Building Interpretable Classifiers with Rules using Bayesian Analysis

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
December, 2012

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 the Bayesian List Machine (BLM), which yields a posterior distribution over possible decision lists. It employs a novel prior structure to encourage sparsity. In terms of predictive accuracy, our experiments show that the Bayesian List Machine is 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.

if total cholesterol $\geq$160 **and** smoke **then** *10 year CHD risk* $\geq$ *5%*
else if smoke **and** systolic blood pressure$\geq$140 **then** *10 year CHD risk* $\geq$ *5%*
else *10 year CHD risk* $<$ *5%*

Figure 1: Example decision list created using the NHBLI Framingham Heart Study Coronary Heart Disease (CHD) inventory for a 45 year old male.

KEY WORDS: Clustering, mixed data, item response theory.

1 Introduction

In many domains, interpretability is a fundamental desirable quality in a predictive model [17, 41]. Domain experts tend not to prefer black-box predictive models. They tend to prefer models that are more transparent, where it is clear exactly which factors were used to make a particular prediction. Interpretable models can be very convincing, particularly when only a few key factors are used, and each of them is meaningful. An interpretable model should be able to pinpoint exactly why a particular prediction was made, and provide the reason in a clear and natural way.

Our goal is to build predictive models that are as accurate as the top machine learning algorithms, yet are highly interpretable. Our predictive models will be in the natural form of sparse *decision lists*, where a decision list consists 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 outcome of interest. Figure 1 presents one possible decision list that we created using the NHBLI Framingham Heart Study Coronary Heart Disease (CHD) inventory [42] for a male patient who is 45 years old. The list provides an explanation of the risk factors that can be used both by healthcare providers and patients; a patient is at risk for CHD based on Figure 1, for example, *because* he has high blood pressure and he smokes. The list in Figure 1 is not the only accurate and interpretable decision list for predicting CHD; in fact there could be many such lists. Our goal is to learn these lists from data.

Our model for producing accurate and interpretable decision lists, called the Bayesian List Machine (BLM), produces a posterior distribution over

decision lists. The decision lists with the highest posterior values tend to be both very accurate and very interpretable, where the interpretability comes from a novel sparsity-inducing prior structure. 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. In Figure 1, the total number of rules is only 3, and the average number of terms on the left is 2, since both rules have 2 terms. In general, humans can handle only a handful of cognitive entities at once [19, 32], and as decision lists get more complicated, users need to weigh more conditional statements before arriving at an outcome. Encouraging sparsity through a Bayesian prior allows us to get the most interpretable solutions possible without sacrificing accuracy. Note that the prior is also useful for computational reasons. Even with a small number of features, the number of possible lists becomes very large - it is the number of possible permutations of rules. The prior helps to reduce the space of lists to be explored.

The motivation for our work lies in developing interpretable patient-level predictive models using massive observational medical data. This represents one of the fundamental challenges in delivering on the promise of evidence-based personalized medicine. In this context, our goal is to predict, given data on a patient's past medical history (encounters with providers, lab tests, imaging, etc.), the risk that each patient has towards a particular outcome (e.g. stroke). Since our predictive models are entirely learned from large amounts of data, they can potentially be more accurate than current medical scoring systems - yet just as concise and convincing. In this work, we demonstrate how this can be accomplished by constructing an alternative to the widely used CHADS_2 score of [16], which predicts stroke in patients with atrial fibrillation. The lists we obtain are as interpretable as CHADS_2, but more accurate.

In the remainder of this section, we discuss related work and other approaches to interpretable modeling. Then we describe our motivating application in further detail. In Section 2 we present the Bayesian List Machine. We describe the prior structure, and illustrate how the method can be adapted for imbalanced data that arise in rare event prediction problems. In Section 3, we provide an alternative to the CHADS_2 score. Section 4 includes experimental results on benchmark datasets, including a comparison to C4.5 [35], classification and regression trees (CART) [5], logistic regression, support vector machines (SVM)[40], boosted decision trees (BDT) [14], and random forests [4].

An argument based on Occam's Razor that is widely used in machine learning and statistics states that if a simpler class of models can be used to describe the data well, it tends to lead to better predictions because a simpler class of models tends not to overfit the data [e.g., see 27]. This means that the choice to use small decision lists can lead both to more accurate predictive models and more interpretable predictive models in many cases.

1.1 The framework of decision list machines

Decision lists are similar to models used in the expert systems literature from the 1970's and 1980's [21]. The knowledge base of an expert system is composed of natural language statements that are *if... then...* rules, and expert systems were among the first successful types of Artificial Intelligence. 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 sorting mechanisms [22, 25, 29, 36, 37, 44, 45]. Some of these sorting mechanisms provably work 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 the resulting classifier is not generally interpretable [examples are 15, 31]. Interpretability is closely related to the concept of explanation; an interpretable predictive model ought to be able to *explain* its predictions. A small literature has explored the concept of explanation in statistical modeling, see, for example, [26].

