecedents are a subset of a pre-selected collection of antecedents. For data with binary or categorical features this can be done using frequent itemset mining, where itemsets are used as antecedents. In our experiments, the features were binary and we used the FP-Growth algorithm [Borgelt, 2005] for antecedent mining, which finds all itemsets that satisfy constraints on minimum support and maximum cardinality. This means each antecedent applies to a sufficiently large amount of data and does not have too many conditions. The particular choice of the itemset mining algorithm is unimportant as the output is an exhaustive list of all itemsets satisfying the constraints. Because the goal is to obtain decision lists with few rules and few conditions per rule, we need not include any itemsets that apply only to a small number of observations or have a large number of conditions. Thus frequent itemset mining allows us to significantly reduce the size of the feature space, compared to considering all possible combinations of features.

The frequent itemset mining that we do in our experiments produces only antecedents with sets of features, such as "diabetes and heart disease." Other techniques could be used for mining antecedents with negation, such as "not diabetes" [Wu et al., 2004]. For data with continuous features, a variety of procedures exist for antecedent mining [Fayyad and Irani, 1993, Dougherty et al., 1995, Srikant and Agrawal, 1996], or one can create categorical features using interpretable thresholds (e.g, ages 40-49, 50-59, etc.) or interpretable quantiles (e.g., quartiles).

We let $\mathcal{A}$ represent the complete, pre-mined collection of antecedents, and suppose that $\mathcal{A}$ contains R antecedents with up to C conditions in each antecedent.

2.3 Generative model

We now sketch the generative model for the labels $\mathbf{y}$ from the observations $\mathbf{x}$ and antecedents $\mathcal{A}$.

- Sample a decision list length $m \sim p(m|\lambda)$.
- Sample the default rule parameter $\theta_0 \sim \text{Dirichlet}(\alpha)$.
- For decision list rule $j = 1, \ldots, m$:
  - Sample the cardinality of antecedent a_j in d as $c_j \sim p(c_j|\eta)$.
  - Sample a_j of cardinality c_j from $p(a_j|a_1, \ldots, a_{j-1}, c_j, \mathcal{A})$.
  - Sample rule consequent parameter $\theta_j \sim \text{Dirichlet}(\alpha)$.
- For observation $i = 1, \ldots, n$:
  - Find the antecedent a_j in d that is the first that applies to x_i.
  - If no antecedents in d apply, set $j = 0$.
  - Sample $y_i \sim \text{Multinomial}(\theta_j)$.

Our goal is to sample from the posterior distribution over antecedent lists:

$$p(d|\mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta) \propto p(\mathbf{y}|\mathbf{x}, d, \alpha)p(d|\mathcal{A}, \lambda, \eta).$$

Given d, we can compute the posterior consequent distributions required to construct a Bayesian decision list as in Section 2.1. There are three prior hyperparameters that must be specified by the user: α, λ, and η. We will see in Sections 2.4 and 2.5 that these hyperparameters have natural interpretations that suggest the values to which they should be set.

2.4 The hierarchical prior for antecedent lists

Suppose the list of antecedents d has length m and antecedent cardinalities $c_1, \ldots, c_m$. The prior probability of d is defined hierarchically as

$$p(d|\mathcal{A}, \lambda, \eta) = p(m|\mathcal{A}, \lambda) \prod_{j=1}^{m} p(c_j|c_1, \ldots, c_{j-1}, \mathcal{A}, \eta)p(a_j|a_1, \ldots, a_{j-1}, c_j, \mathcal{A}).$$

We take the distributions for list length m and antecedent cardinality c_j to be Poisson with parameters λ and η respectively, with proper truncation to account for the finite number of antecedents in $\mathcal{A}$. Specifically, the distribution of m is Poisson truncated at the total number of pre-selected antecedents:

$$p(m|\mathcal{A}, \lambda) = \frac{(\lambda^m/m!)}{\sum_{j=0}^{R}(\lambda^j/j!)}, \quad m = 0, \ldots, R.$$

This truncated Poisson is a proper prior, and is natural choice because of its simple parameterization. Specifically, this prior has the desirable property that when R is large compared

to the desired size of the decision list, as will generally be the case when seeking an interpretable decision list, the prior expected decision list length $\mathbb{E}[m|\mathcal{A}, \lambda]$ is approximately equal to λ. The prior hyperparameter λ can then naturally be set to the prior belief of the list length required to model the data. A Poisson distribution is used in a similar way in the hierarchical prior of Wu et al. [2007].

The distribution of c_j must be truncated at zero and at the maximum antecedent cardinality C. Additionally, any cardinalities that have been exhausted by point j in the decision list sampling must be excluded. Let $R_j(c_1, \ldots, c_j, \mathcal{A})$ be the set of antecedent cardinalities that are available after drawing antecedent j. For example, if $\mathcal{A}$ contains antecedents of size 1, 2, and 4, then we begin with $R_0(\mathcal{A}) = \{1, 2, 4\}$. If $\mathcal{A}$ contains only 2 rules of size 4 and $c_1 = c_2 = 4$, then $R_2(c_1, c_2, \mathcal{A}) = \{1, 2\}$ as antecedents of size 4 have been exhausted. We now take $p(c_j|c_1, \ldots, c_{j-1}, \mathcal{A}, \eta)$ as Poisson truncated to remove values for which there are no rules available with that cardinality:

$$p(c_j|c_1, \ldots, c_{j-1}, \mathcal{A}, \eta) = \frac{(\eta^{c_j}/c_j!)}{\sum_{k \in R_{j-1}(c_1,\ldots,c_{j-1},\mathcal{A})}(\eta^k/k!)}, \quad c_j \in R_{j-1}(c_1, \ldots, c_{j-1}, \mathcal{A}).$$

If the number of rules of different sizes is large compared to λ, and η is small compared to C, the prior expected average antecedent cardinality is close to η. Thus η can naturally be set to the prior belief of the antecedent cardinality required to model the data.

