Journal of Machine Learning Research 18 (2017) 1-37 Submitted 1/16; Revised 2/17; Published 8/17

A Bayesian Framework for Learning Rule Sets for Interpretable Classification

Tong Wang TONG-WANG@UIOWA.EDU *University of Iowa*
Cynthia Rudin CYNTHIA@CS.DUKE.EDU *Duke University*
Finale Doshi-Velez FINALE@SEAS.HARVARD.EDU *Harvard University*
Yimin Liu LIUYIMIN2000@GMAIL.COM *Edward Jones*
Erica Klampfl EKLAMPFL@FORD.COM *Ford Motor Company*
Perry MacNeille PMACNEIL@FORD.COM *Ford Motor Company*

Editor: Maya Gupta

Abstract

We present a machine learning algorithm for building classifiers that are comprised of a *small* number of *short* rules. These are restricted disjunctive normal form models. An example of a classifier of this form is as follows: *If* X satisfies (condition A AND condition B) OR (condition C) OR $\cdots$, *then* $Y = 1$. Models of this form have the advantage of being interpretable to human experts since they produce a set of rules that concisely describe a specific class. We present two probabilistic models with prior parameters that the user can set to encourage the model to have a desired size and shape, to conform with a domain-specific definition of interpretability. We provide a scalable MAP inference approach and develop theoretical bounds to reduce computation by iteratively pruning the search space. We apply our method (Bayesian Rule Sets – *BRS*) to characterize and predict user behavior with respect to in-vehicle context-aware personalized recommender systems. Our method has a major advantage over classical associative classification methods and decision trees in that it does not greedily grow the model.

Keywords: disjunctive normal form, statistical learning, data mining, association rules, interpretable classifier, Bayesian modeling

1. Introduction

When applying machine learning models to domains such as medical diagnosis and customer behavior analysis, in addition to a reliable decision, one would also like to understand how this decision is generated, and more importantly, what the decision says about the data itself. For classification tasks specifically, a few summarizing and descriptive rules will provide intuition about the data that can directly help domain experts understand the decision process.

Our goal is to construct rule set models that serve this purpose: these models provide predictions and also descriptions of a class, which are reasons for a prediction. Here is an example of a rule set model for predicting whether a customer will accept a coupon for a nearby coffee house, where the coupon is presented by their car's mobile recommendation device:

if a customer (goes to coffee houses $\geq$ once per month AND destination = no urgent place AND passenger $\neq$ kids)

OR (goes to coffee houses $\geq$ once per month AND the time until coupon expires $=$ one day)
then

predict the customer will accept the coupon for a coffee house.

This model makes predictions and provides characteristics of the customers and their contexts that lead to an acceptance of coupons.

Formally, a rule set model consists of a set of *rules*, where each rule is a conjunction of *conditions*. Rule set models predict that an observation is in the positive class when at least one of the rules is satisfied. Otherwise, the observation is classified to be in the negative class. Rule set models that have a small number of conditions are useful as non-black-box (interpretable) machine learning models. Observations come with predictions and also reasons (e.g., this observation is positive *because* it satisfies a particular rule).

This form of model appears in various fields with various names. First, they are sparse disjunctive normal form (DNF) models. In operations research, there is a field called "logical analysis of data" (Boros et al., 2000; Crama et al., 1988), which constructs rule sets based on combinatorics and optimization. Similar models are called "consideration sets" in marketing (see, e.g., Hauser et al., 2010), or "disjunctions of conjunctions." Consideration sets are a way for humans to handle information overload. When confronted with a complicated decision (such as which television to purchase), the theory of consideration sets says that humans tend to narrow down the possibilities of what they are willing to consider into a small disjunction of conjunctions ("only consider TV's that are small AND inexpensive, OR large and with a high display resolution"). In PAC learning, the goal is to learn rule set models when the data are non-noisy, meaning there exists a model within the class that perfectly classifies the data. The field of "rule learning" aims to construct rule set models. We prefer the terms "rule set" models or "sparse DNF" for this work as our goal is to create interpretable models for non-experts, which are not necessarily consideration sets, not other forms of logical data analysis, and pertain to data sets that are potentially noisy, unlike the PAC model.

For many decisions, the space of good predictive models is often large enough to include very simple models (Holte, 1993). This means that there may exist very sparse but accurate rule set models; the challenge is determining how to find them efficiently. Greedy methods used in previous works (Malioutov and Varshney, 2013; Pollack et al., 1988; Friedman and Fisher, 1999; Gaines and Compton, 1995), where rules are added to the model one by one, do not generally produce high-quality sparse models.

We create a Bayesian framework for constructing rule sets, which provides priors for controlling the size and shape of a model. These methods strike a nice balance between accuracy and interpretability through user-adjustable Bayesian prior parameters. To control interpretability, we introduce two types of priors on rule set models, one that uses beta-binomials to model the rule selecting process, called BRS-BetaBinomial, and one that uses Poisson distributions to model rule generation, called BRS-Poisson. Both can be adjusted to suit a domain-specific notion of interpretability; it is well-known that interpretability comes in different forms for different domains (Martens and Baesens, 2010; Martens et al., 2011; Huysmans et al., 2011; Allahyari and Lavesson, 2011; Rüping, 2006; Freitas, 2014).

We provide an approximate inference method that uses association rule mining and a randomized search algorithm (which is a form of stochastic coordinate descent or simulated annealing) to find optimal BRS maximum a posteriori (MAP) models. This approach is motivated by a theoretical bound that allows us to iteratively reduce the size of the computationally hard problem of finding a MAP solution. This bound states that we need only mine rules that are sufficiently fre-

quent in the database and as the search continues, the bound becomes tighter and further reduces the search space. This greatly reduces computational complexity and allows the problem to be solved in practical settings. The theoretical result takes advantage of the strength of the Bayesian prior.

Our applied interest is to understand user responses to personalized advertisements that are chosen based on user characteristics, the advertisement, and the context. Such systems are called *context-aware recommender systems* (see surveys Adomavicius and Tuzhilin, 2005, 2008; Verbert et al., 2012; Baldauf et al., 2007, and references therein). One major challenge in the design of recommender systems, reviewed in Verbert et al. (2012), is the *interaction challenge*: users typically wish to know *why* certain recommendations were made. Our work addresses precisely this challenge: our models provide rules in data that describe conditions for a recommendation to be accepted.

The main contributions of our paper are as follows:

- We propose a Bayesian approach for learning rule set (DNF) classifiers. This approach incorporates two important aspects of performance, accuracy, and interpretability, and balance between them via user-defined parameters. Because of the foundation in Bayesian methodology, the prior parameters are meaningful; they represent the desired size and shape of the rule set.

- We derive bounds on the support of rules and number of rules in a MAP solution. These bounds are useful in practice because they safely (and drastically) reduce the solution space, improving computational efficiency. The bound on the size of a MAP model guarantees a sparse solution if prior parameters are chosen to favor smaller sets of rules. More importantly, the bounds become tighter as the search proceeds, reducing the search space until the search finishes.

- The simulation studies demonstrate the efficiency and reliability of the search procedure. Losses in accuracy usually result from a misspecified rule representation, not generally from problems with optimization. Separately, using publicly available data sets, we compare rule set models to interpretable and uninterpretable models from other popular classification methods. Our results suggest that BRS models can achieve competitive performance, and are particularly useful when data are generated from a set of underlying rules.

- We study in-vehicle mobile recommender systems. Specifically, we used Amazon Mechanical Turk to collect data about users interacting with a mobile recommendation system. We used rule set models to understand and analyze consumers' behavior and predict their response to different coupons recommended in different contexts. We were able to generate interpretable results that can help domain experts better understand their customers.

The remainder of our paper is structured as follows. In Section 2, we discuss related work. In Section 3, we present Bayesian Rule Set modeling. In Section 4, we introduce an approximate MAP inference method using associative rule mining and randomized search. We also present theoretical bounds on the support and the number of rules in an optimal MAP solution. In Section 5, we show the simulation studies to justify the inference methods. We then report experimental results using publicly available data, including the in-vehicle recommendation system data set we created, to show that BRS models are on-par with black-box models while under strict size restrictions for the purpose of being interpretable.

A shorter version of this work appeared at the International Conference on Data Mining (Wang et al., 2016).

2. Related Work

The models we are studying have different names in different fields: “disjunctions of conjunctions" in marketing, “classification rules" in data mining, “disjunctive normal forms" (DNF) in artificial intelligence, and “logical analysis of data" in operations research. Learning logical models of this form has an extensive history in various settings. The LAD techniques were first applied to binary data in (Crama et al., 1988) and were later extended to non-binary cases (Boros et al., 2000). In parallel, Valiant (1984) showed that DNFs could be learned in polynomial time in the PAC (probably approximately correct - non-noisy) setting, and recent work has improved those bounds via polynomial threshold functions (Klivans and Servedio, 2001) and Fourier analysis (Feldman, 2012). Other work studies the hardness of learning DNF (Feldman, 2006). None of these theoretical approaches considered the practical aspects of building a *sparse* model with realistic noisy data.

In the meantime, the data-mining literature has developed approaches to building logical conjunctive models. Associative classification and rule learning methods (e.g., Ma et al., 1998; Li et al., 2001; Yin and Han, 2003; Chen et al., 2006; Cheng et al., 2007; McCormick et al., 2012; Rudin et al., 2013; Dong et al., 1999; Michalski, 1969; Clark and Niblett, 1989; Frank and Witten, 1998) mine for frequent rules in the data and combine them in a heuristic way, where rules are ranked by an interestingness criteria. Some of these methods, like CBA, CPAR and CMAR (Li et al., 2001; Yin and Han, 2003; Chen et al., 2006; Cheng et al., 2007) still suffer from a huge number of rules. Inductive logic programming (Muggleton and De Raedt, 1994) is similar, in that it mines (potentially complicated) rules and takes the simple union of these rules as the rule set, rather than optimizing the rule set directly. This is a major disadvantage over the type of approach we take here. Another class of approaches aims to construct rule set models by greedily adding the conjunction that explains the most of the remaining data (Malioutov and Varshney, 2013; Pollack et al., 1988; Friedman and Fisher, 1999; Gaines and Compton, 1995; Cohen, 1995). Thus, again, these methods do not directly aim to produce globally optimal conjunctive models.

There are few recent techniques that do aim to fully learn rule set models. Hauser et al. (2010); Goh and Rudin (2014) applied integer programming approaches to solving the full problems but would face a computational challenge for even moderately sized problems. There are several branch-and-bound algorithms that optimize different objectives than ours (e.g., Webb, 1995). Rijnbeek and Kors (2010) proposed an efficient way to exhaustively search for short and accurate decision rules. That work is different than ours in that it does not take a generative approach, and their global objective does not consider model complexity, meaning they do not have the same advantage of reduction to a smaller problem that we have. Methods that aim to globally find decision lists or rule lists can be used to find optimal DNF models, as long as it is possible to restrict all of the labels in the rule list (except the default) to the same value. A rule list where all labels except the default rule are the same value is exactly a DNF formula. We are working on extending the CORELS algorithm (Angelino et al., 2017) to handle DNF.

Note that logical models are generally robust to outliers and naturally handle missing data, with no imputation needed for missing attribute values. These methods can perform comparably with traditional convex optimization-based methods such as support vector machines or lasso (though linear models are not always considered to be interpretable in many domains).

One could extend any method for DNF learning of binary classification problems to multi-class classification. A conventional way to directly extend a binary classifier is to decompose the multi-class problem into a set of two-class problems. Then for each binary classification problem, one could build a separate BRS model, e.g., by the one-against-all method. One would generate different rule sets for each class. For this approach, the issue of overlapping coverage of rule sets for different classes is handled, for instance, by an error correcting output codes (Schapire, 1997).

