The Annals of Applied Statistics
2015, Vol. 9, No. 3, 1350–1371
DOI: 10.1214/15-AOAS848

INTERPRETABLE CLASSIFIERS USING RULES AND BAYESIAN ANALYSIS: BUILDING A BETTER STROKE PREDICTION MODEL

By Benjamin Letham[*,1], Cynthia Rudin[*,1], Tyler H. McCormick[†,2] and David Madigan[‡,3]

Massachusetts Institute of Technology[*], University of Washington[†] and Columbia University[‡]

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 (e.g., *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

Received October 2013; revised April 2015.
[1]Supported in part by NSF CAREER Grant IIS-1053407 from the National Science Foundation to C. Rudin, and awards from Siemens and Wistron.
[2]Supported in part by a Google Faculty Award and NIAID Grant R01 HD54511.
[3]Supported in part by Grant R01 GM87600-01 from the National Institutes of Health.
Key words and phrases. Bayesian analysis, classification, interpretability.

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%)

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

each prediction that it makes. Figure 1 presents an example decision list that we created using the Titanic data set available in R. This data set 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 Figure 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, premined 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, Hsu and Ma (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 involving splitting and pruning, 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 premined 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; in a sense, this algorithm scales with the sparsity of the data set rather than the number of features. 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 the purpose of not only producing a more interpretable model, but also reducing 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 score was calibrated with stroke risks using a database of 1733 Medicare beneficiaries followed for, on average, about a year.

Our alternative to the $CHADS_2$ was constructed using 12,586 patients and 4148 factors. Because we are using statistical learning, we are able to consider significantly more features; this constitutes over 6000 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 (2001a)], and Bayesian CART [Denison, Mallick and Smith (1998), Chipman, George and McCulloch (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 $a \to b$ 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:

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

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

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

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 distribution*:

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

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

if a_1 **then** $y \sim \text{Multinomial}(\boldsymbol{\theta}_1)$, $\boldsymbol{\theta}_1 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{N}_1)$
else if a_2 **then** $y \sim \text{Multinomial}(\boldsymbol{\theta}_2)$, $\boldsymbol{\theta}_2 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{N}_2)$
$\vdots$
else if a_m **then** $y \sim \text{Multinomial}(\boldsymbol{\theta}_m)$, $\boldsymbol{\theta}_m \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{N}_m)$
else $y \sim \text{Multinomial}(\boldsymbol{\theta}_0)$, $\boldsymbol{\theta}_0 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{N}_0)$.

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 preselected 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. For binary or categorical features the particular choice of the itemset mining algorithm is unimportant, as the output is an exhaustive list of all itemsets satisfying the constraints. Other algorithms, such as Apriori or Eclat [Agrawal and Srikant (1994), Zaki (2000)], would return an identical set of antecedents as FP-Growth if given the same minimum support and maximum cardinality 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, Zhang and Zhang (2004)]. For data with continuous features, a variety of procedures exist for antecedent mining [Fayyad and Irani (1993), Dougherty, Kohavi and Sahami (1995), Srikant and Agrawal (1996)]. Alternatively, one can create categorical features using interpretable thresholds (e.g., ages 40–49, 50–59, etc.) or interpretable quantiles (e.g., quartiles)—we took this approach in our experiments.

We let $\mathcal{A}$ represent the complete, pre-mined collection of antecedents, and suppose that $\mathcal{A}$ contains $|\mathcal{A}|$ 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 $\mathcal{A}$. Define $a_{<j}$ as the antecedents before j in the rule list if there are any, *for example*, $a_{<3} = \{a_1, a_2\}$.