Let us discuss the role of computation in decision lists, and in decision trees. In our framework, the set of possible decision lists is the set of permutations of a pre-computed set of rules. Decision lists are a simple type of decision tree. However decision trees can be much more difficult to construct than decision lists, because the space of possible decision trees is much larger than the space of possible decision lists designed from the pre-computed rules. Because the space of possible decision trees is so large, they are usually constructed greedily from the top down, and then pruned heuristically upwards, and cross-validated to ensure accuracy. For instance, CART [5] and C4.5 [35] trees are constructed this way. 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 [8–10] are also constructed in

an approximate way, where the sampling procedure repeatedly restarts when the samples start to concentrate around a posterior mode, which is claimed to happen quickly. This means that the tree that is actually found is a local posterior maximum.

In contrast, our framework for constructing lists (first find rules, then order them) considers the whole list at once, and explores the set of lists using a Bayesian sampling procedure, and thus does not have the problem with the top of the list that we described for trees. Thus, the heuristic mechanisms used generally in associative classification and in tree construction are replaced in this work by sampling over permutations of rules. The computational hypothesis explored in this framework is that as long as one "mines deep enough" (finds enough useful rules in the database), it is possible to find a small number of good patterns that can be combined into a powerful, concise, and convincing predictive model. This hypothesis is also explored by [2], however, that method produces one list at a time, whereas our method produces the whole posterior of lists, and is more easily parallelized and adapted to larger scale problems. The work of [34] is based on simulated annealing, again producing a single rule list. The methods of [2] and [34] are based on discrete optimization, where the objective trades off between training accuracy and number of rules. In our work, the regularization is replaced with a Bayesian prior.

This work is also related to the *Hierarchical Association Rule Model (HARM)* presented recently by [30]. HARM is a Bayesian model that uses rules, but for a different medical context and a different statistical problem. HARM estimates the conditional probabilities of each rule in a conservative way, and does not explicitly aim to learn the *ordering* of rules, as the BLM does. HARM's estimates of conditional probability are based on the principle of the adjusted confidence [37], where rules that do not appear often enough in the database may not considered to be trustworthy enough to make accurate predictions. HARM is a Bayesian model for these conditional probabilities, and it makes predictions by ranking rules by the posterior means of the (conditional) probabilities. HARM aims at a different medical problem than the BLM: given a patient's medical history and characteristics, HARM predicts the very next condition they will experience. Thus, only the very top of HARM's list needs to make accurate predictions. The patient will likely not need to look down the list, as they will only look at the top few predictions of their next medical condition. In contrast, the BLM is designed for regular classification problems, and the classifier is the whole list. All of

the rules can be used to make predictions, not just the top few. This is why the full list for the BLM needs to be sparse and accurate.

1.2 Interpretable patient-level predictive models

Recent advances in the collection and storing of medical data present unprecedented opportunities to develop models that can predict a wide variety of outcomes [38]. Predictive models are, naturally, of interest to healthcare providers who, given reliable knowledge about a patient's likely healthcare trajectory, can tailor care to be proactive, appropriate, and cost-sensitive.

The front-end user interface of risk assessment tools are increasingly available online (see for example http://swedish.org/Health-Aware-Risk-Assessments/Stroke-Aware) or even as apps for mobile devices (for example http://www.qxmd.com/apps/calculate-by-qxmd). See http://www.lerner.ccf.org/qhs/risk_calculator/ for a collection of online risk calculators. In large part, however, these interfaces rely on risk assessment tools that were developed using statistical models that are not interpretable. At the end of the assessment, a patient may, therefore, be told he or she has a high risk for a particular outcome but have no understanding of why the risk is high or what steps can be taken to reduce risk. This situation may be especially troubling for a patient if the assessment happens without a healthcare provider to interpret the findings with the patient. Further, there is no way of assessing whether the model is trustworthy, given that it is a black-box.

Most widely used medical scoring systems, on the other hand, are designed to be interpretable, but are not necessarily optimized for accuracy, and are derived from few factors. For instance, the Thrombolysis In Myocardial Infarction (TIMI) Score [1], Apache II score for infant mortality in the ICU [20], the CURB-65 score for predicting mortality in community-acquired pneumonia [23], and the $CHADS_2$ score [16] are examples of interpretable scoring systems that are very widely used.

In this work, we will focus on the $CHADS_2$ score for predicting cerebrovascular accidents, or strokes, in patients with atrial fibrillation. A patient's 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 [24] 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 [16], the scores was calibrated with stroke risks using a database of 1,733 Medicare beneficiaries followed for, on average, about a year. These calibration data demonstrate a key challenge in making predictions for (relatively) rare but important events. During the follow-up period, there were 94 strokes across all risk categories (scores 0-6). Most patients were in lower risk categories, leaving very few patients to calibrate risk for patients with the highest scores. There were 65 patients with a score of 5 and only 5 patients with the maximum score of 6, for example. Thus, the $CHADS_2$ score is calibrated using the least data for patients most at risk.

Our mechanism for deriving a medical scoring system contrasts with the original study described above for $CHADS_2$. The $CHADS_2$ score was constructed with 1733 patients, whereas our scoring system was constructed from 12,586 patients chosen out of a database of over 11.1 million Medicaid enrollees. Each patient was followed for 2 years rather than 1 year as in the $CHADS_2$ study. The $CHADS_2$ score was constructed using 5 factors, whereas we considered all drugs and conditions experienced by any of the patients (thousands), and allowed the Bayesian List Machine to choose the most important ones. We do not have a problem with small samples for the highest risk categories.