The main application we consider is in-vehicle context-aware recommender systems. The most similar works to ours include that of Baralis et al. (2011), who present a framework that discovers relationships between user context and services using association rules. Lee et al. (2006) create interpretable context-aware recommendations by using a decision tree model that considers location context, personal context, environmental context and user preferences. However, they did not study some of the most important factors we include, namely contextual information such as the user's destination, relative locations of services along the route, the location of the services with respect to the user's destination, passenger(s) in the vehicle, etc. Our work is related to recommendation systems for in-vehicle context-aware music recommendations (see Baltrunas et al., 2011; Wang et al., 2012), but whether a user will accept a music recommendation does not depend on anything analogous to the location of a business that the user would drive to. The setup of in-vehicle recommendation systems are also different than, for instance, mobile-tourism guides (Noguera et al., 2012; Schwinger et al., 2005; Van Setten et al., 2004; Tung and Soo, 2004) where the user is searching to accept a recommendation, and interacts heavily with the system in order to find an acceptable recommendation. The closest work to ours is probably that of Park et al. (2007) who also consider Bayesian predictive models for context aware recommender systems to restaurants. They also consider demographic and context-based attributes. They did not study advertising, however, which means they did not consider the locations to the coupon's venue, expiration times, etc.

3. Bayesian Rule Sets

We work with standard classification data. The data set S consists of $\{\mathbf{x}_n, y_n\}_{n=1,\ldots N}$, where $y_n \in \{0,1\}$ with N_+ positive and N_- negative, and $\mathbf{x}_n \in \mathcal{X}$. $\mathbf{x}_n$ has J features. Let a represent a rule and $a(\cdot)$ with a corresponding Boolean function

$$a(\cdot) : \mathcal{X} \to \{0,1\}.$$

$a(\mathbf{x})$ indicates if $\mathbf{x}$ satisfies rule a. Let A denote a rule set and let $A(\cdot)$ represent a classifier,

$$A(\mathbf{x}) = \begin{cases} 1 & \exists a \in A, a(\mathbf{x}) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

$\mathbf{x}$ is classified as positive if it satisfies at least one rule in A.

Figure 1 shows an example of a rule set. Each rule is a yellow patch that covers a particular area, and the rule applies to the area it covers. In Figure 1, the white oval in the middle indicates the positive class. Our goal is to find a set of rules A that covers mostly the positive class, but little of the negative class, while keeping A as a small set of short rules.

We present a probabilistic model for selecting rules. Taking a generative approach allows us to flexibly incorporate users' expectations on the "shape" of a rule set through the prior. The user can guide the model toward more domain-interpretable solutions by specifying the desired balance between the size and lengths of rules without committing to any particular value for these parameters.

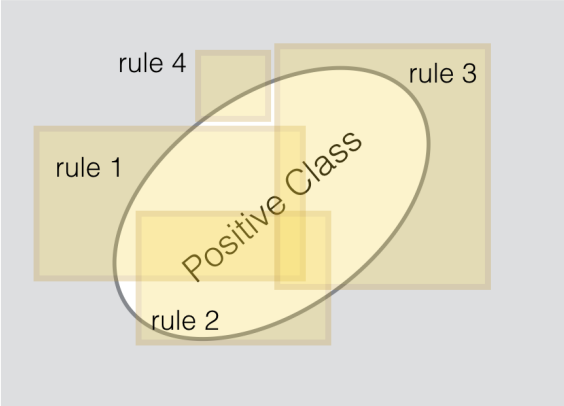


Figure 1: Illustration of Rule Sets. Area covered by any of rules is classified as positive. Area not covered by any rules is classified as negative.

We propose two models for the prior, a Beta-Binomial prior and a Poisson prior, which characterize the interpretability from different perspectives. The likelihood ensures that the model explains the data well and ensures good classification performance. We detail the priors and likelihood below.

3.1 Prior

We propose two priors. In the BRS-BetaBinomial model, the maximum length L of rules is predetermined by a user. Rules of the same length are placed into the same rule *pool*. The model uses L beta priors to control the probabilities of selecting rules from different pools. In a BRS-Poisson model, the "shape" of a rule set, which includes the number of rules and lengths of rules, is decided by drawing from Poisson distributions parameterized with user-defined values. Then the generative process fills it in with conditions by first randomly selecting attributes and then randomly selecting values corresponding to each attribute. We present the two prior models in detail.

3.1.1 BETA-BINOMIAL PRIOR

Rules are drawn randomly from a set $\mathcal{A}$. Assuming the interpretability of a rule is associated with the *length* of a rule (the number of conditions in a rule), the rule space $\mathcal{A}$ is divided into L pools indexed by the lengths, L being the maximum length the user allows.

$$\mathcal{A} = \cup_{l=1}^{L} \mathcal{A}_l. \tag{1}$$

Then, rules are drawn independently in each pool, and we assume in $\mathcal{A}_l$ each rule has probability p_l to be drawn, which we place a beta prior on. For $l \in \{1, ..., L\}$,

$$\text{Select a rule from } \mathcal{A}_l \sim \text{Bernoulli}(p_l) \tag{2}$$

$$p_l \sim \text{Beta}(\alpha_l, \beta_l). \tag{3}$$

In reality, it is not practical to enumerate all possible rules within $\mathcal{A}_l$ when ℓ is even moderately large. However, since we aim for intuitive models, we know that the optimal MAP model will consist of a small number of rules, each of which has sufficiently large support in the data (this

is formalized later within proofs). Our approximate inference technique thus finds and uses high support rules to use within $\mathcal{A}_l$. This effectively makes the beta-binomial model non-parametric.

Let M_l notate the number of rules drawn from $\mathcal{A}_l$, $l \in \{1, ..., L\}$. A BRS model A consists of rules selected from each pool. Then

$$p(A; \{\alpha_l, \beta_l\}_l) = \prod_{l=1}^{L} \int p_l^{M_l} (1-p_l)^{(|\mathcal{A}_l| - M_l)} dp_l$$
$$= \prod_{l=1}^{L} \frac{B(M_l + \alpha_l, |\mathcal{A}_l| - M_l + \beta_l)}{B(\alpha_l, \beta_l)}, \tag{4}$$

where $B(\cdot)$ represents a Beta function. Parameters $\{\alpha_l, \beta_l\}_l$ govern the prior probability of selecting a rule set. To favor smaller models, we would choose α_l, β_l such that $\frac{\alpha_l}{\alpha_l + \beta_l}$ is close to 0. In our experiments, we set $\alpha_l = 1$ for all l.

3.1.2 POISSON PRIOR

We introduce a data independent prior for the BRS model. Let M denote the total number of rules in A and L_m denote the lengths of rules for $m \in \{1, ...M\}$. For each rule, the length needs to be at least 1 and at most the number of all attributes J. According to the model, we first draw M from a Poisson distribution. We then draw L_m from truncated Poisson distributions (which only allow numbers from 1 to J), to decide the "shape" of a rule set. Then, since we know the rule sizes, we can now fill in the rules with conditions. To generate a condition, we first randomly select the attributes then randomly select values corresponding to each attribute. Let $v_{m,k}$ represent the attribute index for the k-th condition in the m-th rule, $v_{m,k} \in \{1, ...J\}$. $K_{v_{m,k}}$ is the total number of values for attribute $v_{m,k}$. Note that a "value" here refers to a category for a categorical attribute, or an interval for a discretized continuous variable. To summarize, a rule set is constructed as follows:

Draw the number of rules: $M \sim$ Poisson(λ)
for $m \in \{1, ...M\}$ **do**
    Draw the number of conditions in m-th rule: $L_m \sim$ Truncated-Poisson(η), $L_m \in \{1, ...J\}$
    Randomly select L_m attributes from J attributes without replacement
    for $k \in \{1, ...L_m\}$ **do**
        Uniformly at random select a value from $K_{v_{m,k}}$ values corresponding to attribute $v_{m,k}$
    end for
end for

We define the normalization constant $\omega(\lambda, \eta)$, so the probability of generating a rule set A is

$$p(A; \lambda, \eta) = \frac{1}{\omega(\lambda, \eta)} \text{Poisson}(M; \lambda) \prod_{m=1}^{M} \text{Poisson}(L_m; \eta) \frac{1}{\binom{J}{L_m}} \prod_{k=1}^{L_m} \frac{1}{K_{v_{m,k}}}. \tag{5}$$

3.2 Likelihood

Let $A(\mathbf{x}_n)$ denote the classification outcome for $\mathbf{x}_n$, and let y_n denote the observed outcome. Recall that $A(\mathbf{x}_n) = 1$ if $\mathbf{x}_n$ obeys any of the rules $a \in A$. We introduce likelihood parameter ρ_+ to represent the prior probability that $y_n = 1$ when $A(\mathbf{x}_n) = 1$, and ρ_- to represent the prior probability

that $y_n = 0$ when $A(\mathbf{x}_n) = 0$. Thus:

$$y_n|\mathbf{x}_n, A \sim \begin{cases} \text{Bernoulli}(\rho_+) & A(\mathbf{x}_n) = 1 \\ \text{Bernoulli}(1-\rho_-) & A(\mathbf{x}_n) = 0 \end{cases},$$

with ρ_+ and ρ_- drawn from beta priors:

$$\rho_+ \sim \text{Beta}(\alpha_+, \beta_+) \text{ and } \rho_- \sim \text{Beta}(\alpha_-, \beta_-). \tag{6}$$

Based on the classification outcomes, the training data are divided into four cases: true positives (TP $= \sum_n A(\mathbf{x}_n) y_n$), false positives (FP $= \sum_n A(\mathbf{x}_n)(1-y_n)$), true negatives (TN $= \sum_n (1-A(\mathbf{x}_n))(1-y_n)$) and false negatives (FN $= \sum_n (1-A(\mathbf{x}_n))\, y_n$). Then, it can be derived that

$$\begin{aligned} p(S|A;\alpha_+,\beta_+,\alpha_-,\beta_-) &= \int \rho_+^{\text{TP}}(1-\rho_+)^{\text{FP}}\rho_-^{\text{TN}}(1-\rho_-)^{\text{FN}} d\rho_+ d\rho_- \\ &= \frac{B(\text{TP}+\alpha_+, \text{FP}+\beta_+)}{B(\alpha_+,\beta_+)} \frac{B(\text{TN}+\alpha_-, \text{FN}+\beta_-)}{B(\alpha_-,\beta_-)}. \end{aligned} \tag{7}$$

According to (6), $\alpha_+, \beta_+, \alpha_-, \beta_-$ govern the probability that a prediction is correct on training data, which determines the likelihood. To ensure the model achieves the maximum likelihood when all data points are classified correctly, we need $\rho_+ \geq 1-\rho_+$ and $\rho_- \geq 1-\rho_-$. So we choose $\alpha_+, \beta_+, \alpha_-, \beta_-$ such that $\frac{\alpha_+}{\alpha_+ + \beta_+} > 0.5$ and $\frac{\alpha_-}{\alpha_- + \beta_-} > 0.5$. Parameter tuning for the prior parameters will be shown in experiments section. Let H denote a set of parameters for a data S,

$$H = \left\{\alpha_+, \beta_+, \alpha_-, \beta_-, \theta_{\text{prior}}\right\},$$

where θ_{prior} depends on the prior model, our goal is to find an optimal rule set A^* that

$$A^* \in \arg\max_A p(A|S;H). \tag{8}$$

Solving for a MAP model is equivalent to maximizing

$$F(A,S;H) = \log p(A;H) + \log p(S|A;H), \tag{9}$$

where either of the two priors provided above can be used for the first term, and the likelihood is in (7). We write the objective as $F(A)$, the prior probability of selecting A as $p(A)$, and the likelihood as $p(S|A)$, omitting dependence on parameters when appropriate.

4. Approximate MAP Inference

In this section, we describe a procedure for approximately solving for the maximum *a posteriori* (MAP) solution to a BRS model.

Inference in Rule Set modeling is challenging because it involves a search over exponentially many possible sets of rules: since each rule is a conjunction of conditions, the number of rules increases exponentially with the number of conditions, and the number of sets of rules increases exponentially with the number of rules. This is the reason learning a rule set has always been a difficult problem in theory.