Similarly, let c_j be the cardinality of antecedent a_j, and $c_{<j}$ the cardinalities of the antecedents before j in the rule list. The generative model is then:

– 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 $c_j \sim p(c_j|c_{<j}, \mathcal{A}, \eta)$.
  Sample a_j of cardinality c_j from $p(a_j|a_{<j}, c_j, \mathcal{A})$.
  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:

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

Given d, we can compute the posterior consequent distributions required to construct a Bayesian decision list as in Section 2.1. Three prior hyperparameters must be specified by the user: $\boldsymbol{\alpha}$, 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

$$p(d|\mathcal{A}, \lambda, \eta) = p(m|\mathcal{A}, \lambda) \prod_{j=1}^{m} p(c_j|c_{<j}, \mathcal{A}, \eta) p(a_j|a_{<j}, c_j, \mathcal{A}). \tag{2.1}$$

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 preselected antecedents:

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

This truncated Poisson is a proper prior, and is a natural choice because of its simple parameterization. Specifically, this prior has the desirable property that when $|\mathcal{A}|$ 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 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, Tjelmeland and West (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 $p(c_j | c_{<j}, \mathcal{A}, \eta)$ as Poisson truncated to remove values for which no rules are available with that cardinality:

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

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 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. Here, we use a uniform distribution over antecedents in $\mathcal{A}$ of size c_j, excluding those in $a_{<j}$:

$$p(a_j | a_{<j}, c_j, \mathcal{A}) \propto 1, \qquad a_j \in \{a \in \mathcal{A} \setminus a_{<j} : |a| = c_j\}. \tag{2.2}$$

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 $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_m)$ be the consequent parameters corresponding to each antecedent in d, together with the default rule parameter θ_0. Then, the likelihood is the product of the multinomial probability mass functions for the observed label counts at each rule:

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

with

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

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

multinomial distribution,

$$p(\mathbf{y}|\mathbf{x}, d, \boldsymbol{\alpha}) = \prod_{j=0}^{m} \frac{\Gamma(\sum_{l=1}^{L} \alpha_l)}{\Gamma(\sum_{l=1}^{L} N_{j,l} + \alpha_l)} \times \prod_{l=1}^{L} \frac{\Gamma(N_{j,l} + \alpha_l)}{\Gamma(\alpha_l)}$$
$$\propto \prod_{j=0}^{m} \frac{\prod_{l=1}^{L} \Gamma(N_{j,l} + \alpha_l)}{\Gamma(\sum_{l=1}^{L} N_{j,l} + \alpha_l)}.$$

The prior hyperparameter $\boldsymbol{\alpha}$ has the usual Bayesian interpretation of pseudocounts. In our experiments, we set $\alpha_l = 1$ for all l, producing a uniform prior. Other approaches for setting prior hyperparameters such as empirical Bayes are also applicable.

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

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

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 for it; 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, George and McCulloch (1998, 2002), Wu, Tjelmeland and West (2007)] and to methods for exploring tree spaces [Madigan, Mittal and Roberts (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 $\tilde{y}$ of a new observation $\tilde{x}$ using either a point estimate (a single Bayesian decision list) or the posterior predictive distribution. Given a point estimate of the antecedent list d, we have that

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

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

$$\theta|\tilde{x}, d, \mathbf{x}, \mathbf{y}, \alpha \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{N}_{j(d,\tilde{x})}). \tag{2.3}$$

Thus,

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

Additionally, (2.3) 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, but the interpretation of "mean" here is not clear since the posterior is a distribution over antecedent lists. 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*.

Another possible point estimate is the decision list with the highest posterior probability—the maximum a posteriori estimate. Given two list lengths, there are many more possible lists of the longer length than of the shorter length, so prior probabilities in (2.1) are generally higher for shorter lists. The maximum a posteriori estimate might yield a list that is much shorter than the posterior mean decision list length, so we prefer the BRL-point.

In addition to point estimates, 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

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

where $\mathbf{D}$ is the set of all ordered subsets of $\mathcal{A}$. The posterior samples obtained by MCMC simulation, after burn-in, 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 data set to show that when data are generated by a decision list model, the BRL (Bayesian Rule Lists; see Section 1) 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 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 data sets. We did MCMC sampling with three chains as described in Section 2 for each data set. For all data sets, 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,