2 The Bayesian List Machine

We begin by presenting our method as a generative model. We are in the setting of multi-class classification, where the set of possible labels is $1, \ldots, L$. In the case of predicting stroke risk, there are only two possible labels: stroke or no stroke. In the more general multi-class setting, one could predict the class of each observation (e.g., which type of product a customer is likely to purchase given their past purchase characteristics). 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, within classes $1, ..., L$. Our predictions are based on a set of rules $r = 1, \ldots, R$ (left hand sides) which are constructed from the training data and used to form lists.

Informally, we can describe the generative model as follows, where each step is discussed in more detail below. To generate a class label for the ith observation x_i, proceed via the following:

1. Generate an exhaustive list of rules $r = 1, \ldots, R$ (left hand sides) using

a rule-mining algorithm. These rules come from the set of features and are used to make lists.

2. Generate a random permutation over rules π from a prior $\text{Prior}(p, C)$.

3. Using this ordering, select the first rule that applies to observation, in that it matches the observed features x_i. Call the rule $\tilde{r}_i$.

4. Generate a label y_i as a draw from a Dirichlet-Multinomial distribution $\boldsymbol{\theta}^{(\tilde{r}_i)}$, with Dirichlet parameters $\alpha_1, \ldots, \alpha_L$ and counts $n_{\tilde{r}_i 1}, \ldots, n_{\tilde{r}_i L}$ for rule $\tilde{r}_i$ chosen in the previous step.

The posterior and the prior are distributions over rule lists. To obtain a single rule list, we could choose, for instance, the rule list having the highest value (the mode) of the posterior distribution (the maximum a posteriori estimator).

Let us discuss the steps in more detail. The first step in our model is to generate the set of rules. For situations where the dimensionality of the features is fairly low, we may consider all possible candidate rules; however in most applications we select a smaller number of rules using an algorithm for frequent itemset mining. In our experiments we used the FP-Growth algorithm [3] which finds all itemsets that satisfy constraints on minimum support and maximum cardinality. As long as the set of rules is large enough, we should be able to find subsets of them, and permutations of the subsets, that form useful decision lists.

After the set of rules is constructed, we draw an ordering over rules randomly from a prior distribution over permutations of rules, $\pi \sim \text{Prior}(p, C)$. The prior favors shorter decision lists (small total number of rules, sparse in the vertical direction of the list), and prefers rules with a small number of conditional statements (small left-hand-sides of rules, sparse in the horizontal direction of the list). The user-specified parameter C in the prior trades off between horizontal and vertical sparseness. A separate user-defined parameter p controls the overall strength of the prior. Specifically, the prior is:

$$\text{Prior}(\pi) \propto \frac{1}{\left(R_\pi + C\frac{A_\pi}{M}\right)^p}, \tag{1}$$

where R_π is the number of rules in the list. The A_π term is the average size of the left-hand-sides of the rules. M is the maximum allowed size of the left-hand-sides of rules. Thus, A_π/M is a fraction between 0 and 1, allowing

the constant C to be calibrated in a more intuitive way. For instance, if we chose C to be 1, the largest values of the prior would be achieved for lists with a smaller number of rules (smaller R_π, which is a positive integer). Then among lists of the same number of rules, the prior would favor lists with smaller left hand sides A_π. But since $A_\pi/M \leq 1$, a shorter rule list would be favored over shorter left hand sides of rules when $C = 1$. The constant C can be adjusted to the user's view of interpretability, or can be cross-validated. To promote sparsity, one can mine only rules with small left-hand-sides, in which case M would be relatively small. In our experiments we set $C = 1$. We then used the single prior hyperparameter p to directly control the length of the decision list, and set it either using cross-validation or to a specific value to obtain a list of desired length. M was chosen to be 2 or 5 in our experiments.

Continuing to follow the outline of our generative model, after the permutation of left-hand-sides of rules is chosen from the prior, for our given observation x_i we traverse down the list until we find the first rule that applies. We denote this rule by $\tilde{r}_i$. For instance, if the observation is $x_i = [10, 12, 7, 8]$ and the left hand side of the rule is "*if* $x_{i2} = 12$ *and* $x_{i3} = 7$" then this rule applies to the observation.

We then generate an outcome y_i as a single draw from a Multinomial distribution with $\boldsymbol{\theta}^{(\tilde{r}_i)} = \theta_1^{(\tilde{r}_i)}, \ldots, \theta_L^{(\tilde{r}_i)}$ the vector of class probabilities. $\boldsymbol{\theta}^{(\tilde{r}_i)}$ in turn follows a Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_L$, which are set to be weakly informative. Our experiments were on binary classification problems, and we fixed $\alpha_1 = \alpha_2 = 5$.