Inference becomes easier, however, for our BRS model, due to the computational benefits brought by the Bayesian prior, which can effectively reduce the original problem to a much more manageable size and significantly improve computational efficiency. Below, we detail how to exploit the Bayesian prior to derive useful bounds during any search process. Here we use a stochastic coordinate descent search technique, but the bounds hold regardless of what search technique is used.

4.1 Search Procedure

Given an objective function $F(A)$ over discrete search space of different sets of rules and a temperature schedule function over time steps, $T^{[t]} = T_0^{1-\frac{t}{N_{\text{iter}}}}$, a simulated annealing (Hwang, 1988) procedure is a discrete time, discrete state Markov Chain where at step t, given the current state $A^{[t]}$, the next state $A^{[t+1]}$ is chosen by first randomly selecting a proposal from the neighbors, and the proposal is accepted with probability $\min\left\{1, \exp\left(\frac{F(A^{[t+1]})-F(A^{[t]})}{T^{[t]}}\right)\right\}$, in order to find an optimal rule set A^*. Our version of simulated annealing is similar also to the ϵ-greedy algorithm used in multi-armed bandits (Sutton and Barto, 1998).

Similar to the Gibbs sampling steps used by Letham et al. (2015); Wang and Rudin (2015) for rule-list models, here a "neighboring" solution is a rule set whose edit distance is 1 to the current rule set (one of the rules is different). Our simulated annealing algorithm proposes a new solution via two steps: choosing an action from ADD, CUT and REPLACE, and then choosing a rule to perform the action on.

In the first step, the selection of which rules to ADD, CUT or REPLACE is not chosen uniformly. Instead, the simulated annealing proposes a neighboring solution that aims to do better than the current model with respect to a randomly chosen misclassified point. At iteration $t+1$, an example k is drawn from data points misclassified by the current model $A^{[t]}$. Let $\mathcal{R}_1(\mathbf{x}_k)$ represent a set of rules that $\mathbf{x}_k$ satisfies, and let $\mathcal{R}_0(\mathbf{x}_k)$ represent a set of rules that $\mathbf{x}_k$ does not satisfy. If example k is positive, it means the current rule set fails to cover it so we propose a new model that covers example k, by either adding a new rule from $\mathcal{R}_1(\mathbf{x}_k)$ to $A^{[t]}$, or replacing a rule from $A^{[t]}$ with a rule from $\mathcal{R}_1(\mathbf{x}_k)$. The two actions are chosen with equal probabilities. Similarly, if example k is negative, it means the current rule set covers wrong data, and we then find a neighboring rule set that covers less, by removing or replacing a rule from $A^{[t]} \cap \mathcal{R}_0(\mathbf{x}_k)$, each with probability 0.5.

In the second step, a rule is chosen to perform the action on. To choose the best rule, we evaluate the *precision* of tentative models obtained by performing the selected action on each candidate rule. This precision is:

$$Q(A) = \frac{\sum_i A(\mathbf{x}_i) y_i}{\sum_i A(\mathbf{x}_i)}.$$

Then a choice is made between exploration, which means choosing a random rule (from the ones that improve on the new point), and exploitation, which means choosing the best rule (the one with the highest precision). We denote the probability of exploration as p in our search algorithm. This randomness helps to avoid local minima and helps the Markov Chain to converge to a global minimum. We detail the three actions below.

- ADD

    1. Select a rule z according to $\mathbf{x}_k$

- With probability p, draw z randomly from $\mathcal{R}_1(\mathbf{x}_k)$. (Explore)
- With probability $1-p$, $z = \underset{a \in \mathcal{R}_1(\mathbf{x}_k)}{\arg\max}\, Q(A^{[t]} \cup a)$. (Exploit)

2. Then $A^{[t+1]} \leftarrow A^{[t]} \cup z$.

- CUT

1. Select a rule z according to $\mathbf{x}_k$
   - With probability p, draw z randomly from $A^{[t]} \cap \mathcal{R}_0(\mathbf{x}_k)$. (Explore)
   - With probability $1-p$, $z = \underset{a \in A^{[t]} \cap \mathcal{R}_0(\mathbf{x}_k)}{\arg\max}\, Q(A^{[t]} \backslash a)$. (Exploit)
2. Then $A^{[t+1]} \leftarrow A^{[t]} \backslash z$.

- REPLACE: first CUT, then ADD.

To summarize: the proposal strategy uses an ϵ-greedy strategy to determine when to explore and when to exploit, and when it exploits, the algorithm uses a randomized coordinate descent approach by choosing a rule to maximize improvement.

4.2 Iteratively reducing rule space

Another approach to reducing computation is to directly reduce the set of rules. This dramatically reduces the search space, which is the power set of the set of rules. We take advantage of the Bayesian prior and derive a deterministic bound on MAP BRS models that exclude rules that fail to satisfy the bound.

Since the Bayesian prior is constructed to penalize large models, a BRS model tends to contain a small number of rules. As a small number of rules must cover the positive class, each rule in the MAP model must cover enough observations. We define the number of observations that satisfy a rule as the *support* of the rule:

$$\text{supp}(a) = \sum_{n=1}^{N} \mathbf{1}(a(\mathbf{x}_n) = 1). \tag{10}$$

Removing a rule will yield a simpler model (and a smaller prior) but may lower the likelihood. However, we can prove that the loss in likelihood is bounded as a function of the support. For a rule set A and a rule $z \in A$, we use $A_{\backslash z}$ to represent a set that contains all rules from A except z, i.e.,

$$A_{\backslash z} = \{a | a \in A, a \neq z\}.$$

Define

$$\Upsilon = \frac{\beta_-(N_+ + \alpha_+ + \beta_+ - 1)}{(N_- + \alpha_- + \beta_-)(N_+ + \alpha_+ - 1)}.$$

Then the following holds:

Lemma 1 $p(S|A) \geq (\Upsilon)^{\text{supp}(z)} p(S|A_{\backslash z})$.

All proofs in this paper are in Appendix A. This lemma shows that if $\Upsilon \leq 1$, removing a rule with high support lowers the likelihood.

We wish to derive lower bounds on the support of rules in a MAP model. This will allow us to remove (low-support) rules from the search space without any loss in posterior. This will simultaneously yield an upper bound on the maximum number of rules in a MAP model. The bounds will be updated as the search proceeds.

Recall that $A^{[\tau]}$ denotes the rule set at time τ, and let $v^{[t]}$ denote the best solution found until iteration t, i.e.,

$$v^{[t]} = \max_{\tau \leq t} F(A^{[\tau]}).$$

We first look at BRS-BetaBinomial models. In a BRS-BetaBinomial model, rules are selected based on their lengths, so there are separate bounds for rules of different lengths, and we notate the upper bounds of the number of rules at step t as $\{m_l^{[t]}\}_{l=1,\ldots,L}$, where $m_l^{[t]}$ represents the upper bound for the number of rules of length l.

We then introduce some notations that will be used in the theorems. Let $\mathcal{L}^*$ denote the maximum likelihood of data S, which is achieved when all data are classified correctly (this holds when $\alpha_+ > \beta_+$ and $\alpha_- > \beta_-$), i.e. TP $= N_+$, FP $= 0$, TN $= N_-$, and FN $= 0$, giving:

$$\mathcal{L}^* := \max_A p(S|A) = \frac{B(N_+ + \alpha_+, \beta_+)}{B(\alpha_+, \beta_+)} \frac{B(N_- + \alpha_-, \beta_-)}{B(\alpha_-, \beta_-)}. \tag{11}$$

For a BRS-BetaBinomial model, the following is true.

Theorem 1 *Take a BRS-BetaBinomial model with parameters*

$$H = \{L, \alpha_+, \beta_+, \alpha_-, \beta_-, \{\mathcal{A}_l, \alpha_l, \beta_l\}_{l=1,..L}\},$$

where $L, \alpha_+, \beta_+, \alpha_-, \beta_-, \{\alpha_l, \beta_l\}_{l=1,\ldots L} \in \mathbb{N}^+$, $\alpha_+ > \beta_+, \alpha_- > \beta_-$ *and* $\alpha_l < \beta_l$. *Define* $A^* \in \arg\max_A F(A)$ *and* $M^* = |A^*|$. *If* $\Upsilon \leq 1$, *we have:* $\forall a \in A^*$, *supp*$(a) \geq C^{[t]}$ *and*

$$C^{[t]} = \left\lceil \frac{\log \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t-1]} + \beta_l}{m_l^{[t-1]} - 1 + \alpha_l} \right)}{\log \Upsilon} \right\rceil, \tag{12}$$

where

$$m_l^{[t]} = \left\lfloor \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log \frac{|\mathcal{A}_l| + \beta_l - 1}{\alpha_l + m_l^{[t-1]} - 1}} \right\rfloor,$$

and $m_l^{[0]} = \left\lfloor \frac{\log \mathcal{L}^* - \log p(S|\emptyset)}{\log \frac{|\mathcal{A}_l| + \beta_l - 1}{\alpha_l + |\mathcal{A}_l| - 1}} \right\rfloor$ *and* $v^{[0]} = F(\emptyset)$. *The size of* A^* *is upper bounded by*

$$M^* \leq \sum_{l=1}^{L} m_l^{[t]}.$$

In the equation for $m_l^{[t]}$, $p(\emptyset)$ is the prior of an empty set, which is also the maximum prior for a rule set model, which occurs when we set $\alpha_l = 1$ for all l. Since $\mathcal{L}^*$ and $p(\emptyset)$ upper bound the likelihood and prior, respectively, $\log \mathcal{L}^* + \log p(\emptyset)$ upper bounds the optimal objective value. The difference between this value and $v^{[t]}$, the numerator, represents the room for improvement from the current solution $v^{[t]}$. The smaller the difference, the smaller the $m_l^{[t]}$, and thus the larger the minimum support bound. In the extreme case, when an empty set is the optimal solution, i.e., the likelihood and the prior achieve the upper bound $\mathcal{L}^*$ and $p(\emptyset)$ at the same time, and the optimal solution is discovered at time t, then the numerator becomes zero and $m_l^{[t]} = 0$ which precisely refers to an empty set. Additionally, parameters α_l and β_l jointly govern the probability of selecting rules of length l, according to formula (3). When α_l is set to be small and β_l is set to be large, p_l is small since $\mathbb{E}(p_l) = \frac{\alpha_l}{\alpha_l+\beta_l}$. Therefore the resulting $m_l^{[t]}$ is small, guaranteeing that the algorithm will choose a smaller number of rules overall.

Specifically, at step 0 when there is no $m_l^{[t-1]}$, we set $m_l^{[t-1]} = |\mathcal{A}_l|$, yielding

$$m_l^{[0]} = \left\lfloor \frac{\log \mathcal{L}^* - \log p(S|\emptyset)}{\log \frac{|\mathcal{A}_l|+\beta_l-1}{\alpha_l+|\mathcal{A}_l|-1}} \right\rfloor, \tag{13}$$

which is the upper bound we obtain before we start learning for A^*. This bound uses an empty set as a comparison model, which is a reasonable benchmark since the Bayesian prior favors smaller models. (It is possible, for instance, that the empty model is actually close to optimal, depending on the prior.) As $\log p(S|\emptyset)$ increases, $m_l^{[0]}$ becomes smaller, which means the model's maximum possible size is smaller. Intuitively, if an empty set already achieves high likelihood, adding rules will often hurt the prior term and achieve little gain on the likelihood.

Using this upper bound, we get a minimum support which can be used to reduce the search space before simulated annealing begins. As $m^{[t]}$ decreases, the minimum support increases since the rules need to jointly cover positive examples. Also, if fewer rules are needed it means each rule needs to cover more examples.