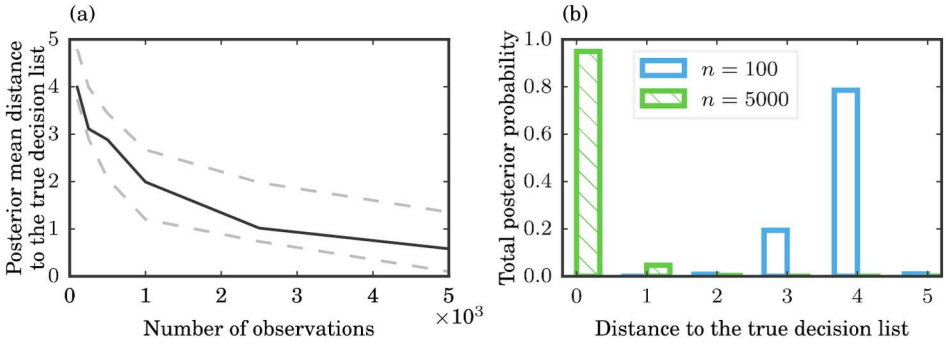

FIG. 2. (a) *Average Levenshtein distance from posterior samples to the true decision list, for differing numbers of observations. The black solid line indicates the median value across the* 100 *simulated data sets of each size, and the gray dashed lines indicate the first and third quartiles.* (b) *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* $n = 5000$.

using the Levenshtein string edit distance [Levenshtein (1965)] 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 Figure 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 data set from the UCI Machine Learning Repository [Bache and Lichman (2013)] of benchmark data sets. The Tic–Tac–Toe Endgame data set provides all possible end board configurations for the game Tic–Tac–Toe, with the task of determining if player "X" won or not. The data set is deterministic, with exactly 8 ways that player "X" can win, each one of the 8 ways to get 3 "X"'s in a row on a 3×3 grid. We split the data set 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 was able to achieve perfect

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 data set

	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

accuracy. Decision trees in particular are capable of providing a perfect classifier for this problem, but the greedy learning done by C5.0 and CART did not find the perfect classifier.

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, 1786 (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 full longitudinal record prior to the atrial fibrillation diagnosis. These totaled 4146 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 data set that is over 6000 times larger than that originally used to develop the CHADS_2 score (which had $n = 1733$ 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

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%)

FIG. 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.*

constructing a receiver operating characteristic (ROC) curve and measuring area under the curve (AUC) for each fold.

In Figure 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 [Letham et al. (2015)]. There is significant overlap in the antecedents in the point estimates across the folds. This suggests that the model may be more stable in practice than decision trees, which are notorious for producing entirely different models after small changes to the training set [Breiman (1996a, 1996b)].

In Figure 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,

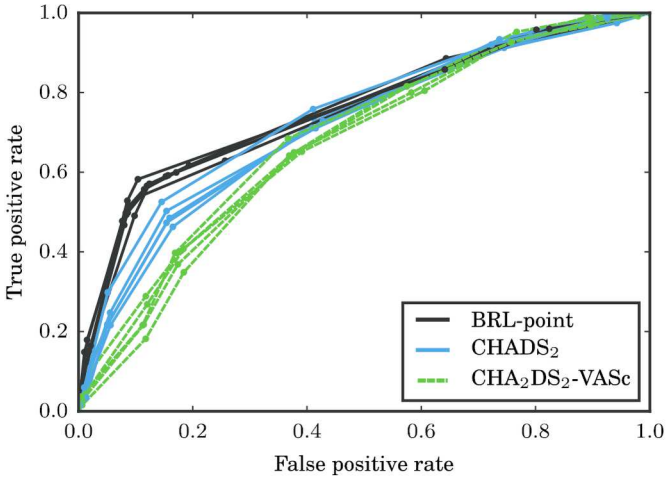

FIG. 4. *ROC curves for stroke prediction on the MDCD database for each of 5 folds of cross-validation, for the BRL point estimate, CHADS$_2$ and CHA$_2$DS$_2$-VASc.*

is a poor model of actual stroke risk. For instance, the stroke risks estimated by CHA$_2$DS$_2$-VASc are not a monotonic function of score. Within the original CHA$_2$DS$_2$-VASc calibration study, Lip et al. (2010a) estimate a stroke risk of 9.6% with a CHA$_2$DS$_2$-VASc score of 7, and a 6.7% risk with a score of 8. The indication that more stroke risk factors can correspond to a lower stroke risk suggests that the CHA$_2$DS$_2$-VASc model may be misspecified, and highlights the difficulty in constructing these interpretable models manually.