2.1 Likelihood

Define $\tilde{\mathbf{r}} \in \mathbb{R}^n$ as a vector of rule labels such that element $\tilde{r}_i = r$ if x_i is classified by rule r. That is, the vector $\tilde{\mathbf{r}}$ partitions the set of outcomes y_i so that the likelihood for each response is computed under exactly one rule. We then use these rule assignments to construct multinomial counts $n_{r\ell}$ for each rule $r = 1, \ldots, R$ and for each class $\ell = 1, \ldots, L$ by tallying the number of times rule r was associated with an outcome in class ℓ. That is, $n_{r\ell}$ is the number of observations x for which r was the first rule in the list that applied, and which have label $y = \ell$. Let $n_r = \sum_{\ell=1}^{L} n_{r\ell}$ be the total number

of observations classified by rule r. The likelihood is then

$$\mathcal{L}(y_1, \ldots, y_n | \boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(R)}, \tilde{\mathbf{r}}) = \prod_{r=1}^{R} \text{Multinomial}(n_{r1}, \ldots, n_{rL} | n_r, \boldsymbol{\theta}^{(r)})$$

where

$$\boldsymbol{\theta}^{(r)} \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_L).$$

Since the $\boldsymbol{\theta}^{(r)}$ are not of primary interest, we marginalized over $\boldsymbol{\theta}^{(r)}$ in each Multinomial distribution in the above product. Thus, conditional on the rule indicators we have, through the standard derivation of the Dirichlet-Multinomial distribution,

$$\begin{aligned} p(y_1, \ldots, y_n | \alpha_1, \ldots, \alpha_L, \tilde{\mathbf{r}}) &= \prod_{r=1}^{R} \frac{\Gamma(\sum_{\ell=1}^{L} \alpha_\ell)}{\Gamma(\sum_{\ell=1}^{L} n_{r\ell} + \alpha_\ell)} \\ &\quad \times \prod_{\ell=1}^{L} \frac{\Gamma(n_{r\ell} + \alpha_\ell)}{\Gamma(\alpha_\ell)} \\ &\propto \prod_{r=1}^{R} \frac{\prod_{\ell=1}^{L} \Gamma(n_{r\ell} + \alpha_\ell)}{\Gamma(\sum_{\ell=1}^{L} n_{r\ell} + \alpha_\ell)}, \end{aligned} \tag{2}$$

Note that the above equation depends on the rule indicators $\tilde{r}_i$ through $n_{r\ell}$.

2.2 Handling Imbalanced Data

In practice, many datasets are extremely imbalanced. For example, many fewer medical patients have a stroke than do not have a stroke. In such circumstances, without an appropriate correction, the likelihood can be dominated by negative responses and, as a result the method will simply predict "no stroke" for each patient using a single default rule; i.e., the full model will be "else predict majority class." We may instead desire to trade off between sensitivity and specificity of the classifier. To do this, we introduce an altered likelihood for imbalanced data:

$$p(y_1, \ldots, y_n | \alpha_1, \ldots, \alpha_L, \tilde{\mathbf{r}}) \propto \prod_{r=1}^{R} \frac{\prod_{\ell=1}^{L} \Gamma(\upsilon_\ell n_{r\ell} + \alpha_\ell)}{\Gamma(\sum_{\ell=1}^{L} \upsilon_\ell n_{r\ell} + \alpha_\ell)}, \tag{3}$$

where $\upsilon_\ell = L/P(y = \ell)$. The υ_ℓ terms thus re-weight the observations in each class to introduce additional weight in the likelihood for underrepresented cases. For imbalanced datasets, we apply the rule mining algorithm

separately to each class to ensure that rules that are powerful for a particular underrepresented class are not rejected by the minimum support threshold that is typically used in the rule-mining algorithms.

2.3 Model Fitting

The rule that ends the useable part of the list is called the "default" rule. The default rule has an empty left hand side, so that every observation applies to it. In the example of Figure 1, the default rule is "*else 10 year CHD risk* $< 5\%$." Any observation that has not already gotten a prediction from an earlier rule will get a prediction from the default rule.

We do Metropolis sampling, generating the proposed π^* from the current π_t using one of three options:

- (Move) Choose a position within the rule list at random, and move that rule to a randomly selected position above the default rule.
- (Add) Choose a rule that is not currently in the list at random, and move it to a randomly selected position within the list, above the default rule.
- (Cut) Choose a position above the default rule at random, remove that rule from the list, and include it in the collection of unused rules.

At each iteration, the option to Move, Add, or Cut is chosen randomly. As long as the probabilities of Add and Cut are the same, the proposal distribution is symmetric. In our experiments we used a uniform distribution over these choices.

This sampling algorithm is related to those used for Bayesian Decision Tree models [see for example 8, 9, 43]. We assess chain convergence using the method of [6] with the novel addition of a randomization test on the chi-squared statistic. The details of our convergence diagnostic are in the Appendix A. We make predictions in our experiments using the decision list with highest posterior probability. The work of [34] also uses a similar scheme (swap, add, cut) for rules to create a decision list using simulated annealing.

3 Stroke prediction compared to $CHADS_2$

We use the Bayesian List Machine to derive a competitor to $CHADS_2$ 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 that the Observational Medical Outcomes Partnership (OMOP, http://omop.fnih.org) has mapped to a common data model [39]. We extracted every patient in the MDCD database with a diagnosis of atrial fibrillation, one-year of atrial fibrillation-free 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. This is a much larger dataset than the one originally used to develop the $CHADS_2$ score (n=1,733).

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 50, 60, 70, and 80 years of age as split points, and for each split point introduced a pair of binary variables indicating whether the patient's age is less than or greater than the split point. We mined rules separately for each class (stroke or no stroke) using a minimum support threshold of 10% and a maximum cardinality M of 2. We used the likelihood model for imbalanced data, (3), and set the BLM prior hyperparameter at $p = 700$ to obtain a list of similar complexity to the $CHADS_2$ score. We fit the models and evaluated their performance using 5-fold cross-validation, constructing an ROC curve and measuring AUC for each fold.