Similarly, for BRS-Poisson models, we have:

Theorem 2 *Take a BRS-Poisson model with parameters*

$$H = \{L, \alpha_+, \beta_+, \alpha_-, \beta_-, \lambda, \eta\},$$

where $L, \alpha_+, \beta_+, \alpha_-, \beta_- \in \mathbb{N}^+$, $\alpha_+ > \beta_+, \alpha_- > \beta_-$. *Define* $A^* \in \arg\max_A F(A)$ *and* $M^* = |A^*|$. *If* $\Upsilon \leq 1$, *we have:* $\forall a \in A^*$, *supp*$(a) \geq C^{[t]}$ *and*

$$C^{[t]} = \left\lceil \frac{\log \frac{x}{M^{[t]}}}{\log \Upsilon} \right\rceil, \tag{14}$$

where

$$x = \frac{e^{-\eta}\lambda \max\left\{\left(\frac{\eta}{2}\right)\Gamma(J+1), \left(\frac{\eta}{2}\right)^J\right\}}{\Gamma(J+1)},$$

and

$$M^{[t]} = \left\lfloor \frac{\log \frac{(\lambda+1)^\lambda}{\Gamma(\lambda+1)}}{\log \frac{\lambda+1}{x}} + \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log \frac{\lambda+1}{x}} \right\rfloor.$$

The size of A^* is upper bounded by

$$M^* \leq M^{[t]}.$$

Similar to Theorem 1, $\log \mathcal{L}^* + \log p(\emptyset)$ upper bounds the optimal objective value and $\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}$ is an upper bound on the room for improvement from the current solution $v^{[t]}$. The smaller the bounds becomes, the larger is the minimum support, thus reducing the rule space iteratively whenever a new maximum solution is discovered.

The algorithm, which applies the bounds above, is given as Algorithm 1.

Algorithm 1 Inference algorithm.

procedure SIMULATED ANNEALING(N_{iter})
  $\mathcal{A} \leftarrow$ FP-Growth(S)
  $A^{[0]} \leftarrow$ a randomly generated rule set
  $A_{\text{max}} \leftarrow A^{[0]}$
  for $t = 0 \rightarrow N_{\text{iter}}$ **do**
    $\mathcal{A} \leftarrow \{a \in \mathcal{A}, \text{supp}(a) \geq C^{[t]}\}$ ($C^{[t]}$ is from (12) or (14), depending on the model choice)
    $S_\epsilon \leftarrow$ misclassified examples by $A^{[t]}$ (Find misclassified examples)
    $(\mathbf{x}_k, y_k) \leftarrow$ a random example drawn from S_ϵ
    if $y_k = 1$ **then**
      $$\text{action} \leftarrow \begin{cases} \text{ADD, with probability } 0.5 \\ \text{REPLACE, with probability } 0.5 \end{cases}$$
    else
      $$\text{action} \leftarrow \begin{cases} \text{CUT, with probability } 0.5 \\ \text{REPLACE, with probability } 0.5 \end{cases}$$
    $A^{[t+1]} \leftarrow \text{action}(\mathcal{A}, A^{[t]}, \mathbf{x}_k)$
    $A_{\text{max}} \leftarrow \arg\max(F(A_{\text{max}}), F(A^{[t+1]}))$ (Check for improved optimal solution)
    $\alpha = \min\left\{1, \exp\left(\frac{F(A^{[t+1]}) - F(A^{[t]})}{T^{[t]}}\right)\right\}$ (Probability of an annealing move)
    $$A^{[t+1]} \leftarrow \begin{cases} A^{[t+1]}, \text{ with probability } \alpha \\ A^{[t]}, \text{ with probability } 1-\alpha \end{cases}$$
  end for
  return A_{max}
end procedure

4.3 Rule Mining

We mine a set of candidate rules $\mathcal{A}$ before running the search algorithm and only search within $\mathcal{A}$ to create an approximate but more computationally efficient inference algorithm. For categorical attributes, we consider both positive associations (e.g., x_j='blue') and negative associations (x_j='not green') as conditions. (The importance of negative conditions is stressed, for instance, by Brin et al., 1997; Wu et al., 2002; Teng et al., 2002). For numerical attributes, we create a set of binary variables for each numerical attribute, by comparing it with a set of thresholds (usually quantiles), and add their negations as separate attributes. For example, we discretize age with three thresholds, creating six binary variables in total, i.e., age$\geq$25, age$\geq$50, age$\geq$ 75 and the negations for each of

them, i.e., age<25, age<50 and age<75. We then mine for frequent rules within the set of positive observations S^+. To do this, we use the FP-growth algorithm (Borgelt, 2005), which can in practice be replaced with any desired frequent rule-mining method.

Sometimes for large data sets with many features, even when we restrict the length of rules and the minimum support, the number of generated rules could still be too large to handle, as it can be exponential in the number of attributes. For example, for one of the advertisement data sets, a million rules are generated when the minimum support is 5% and the maximum length is 3. In that case, we use a second criterion to screen for the most potentially useful M_0 rules, where M_0 is user-defined and depends on the user's computational capacity. We first filter out rules on the lower right plane of ROC space, i.e., their false positive rate is greater than true positive rate. Then we use *information gain* to screen rules, similarly to other works (Chen et al., 2006; Cheng et al., 2007). For a rule a, the information gain is

$$\text{InfoGain}(S|a) = H(S) - H(S|a), \tag{15}$$

where $H(S)$ is the entropy of the data and $H(S|a)$ is the conditional entropy of data that split on rule a. Given a data set S, entropy $H(S)$ is constant; therefore our screening technique chooses the M_0 rules that have the smallest $H(S|a)$.

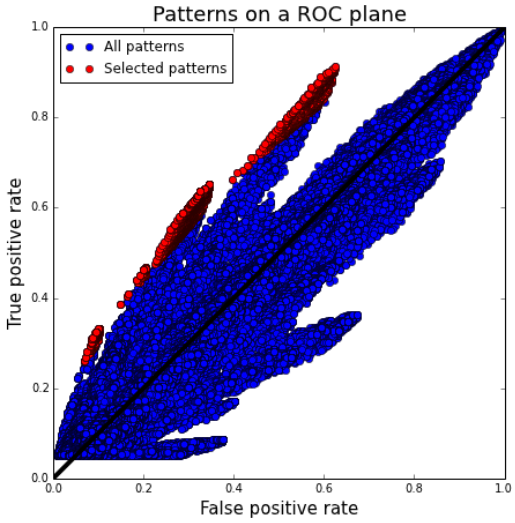


Figure 2: All rules and selected rules on a ROC plane

We illustrate the effect of screening on one of our mobile advertisement data sets. We mined all rules with minimum support 5% and maximum length 3. For each rule, we computed its true positive rate and false positive rate on the training data and plotted it as a dot in Figure 2. The top 5000 rules with highest information gains are colored in red, and the rest are in blue. As shown in the figure, information gain indeed selected good rules as they are closer to the upper left corner in ROC space. For many applications, this screening technique is not needed, and we can simply use the entire set of pre-mined rules that have support above the required threshold for the optimal solution provided in the theorems above.

5. Simulation Studies

We present simulation studies to show the interpretability and accuracy of our model and the efficiency of the simulated annealing procedure. We first demonstrate the efficiency of the simulated annealing algorithm by searching within candidate rules for a "true rule set" that generates the data. Simulations show that our algorithm can recover the true rule set with high probability within a small number of iterations despite the large search space. We then designed the second set of simulations to study the trade-off between accuracy and simplicity. BRS models may lose accuracy to gain model simplicity when the number of attributes is large. Combining the two simulations studies, we were able to hypothesize that possible losses in accuracy are often due to the choice of model representation as a rule set, rather than simulated annealing. In this section, we choose the BRS-BetaBinomial model for the experiments. BRS-Poisson results would be similar; the only difference is the choice of prior.

5.1 Simulation Study 1: Efficiency of Simulated Annealing

In the first simulation study, we would like to test if simulated annealing is able to recover a true rule set given that these rules are in a candidate rule set, and we would like to know how efficiently simulated annealing finds them. Thus we omit the step of rule mining for this experiment and directly work with generated candidate rules, which we know in this case contains a true rule set.

Let there be N observations, $\{\mathbf{x}_n\}_{n=1,...,N}$ and a collection of M candidate rules, $\{a_m\}_{m=1,...M}$. We can construct an $N \times M$ binary matrix where the entry in the n-th row and m-th column is the Boolean function $a_m(x_n)$ that represents whether the n-th observation satisfies the m-th rule. Let the true rule set be a subset of the candidate rules, $A^* \subset \{a_m\}_{m=1,...M}$. The labels $\{y_n\}_{n=1,...N}$ satisfy

$$y_n = \begin{cases} 1 & \exists a \in A^*, a(\mathbf{x}_n) = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

We ran experiments with varying N and M. Each time, we first generated an $N \times M$ binary matrix by setting the entries to 1 with probability 0.035, and then selected 20 columns to form A^*. (Values 0.035 and 20 are chosen so that the positive class contains roughly 50% of the observations. We also ensured that the 20 rules do not include each other.) Then $\{y_n\}_{n=1,...N}$ were generated as above. We randomly assigned lengths from 1 to 5 to $\{a_m\}_{m=1,...M}$. Finally, we ran Algorithm 1 on the data set. The prior parameters were set as below for all simulations:

$$\alpha_l = 1, \beta_l = |\mathcal{A}_l| \text{ for } l = 1, ..., 3,$$
$$\alpha_+ = \alpha_- = 1000, \beta_+ = \beta_- = 1.$$

For each pair of N and M, we repeated the experiment 100 times and recorded intermediate output models at different iterations. Since multiple different rule sets can cover the same points, we compared labels generated from the true rule set and the recovered rule set: a label is 1 if the instance satisfies the rule set and 0 otherwise. We recorded the training error for each intermediate output model. The mean and standard deviation of training error rate are plot Figure 3, along with run time to convergence in seconds on the right figure.

Comparing the three sets of experiments, we notice that different sizes of the binary matrix led to differing convergence times due to different computation cost of handling the matrices, yet the three error rate curves in Figure 3 almost completely overlap. Neither the size of the data set N nor

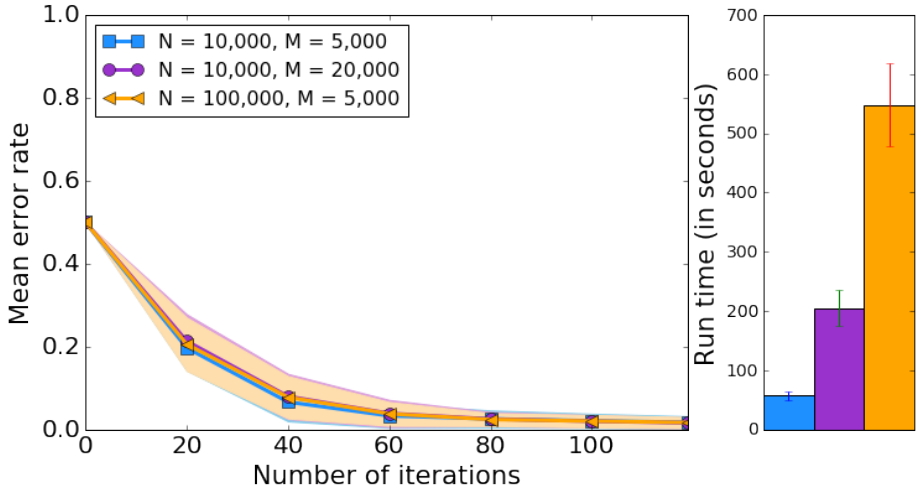


Figure 3: Convergence of mean error rate with number of iterations for data with varying N and M and running time (in seconds), obtained by the BRS-BetaBinomial.

the size of the rule space M affected the search. This is because the strategy based on curiosity and bounds chose the best solution efficiently, regardless of the size of data or the rule space.

More importantly, this study shows that simulated annealing is able to recover a true rule set A^*, or an equivalent rule set with optimal performance, with high probability, with few iterations.