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$_2$DS$_2$-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$_2$DS$_2$-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)

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 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 2.5 GHz and 4 GB 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, using standard implementations of these methods as described in the Appendix.

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 data set contained 8368 observations, and the number of pre-mined antecedents in each of 5 folds ranged from 1982 to 2197. The male data set 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 data sets 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 [Letham et al. (2015)].

5. Related work and discussion. 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

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)

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 medical 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 predicted risk is high, particularly if many pieces of information were used to make the prediction.

In general, humans can handle only a handful of cognitive entities at once [Miller (1956), Jennings, Amabile and Ross (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, Mosurski and Almond (1997), Giraud-Carrier (1998), Rüping (2006), Huysmans et al. (2011), Vellido, Martín-Guerrero and Lisboa (2012), Freitas (2014), e.g.].

Decision lists have the same form as models used in the expert systems literature from the 1970s and 1980s [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, Hsu and Ma (1998), Marchand and Sokolova (2005), Rudin, Letham and Madigan (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, or having the rules each vote on the label and then combining the votes, but the resulting classifier is not generally interpretable [Li, Han and Pei (2001), Yin and Han (2003), Friedman and Popescu (2008), Meinshausen (2010)].

In a previous paper we proved that the VC dimension of decision list classifiers equals $|\mathcal{A}|$, the number of antecedents used to learn the model [Theorem 3, Rudin, Letham and Madigan (2013)]. This result leads to a uniform generalization bound for decision lists [Corollary 4, Rudin, Letham and Madigan (2013)]. This is the same as the VC dimension obtained by using the antecedents as features in a linear model, thus we have the same prediction guarantees. We then expect similar generalization behavior for decision lists and weighted linear combination models.

BRL interacts with the feature space only through the collection of antecedents $\mathcal{A}$. The computational effort scales with the number of antecedents, not the number of features, meaning there will generally be less computation when the data are sparse. This means that BRL tends to scale with the sparsity of the data rather than the number of features.

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 upward 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, George and McCulloch (1998, 2002), Denison, Mallick and Smith (1998)] 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, Tjelmeland and West (2007), Chipman, George and McCulloch (2010), Taddy, Gramacy and Polson (2011)]. The space of decision lists using pre-mined rules is significantly smaller than the space of decision trees, making it substantially easier to obtain MCMC convergence and to avoid the pitfalls of local optima. Moreover, rule mining allows for the rules to be individually powerful. Constructing a single decision tree is extremely fast, but sampling over the space of decision trees is extremely difficult (unless one is satisfied with local maxima). To

contrast this with our approach, the rule mining step is extremely fast, yet sampling over the space of decision lists is very practical.

There is a subfield of artificial intelligence, Inductive Logic Programming [Muggleton and De Raedt (1994)], whose goal is to mine individual conjunctive rules. It is possible to replace the frequent itemset miner with an inductive logic programming technique, but this generally leads to losses in predictive accuracy; ideally, we would use a large number of diverse rules as antecedents, rather than a few (highly overlapping) complex rules as would be produced by an ILP algorithm. In our experiments to a follow-up work [Wang and Rudin (2015)], the use of an ILP algorithm resulted in a substantial loss in performance.

Interpretable models are generally not unique (stable), in the sense that there may be many equally good models, and it is not clear in advance which one will be returned by the algorithm. For most problems, the space of high quality predictive models is fairly large [called the "Rashomon Effect" Breiman (2001b)], so we cannot expect uniqueness. In practice, as we showed, the rule lists across test folds were very similar, but if one desires stability to small perturbations in the data generally, we recommend using the full posterior rather than a point estimate. The fact that many high performing rule lists exist can be helpful, since it means the user has many choices of which model to use.

This work is related to the Hierarchical Association Rule Model (HARM), a Bayesian model that uses rules [McCormick, Rudin and Madigan (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.

There are related works on learning decision lists from an optimization perspective. In particular, the work of Rudin and Ertekin (2015) uses mixed-integer programming to build a rule list out of association rules, which has guarantees on optimality of the solution. Similarly to that work, Goh and Rudin (2014) fully learn sparse disjunctions of conjunctions using optimization methods.