Once the antecedent cardinality c_j has been selected, the antecedent a_j must be sampled from all available antecedents in $\mathcal{A}$ of size c_j. If there were a preference towards specific antecedents, *e.g.* antecedents that are particularly interpretable, this preference could be expressed in $p(a_j|a_1, \ldots, a_{j-1}, c_j, \mathcal{A})$. Here, we use a uniform distribution over antecedents in $\mathcal{A}$ of size c_j, excluding those in $\{a_1, \ldots, a_{j-1}\}$. Let $Q_k(\mathcal{A}, a_1, \ldots, a_{j-1}) = \{a \in \mathcal{A} \setminus \{a_1, \ldots, a_{j-1}\} : |a| = k\}$. Then,

$$p(a_j|a_1, \ldots, a_{j-1}, c_j, \mathcal{A}) = \frac{1}{|Q_{c_j}(\mathcal{A}, a_1, \ldots, a_{j-1})|}, \quad a_j \in Q_{c_j}(\mathcal{A}, a_1, \ldots, a_{j-1}).$$

It is straightforward to sample an ordered antecedent list d from the prior by following the generative model, using the provided distributions.

2.5 The likelihood function

The likelihood function follows directly from the generative model. Let $\theta = (\theta_0, \theta_1, \ldots, \theta_m)$ be the consequent parameters corresponding to each antecedent in d, together with the default rule parameter θ_0. Then,

$$p(\mathbf{y}|\mathbf{x}, d, \theta) = \prod_{\substack{j=0,\ldots,m,\\ \sum_l N_{j,l}>0}} \text{Multinomial}(\mathbf{N}_j|\theta_j),$$

with

$$\theta_j \sim \text{Dirichlet}(\alpha).$$

We can marginalize over θ_j in each Multinomial distribution in the above product, obtaining, through the standard derivation of the Dirichlet-Multinomial distribution,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, d, \alpha) &= \prod_{\substack{j=0,\ldots,m,\\ \sum_l N_{j,l}>0}} \frac{\Gamma(\sum_{l=1}^{L} \alpha_l)}{\Gamma(\sum_{l=1}^{L} N_{j,l} + \alpha_l)} \times \prod_{l=1}^{L} \frac{\Gamma(N_{j,l} + \alpha_l)}{\Gamma(\alpha_l)} \\ &\propto \prod_{\substack{j=0,\ldots,m,\\ \sum_l N_{j,l}>0}} \frac{\prod_{l=1}^{L} \Gamma(N_{j,l} + \alpha_l)}{\Gamma(\sum_{l=1}^{L} N_{j,l} + \alpha_l)}. \end{aligned}$$

The prior hyperparameter α has the usual Bayesian interpretation of pseudocounts. In our experiments, we set $\alpha_l = 1$ for all l, producing a uniform prior.

2.6 Markov chain Monte Carlo sampling

We do Metropolis-Hastings sampling of d, generating the proposed d^* from the current d^t using one of three options: 1) Move an antecedent in d^t to a different position in the list. 2) Add an antecedent from $\mathcal{A}$ that is not currently in d^t into the list. 3) Remove an antecedent from d^t. Which antecedents to adjust and their new positions are chosen uniformly at random at each step. The option to move, add, or remove is also chosen uniformly. The probabilities for the proposal distribution $Q(d^*|d^t)$ depend on the size of the antecedent list, the number of pre-mined antecedents, and whether the proposal is a move, addition, or removal. For the uniform distribution that we used, the proposal probabilities for a d^* produced by one of the three proposal types is:

$$Q(d^*|d^t, \mathcal{A}) = \begin{cases} \frac{1}{(|d^t|)(|d^t|-1)} & \text{if move proposal,} \\ \frac{1}{(|\mathcal{A}|-|d^t|)(|d^t|+1)} & \text{if add proposal,} \\ \frac{1}{|d^t|} & \text{if remove proposal.} \end{cases}$$

To explain these probabilities, if there is a move proposal, we consider the number of possible antecedents to move and the number of possible positions to move to; if there is an add proposal, we consider the number of possible antecedents to add to the list and the number of positions to place a new antecedent; for remove proposals we consider the number of possible antecedents to remove. This sampling algorithm is related to those used for Bayesian Decision Tree models [Chipman et al., 2002, 1998, Wu et al., 2007] and to methods for exploring tree spaces [Madigan et al., 2011].

For every MCMC run, we ran 3 chains, each initialized independently from a random sample from the prior. We discarded the first half of simulations as burn-in, and then assessed chain convergence using the Gelman-Rubin convergence diagnostic applied to the log posterior density [Gelman and Rubin, 1992]. We considered chains to have converged when the diagnostic $\hat{R} < 1.05$.

2.7 The posterior predictive distribution and point estimates

Given the posterior $p(d|\mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta)$, we consider estimating the label y of a new observation x using either a point estimate (a single interpretable model) or the posterior predictive

distribution. Given a point estimate of the antecedent list d, we have that

$$\begin{aligned} p(y = l|x, d, \mathbf{x}, \mathbf{y}, \alpha) &= \int_\theta \theta_l p(\theta|x, d, \mathbf{x}, \mathbf{y}, \alpha) d\theta \\ &= \mathbb{E}[\theta_l|x, d, \mathbf{x}, \mathbf{y}, \alpha]. \end{aligned}$$

Let $j(d, x)$ be the index of the first antecedent in d that applies to x. The posterior consequent distribution is

$$\theta|x, d, \mathbf{x}, \mathbf{y}, \alpha \sim \text{Dirichlet}\left(\alpha + \mathbf{N}_{j(d,x)}\right). \tag{1}$$

Thus,

$$p(y = l|x, d, \mathbf{x}, \mathbf{y}, \alpha) = \frac{\alpha_l + N_{j(d,x),l}}{\sum_{k=1}^{L}\left(\alpha_k + N_{j(d,x),k}\right)}.$$

Additionally, (1) allows for the estimation of 95% credible intervals using the Dirichlet distribution function..