In Figure 2 we show the decision list recovered from one of the folds. For each rule we give the stroke risk estimated from the training data as the number of patients satisfying that rule (and no preceding rule) that had a stroke. We give in parentheses the stroke risk across all patients that did not satisfy any of the preceding rules in the list. For example, the second line in the list indicates that among patients without hemiplegia the stroke risk was 12.5%, which increased to 46.6% when patients had a cerebrovascular disorder.

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

if hemiplegia **then** *stroke risk 58.0%* (14.5%)
else if cerebrovascular disorder **then** *stroke risk 46.6%* (12.5%)
else if transient ischaemic attack **and** essential hypertension
then *stroke risk 23.2%* (8.3%)
else if occlusion and stenosis of carotid artery without mention of cerebral infarction
then *stroke risk 16.4%* (7.8%)
else if age≤60 **then** *stroke risk 3.7%* (7.4%)
else *stroke risk 8.5%*

Figure 2: Decision list for determining 1-year stroke risk following diagnosis of atrial fibrillation from patient medical history. For each rule we give in parentheses the base risk for all patients that make it to that depth on the list.

	BLM	$CHADS_2$	CHA_2DS_2-VASc
AUC	0.750 (0.007)	0.721 (0.014)	0.677 (0.007)

Table 1: Mean AUC for stroke prediction with standard deviationin parentheses, across 5 folds of cross-validation.

focuses on a history of stroke, in order of severity. Hemiplegia, the paralysis of an entire side of the body, is a symptom of a severe stroke. 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 lists that we obtained in the 5 folds of cross-validation were of similar complexity to the $CHADS_2$ score: the mean list length was 6.8 (standard deviation 0.8). For comparison, $CHADS_2$ uses 5 features and CHA_2DS_2-VASc uses 8 features.

	SVM	C4.5	CART	Logistic Reg.	BDT	Rand. Forest
Mean AUC	0.763	0.553	0.703	0.767	0.780	0.776
Standard Dev.	0.013	0.019	0.010	0.011	0.017	0.012

Table 2: Mean AUC for stroke prediction and standard deviation across 5 folds of cross-validation for standard machine learning algorithms.

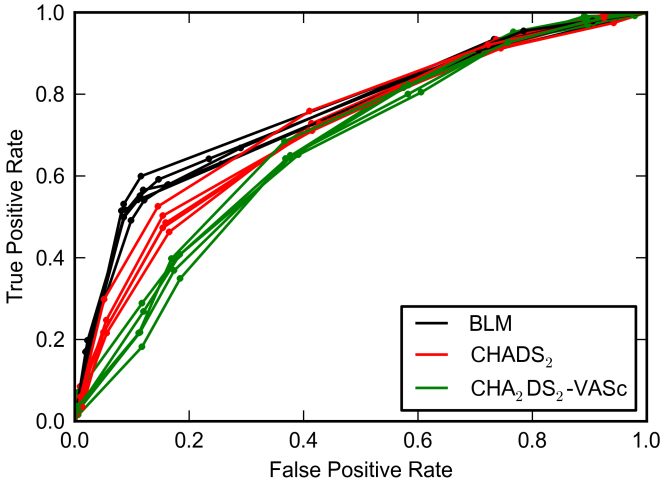

Figure 3: ROC curves for stroke prediction on the MDCD database for each of 5 folds of cross-validation, for BLM (black), $CHADS_2$ (red), and CHA_2DS_2-VASc (green).

In Figure 3 we give ROC curves for all 5 folds for BLM, $CHADS_2$, and CHA_2DS_2-VASc, and in Table 1 we report mean AUC (and in parentheses, standard deviation) across the folds. These results show that with complexity and interpretability similar to $CHADS_2$, the BLM decision lists performed significantly better at stroke prediction than both $CHADS_2$ and CHA_2DS_2-VASc ($p < 0.01$, t-test). Interestingly, we also found that $CHADS_2$ outperformed CHA_2DS_2-VASc despite CHA_2DS_2-VASc being an extension to $CHADS_2$. 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%. The fact that more stroke risk factors can correspond to a lower stroke risk suggests that the CHA_2DS_2-VASc model is misspecified, and highlights the difficulty in constructing these interpretable models manually. We also ran the decision tree algorithms C4.5 and CART on the stroke data, and the BLM's performance was substantially better (0.553 for C4.5 and 0.703 for CART, vs. 0.750 for the BLM). On these data, C4.5 tended to overfit the data, as its accuracy was well below the $CHADS_2$ baseline, and its trees were uninterpretable, with an avergae of 372 leaves (standard

deviation, 16.3). CART produced very concise lists, with an average of 3.2 leaves (standard deviation, 0.4), but it also did not perform nearly as well as the CHADS_2 baseline.

The performance of the BLM is comparable with the set of state-of-the-art (uninterpretable) methods. In Table 2 we provide the prediction performance results for several other machine learning algorithms, the implementation details of which are found in Appendix B.

We remark that there are potentially many different ways to produce an accurate and interpretable classifier for the same problem. For stroke prediction, as we ran the BLM over 5 folds, we produced 5 different lists. The other lists are found in Appendix C.