5.2 Simulation Study 2: Accuracy-Interpretability Trade-off

We observe from the first simulation study that simulated annealing does not tend to cause losses in accuracy. In the second simulation study, we would like to identify whether the rule representation causes losses in accuracy. In this study, we used the rule mining step and directly worked with data $\{\mathbf{x}_n\}_{n=1,\ldots,N}$. Without loss of generality, we assume $\mathbf{x}_n \in \{0,1\}^J$.

In each experiment, we first constructed an $N \times J$ binary matrix $\{\mathbf{x}_n\}_{n=1,\ldots N}$ by setting the entries to 1 with probability 0.5, then randomly generated 20 rules from $\{\mathbf{x}_n\}_{n=1,\ldots,N}$ to form A^* and finally generated labels $\{y_n\}_{n=1,\ldots,N}$ following formula (16). We checked the balance of the data before pursuing the experiment. The data set was used only if the positive class was within $[30\%, 70\%]$ of the total data, and otherwise regenerated.

To constrain computational complexity, we set the maximum length of rules to be 7 and then selected the top 5000 rules with highest information gain. Then we performed Algorithm 1 on these candidate rules to generate a BRS model $\tilde{A}$. Ideally, $\tilde{A}$ should be simpler than A^*, with a possible loss in its accuracy as a trade-off.

To study the influence of the dimensions of data on accuracy and simplicity, we varied N and J and repeated each experiment 100 times. Figure 4 shows the accuracy and number of rules in $\tilde{A}$ in each experiment. The number of rules in $\tilde{A}$ was almost always less than the number of rules in A^* (which was 20). On average, $\tilde{A}$ contained 12 rules, which was 60% of the size of A^*, slightly compromising accuracy. BRS needed to compromise more when J was large since it became harder to maintain the same level of simplicity.

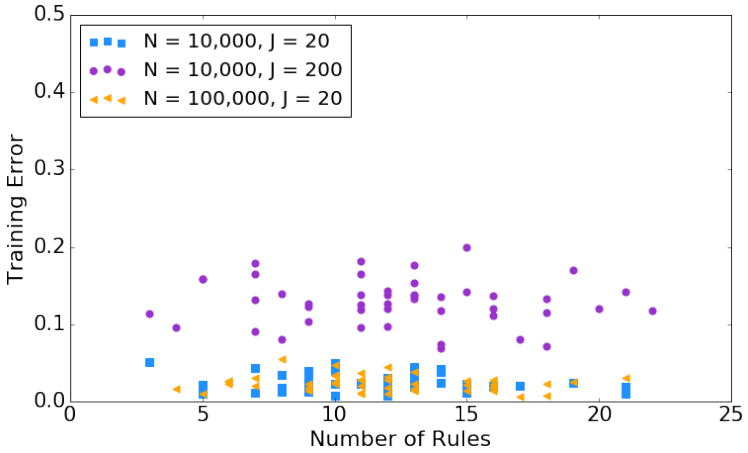


Figure 4: Accuracy and complexity of output BRS models for data with varying N and J, obtained from the BRS-BetaBinomial.

We show in Figure 5 the average training error at different iterations to illustrate the convergence rate. Simulated annealing took less than 50 iterations to converge. Note that it took fewer iterations than simulation study 1 since the error rate curve started from less than 0.5, due to rule pre-screening using information gain.

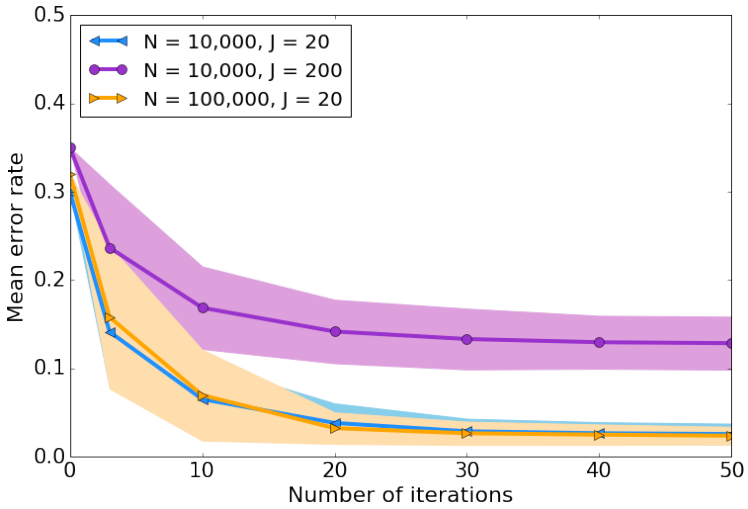


Figure 5: Convergence of mean error rate with varying N and J, obtained by BRS-BetaBinomial.

Figure 4 and Figure 5 both indicate that increasing the number of observations did not affect the performance of the model. BRS is more sensitive to the number of attributes. The larger J is, the less informative a short rule becomes to describe and distinguish a class in the data set, and the more BRS has to trade-off, in order to keep simple rules within the model.

To summarize: when we model using fewer rules, it benefits interpretability but can lose information. Rules with a constraint on the maximum length tend to be less accurate when dealing with

high dimensional feature spaces. The loss increases with the number of features. This is the price paid for using an interpretable model. There is little price paid for using our optimization method.

6. Experiments

Our main application of interest is mobile advertising. We collected a large advertisement data set using Amazon Mechanical Turk that we made publicly available (Wang et al., 2015), and we tested BRS on other publicly available data sets. We show that our model can generate interpretable results that are easy for humans to understand, without losing too much (if any) predictive accuracy. In situations where the ground truth consists of deterministic rules (similarly to the simulation study), our method tends to perform better than other popular machine learning techniques.

6.1 UCI Data Sets

We test BRS on ten data sets from the UCI machine learning repository (Bache and Lichman, 2013) with different types of attributes; four of them contain categorical attributes, three of them contain numerical attributes and three of them contain both. We performed 5-fold cross validation on each data set. Since our goal was to produce models that are accurate and interpretable, we set the maximum length of rules to be 3 for all experiments. We compared with interpretable algorithms: Lasso (without interaction terms to preserve interpretability), decision trees (C4.5 and CART), and inductive rule learner RIPPER (Cohen, 1995), which produces a rule list greedily and anneals locally afterward. In addition to interpretable models, we also compare BRS with uninterpretable models from random forests and SVM to set a benchmark for expected accuracy on each data set. We used implementations of baseline methods from R packages, and tuned model parameters via 5-fold nested cross validation.

Table 1: Accuracy of each comparing algorithm (mean ± std) on ten UCI data sets.

	Data Type	**BRS1**	**BRS2**	Lasso	C4.5	CART	RIPPER	Random Forest	SVM
connect-4	Categorical	.76±.01	.75±.01	.70±.00	.83±.00	.69±.01	———	.86±.00	.82±.00
mushroom		1.00±.00	1.00±.00	.96±.00	1.00±.00	.99±.00	.99±.00	1.00±.00	.93±.00
tic-tac-toe		1.00±.00	1.00±.00	.71±.02	.92±.03	.93±.02	.98±.01	.99±.00	.99±.00
chess		.89±.01	.89±.01	.83±.01	.97±.00	.92±.04	.91±.01	.91±.00	.99±.00
magic	Numerical	.80±.02	.79±.01	.76±.01	.79±.00	.77±.00	.78±.00	.78±.01	.78±.01
banknote		.97±.01	.97±.01	1.00±.00	.90±.01	.90±.02	.91±.01	.91±.01	1.00±.00
indian-diabetes		.72±.03	.73±.03	.67±.01	.66±.03	.67±.01	.67±.02	.75±.03	.69±.01
adult	Mixed	.83±.00	.84±.01	.82±.02	.83±.00	.82±.01	.83±.00	.84±.00	.84±.00
bank-marketing		.91±.00	.91±.00	.90±.00	.90±.00	.90±.00	.90±.00	.90±.00	.90±.00
heart		.85±.03	.84±.03	.85±.04	.76±.06	.77±.06	.78±.04	.81±.06	.86±.06
Rank of accuracy		2.6±1.7	2.6±1.8	5.4±2.8	4.3±2. 9	5.6±1.8	4.8±1.5	2.7±1.7	2.8±2.2

Table 1 displays the means and standard deviations of the test accuracy for all algorithms (the lead author's laptop ran out of memory when applying RIPPER to the connect-4 data set),

Table 2: Runtime (in seconds) of each comparing algorithm (mean ± std) on ten UCI data sets.

	BRS1	**BRS2**	Lasso	C4.5	CART	RIPPER	Random Forest	SVM
connect-4	288.1±22.2	279.1±20.9	0.9±0.0	30.4±0.6	20.5±0.4	———-	746.6±14.9	1427.8±35.1
mushroom	8.8±1.3	8.18±1.3	0.1±0.0	0.5±0.0	0.6±0.0	1.2±0.1	15.6±0.8	23.5±1.2
tic-tac-toe	1.0±0.0	1.0±0.0	0.0±0.0	0.1±0.0	0.1±0.0	0.3±0.0	1.3±.0.0	0.2±0.0
chess	44.8±8.2	41.6±8.2	0.1±0.0	1.3±0.4	0.3±0.1	50.6±18.0	47.5±1.9	1193.3±79.6
magic	31.3±4.4	29.3±4.4	0.1±0.0	1.4±0.1	2.3±0.2	8.8±1.7	43.5±1.3	139.7±3.2
banknote	1.0±0.1	0.9±0.1	0.0±0.0	0.1±0.0	0.1±0.0	0.2±0.0	1.1±0.0	0.1±0.0
indian-diabetes	0.6±0.1	0.6±0.1	0.0±0.0	0.1±0.0	0.1±0.0	0.2±0.0	0.7±0.0	0.2±0.0
adult	50.2±6.2	47.1±6.2	0.4±0.0	9.8±0.3	10.7±0.4	66.0±8.8	344.4±7.7	174.6±4.7
bank-marketing	55.2±2.7	50.8±2.7	0.4±0.0	11.6±0.6	7.6±0.3	23.9±4.6	320.0±9.8	395.1±28.1
heart	0.5±0.1	0.5±0.1	0.0±0.0	0.1±0.0	0.0±0.0	0.2±0.1	0.3±0.0	0.0±0.0

and the rank of their average performance. Table 2 displays the runtime. BRS1 represents BRS-BetaBinomial and BRS2 represents BRS-Poisson. While BRS models were under strict restriction for interpretability purposes ($L = 3$), BRS's performance surpassed that of the other interpretable methods and was on par with uninterpretable models. This is because, firstly, BRS has Bayesian priors that favor rules with a large support (Theorem 1, 2) which naturally avoids overfitting of the data; and secondly, BRS optimizes a global objective while decision trees and RIPPER rely on local greedy splitting and pruning methods, and interpretable versions of Lasso are linear in the base features. Decision trees and RIPPER are local optimization methods and tend not to have the same level of performance as globally optimal algorithms, such as SVM, Random Forests, BRS, etc. The class of rule set models and decision tree models are the same: both create regions within input space consisting of a conjunction of conditions. The difference is the choice of optimization method: global vs. local.

For the tic-tac-toe data set, the positive class can be classified using exactly eight rules. BRS has the capability to exactly learn these conditions, whereas the greedy splitting and pruning methods that are pervasive throughout the data mining literature (e.g., CART, C4.5) and convexified approximate methods (e.g., SVM) have difficulty with this. Both linear models and tree models exist that achieve perfect accuracy, but the heuristic splitting/pruning and convexification of the methods we compared with prevented these perfect solutions from being found.

BRS achieves accuracies competitive to uninterpretable models while requiring a much shorter runtime, which grows slowly with the size of the data, unlike Random forest and SVM. This indicates that BRS can reliably produce a good model within a reasonable time for large data sets.