The posterior mean is often a good choice for a point estimate, however since the posterior is a distribution over antecedent lists there is not a clear notion of its mean. We thus look for an antecedent list whose statistics are similar to the posterior mean statistics. Specifically, we are interested in finding a point estimate $\hat{d}$ whose length m and whose average antecedent cardinality $\bar{c} = \frac{1}{m}\sum_{j=1}^{m} c_j$ are close to the posterior mean list length and average cardinality. Let $\bar{m}$ be the posterior mean decision list length and $\bar{\bar{c}}$ the posterior mean average antecedent cardinality, as estimated from the MCMC samples. Then, we choose our point estimate $\hat{d}$ as the list with the highest posterior probability among all samples with $m \in \{\lfloor \bar{m} \rfloor, \lceil \bar{m} \rceil\}$ and $\bar{c} \in [\lfloor \bar{\bar{c}} \rfloor, \lceil \bar{\bar{c}} \rceil]$. We call this point estimate *BRL-point*.

Alternatively, we can use the entire posterior $p(d|\mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta)$ to estimate y. The posterior predictive distribution for y is

$$\begin{aligned} p(y = l|x, \mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta) &= \sum_{d \in \mathbf{D}} p(y = l|x, d, \mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha) p(d|\mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta) \\ &= \sum_{d \in \mathbf{D}} \frac{\alpha_l + N_{j(d,x),l}}{\sum_{k=1}^{L}\left(\alpha_k + N_{j(d,x),k}\right)} p(d|\mathbf{x}, \mathbf{y}, \mathcal{A}, \alpha, \lambda, \eta) \end{aligned}$$

where $\mathbf{D}$ is the set of all ordered subsets of $\mathcal{A}$. The posterior samples obtained by MCMC simulation can be used to approximate this sum. We call the classifier that uses the full collection of posterior samples *BRL-post*. Using the entire posterior distribution to make a prediction means the classifier is no longer interpretable. One could, however, use the posterior predictive distribution to classify, and then provide several point estimates from the posterior to the user as example explanations for the prediction.

3 Simulation studies

We use simulation studies and a deterministic dataset to show that when data are generated by a decision list model, the BRL method is able to recover the true decision list.

3.1 Simulated data sets

Given observations with arbitrary features, and a collection of rules on those features, we can construct a binary matrix where the rows represent observations and the columns represent

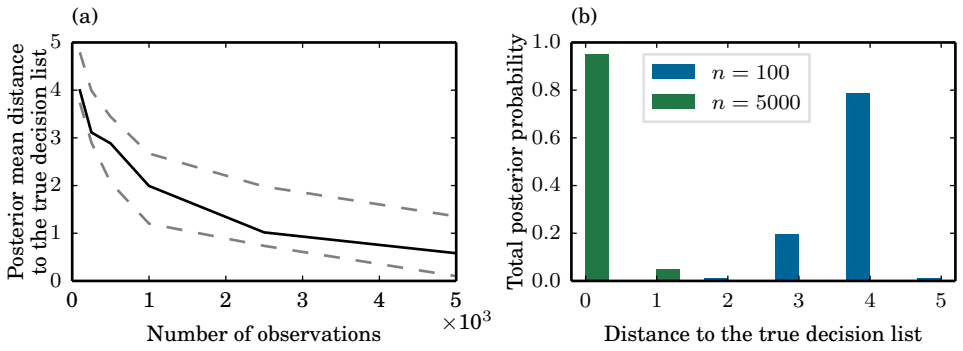

Figure 2: (a) Average Levenshtein distance from posterior samples to the true decision list, for differing numbers of observations. The black solid line indicates the median value across the 100 simulated datasets of each size, and the gray dashed lines indicate the first and third quartiles. (b) The proportion of posterior samples with the specified distance to the true decision list, for a randomly selected simulation with $n = 100$ observations and a randomly selected simulation with $n = 5000$.

rules, and the entry is 1 if the rule applies to that observation and 0 otherwise. We need only simulate this binary matrix to represent the observations without losing generality. For our simulations, we generated independent binary rule sets with 100 rules by setting each feature value to 1 independently with probability 1/2.

We generated a random decision list of size 5 by selecting 5 rules at random, and adding the default rule. Each rule in the decision list was assigned a consequent distribution over labels using a random draw from the Beta$(1/2, 1/2)$ distribution, which ensures that the rules are informative about labels. Labels were then assigned to each observation using the decision list: For each observation, the label was taken as a draw from the label distribution corresponding to the first rule that applied to that observation.

For each number of observations $N \in \{100, 250, 500, 1000, 2500, 5000\}$, we generated 100 independent data sets $(\mathbf{x}, \mathbf{y})$, for a total of 600 simulated datasets. We did MCMC sampling with three chains as described in Section 2 for each dataset. For all datasets, 20,000 samples were sufficient for the chains to converge.

To appropriately visualize the posterior distribution, we binned the posterior antecedent lists according to their distance from the true antecedent list, using the Levenshtein string edit distance [Levenshtein, 1966] to measure the distance between two antecedent lists. This metric measures the minimum number of antecedent substitutions, additions, or removals to transform one decision list into the other. The results of the simulations are given in Fig. 2. Figure 2(a) shows that as the number of observations increases, the posterior mass concentrates on the true decision list. Figure 2(b) illustrates this concentration with two choices of the distribution of posterior distances to the true decision list, for n small and for n large.

3.2 A deterministic problem

We fit BRL to the Tic-Tac-Toe Endgame dataset from the UCI Machine Learning Repository [Bache and Lichman, 2013] of benchmark datasets. The Tic-Tac-Toe Endgame dataset provides all possible end board configurations for the game Tic-Tac-Toe, with the task of

Table 1: Mean classification accuracy in the top row, with standard deviation in the second row, for machine learning algorithms using 5 folds of cross-validation on the Tic-Tac-Toe Endgame dataset.