There have been several follow-up works that directly extend and apply Bayesian Rule Lists. The work of Wang and Rudin (2015) on Falling Rule Lists provides a nontrivial extension to BRL whereby the probabilities for the rules are monotonically decreasing down the list. Wang et al. (2015) build disjunctions of conjunctive rules using a Bayesian framework similar to the one in this work. Zhang et al. (2015) have taken an interesting approach to constructing optimal treatment regimes using a BRL-like method, where, in addition to the criteria of accuracy, the rule list has a decision cost for evaluating it. It is possible to use BRL itself for that purpose as well, as one could give preference to particular antecedents that cost less. This sort of preference could be expressed in the antecedent prior distribution in (2.2).

King, Lam and Roberts (2014) have taken a Bayesian Rule List approach to handle a challenging problem in text analysis, which is to build a keyword-based classifier that is easier to understand in order to solicit high quality human input. Souillard-Mandar et al. (2015) applied Bayesian Rule Lists and Falling Rule Lists to the problem of screening for cognitive disorders such as Alzheimer's disease based on the digitized pen strokes of patients during the Clock Drawing test.

Shorter preliminary versions of this work are those of Letham et al. (2013, 2014). Letham et al. (2013) used a different prior and called the algorithm the Bayesian List Machine.

6. Conclusion. We are working under the hypothesis that many real data sets permit predictive models that can be surprisingly small. This was hypothesized over two decades 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 allowing 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, and there is a much wider 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 $C_{\mathrm{SVM}} \in \{2^{-2}, 2^0, \ldots, 2^6\}$ 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.

Acknowledgments. The authors thank Zachary Shahn and the OMOP team for help with the data.

SUPPLEMENTARY MATERIAL

Computer code (DOI: 10.1214/15-AOAS848SUPPA; .zip). Our Python code used to fit decision lists to data, along with an example data set.

BRL point estimates (DOI: 10.1214/15-AOAS848SUPPB; .pdf). The BRL point estimates for all of the cross-validation folds for the stroke prediction experiment, and BRL-point estimates for the female-only and male-only experiments.

REFERENCES

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *VLDB'94 Proceedings of the 20th International Conference on Very Large Databases* 487–499. Morgan Kaufmann, San Francisco, CA.

Antman, E. M., Cohen, M., Bernink, P. J. L. M., McCabe, C. H., Horacek, T., Papuchis, G., Mautner, B., Corbalan, R., Radley, D. and Braunwald, E. (2000). The TIMI risk score for unstable angina/non-ST elevation MI: A method for prognostication and therapeutic decision making. *JAMA* **284** 835–842.

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

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

Bratko, I. (1997). Machine learning: Between accuracy and interpretability. In *Learning, Networks and Statistics* (G. Della Riccia, H.-J. Lenz and R. Kruse, eds.). *International Centre for Mechanical Sciences* **382** 163–177. Springer, Vienna.

Breiman, L. (1996a). Bagging predictors. *Mach. Learn.* **24** 123–140.

Breiman, L. (1996b). Heuristics of instability and stabilization in model selection. *Ann. Statist.* **24** 2350–2383. MR1425957

Breiman, L. (2001a). Random forests. *Mach. Learn.* **45** 5–32.

Breiman, L. (2001b). Statistical modeling: The two cultures. *Statist. Sci.* **16** 199–231. MR1874152

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2** 27:1–27:27.

Chipman, H. A., George, E. I. and McCulloch, R. E. (1998). Bayesian CART model search. *J. Amer. Statist. Assoc.* **93** 935–948.

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

CHIPMAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2010). BART: Bayesian additive regression trees. *Ann. Appl. Stat.* **4** 266–298. MR2758172

DAWES, R. M. (1979). The robust beauty of improper linear models in decision making. *American Psychologist* **34** 571–582.

DENISON, D. G. T., MALLICK, B. K. and SMITH, A. F. M. (1998). A Bayesian CART algorithm. *Biometrika* **85** 363–377. MR1649118

DOUGHERTY, J., KOHAVI, R. and SAHAMI, M. (1995). Supervised and unsupervised discretization of continuous features. In *ICML'95 Proceedings of the 12th International Conference on Machine Learning* 194–202. Morgan Kaufmann, San Francisco, CA.