6.1.1 PARAMETER TUNING

A MAP solution maximizes the sum of logs of prior and likelihood. The scale of the prior and likelihood directly determines how the model trades off between fitting the data and achieving the desired sparsity level. Again, we choose the BRS-BetaBinomial for this demonstration. The Bayesian model uses parameters $\{\alpha_l, \beta_l\}_{l=1}^{L}$ to govern the prior for selecting rules and parameters $\alpha_+, \beta_+, \alpha_-, \beta_-$ to govern the likelihood of data. We study how sensitive the results are to different

parameters and how to tune the parameters to get the best performing model. We choose one data set from each category from Table 1.

We fixed the prior parameters $\alpha_l = 1, \beta_l = |\mathcal{A}_l|$ for $l = 1, .., L$, and only varied likelihood parameters $\alpha_+, \beta_+, \alpha_-, \beta_-$ to analyze how the performance changes as the magnitude and ratio of the parameters change. To simplify the study, we took $\alpha_+ = \alpha_- = \alpha$, and $\beta_+ = \beta_- = \beta$. We let $s = \alpha + \beta$ and s was chosen from $\{100, 1000, 10000\}$. We let $\rho = \frac{\alpha}{\alpha+\beta}$ and ρ varied within $[0, 1]$. Here, s and ρ uniquely define α and β. We constructed models for different s and ρ and plotted the average out-of-sample accuracy as s and ρ changed in Figure 6 for three data sets representative of categorical, numerical and mixed data types. The X axis represents ρ and Y axis represents s. Accuracies increase as ρ increases, which is consistent with the intuition of ρ_+ and ρ_-. The highest accuracy is always achieved at the right half of the curve. The performance is less sensitive to the magnitude s and ratio ρ, especially when $\rho > 0.5$. For tic-tac-toe and diabetes, the accuracy became flat once ρ becomes greater than a certain threshold, and the performance was not sensitive to ρ either after that point; it is important as it makes tuning the algorithm easy in practice. Generally, taking ρ close to 1 leads to a satisfactory output. Table 1 and 2 were generated by models with $\rho = 0.9$ and $s = 1000$.

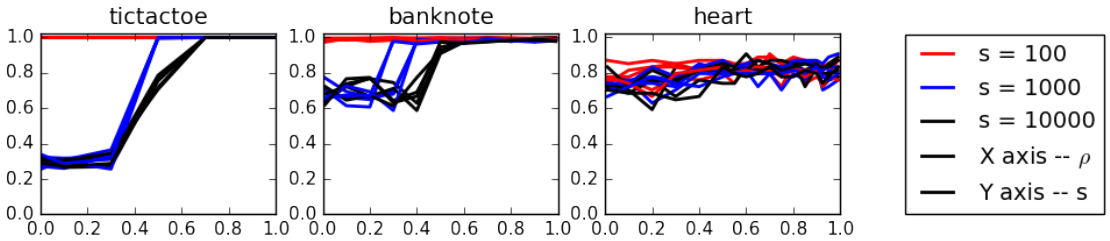


Figure 6: Parameter tuning experiments. Accuracy vs. ρ for all data sets. X axis represents ρ and Y axis represents accuracy.

6.1.2 DEMONSTRATION WITH TIC-TAC-TOE DATA

Let us illustrate the reduction in computation that comes from the bounds in the theorems in Section 4. For this demonstration, we chose the tic-tac-toe data set since the data can be classified exactly by eight rules which are naturally an underlying true rule set, and our method recovered the rules successfully, as shown in Table 1. We use BRS-BetaBinomial for this demonstration.

First, we used FP-Growth (Borgelt, 2005) to generate rules. At this step, we find all possible rules with length between 1 to 9 (the data has 9 attributes) and has minimum support of 1 (the rule's itemset must appear at least once int the data set). FP-growth generates a total of 84429 rules. We then set the hyper-parameters as:

$$\alpha_l = 1, \beta_l = |\mathcal{A}_l| \text{ for } l = 1, ..., 9,$$
$$\alpha_+ = N_+ + 1, \beta_+ = N_+, \alpha_- = N_- + 1, \beta_- = N_-.$$

We ran Algorithm 1 on this data set. We kept a list of the best solutions $v^{[t]}$ found until time t and updated $m_l^{[t-1]}$ and the minimum support with the new $v^{[t]}$ according to Theorem 1. We show in Figure 7 the minimum support at different iterations whenever a better $v^{[t]}$ is obtained. Within

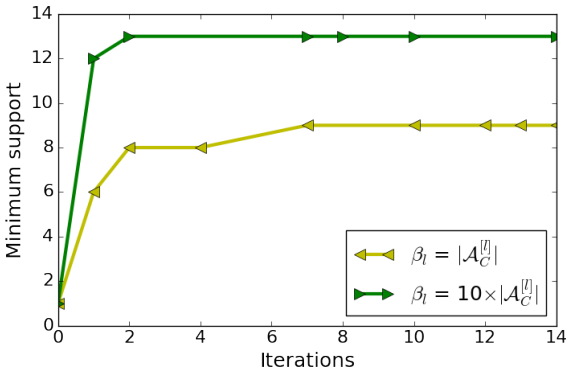
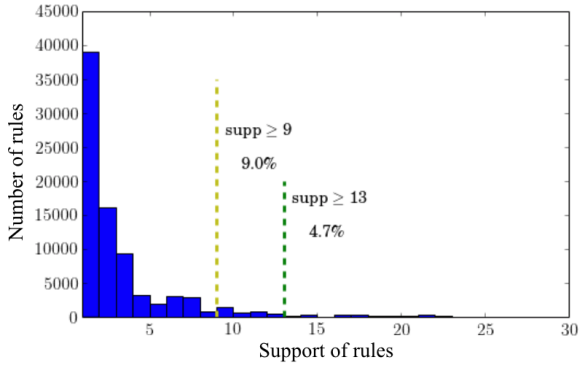


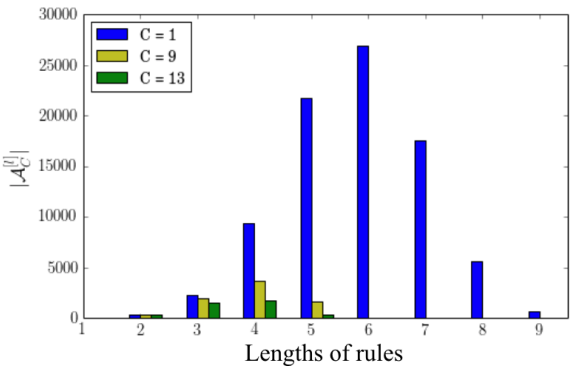
Figure 7: Updated minimum support at different iterations

ten iterations, the minimum support increased from 1 to 9. If, however, we choose a stronger prior for small models by increasing β_l to 10 times $|\mathcal{A}_l|$ for all l and keeping all the other parameters unchanged, then the minimum support increases even faster and reached 13 very quickly. This is consistent with the intuition that if the prior penalizes large models more heavily, the output tends to have a smaller size. Therefore each rule will need to cover more data.

Placing a bound on the minimum support is critical for computation since the number of possible rules decays exponentially with the support, as shown in Figure 8a. As the minimum support is increased, more rules are removed from the original rule space. Figure 8a shows the percentage of rules left in the solution space as the minimum threshold is increased from 1 to 9 and to 13. At a minimum support of 9, the rule space is reduced to 9.0% of its original size; at a minimum support of 13, the rule space is reduced to 4.7%. This provides intuition of the benefit of Theorem 1.



(a) Reduced rule space as the minimum support increases



(b) Number of rules of different lengths mined at different minimum support.

Figure 8: Demonstration with Tic-Tac-Toe data set

Although rules are filtered out based on their support, we observe that longer rules are more heavily reduced than shorter rules. This is because the support of a rule is negatively correlated with its length, as the more conditions a rule contains, the fewer observations can satisfy it. Figure 8b shows distributions of rules across different lengths when mined at different minimum support lev-

els. Before any filtering (i.e., C = 1), more than half of the rules have lengths greater than 5. As C is increased to 9 and 13, these long rules are almost completely removed from the rule space.

To summarize, the strength of the prior is important for computation, and changes the nature of the problem: a stronger prior can dramatically reduce the size of the search space. Any increase in minimum support, that the prior provides, leads to a smaller search by the rule mining method and a smaller optimization problem for the BRS algorithm. These are multiple order-of-magnitude decreases in the size of the problem.

6.2 Application to In-vehicle Recommendation System

For this experiment, our goal was to understand customers' response to recommendations made in an in-vehicle recommender system that provides coupons for local businesses. The coupons would be targeted to the user in his/her particular context. Our data were collected on Amazon Mechanical Turk via a survey that we will describe shortly. We used Turkers with high ratings (95% or above). Out of 752 surveys, 652 were accepted, which generated 12684 data cases (after removing rows containing missing attributes).

The prediction problem is to predict if a customer is going to accept a coupon for a particular venue, considering demographic and contextual attributes. Answers that the user will drive there 'right away' or 'later before the coupon expires' are labeled as '$Y = 1$' and answers 'no, I do not want the coupon' are labeled as '$Y = 0$'. We are interested in investigating five types of coupons: bars, takeaway food restaurants, coffee houses, cheap restaurants (average expense below $20 per person), expensive restaurants (average expense between $20 to $50 per person). In the first part of the survey, we asked users to provide their demographic information and preferences. In the second part, we described 20 different driving scenarios (see examples in Appendix B) to each user along with additional context information and coupon information (see Appendix B for a full description of attributes) and asked the user if s/he will use the coupon.

For this problem, we wanted to generate simple BRS models that are easy to understand. So we restricted the lengths of rules and the number of rules to yield very sparse models. Before mining rules, we converted each row of data into an item which is an attribute and value pair. For categorical attributes, each attribute-value pair was directly coded into a condition. Using marital status as an example, 'marital status is single' was converted into (MaritalStatus: Single), (MaritalStatus: Not Married partner), and (MaritalStatus: Not Unmarried partner), (MaritalStatus: Not Widowed). For discretized numerical attributes, the levels are ordered, such as: age is '20 to 25', or '26 to 30', etc.; additionally, each attribute-value pair was converted into two conditions, each using one side of the range. For example, age is '20 to 25' was converted into (Age:>=20) and (Age:<=25). Then each condition is a half-space defined by threshold values. For the rule mining step, we set the minimum support to be 5% and set the maximum length of rules to be 4. We used information gain in Equation (15) to select the best 5000 rules to use for BRS. We ran simulated annealing for 50000 iterations to obtain a rule set.

We compared with interpretable classification algorithms that span the space of widely used methods that are known for interpretability and accuracy, C4.5, CART, Lasso, RIPPER, and a naïve baseline using top K rules, referred to as Top$K$. For C4.5, CART, Lasso, and RIPPER, we used the RWeka package in R and tuned the hyper-parameters to generate different models on a ROC plane. For the Top$K$ method, we varied K from 1 to 10 to produce ten models using best K pre-mined rules ranked by the information gain in Equation (15). For BRS, We varied the hyperparameters

$\alpha_+, \beta_+, \alpha_-, \beta_-$ to obtain different sets of rules. For all methods, we picked models on their ROC frontiers and reported their performance below.

Performance in accuracy To compare their predictive performance, we measured out-of-sample AUC (the Area Under The ROC Curve) from 5-fold testing for all methods, reported in Table 3. The BRS classifiers, while restricted to produce sparse disjunctions of conjunctions, had better performance than decision trees and RIPPER, which use greedy splitting and pruning methods, and do not aim to globally optimize. Top$K$'s performance was substantially below that of other methods. BRS models are also comparable to Lasso, but the form of the model is different. BRS models do not require users to cognitively process coefficients.