	BRL	C5.0	CART	ℓ_1-LR	SVM	RF	BCART
Mean accuracy	1.00	0.94	0.90	0.98	0.99	0.99	0.71
Standard deviation	0.00	0.01	0.04	0.01	0.01	0.01	0.04

determining if player "X" won or not. The dataset is deterministic, and there are exactly 8 ways that player "X" can win, which are the 8 ways of getting 3 "X"'s in a row on a 3x3 grid. We split the dataset into 5 folds and did cross-validation to estimate test accuracy. For each fold of cross-validation, we fit BRL with prior hyperparameters $\lambda = 8$ and $\eta = 3$, and the point estimate decision list contained the 8 ways to win and thus achieved perfect accuracy. In Table 1, we compare accuracy on the test set with C5.0, CART, ℓ_1-regularized logistic regression (ℓ_1-LR), RBF kernel support vector machines (SVM), random forests (RF), and Bayesian CART (BCART). The implementation details for these comparison algorithms are in the appendix. None of these other methods were able to achieve perfect accuracy.

4 Stroke prediction

We used Bayesian Rule Lists to derive a stroke prediction model using the MarketScan Medicaid Multi-State Database (MDCD). MDCD contains administrative claims data for 11.1 million Medicaid enrollees from multiple states. This database forms part of the suite of databases from the Innovation in Medical Evidence Development and Surveillance (IMEDS, `http://imeds.reaganudall.org/`) program that have been mapped to a common data model [Stang et al., 2010].

We extracted every patient in the MDCD database with a diagnosis of atrial fibrillation, one year of observation time prior to the diagnosis, and one year of observation time following the diagnosis (n=12,586). Of these, 1,786 (14%) had a stroke within a year of the atrial fibrillation diagnosis.

As candidate predictors, we considered all drugs and all conditions. Specifically, for every drug and condition, we created a binary predictor variable indicating the presence or absence of the drug or condition in the longitudinal record prior to the atrial fibrillation diagnosis. These totaled 4,146 unique medications and conditions. We included features for age and gender. Specifically, we used the natural values of 50, 60, 70, and 80 years of age as split points, and for each split point introduced a pair of binary variables indicating if age was less than or greater than the split point. Considering both patients and features, here we apply our method to a dataset that is over 5000 times larger than that originally used to develop the CHADS_2 score (which had n=1,733 and considered 5 features).

We did five folds of cross-validation. For each fold, we pre-mined the collection of possible antecedents using frequent itemset mining with a minimum support threshold of 10% and a maximum cardinality of 2. The total number of antecedents used ranged from 2162 to 2240 across the folds. We set the antecedent list prior hyperparameters λ and η to 3 and 1 respectively, to obtain a Bayesian decision list of similar complexity to the CHADS_2 score. For each fold, we evaluated the performance of the BRL point estimate by constructing a receiver operating characteristic (ROC) curve and measuring area under the curve (AUC) for each fold.

if hemiplegia **and** age>60 **then** *stroke risk* 58.9% (53.8% - 63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8% - 50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5% - 28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2% - 19.6%)
else if altered state of consciousness **and** age>60 **then** *stroke risk* 16.0% (12.2% - 20.2%)
else if age≤70 **then** *stroke risk* 4.6% (3.9% - 5.4%)
else *stroke risk* 8.7% (7.9% - 9.6%)

Figure 3: Decision list for determining 1-year stroke risk following diagnosis of atrial fibrillation from patient medical history. The risk given is the mean of the posterior consequent distribution, and in parentheses is the 95% credible interval.

In Fig. 3 we show the BRL point estimate recovered from one of the folds. The list indicates that past history of stroke reveals a lot about the vulnerability toward future stroke. In particular, the first half of the decision list focuses on a history of stroke, in order of severity. Hemiplegia, the paralysis of an entire side of the body, is often a result of a severe stroke or brain injury. Cerebrovascular disorder indicates a prior stroke, and transient ischaemic attacks are generally referred to as "mini-strokes." The second half of the decision list includes age factors and vascular disease, which are known risk factors and are included in the CHA_2DS_2-VASc score. The BRL-point lists that we obtained in the 5 folds of cross-validation were all of length 7, a similar complexity to the $CHADS_2$ and CHA_2DS_2-VASc scores which use 5 and 8 features respectively.

The point estimate lists for all five of the folds of cross-validation are given in the supplemental material. There is significant overlap in the antecedents in the point estimates across the folds. This suggests that the model may be more stable than decision trees, which are notorious for producing entirely different models after small changes to the training set [Breiman, 1996b,a].

In Fig. 4 we give ROC curves for all 5 folds for BRL-point, $CHADS_2$, and CHA_2DS_2-VASc, and in Table 2 we report mean AUC across the folds. These results show that with complexity and interpretability similar to $CHADS_2$, the BRL point estimate decision lists performed significantly better at stroke prediction than both $CHADS_2$ and CHA_2DS_2-VASc. Interestingly, we also found that $CHADS_2$ outperformed CHA_2DS_2-VASc despite CHA_2DS_2-VASc being an extension of $CHADS_2$. This is likely because the model for the CHA_2DS_2-VASc score, in which risk factors are added linearly, is a poor model of actual stroke risk. For instance, the stroke risk percentages calibrated to the CHA_2DS_2-VASc scores are not a monotonic function of score: The stroke risk with a CHA_2DS_2-VASc score of 7 is 9.6%, whereas a score of 8 corresponds to a stroke risk of 6.7% [Lip et al., 2010a]. The fact that more stroke risk factors can correspond to a lower stroke risk suggests that the CHA_2DS_2-VASc model may be misspecified, and highlights the difficulty in constructing these interpretable models manually.