4 Benchmark dataset studies

The Bayesian list machine can produce powerful decision lists for general problems. We compared algorithm performance on a collection of datasets from the UCI Machine Learning Repository [13]: Tic-Tac-Toe Endgame, Mammographic Mass [11], Titanic, Breast Cancer Wisconsin (Original) [28], and Adult. For each dataset, categorical features were separated into binary features and real-valued features were split at their median into two binary features each. We used 5-fold cross validation and measured classification accuracy on each fold. None of these datasets suffer from extreme class imbalance, so we used the form of the likelihood given in (2). For all datasets except Adult, the parameters for rule mining were 5% minimum support and maximum cardinality of 5. For Adult the minimum support threshold was increased to 20% due to a large number of itemsets. We chose the strength of the prior p using 5-fold cross validation on each training set with $p = 0.5$, 2, and 5, and set p at the value that maximized AUC over the validation sets. The implementation details for the comparison algorithms are in Appendix B.

The Tic-Tac-Toe Endgame dataset provides all possible end board configurations for the game Tic-Tac-Toe. Tic-Tac-Toe is a two player game (player "X" and player "O") played on a 3x3 grid, such as that shown in Figure 4. Players sequentially mark spaces in the grid, and a player wins by marking three spaces in a row. The task is to use the end board configuration to identify whether "X" won or not. Figure 5 gives a fitted decision list, which simply identifies the 8 possible ways that "X" can have three marks in a row.

1	2	3
4	5	6
7	8	9

Figure 4: Grid labeling for the Tic-Tac-Toe dataset.

if X in 2 **and** X in 8 **and** X in 5 **then** *X wins* (100%)
else if X in 1 **and** X in 3 **and** X in 2 **then** *X wins* (100%)
else if X in 9 **and** X in 3 **and** X in 6 **then** *X wins* (100%)
else if X in 7 **and** X in 4 **and** X in 1 **then** *X wins* (100%)
else if X in 7 **and** X in 9 **and** X in 8 **then** *X wins* (100%)
else if X in 9 **and** X in 5 **and** X in 1 **then** *X wins* (100%)
else if X in 4 **and** X in 5 **and** X in 6 **then** *X wins* (100%)
else if X in 7 **and** X in 3 **and** X in 5 **then** *X wins* (100%)
else *X does not win* (100%)

Figure 5: Decision list for Tic-Tac-Toe. Following each class prediction in parentheses is the confidence of the prediction.

The accuracy of this decision list is thus perfect, whereas neither CART nor C4.5 achieve nearly this level, at 88% and 86% accuracy respectively. Performance on the Tic-Tac-Toe dataset is a good sanity check on the performance of an interpretable classifier - and it is a test that neither CART nor C4.5 can pass.

The Mammographic Mass dataset contains descriptions of the shape, margin, and density of 961 mammographic masses, together with the patient's age. The task is to predict whether a mass is benign or malignant, and a fitted decision list is given in Figure 6.

The Titanic dataset contains categorical descriptions of ticket class, sex,

if irregular **then** *malignant* (80%)
else if spiculated **then** *malignant* (95%)
else if age> 57 **and** ill-defined **then** *malignant* (70%)
else if lobular **and** low density **then** *benign* (57%)
else *benign* (88%)

Figure 6: Decision list for Mammographic Mass.

if male **and** adult **then** *died* (80%)
else if 3rd class **then** *died* (59%)
else *survived* (93%)

Figure 7: Decision list for Titanic.

and age for the 2201 passengers on the Titanic, with the task of predicting whether or not a passenger survived. The fitted decision list in Figure 7 is consistent with historical accounts of space on lifeboats being limited to women and children, particularly those with higher-class tickets.

The Breast Cancer Wisconsin (Original) dataset contains 10 features describing the properties of cells in a breast mass, for each of 699 patients. As with the Mammographic Mass dataset, the task is to predict whether the mass is benign or malignant.

The Adult dataset contains demographic information such as age, education, gender, and marital status for 48,842 individuals, with the task of predicting whether or not an individual makes more than $50K per year. We limited our experiments to a randomly sampled collection of 5000 individuals.

Table 3 gives the classification accuracy for each of the algorithms we tried for each of the baseline datasets. For the interpretable classifiers (BLM, C4.5, and CART), we provide in Table 4 the average size of the classifier across the 5 folds: length of the decision list for BLM and number of leaves for the decision trees. With size similar to that of C4.5 and CART, the BLM decision lists outperformed these decision tree algorithms: For every experiment the mean accuracy were either significantly higher with BLM ($p < 0.05$, t-test) or statistically indistinguishable. For many datasets the BLM performance was comparable to that of the uninterpretable methods (Logistic regression, SVM, BDT, and random forests).