FAN, R.-E., CHANG, K.-W., HSIEH, C.-J., WANG, X.-R. and LIN, C.-J. (2008). LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **9** 1871–1874.

FAYYAD, U. M. and IRANI, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI'93 Proceedings of the 1993 International Joint Conference on Artificial Intelligence* 1022–1027. Morgan Kaufmann, San Francisco, CA.

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

FRIEDMAN, J. H. and POPESCU, B. E. (2008). Predictive learning via rule ensembles. *Ann. Appl. Stat.* **2** 916–954. MR2522175

GAGE, B. F., WATERMAN, A. D., SHANNON, W., BOECHLER, M., RICH, M. W. and RADFORD, M. J. (2001). Validation of clinical classification schemes for predicting stroke. *Journal of the American Medical Association* **285** 2864–2870.

GELMAN, A. and RUBIN, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statist. Sci.* **7** 457–472.

GIRAUD-CARRIER, C. (1998). Beyond predictive accuracy: What? Technical report, Univ. Bristol, Bristol, UK.

GOH, S. T. and RUDIN, C. (2014). Box drawings for learning with imbalanced data. In *KDD'14 Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 333–342. DOI:10.1145/2623330.2623648.

HOLTE, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.* **11** 63–91.

HUYSMANS, J., DEJAEGER, K., MUES, C., VANTHIENEN, J. and BAESENS, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* **51** 141–154.

JENNINGS, D. L., AMABILE, T. M. and ROSS, L. (1982). Informal covariation assessments: Data-based versus theory-based judgements. In *Judgment Under Uncertainty: Heuristics and Biases,* (D. KAHNEMAN, P. SLOVIC and A. TVERSKY, eds.) 211–230. Cambridge Univ. Press, Cambridge, MA.

KING, G., LAM, P. and ROBERTS, M. (2014). Computer-assisted keyword and document set discovery from unstructured text. Technical report, Harvard.

KNAUS, W. A., DRAPER, E. A., WAGNER, D. P. and ZIMMERMAN, J. E. (1985). APACHE II: A severity of disease classification system. *Critical Care Medicine* **13** 818–829.

LEONDES, C. T. (2002). *Expert Systems: The Technology of Knowledge Management and Decision Making for the 21st Century.* Academic Press, San Diego, CA.

LETHAM, B., RUDIN, C., MCCORMICK, T. H. and MADIGAN, D. (2013). An interpretable stroke prediction model using rules and Bayesian analysis. In *Proceedings of AAAI Late Breaking Track.* MIT, Cambridge, MA.

LETHAM, B., RUDIN, C., MCCORMICK, T. H. and MADIGAN, D. (2014). An interpretable model for stroke prediction using rules and Bayesian analysis. In *Proceedings of 2014 KDD Workshop on Data Science for Social Good.* MIT, Cambridge, MA.

Letham, B., Rudin, C., McCormick, T. H. and Madigan, D. (2015). Supplement to "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model." DOI:10.1214/15-AOAS848SUPPA, DOI:10.1214/15-AOAS848SUPPB.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Dokl.* **10** 707–710. MR0189928

Li, W., Han, J. and Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the IEEE International Conference on Data Mining* 369–376. IEEE, New York.

Lim, W. S., van der Eerden, M. M., Laing, R., Boersma, W. G., Karalus, N., Town, G. I., Lewis, S. A. and Macfarlane, J. T. (2003). Defining community acquired pneumonia severity on presentation to hospital: An international derivation and validation study. *Thorax* **58** 377–382.

Lip, G. Y. H., Frison, L., Halperin, J. L. and Lane, D. A. (2010a). 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.

Lip, G. Y. H., Nieuwlaat, R., Pisters, R., Lane, D. A. and Crijns, H. J. G. M. (2010b). 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.

Liu, B., Hsu, W. and Ma, Y. (1998). Integrating classification and association rule mining. In *KDD'98 Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* 80–96. AAAI Press, Palo Alto, CA.