The results in Table 2 give the AUC for BRL, $CHADS_2$, CHA_2DS_2-VASc, along with the same collection of machine learning algorithms used in Section 3.2. The decision tree algorithms CART and C5.0, the only other interpretable classifiers, were outperformed even by $CHADS_2$. The BRL-point performance was comparable to that of SVM, and not

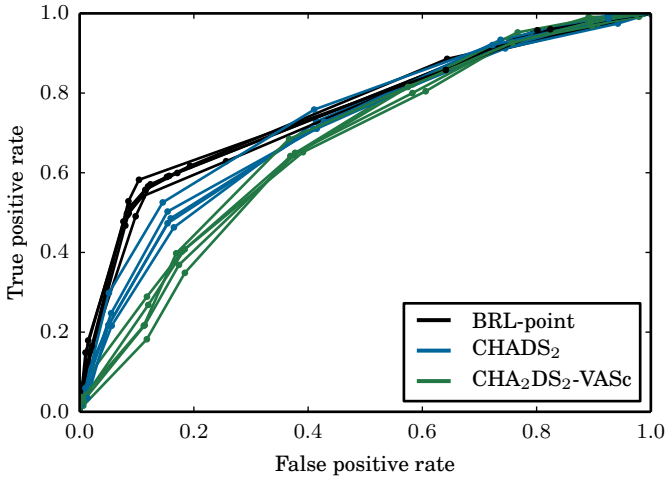

Figure 4: ROC curves for stroke prediction on the MDCD database for each of 5 folds of cross-validation, for the BRL point estimate, $CHADS_2$, and CHA_2DS_2-VASc.

Table 2: Mean, and in parentheses standard deviation, of AUC and training time across 5 folds of cross-validation for stroke prediction. Note that the $CHADS_2$ and CHA_2DS_2-VASc models are fixed, so no training time is reported.

	AUC	Training time (mins)
BRL-point	0.756 (0.007)	21.48 (6.78)
$CHADS_2$	0.721 (0.014)	no training
CHA_2DS_2-VASc	0.677 (0.007)	no training
CART	0.704 (0.010)	12.62 (0.09)
C5.0	0.704 (0.011)	2.56 (0.27)
ℓ_1 logistic regression	0.767 (0.010)	0.05 (0.00)
SVM	0.753 (0.014)	302.89 (8.28)
Random forests	0.774 (0.013)	698.56 (59.66)
BRL-post	0.775 (0.015)	21.48 (6.78)

substantially worse than ℓ_1 logistic regression and random forests. Using the full posterior, BRL-post matched random forests for the best performing method.

All of the methods were applied to the data on the same, single Amazon Web Services virtual core with a processor speed of approximately 2.5Ghz and 4GB of memory. Bayesian CART was unable to fit the data since it ran out of memory, and so it is not included in Table 2.

The BRL MCMC chains were simulated until convergence, which required 50,000 iterations for 4 of the 5 folds, and 100,000 for the fifth. The three chains for each fold were simulated in serial, and the total CPU time required per fold is given in Table 2, together with the CPU times required for training the comparison algorithms on the same processor. Table 2 shows that the BRL MCMC simulation was more than ten times faster than training SVM, and more than thirty times faster than training random forests.

Table 3: Mean, and in parentheses standard deviation, of AUC and training time (mins) across 5 folds of cross-validation for stroke prediction

	Female patients		Male patients	
	AUC	Training time	AUC	Training time
BRL-point	0.747 (0.028)	9.12 (4.70)	0.738 (0.027)	6.25 (3.70)
$CHADS_2$	0.717 (0.018)	no training	0.730 (0.035)	no training
CHA_2DS_2-VASc	0.671 (0.021)	no training	0.701 (0.030)	no training
CART	0.704 (0.024)	7.41 (0.14)	0.581 (0.111)	2.69 (0.04)
C5.0	0.707 (0.023)	1.30 (0.09)	0.539 (0.086)	0.55 (0.01)
ℓ_1 logistic regression	0.755 (0.025)	0.04 (0.00)	0.739 (0.036)	0.01 (0.00)
SVM	0.739 (0.021)	56.00 (0.73)	0.753 (0.035)	11.05 (0.18)
Random forests	0.764 (0.022)	389.28 (33.07)	0.773 (0.029)	116.98 (12.12)
BRL-post	0.765 (0.025)	9.12 (4.70)	0.778 (0.018)	6.25 (3.70)

4.1 Additional experiments

We further investigated the properties and performance of the BRL by applying it to two subsets of the data, female patients only and male patients only. The female dataset contained 8368 observations, and the number of pre-mined antecedents in each of 5 folds ranged from 1982 to 2197. The male dataset contained 4218 observations, and the number of pre-mined antecedents in each of 5 folds ranged from 1629 to 1709. BRL MCMC simulations and comparison algorithm training were done on the same processor as the full experiment. The AUC and training time across five folds for each of the datasets is given in Table 3.

The BRL point estimate again outperformed the other interpretable models ($CHADS_2$, CHA_2DS_2-VASc, CART, and C5.0), and the BRL-post performance matched that of random forests for the best performing method. As before, BRL MCMC simulation required significantly less time than SVM or random forests training. Point estimate lists for these additional experiments are given in the supplemental materials.

5 Related Work

Most widely used medical scoring systems are designed to be interpretable, but are not necessarily optimized for accuracy, and generally are derived from few factors. The Thrombolysis In Myocardial Infarction (TIMI) Score [Antman et al., 2000], Apache II score for infant mortality in the ICU [Knaus et al., 1985], the CURB-65 score for predicting mortality in community-acquired pneumonia [Lim et al., 2003], and the $CHADS_2$ score [Gage et al., 2001] are examples of interpretable predictive models that are very widely used. Each of these scoring systems involves very few calculations, and could be computed by hand during a doctor's visit. In the construction of each of these models, heuristics were used to design the features and coefficients for the model; none of these models was fully learned from data.