5 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 [18], however, we now are starting to have the computational tools to truly test this hypothesis. The BLM method introduced in this work aims to hit the "sweet spot" between predictive accuracy, interpretability, and

Accuracy	Tic-Tac-Toe	Mammogram	Titanic	Wisconsin	Adult
BLM	1.00 (0.00)	0.81 (0.05)	0.79 (0.02)	0.95 (0.02)	0.82 (0.01)
C4.5	0.86 (0.04)	0.81 (0.05)	0.77 (0.02)	0.93 (0.03)	0.83 (0.01)
CART	0.88 (0.04)	0.79 (0.06)	0.79 (0.02)	0.93 (0.02)	0.82 (0.01)
Logistic Reg.	0.98 (0.01)	0.81 (0.06)	0.78 (0.02)	0.96 (0.01)	0.85 (0.01)
SVM	0.99 (0.01)	0.79 (0.06)	0.78 (0.02)	0.96 (0.02)	0.84 (0.02)
BDT	0.86 (0.04)	0.81 (0.06)	0.78 (0.02)	0.96 (0.02)	0.85 (0.01)
Rand. Forest	0.98 (0.01)	0.81 (0.06)	0.79 (0.02)	0.96 (0.02)	0.84 (0.02)

Table 3: Mean classification accuracy across 5 folds of cross-validation, and in parentheses standard deviation, for the various machine learning algorithms applied to UCI datasets.

	Tic-Tac-Toe	Mammogram	Titanic	Wisconsin	Adult
BLM length	8 (0)	5.6 (0.5)	4 (0.7)	3.8 (0.4)	8.4 (0.9)
C4.5 leaves	39.6 (6.7)	8.2 (0.8)	5 (0)	8.2 (1.9)	85.2 (11.1)
CART leaves	20.4 (2.1)	5.4 (1.3)	4 (1.4)	5 (1)	3.8 (0.4)

Table 4: Mean size, and in parentheses standard deviation, of the interpretable classifiers on the UCI datasets.

tractability.

For problems where interpretability requires extra constraints on the ordering or form of the rules, the framework introduced here can be adapted to handle that, and there are several ways to do this. First, the prior can be set to zero for lists that are "uninterpretable" according to a given definition. Second, post-processing on the lists can be performed in order to engineer the lists towards the desired level of interpretability. In that case, one should beware of changing the accuracy level when working manually with the lists. Third, one can explore the set of lists having high posterior values, and can choose among those lists for the one that is the most interpretable.

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.

A Convergence diagnostic

We follow the convergence diagnostic of [6]. We begin J chains from randomly selected initial conditions and run them for N iterations. We discard the first half of the samples as burn-in, and thin the remaining samples at a rate of 100. Suppose that the J chains visited a total of c decision lists. We define N_ν^j as the number of times chain j visited decision list ν in the thinned samples, $j = 1, \ldots, J$ and $\nu = 1, \ldots, c$. We then implement a chi-square test of homogeneity across the chains. If the chains were homogenous, the expected number of visits per chain to each decision list ν would be $E_\nu = \frac{1}{J}\sum_{j=1}^{J} N_\nu^j$ and the chi-squared statistic is

$$\chi^2 = \sum_{\nu=1}^{c}\sum_{j=1}^{J}\frac{(N_\nu^j - E_n u)^2}{E_\nu}.$$

[6] use Pearson's chi-squared test to compute a p-value. If the p-value is sufficiently large (*e.g.*, greater than 0.05) then the null hypothesis of chain

homogeneity cannot be rejected, and the chains can be considered converged. Pearson's chi-squared test tends to perform poorly for when counts are less than around 5, which is often the case for chains over decision lists because the space of decision lists is very large. Thus rather than use the χ^2 distribution which is only asymptotically accurate, here we here we empirically estimate the actual distribution of the χ^2 statistic. This is done by randomly sampling a large number of contingency tables with the same marginals as

$$\begin{pmatrix} N_1^1 & \dots & N_c^1 \\ \vdots & & \vdots \\ N_1^J & \dots & N_c^J \end{pmatrix}$$

and computing their χ^2 statistic. Random contigency tables with fixed marginals can be efficiently sampled using Patefield's algorithm [33], which is available as the R function "r2dtable." This provides an empirical distribution for the χ^2 statistic and the p-value can be estimated directly as the fraction of randomly generated tables with a χ^2 value larger than that of the MCMC chains.

In our experiments, we used 3 chains and determined chains were converged when $p > 0.05$.

B Comparison algorithm implementations

SVM: LIBSVM [7] 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 $C_{\text{SVM}} \in \{2^{-4}, 2^{-3}, \dots, 2^4\}$ and $\gamma \in \{2^{-6}, 2^{-5}, \dots, 2^0\}$ to find the parameters that maximized AUC over a 5-fold cross-validation over each training set.
C4.5: C4.5 Release 8, distributed by Quinlan.
CART: The R library "rpart" with default parameters and pruned using the complexity parameter that minimized cross-validation error.
Logistic regression: The LIBLINEAR [12] implementation of logistic regression with ℓ_1 regularization. We selected the regularization parameter C_{LR} from $\{2^{-4}, 2^{-3}, \dots, 2^4\}$ using 5-fold cross-validation over each training set to find the parameter that maximized AUC.
Boosted decision trees: The R library "gbm" with shrinkage = 0.005, ntrees = 10000, and the number of iterations selected with 5-fold cross-validation.

Random forests: The R library "randomForest" with $10,000$ trees.

C Other $CHADS_2$ Lists

Figure 8 provides the decision list for the other four folds for the $CHAD_2$ experiments.

Acknowledgements

We would like to thank Zachary Shahn and the OMOP team for help with the data. Cynthia Rudin's research was partly funded by grant IIS-1053407 from the National Science Foundation. 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.

References

[1] E. M. Antman, M. Cohen, P. M. Bernink, and et al. The TIMI risk score for unstable angina/nonST elevation MI: A method for prognostication and therapeutic decision making. *The Journal of the American Medical Association*, 284(7):835–842, 2000.