	Bar	Takeaway Food	Coffee House	Cheap Restaurant	Expensive Restaurant
BRS1	0.773 (0.013)	0.682 (0.005)	0.760 (0.010)	0.736 (0.022)	0.705 (0.025)
BRS2	0.776 (0.011)	0.667 (0.023)	0.762 (0.007)	0.736 (0.019)	0.707 (0.030)
C4.5	0.757 (0.015)	0.602 (0.051)	0.751 (0.018)	0.692 (0.033)	0.639 (0.027)
CART	0.772 (0.019)	0.615 (0.035)	0.758 (0.013)	0.732 (0.018)	0.657 (0.010)
Lasso	0.795 (0.014)	0.673 (0.042)	0.786 (0.011)	0.769 (0.024)	0.706 (0.017)
RIPPER	0.762 (0.015)	0.623 (0.048)	0.762(0.012)	0.705(0.023)	0.689 (0.034)
Top$K$	0.562 (0.015)	0.523 (0.024)	0.502(0.012)	0.582(0.023)	0.508 (0.011)

Table 3: AUC comparison for mobile advertisement data set, means and standard deviations over folds are reported.

Performance in complexity For the same experiment, we would also like to know the complexity of all methods at different accuracy levels. Since the methods we compare have different structures, there is not a straightforward way to directly compare complexity. However, decision trees can be converted into equivalent rule set models. For a decision tree, an example is classified as positive if it falls into any positive leaf. Therefore, we can generate equivalent models in rule set form for each decision tree, by simply collecting branches with positive leaves. Therefore, the four algorithms we compare, C4.5, CART, RIPPER, and BRS have the same form. To measure the complexity, we count the total number of conditions in the model, which is the sum of lengths for all rules. This loosely represents the cognitive load needed to understand a model. For each method, we take the models that are used to compute the AUC in Table 3 and plot their accuracy and complexity in Figure 9. BRS models achieved the highest accuracy at the same level of complexity. This is not surprising given that BRS performs substantially more optimization than other methods.

To show that the benefits of BRS did not come from rule mining or screening using heuristics, we compared the BRS models with Top$K$ models that rely solely on rule mining and ranking with heuristics. Figure 10 shows there is a substantial gain in the accuracy of BRS models compared to Top$K$ models.

From Table 3 we observe that Lasso achieved consistently good AUC. For the five coupons, the average number of nonzero coefficients for lasso models in different folds are 93, 94.6, 90.2, 93.2 and 93.4, which is on the order of $\sim$20 times larger than the number of conditions used in BRS and other rule based models; in this case, the lasso models are not interpretable.

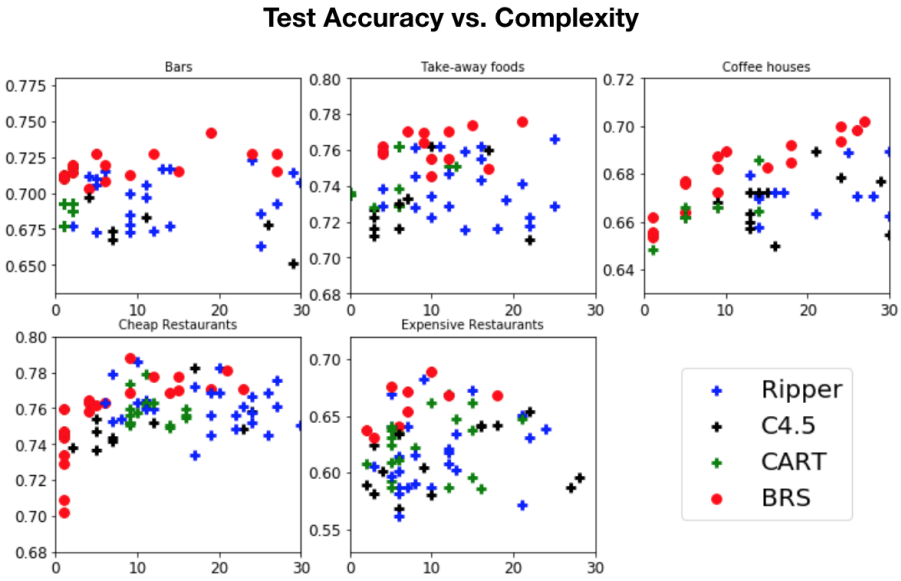


Figure 9: Test accuracy vs complexity for BRS and other models on mobile advertisement data sets for different coupons

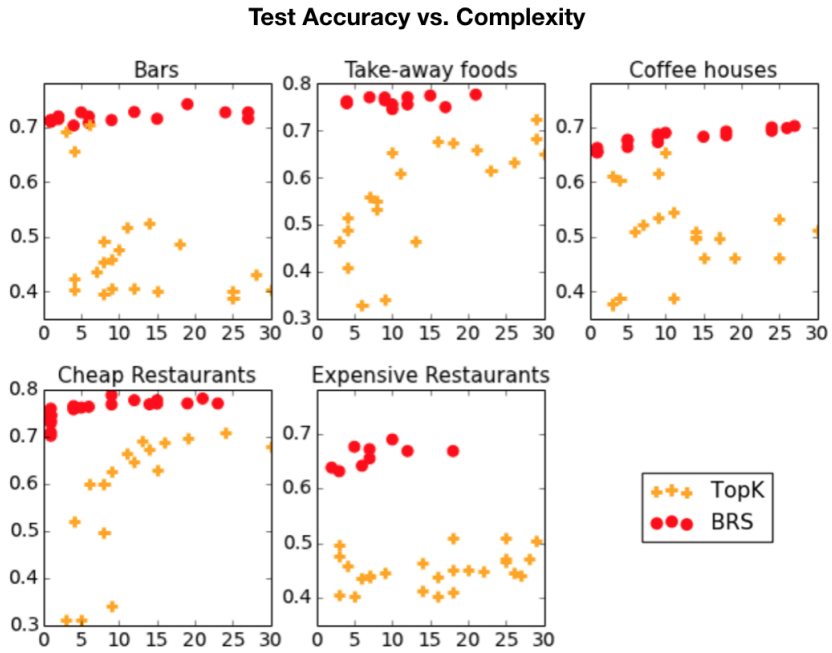
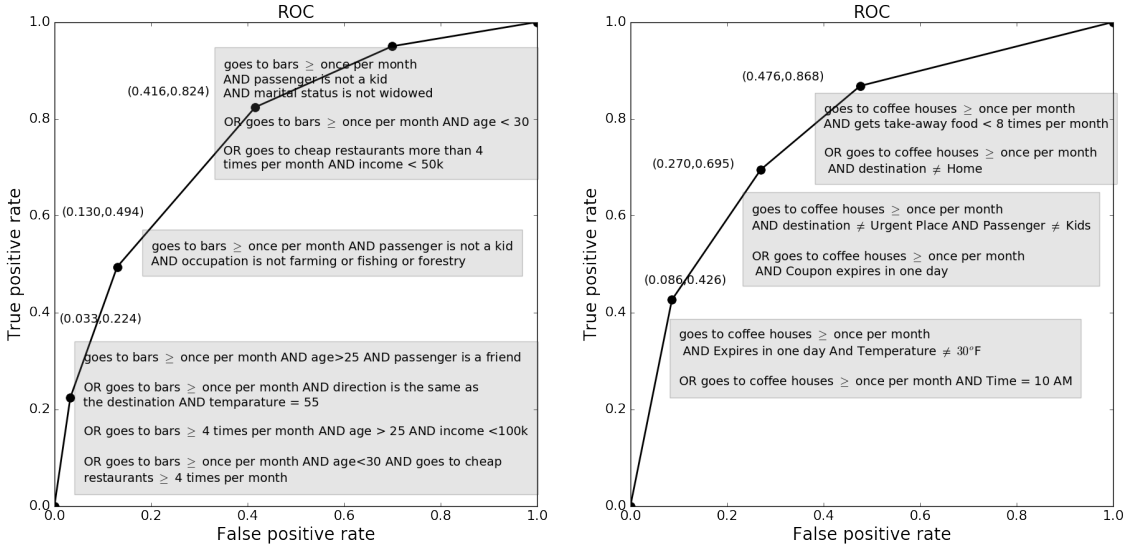


Figure 10: Test accuracy vs complexity for BRS and Top$K$ on mobile advertisement data sets for different coupons

Examples of BRS models In practice, for this particular application, the benefits of interpretability far outweigh small improvements in accuracy. An interpretable model can be useful to a vender

choosing whether to provide a coupon and what type of coupon to provide, it can be useful to users of the recommender system, and it can be useful to the designers of the recommender system to understand the population of users and correlations with successful use of the system. As discussed in the introduction, rule set classifiers are particularly natural for representing consumer behavior, particularly consideration sets, as modeled here.

We show several classifiers produced by BRS in Figure 11, where the curves were produced by models from the experiments discussed above. Example rule sets are listed in each box along the curve. For instance, the classifier near the middle of the curve in Figure 11 (a) has one rule, and reads "If a person visits a bar at least once per month, is not traveling with kids, and their occupation is not farming/fishing/forestry, then predict the person will use the coupon for a bar before it expires." In these examples (and generally), we see that a user's general interest in a coupon's venue (bar, coffee shop, etc.) is the most relevant attribute to the classification outcome; it appears in every rule in the two figures.



(a) Coupons for bars

(b) Coupons for coffee houses

Figure 11: ROC for data set of coupons for bars and coffee houses.

7. Conclusion

We presented a method that produces rule set (disjunctive normal form) models, where the shape of the model can be controlled by the user through Bayesian priors. In some applications, such as those arising in customer behavior modeling, the form of these models may be more useful than traditional linear models. Since finding sparse models is computationally hard, most approaches take severe heuristic approximations (such as greedy splitting and pruning in the case of decision trees, or convexification in the case of linear models). These approximations can severely hurt performance, as is easily shown experimentally, using data sets whose ground truth formulas are not difficult to find. We chose a different type of approximation, where we make an up-front statistical assumption in building our models out of pre-mined rules, and aim to find the globally optimal

solution in the reduced space of rules. We then find theoretical conditions under which using pre-mined rules provably does not change the set of MAP optimal solutions. These conditions relate the size of the data set to the strength of the prior. If the prior is sufficiently strong and the data set is not too large, the set of pre-mined rules is provably sufficient for finding an optimal rule set. We showed the benefits of this approach on a consumer behavior modeling application of current interest to "connected vehicle" projects. Our results, using data from an extensive survey taken by several hundred individuals, show that simple rules based on a user's context can be directly useful in predicting the user's response.

Appendix A.

Proof 1 *(of Lemma 1) Let TP, FP, TN, and FN be the number of true positives, false positives, true negatives and false negatives in S classified by A. We now compute the likelihood for model $A_{\setminus z}$. The most extreme case is when rule z is an 100% accurate rule that applies only to real positive data points and those data points satisfy only z. Therefore once removing it, the number of true positives decreases by $supp(z)$ and the number of false negatives increases by $supp(z)$. That is,*

$$\begin{aligned} p(S|A_{\setminus z}) \geq & \frac{B(TP - supp(z) + \alpha_+, FP + \beta_+)}{B(\alpha_+, \beta_+)} \frac{B(TN + \alpha_-, FN + supp(z) + \beta_-)}{B(\alpha_-, \beta_-)} \\ = & p(S|A) \cdot g_3(supp(z)), \end{aligned} \tag{17}$$

where

$$g_3(supp(z)) = \frac{\Gamma(TP + \alpha_+ - supp(z))}{\Gamma(TP + \alpha_+)} \frac{\Gamma(TP + FP + \alpha_+ + \beta_+)}{\Gamma(TP + FP + \alpha_+ + \beta_+ - supp(z))} \tag{18}$$

$$\frac{\Gamma(FN + \beta_- + supp(z))}{\Gamma(FN + \beta_-)} \frac{\Gamma(TN + FN + \alpha_- + \beta_-)}{\Gamma(TN + FN + \alpha_- + \beta_- + supp(z))}. \tag{19}$$