In contrast with these hand-designed interpretable medical scoring systems, recent advances in the collection and storing of medical data present unprecedented opportunities to develop powerful models that can predict a wide variety of outcomes [Shmueli, 2010]. The front-end user interface of risk assessment tools are increasingly available online (e.g., `http://www.r-calc.com`). At the end of the assessment, a patient may be told he or she has a high risk for a particular outcome but without understanding why the risk is high or

what steps can be taken to reduce risk.

In general, humans can handle only a handful of cognitive entities at once [Miller, 1956, Jennings et al., 1982]. It has long since been hypothesized that simple models predict well, both in the machine learning literature [Holte, 1993], and in the psychology literature [Dawes, 1979]. The related concepts of explanation and comprehensibility in statistical modeling have been explored in many past works [Bratko, 1997, Madigan et al., 1997, Giraud-Carrier, 1998, Rüping, 2006, Nowozin et al., 2007, Huysmans et al., 2011, Vellido et al., 2012, Freitas, 2014, for example].

Decision lists have the same form as models used in the expert systems literature from the 1970's and 1980's [Leondes, 2002], which were among the first successful types of artificial intelligence. The knowledge base of an expert system is composed of natural language statements that are *if... then...* rules. Decision lists are a type of associative classifier, meaning that the list is formed from association rules. In the past, associative classifiers have been constructed from heuristic greedy sorting mechanisms [Rivest, 1987, Liu et al., 1998, Li et al., 2001, Yin and Han, 2003, Marchand and Sokolova, 2005, Yi and Hüllermeier, 2005, Rudin et al., 2013]. Some of these sorting mechanisms work provably well in special cases, for instance when the decision problem is easy and the classes are easy to separate, but are not optimized to handle more general problems. Sometimes associative classifiers are formed by averaging several rules together, but the resulting classifier is not generally interpretable [Friedman and Popescu, 2008, Meinshausen, 2010]. Chang [2012] orders rules using discrete optimization.

Decision trees are closely related to decision lists, and are in some sense equivalent: any decision tree can be expressed as a decision list, and any decision list is a one-sided decision tree. Decision trees are almost always constructed greedily from the top down, and then pruned heuristically upwards and cross-validated to ensure accuracy. Because the trees are not fully optimized, if the top of the decision tree happened to have been chosen badly at the start of the procedure, it could cause problems with both accuracy and interpretability. Bayesian decision trees [Chipman et al., 1998, Dension et al., 1998, Chipman et al., 2002] use Markov chain Monte Carlo (MCMC) to sample from a posterior distribution over trees. Since they were first proposed, several improvements and extensions have been made in both sampling methods and model structure [Wu et al., 2007, Chipman et al., 2010, Taddy et al., 2011]. The space of decision lists using pre-mined rules is significantly smaller than the space of decision trees, making it easier to obtain MCMC convergence. Moreover, rule mining allows for the rules to be individually powerful.

This work is related to the Hierarchical Association Rule Model (HARM), a Bayesian model that uses rules [McCormick et al., 2012]. HARM estimates the conditional probabilities of each rule jointly in a conservative way. Each rule acts as a separate predictive model, so HARM does not explicitly aim to learn an ordering of rules.

A theoretical work [Rudin et al., 2013] by the same authors provides guarantees on prediction quality for decision lists using statistical learning theory.

6 Discussion and Conclusion

We are working under the hypothesis that many real datasets permit predictive models that can be surprisingly small. This was hypothesized over a decade ago [Holte, 1993], however, we now are starting to have the computational tools to truly test this hypothesis. The BRL method introduced in this work aims to hit the "sweet spot" between predictive accuracy,

interpretability, and tractability.

Interpretable models have the benefits of being both concise and convincing. A small set of trustworthy rules can be the key to communicating with domain experts and to allow machine learning algorithms to be more widely implemented and trusted. In practice, a preliminary interpretable model can help domain experts to troubleshoot the inner workings of a complex model, in order to make it more accurate and tailored to the domain. We demonstrated that interpretable models lend themselves to the domain of predictive medicine, but there are a wide variety of domains in science, engineering, and industry, where these models would be a natural choice.

Appendix

Comparison algorithm implementations

Support vector machines: LIBSVM [Chang and Lin, 2011] with a radial basis function kernel. We selected the slack parameter C_{SVM} and the kernel parameter γ using a grid search over the ranges $C_{\text{SVM}} \in \{2^{-2}, 2^0, \ldots, 2^6\}$ and $\gamma \in \{2^{-6}, 2^{-4}, \ldots, 2^2\}$. We chose the set of parameters with the best 3-fold cross-validation performance using LIBSVM's built-in cross-validation routine. *C5.0*: The R library "C50" with default settings. *CART*: The R library "rpart" with default parameters and pruned using the complexity parameter that minimized cross-validation error. *Logistic regression*: The LIBLINEAR [Fan et al., 2008] implementation of logistic regression with ℓ_1 regularization. We selected the regularization parameter C_{LR} from $\{2^{-6}, 2^{-4}, \ldots, 2^6\}$ as that with the best 3-fold cross-validation performance, using LIBLINEAR's built-in cross-validation routine. *Random forests*: The R library "randomForest." The optimal value for the parameter "mtry" was found using "tuneRF," with its default 50 trees. The optimal "mtry" was then used to fit a random forests model with 500 trees, the library default. *Bayesian CART*: The R library "tgp," function "bcart" with default settings.

Acknowledgement

Ben Letham and Cynthia Rudin were partly funded by NSF CAREER IIS-1053407 from the National Science Foundation to C. Rudin. Tyler McCormick's research was partially funded by a Google Faculty Award and NIAID grant R01 HD54511. David Madigan's research was partly funded by grant R01 GM87600-01 from the National Institutes for Health. The authors thank Zachary Shahn and the OMOP team for help with the data.

References

Elliott M. Antman, Marc Cohen, Peter J.L.M. Bernink, Carolyn H. McCabe, Thomas Horacek, Gary Papuchis, Branco Mautner, Ramon Corbalan, David Radley, and Eugene Braunwald. The TIMI risk score for unstable angina/non-ST elevation MI: a method for prognostication and therapeutic decision making. *The Journal of the American Medical Association*, 284(7):835–842, 2000.