[2] Dimitris Bertsimas, Allison Chang, and Cynthia Rudin. An integer optimization approach to associative classification. In *Proceedings of Neural Information Processing Systems*, 2012.

[3] 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.

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

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

Fold 2. AUC = 0.74

if hemiplegia **then** *stroke risk 57.8%*
else if cerebrovascular disorder **then** *stroke risk 44.5%*
else if cardiac failure congestive **and** hydrocodone **then** *stroke risk 9.9%*
else if transient ischaemic attack **then** *stroke risk 30.4%*
else if age≤ 70 **then** *stroke risk 4.0%*
else *stroke risk 9.1%*

Fold 3. AUC = 0.76

if hemiplegia **then** *stroke risk 56.0%*
else if cerebrovascular disorder **then** *stroke risk 44.8%*
else if transient ischaemic attack **and** age>50 **then** *stroke risk 25.8%*
else if arteriosclerosis coronary artery **and** hydrocodone **then** *stroke risk 9.8%*
else if age>70 **then** *stroke risk 8.7%*
else if age>60 **and** abdominal pain **then** *stroke risk 10.8%*
else *stroke risk 3.4%*

Fold 4. AUC = 0.76

if hemiplegia **then** *stroke risk 56.1%*
else if cerebrovascular disorder **then** *stroke risk 44.5%*
else if transient ischaemic attack **and** age>60 **then** *stroke risk 26.0%*
else if occlusion and stenosis of carotid artery without mention of cerebral infarction
then *stroke risk 14.4%*
else if gabapentin **and** age>50 **then** *stroke risk 10.7%*
else if age≤70 **and** essential hypertension **then** *stroke risk 4.9%*
else if age≤60 **then** *stroke risk 1.6%*
else *stroke risk 8.3%*

Fold 5. AUC = 0.75

if cerebrovascular disorder **then** *stroke risk 49.1%*
else if transient ischaemic attack **then** *stroke risk 24.9%*
else if age≤60 **then** *stroke risk 3.9%*
else if age>50 **and** hemiplegia **then** *stroke risk 41.7%*
else *stroke risk 8.8%*

Figure 8: Decision lists for folds 2, 3, 4, and 5. These are all alternatives to the $CHADS_2$ score. Note that there are drugs included in the lists. The presence of a drug or condition on the list is not necessarily causal, but is correlative (given current medical practices), and can thus be used to determine the risk.

[6] S. P. Brooks, P. Giudici, and A. Philippe. Nonparametric convergence assessment for MCMC model selection. *Journal of Computational and Graphical Statistics*, 12(1):1–22, 2003.

[7] 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.

[8] 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.

[9] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian Treed Models . *Machine Learning*, 48(1/3):299–320, 2002.

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

[11] M. Elter, R. Schulz-Wendtland, and T. Wittenberg. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, 34:4164–4172, 2007.

[12] 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.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory*, 904:23–37, 1995.

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

[16] B Gage, A Waterman, W Shannon, M Boechler, M Rich, and M Radford. Comparing Hospitals on Stroke Care¡subtitle¿The Need to Account for Stroke Severity¡/subtitle¿¡alt-title¿Comparing Hospitals on Stroke Care¡/alt-title¿. *Journal of the American Medical Association*, 285:2864–2870, 2001.

[17] 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.

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

[19] 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.

[20] WILLIAM A. KNAUS, ELIZABETH A. DRAPER, DOUGLAS P. WAGNER, and JACK E. ZIMMERMAN. APACHE II: A severity of disease classification system. *Critical Care Medicine*, 13:818–829, 1985.

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

[22] 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.

[23] 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.

[24] 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, 2010.

[25] 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*, pages 80–96, 1998.

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

[27] David Madigan and Adrian Raftery. Model selection and accounting for uncertainty in graphical models using occamś window. *Journal of the American Statistical Association*, 89:1535–1546, 1994.

[28] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. 23(5):1–18, 1990.

[29] Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent features. *Journal of Machine Learning Research*, 2005.

[30] 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.

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

[32] 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.

[33] W. M. Patefield. Algorithm as159. an efficient method of generating r x c tables with given row and column totals. *Applied Statistics*, 30:91–97, 1981.

[34] Shawn Qian, Cynthia Rudin, and Allison Chang. How to build interpretable classifiers with rules and global optimization. in progress, 2013.

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

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

[37] Cynthia Rudin, Benjamin Letham, Ansaf Salleb-Aouissi, Eugen Kogan, and David Madigan. Sequential event prediction with association rules. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, 2011.

[38] Galit Shmueli. To Explain or to Predict? *Statistical Science*, 25(3):289–310, August 2010.

[39] 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.

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

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

[42] Peter W. F. Wilson, Ralph B. D'Agostino, Daniel Levy, Albert M. Belanger, Halit Silbershatz, and William B. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

[43] 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.

[44] Yu Yi and Eyke Hüllermeier. Learning complexity-bounded rule-based classifiers by combining association analysis and genetic algorithms. In *Proc. Joint 4th Int. Conf. in Fuzzy Logic and Technology and 11th French Days on Fuzzy Logic and Applications*, pages 47–52, 2005.

[45] Xiaoxin Yin and Jiawei Han. CPAR: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335, 2003.