Madigan, D., Mittal, S. and Roberts, F. (2011). Efficient sequential decision-making algorithms for container inspection operations. *Naval Res. Logist.* **58** 637–654. MR2842551

Madigan, D., Mosurski, K. and Almond, R. G. (1997). Explanation in belief networks. *J. Comput. Graph. Statist.* **6** 160–181.

Marchand, M. and Sokolova, M. (2005). Learning with decision lists of data-dependent features. *J. Mach. Learn. Res.* **6** 427–451. MR2249827

McCormick, T. H., Rudin, C. and Madigan, D. (2012). Bayesian hierarchical rule modeling for predicting medical conditions. *Ann. Appl. Stat.* **6** 622–668. MR2976486

Meinshausen, N. (2010). Node harvest. *Ann. Appl. Stat.* **4** 2049–2072. MR2829946

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

Muggleton, S. and De Raedt, L. (1994). Inductive logic programming: Theory and methods. *J. Logic Programming* **19** 629–679. MR1279936

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo.

Rivest, R. L. (1987). Learning decision lists. *Mach. Learn.* **2** 229–246.

Rudin, C. and Ertekin, Ş. (2015). Learning optimized lists of classification rules. Technical report, MIT, Cambridge, MA.

Rudin, C., Letham, B. and Madigan, D. (2013). Learning theory analysis for association rules and sequential event prediction. *J. Mach. Learn. Res.* **14** 3441–3492. MR3144468

Rüping, S. (2006). Learning interpretable models. Ph.D. thesis, Univ. Dortmund.

Shmueli, G. (2010). To explain or to predict? *Statist. Sci.* **25** 289–310. MR2791669

Souillard-Mandar, W., Davis, R., Rudin, C., Au, R., Libon, D. J., Swenson, R., Price, C. C., Lamar, M. and Penney, D. L. (2015). Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. *Machine Learning*. To appear.

Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *SIGMOD'96 Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* 1–12. ACM, New York.

Stang, P. E., Ryan, P. B., Racoosin, J. A., Overhage, J. M., Hartzema, A. G., Reich, C., Welebob, E., Scarnecchia, T. and Woodcock, J. (2010). Advancing the science for active surveillance: Rationale and design for the observational medical outcomes partnership. *Ann. Intern. Med.* **153** 600–606.

Taddy, M. A., Gramacy, R. B. and Polson, N. G. (2011). Dynamic trees for learning and design. *J. Amer. Statist. Assoc.* **106** 109–123. MR2816706

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York. MR1367965

Vellido, A., Martín-Guerrero, J. D. and Lisboa, P. J. G. (2012). Making machine learning models interpretable. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. ESANN, Bruges.

Wang, F. and Rudin, C. (2015). Falling rule lists. In *JMLR Workshop and Conference Proceedings* **38** 1013–1022. San Diego, CA.

Wang, T., Rudin, C., Doshi, F., Liu, Y., Klampfl, E. and MacNeille, P. (2015). Bayesian or's of and's for interpretable classification with application to context aware recommender systems. Available at arXiv:1504.07614.

Wu, Y., Tjelmeland, H. and West, M. (2007). Bayesian CART: Prior specification and posterior simulation. *J. Comput. Graph. Statist.* **16** 44–66. MR2345747

Wu, X., Zhang, C. and Zhang, S. (2004). Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems* **22** 381–405.

Yin, X. and Han, J. (2003). CPAR: Classification based on predictive association rules. In *ICDM'03 Proceedings of the 2003 SIAM International Conference on Data Mining* 331–335. SIAM, Philadelphia, PA.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* **12** 372–390.

Zhang, Y., Laber, E. B., Tsiatis, A. and Davidian, M. (2015). Using decision lists to construct interpretable and parsimonious treatment regimes. Available at arXiv:1504.07715.

B. Letham
Operations Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
USA
E-mail: bletham@mit.edu

C. Rudin
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
USA
E-mail: rudin@mit.edu

T. H. McCormick
Department of Statistics
Department of Sociology
University of Washington
Seattle, Washington 98105
USA
E-mail: tylermc@u.washington.edu

D. Madigan
Department of Statistics
Columbia University
New York, New York 10027
USA
E-mail: madigan@stat.columbia.edu