K. Bache and M. Lichman. UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml.

Christian Borgelt. An implementation of the FP-growth algorithm. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, OSDM '05, pages 1–5, 2005.

I. Bratko. Machine learning: between accuracy and interpretability. In Giacomo Della Riccia, Hans-Joachim Lenz, and Rudolf Kruse, editors, *Learning, Networks and Statistics*, volume 382 of *International Centre for Mechanical Sciences*, pages 163–177. Springer Vienna, 1997. ISBN 978-3-211-82910-3. doi: 10.1007/978-3-7091-2668-4_10. URL http://dx.doi.org/10.1007/978-3-7091-2668-4_10.

Leo Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24:2350–2383, 1996a.

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996b.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

Allison Chang. *Integer Optimization Methods for Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2012.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian treed models. *Machine Learning*, 48(1/3):299–320, 2002.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Robyn M Dawes. The robust beauty of improper linear models in decision making. *American Psychologist*, 34(7):571–582, 1979.

D Dension, B Mallick, and A.F.M. Smith. A Bayesian CART algorithm. *Biometrika*, 85(2): 363–377, 1998.

James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, ICML '95, pages 194–202, 1995.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 1993 International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI '93*, pages 1022–1027, 1993.

Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

Brian F. Gage, Amy D. Waterman, William Shannon, Michael Boechler, Michael W. Rich, and Martha J. Radford. Validation of clinical classification schemes for predicting stroke. *Journal of the American Medical Association*, 285(22):2864–2870, 2001.

Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, November 1992.

Christophe Giraud-Carrier. Beyond predictive accuracy: what? In *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, pages 78–85, 1998.

Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–91, 1993.

Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

Dennis L. Jennings, Teresa M. Amabile, and Lee Ross. Informal covariation assessments: data-based versus theory-based judgements. In Daniel Kahneman, Paul Slovic, and Amos Tversky, editors, *Judgment Under Uncertainty: Heuristics and Biases,*, pages 211–230. Cambridge Press, Cambridge, MA, 1982.

William A. Knaus, Ellizabeth A. Draper, Douglas P. Wagner, and Jack E. Zimmerman. APACHE II: a severity of disease classification system. *Critical Care Medicine*, 13:818–829, 1985.

Cornelius T. Leondes. *Expert systems: the technology of knowledge management and decision making for the 21st century.* Academic Press, 2002.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966.

Wenmin Li, Jiawei Han, and Jian Pei. CMAR: accurate and efficient classification based on multiple class-association rules. *IEEE International Conference on Data Mining*, pages 369–376, 2001.

WS Lim, MM van der Eerden, R Laing, WG Boersma, N Karalus, GI Town, SA Lewis, and JT Macfarlane. Defining community acquired pneumonia severity on presentation to hospital: an international derivation and validation study. *Thorax*, 58(5):377–382, 2003.

Gregory Y.H. Lip, Lars Frison, Jonathan L. Halperin, and Deirdre A. Lane. Identifying patients at high risk for stroke despite anticoagulation: a comparison of contemporary stroke risk stratification schemes in an anticoagulated atrial fibrillation cohort. *Stroke*, 41:2731–2738, 2010a.

GY Lip, R Nieuwlaat, R Pisters, DA Lane, and HJ Crijns. Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: the Euro heart survey on atrial fibrillation. *Chest*, 137:263–272, 2010b.

Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, KDD '98, pages 80–96, 1998.

D Madigan, K Mosurski, and RG Almond. Explanation in belief networks. *Journal of Computational and Graphical Statistics*, 6:160–181, 1997.

David Madigan, Sushil Mittal, and Fred Roberts. Efficient sequential decision making algorithms for container inspection operations. *Naval Research Logistics*, 58:637–654, 2011.

Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent features. *Journal of Machine Learning Research*, 6:427–451, 2005.

Tyler H. McCormick, Cynthia Rudin, and David Madigan. Bayesian hierarchical rule modeling for predicting medical conditions. *The Annals of Applied Statistics*, 6:652–668, 2012.

Nicolai Meinshausen. Node harvest. *The Annals of Applied Statistics*, 4(4):2049–2072, 2010.

George A. Miller. The magical number seven, plus or minus two: some limits to our capacity for processing information. *The Psychological Review*, 63(2):81–97, 1956.

Sebastian Nowozin, Koji Tsuda, Takeaki Uno, Taku Kudo, and Gokhan Bakir. Weighted substructure mining for image analysis. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '07, 2007.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

Cynthia Rudin, Benjamin Letham, and David Madigan. Learning theory analysis for association rules and sequential event prediction. *Journal of Machine Learning Research*, 14: 3384–3436, 2013.

Stefan Rüping. *Learning interpretable models*. PhD thesis, Universität Dortmund, 2006.

Galit Shmueli. To explain or to predict? *Statistical Science*, 25(3):289–310, August 2010. ISSN 0883-4237. URL http://dx.doi.org/10.1214/10-STS330.

Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 1–12, 1996.

PE Stang, PB Ryan, JA Racoosin, JM Overhage, AG Hartzema, C Reich, E Welebob, T Scarnecchia, and J Woodcock. Advancing the science for active surveillance: rationale and design for the observational medical outcomes partnership. *Annals of Internal Medicine*, 153:600–606, 2010.

Matthew A. Taddy, Robert B. Gramacy, and Nicholas G. Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493):109–123, 2011.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

Alfredo Vellido, José D. Martín-Guerrero, and Paulo J.G. Lisboa. Making machine learning models interpretable. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2012.

Xindong Wu, Chengqi Zhang, and Shichao Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, July 2004.

Yuhong Wu, Håkon Tjelmeland, and Mike West. Bayesian CART: prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, 16(1):44–66, 2007.

Yu Yi and Eyke Hüllermeier. Learning complexity-bounded rule-based classifiers by combining association analysis and genetic algorithms. In *Proceedings of the Joint 4th International Conference in Fuzzy Logic and Technology*, EUSFLAT '05, pages 47–52, 2005.

Xiaoxin Yin and Jiawei Han. Cpar: classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, ICDM '03, pages 331–335, 2003.

Supplement to "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model"

Benjamin Letham, Operations Research Center, Massachusetts Institute of Technology. bletham@mit.edu

Cynthia Rudin, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. rudin@mit.edu

Tyler H. McCormick, Department of Statistics, Department of Sociology, Center for Statistics and the Social Sciences, University of Washington. tylermc@u.washington.edu

David Madigan, Department of Statistics, Columbia University. madigan@stat.columbia.edu

This supplement provides the Bayesian Rule Lists (BRL) point estimates recovered from the five folds of cross-validation on the full stroke-prediction experiment in Figs 1-5. Figs 6 and 7 give BLR point estimates from the female-only and male-only experiments, respectively.

if hemiplegia **and** age>60 **then** *stroke risk* 58.9% (53.8% - 63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8% - 50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5% - 28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2% - 19.6%)
else if altered state of consciousness **and** age>60 **then** *stroke risk* 16.0% (12.2% - 20.2%)
else if age≤70 **then** *stroke risk* 4.6% (3.9% - 5.4%)
else *stroke risk* 8.7% (7.9% - 9.6%)

Figure 1: Decision list for determining 1-year stroke risk following diagnosis of atrial fibrillation from patient medical history. The risk given is the mean of the posterior consequent, and in parentheses is the 95% credible interval. Obtained from the first of five folds of cross-validation.

if hemiplegia **and** cerebrovascular disorder **then** *stroke risk* 64.7% (59.6% - 69.6%)
else if cerebrovascular disorder **then** *stroke risk* 44.5% (41.6% - 47.5%)
else if hemiplegia **then** *stroke risk* 32.7% (23.8% - 42.2%)
else if congestive cardiac failure **and** hydrocodone **then** *stroke risk* 9.9% (8.4% - 11.5%)
else if transient ischaemic attack **then** *stroke risk* 30.5% (25.1% - 36.2%)
else if age>70 **then** *stroke risk* 9.1% (8.3% - 10.0%)
else *stroke risk* 4.0% (3.3% - 4.8%)

Figure 2: Stroke prediction decision list obtained from the second fold of cross-validation.

if hemiplegia **and** cerebrovascular disorder **then** *stroke risk* 61.7% (56.5% - 66.9%)
else if cerebrovascular disorder **then** *stroke risk* 44.8% (41.8% - 47.8%)
else if transient ischaemic attack **then** *stroke risk* 26.1% (21.7% - 30.7%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.2% (11.8% - 18.9%)
else if hemiplegia **then** *stroke risk* 37.8% (27.7% - 48.5%)
else if age≤60 **then** *stroke risk* 3.5% (2.8% - 4.4%)
else *stroke risk* 8.1% (7.4% - 8.8%)

Figure 3: Stroke prediction decision list obtained from the third fold of cross-validation.

if hemiplegia **and** cerebrovascular disorder **then** *stroke risk* 61.3% (56.2% - 66.3%)
else if cerebrovascular disorder **then** *stroke risk* 44.5% (41.5% - 47.5%)
else if sodium chloride **and** chronic obstructive pulmonary disease **then** *stroke risk* 10.6% (8.3% - 13.1%)
else if transient ischaemic attack **then** *stroke risk* 27.6% (22.8% - 32.7%)
else if hemiplegia **then** *stroke risk* 41.6% (30.9% - 52.7%)
else if age≤60 **then** *stroke risk* 3.2% (2.4% - 4.1%)
else *stroke risk* 8.2% (7.5% - 8.9%)

Figure 4: Stroke prediction decision list obtained from the fourth fold of cross-validation.

if hemiplegia **and** cerebrovascular disorder **then** *stroke risk* 64.5% (59.3% - 69.5%)
else if cerebrovascular disorder **then** *stroke risk* 44.2% (41.2% - 47.2%)
else if chronic obstructive pulmonary disease **and** chest pain **then** *stroke risk* 8.4% (7.2% - 9.8%)
else if transient ischaemic attack **then** *stroke risk* 30.2% (24.8% - 35.8%)
else if age≤60 **then** *stroke risk* 3.1% (2.3% - 4.0%)
else if hemiplegia **then** *stroke risk* 44.6% (32.8% - 56.7%)
else *stroke risk* 8.9% (8.1% - 9.7%)

Figure 5: Stroke prediction decision list obtained from the fifth fold of cross-validation.

if hemiplegia **then** *stroke risk* 59.0% (53.4% - 64.6%)
else if cerebrovascular disorder **then** *stroke risk* 44.7% (41.2% - 48.3%)
else if hypovolaemia **and** chest pain **then** *stroke risk* 14.6% (11.6% - 17.9%)
else if transient ischaemic attack **then** *stroke risk* 29.9% (24.0% - 36.2%)
else if age$\leq$70 **then** *stroke risk* 4.5% (3.6% - 5.5%)
else *stroke risk* 9.0% (8.0% - 10.0%)

Figure 6: Stroke prediction decision list obtained from the first fold of cross-validation on the females-only dataset.

if hemiplegia **and** age>70 **then** *stroke risk* 57.6% (47.8% - 67.1%)
else if transient ischaemic attack **and** chest pain **then** *stroke risk* 39.1% (31.6% - 46.9%)
else if occlusion and stenosis of carotid artery without infarction **and** coronary artery arteriosclerosis **then** *stroke risk* 21.1% (14.9% - 28.0%)
else if cerebrovascular disorder **then** *stroke risk* 49.6% (43.8% - 55.5%)
else *stroke risk* 6.8% (5.8% - 7.7%)

Figure 7: Stroke prediction decision list obtained from the first fold of cross-validation on the males-only dataset.