Now we break down $g_3(supp(z))$ to find a lower bound for it. The first two terms in (18) become

$$\begin{aligned} & \frac{\Gamma(TP + \alpha_+ - supp(z))}{\Gamma(TP + \alpha_+)} \frac{\Gamma(TP + FP + \alpha_+ + \beta_+)}{\Gamma(TP + FP + \alpha_+ + \beta_+ - supp(z))} \\ = & \frac{(TP + FP + \alpha_+ + \beta_+ - supp(z)) \ldots (TP + FP + \alpha_+ + \beta_+ - 1)}{(TP + \alpha_+ - supp(z)) \ldots (TP + \alpha_+ - 1)} \\ \geq & \left(\frac{TP + FP + \alpha_+ + \beta_+ - 1}{TP + \alpha_+ - 1} \right)^{supp(z)} \\ \geq & \left(\frac{N_+ + \alpha_+ + \beta_+ - 1}{N_+ + \alpha_+ - 1} \right)^{supp(z)}. \end{aligned} \tag{20}$$

Equality holds in (20) when TP $= N_+, FP = 0$*. Similarly, the last two terms in (18) become*

$$\begin{aligned}
&\frac{\Gamma(FN+\beta_-+supp(z))}{\Gamma(FN+\beta_-)}\frac{\Gamma(TN+FN+\alpha_-+\beta_-)}{\Gamma(TN+FN+\alpha_-+\beta_-+supp(z))}\\
=&\frac{(FN+\beta_-)\dots(FN+\beta_-+supp(z)-1)}{(TN+FN+\alpha_-+\beta_-)\dots(TN+FN+\alpha_-+\beta_-+supp(z)-1)}\\
\geq&\left(\frac{FN+\beta_-}{FN+TN+\alpha_-+\beta_-}\right)^{supp(z)}\\
\geq&\left(\frac{\beta_-}{N_-+\alpha_-+\beta_-}\right)^{supp(z)}. \qquad (21)
\end{aligned}$$

Equality in (21) holds when TN $= N_-, FN = 0$*. Combining (17), (18), (20) and (21), we obtain*

$$\begin{aligned}
P(S|A_{\setminus z}) &\geq \left(\frac{N_+ + \alpha_+ + \beta_+ - 1}{N_+ + \alpha_+ - 1}\frac{\beta_-}{N_- + \alpha_- + \beta_-}\right)^{supp(z)} \cdot p(S|A)\\
&= \Upsilon^{supp(z)} p(S|A). \qquad (22)
\end{aligned}$$

Proof 2 *(of Theorem 1)*

Step 1 *We first prove the upper bound* $m_l^{[t]}$*. Since* $A^* \in \arg\max_A F(A)$*,* $F(A^*) \geq v^{[t]}$*, i.e.,*

$$\log p(S|A^*) + \log p(A^*) \geq v^{[t]}. \qquad (23)$$

We then upper bound the two terms on the left-hand-side.

Let M_l^* *denote the number of rules of length* l *in* A^**. The prior probability of selecting* A^* *from* $\mathcal{A}$ *is*

$$p(A^*) = p(\emptyset)\prod_{l=1}^{L}\frac{B(M_l+\alpha_l, |\mathcal{A}_l| - M_l + \beta_l)}{B(\alpha_l, |\mathcal{A}_l|+\beta_l)}.$$

We observe that when $0 \leq M_l^* \leq |\mathcal{A}_l|$*,*

$$B(M_l+\alpha_l, |\mathcal{A}_l| - M_l + \beta_l) \leq B(\alpha_l, |\mathcal{A}_l|+\beta_l),$$

giving

$$\begin{aligned}
p(A^*) &\leq p(\emptyset)\frac{B(M_{l'}+\alpha_{l'}, |\mathcal{A}_{l'}| - M_{l'} + \beta_{l'})}{B(\alpha_{l'}, |\mathcal{A}_{l'}|+\beta_{l'})}\\
&= p(\emptyset)\frac{\alpha_{l'}\cdot(\alpha_{l'}+1)\cdots(M_{l'}^*+\alpha_{l'}-1)}{(|\mathcal{A}_{l'}| - M_{l'}^* + \beta_{l'})\cdots(|\mathcal{A}_{l'}|+\beta_{l'}-1)}\\
&\leq p(\emptyset)\left(\frac{M_{l'}^*+\alpha_{l'}-1}{|\mathcal{A}_{l'}|+\beta_{l'}-1}\right)^{M_{l'}^*} \qquad (24)
\end{aligned}$$

for all $l' \in \{1, ..., L\}$*. The likelihood is upper bounded by*

$$p(S|A^*) \leq \mathcal{L}^*. \qquad (25)$$

Substituting (24) and (25) into (23) we get

$$\log \mathcal{L}^* + \log p(\emptyset) + M_{l'}^* \log\left(\frac{M_{l'}^*+\alpha_{l'}-1}{|\mathcal{A}_{l'}|+\beta_{l'}-1}\right) \geq v^{[t]}, \qquad (26)$$

which gives

$$M_{l'}^* \leq \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log\left(\frac{|\mathcal{A}_{l'}|+\beta_{l'}-1}{M_{l'}^*+\alpha_{l'}-1}\right)} \leq \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log\left(\frac{|\mathcal{A}_{l'}|+\beta_{l'}-1}{m_{l'}^{[t-1]}+\alpha_{l'}-1}\right)}, \tag{27}$$

where (27) follows because $m_{l'}^{[t-1]}$ is an upper bound on $M_{l'}^*$. Since $M_{l'}^*$ has to be an integer, we get

$$M_{l'}^* \leq \left\lfloor \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log\left(\frac{|\mathcal{A}_{l'}|+\beta_{l'}-1}{m_{l'}^{[t-1]}+\alpha_{l'}-1}\right)} \right\rfloor = m_{l'}^{[t]} \text{ for all } l' \in \{1,...,L\}. \tag{28}$$

Thus

$$M^* = \sum_{l=1}^{L} M_l^* \leq \sum_{l=1}^{L} m_l^{[t]}. \tag{29}$$

Step 2: *Now we prove the lower bound on the support. We would like to prove that a MAP model does not contain rules of support less than a threshold. To show this, we prove that if any rule z has support smaller than some constant C, then removing it yields a better objective, i.e.,*

$$F(A) \leq F(A_{\setminus z}). \tag{30}$$

Our goal is to find conditions on C such that this inequality holds. Assume in rule set A, M_l comes from pool $\mathcal{A}_l$ of rules with length l, $l \in \{1,...L\}$, and rule z has length l' so it is drawn from $\mathcal{A}_{l'}$.

$A_{\setminus z}$ consists of the same rules as A except missing one rule from $\mathcal{A}_{l'}$. We must have:

$$\begin{aligned} p(A_{\setminus z}) =& \frac{B(M_{l'}+\alpha_{l'}, |\mathcal{A}_{l'}| - M_{l'} + \beta_{l'})}{B(\alpha_{l'},\beta_{l'})} \cdot \prod_{l\neq l'} \frac{B(M_l+\alpha_l, |\mathcal{A}_l| - M_l + \beta_l)}{B(\alpha_l,\beta_l)} \\ =& p(A) \cdot \frac{B(M_{l'}+\alpha_{l'}, |\mathcal{A}_{l'}| - M_{l'} + \beta_{l'})}{B(M_{l'}+1+\alpha_{l'}, |\mathcal{A}_{l'}| - M_{l'} - 1 + \beta_{l'})} \\ =& p(A) \cdot \frac{|\mathcal{A}_{l'}| - M_{l'} + \beta_{l'}}{M_{l'} - 1 + \alpha_{l'}}. \end{aligned}$$

$\frac{|\mathcal{A}_{l'}| - M_{l'} + \beta_{l'}}{M_{l'} - 1 + \alpha_{l'}}$ decreases monotonically as $M_{l'}$ increases, so it is lower bounded at the upper bound on $M_{l'}$, $m_{l'}^{[t]}$, obtained in step 1. Therefore

$$\frac{|\mathcal{A}_{l'}| - M_{l'} + \beta_{l'}}{M_{l'} - 1 + \alpha_{l'}} \geq \frac{|\mathcal{A}_{l'}| - m_{l'}^{[t]} + \beta_{l'}}{m_{l'}^{[t]} - 1 + \alpha_{l'}} \text{ for } l' \in \{1,...L\}.$$

Thus

$$p(A_{\setminus z}) \geq \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right) p(A). \tag{31}$$

Combining (31) and Lemma 1, the joint probability of S and $A_{\setminus z}$ is bounded by

$$\begin{aligned} P(S, A_{\setminus z}) =& p(A_{\setminus z}) P(S|A_{\setminus z}) \\ \geq& \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right) (\Upsilon)^{supp(z)} \cdot P(S, A). \end{aligned}$$

In order to get $P(S, A_{\setminus z}) \geq P(S, A)$, and with $\Upsilon \leq 1$ from the assumption in the theorem's statement, we must have

$$supp(z) \leq \frac{\log \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right)}{\log \Upsilon}.$$

This means if $\exists z \in A$ such that $supp(z) \leq \frac{\log \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right)}{\log \Upsilon}$, then $A \notin \arg\max_{A'} F(A')$, which is equivalent to saying, for any $z \in A^*$,

$$supp(z) \geq \frac{\log \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right)}{\log \Upsilon}.$$

Since the support of a rule has to be integer,

$$supp(z) \geq \left\lceil \frac{\log \max_l \left(\frac{|\mathcal{A}_l| - m_l^{[t]} + \beta_l}{m_l^{[t]} - 1 + \alpha_l} \right)}{\log \Upsilon} \right\rceil.$$

Before proving Theorem 2, we first prove the following lemma that we will need later.

Lemma 2 *Define a function $g(l) = \left(\frac{\lambda}{2}\right)^l \Gamma(J - l + 1), \lambda, J \in \mathbb{N}^+$. If $1 \leq l \leq J$,*

$$g(l) \leq \max \left\{ \left(\frac{\lambda}{2} \right) \Gamma(J+1), \left(\frac{\lambda}{2} \right)^J \right\}.$$

Proof 3 *(Of Lemma 2) In order to bound $g(l)$, we will show that $g(l)$ is convex, which means its maximum value occurs at the endpoints of the interval we are considering. The second derivative of $g(l)$ respect to l is*

$$g''(l) = g(l) \left[\left(\ln \frac{\lambda}{2} + \gamma - \sum_{k=1}^{\infty} \frac{1}{k} - \frac{1}{k + J - L_m} \right)^2 + \sum_{k=1}^{\infty} \frac{1}{(k + J - l)^2} \right] > 0, \tag{32}$$

since at least one of the terms $\frac{1}{(k+J-l)^2} > 0$. *Thus* $g(l)$ *is strictly convex. Therefore the maximum of* $g(l)$ *is achieved at the boundary of* L_m, *namely 1 or* J. *So we have*

$$\begin{aligned} g(l) &\leq \max\{g(1), g(J)\} \\ &= \max\left\{\left(\frac{\lambda}{2}\right)\Gamma(J+1), \left(\frac{\lambda}{2}\right)^J\right\}. \end{aligned} \tag{33}$$

Proof 4 *(Of Theorem 2) We follow similar steps in the proof for Theorem 1.*
Step 1 *We first derive an upper bound* M^* *at time* t, *denoted as* $M^{[t]}$.

The probability of selecting a rule set A^* *depends on the number of rules* M^* *and the length of each rule, which we denote as* $L_m, m \in \{1, ..., M^*\}$, *so the prior probability of selecting* A^* *is*

$$\begin{aligned} P(A^*;\theta) &= \omega(\lambda,\eta)\textit{Poisson}(M^*;\lambda)\prod_m^{M^*}\textit{Poisson}(L_m;\eta)\frac{1}{\binom{J}{L_m}}\prod_k^{L_m}\frac{1}{K_{v_{m,k}}} \\ &= p(\emptyset)\frac{\lambda^{M^*}}{\Gamma(M^*+1)}\prod_m^{M^*}\frac{e^{-\eta}\eta^{L_m}\Gamma(J-L_m+1)}{\Gamma(J+1)}\prod_k^{L_m}\frac{1}{K_{v_{m,k}}} \\ &\leq p(\emptyset)\frac{1}{\Gamma(M^*+1)}\left(\frac{e^{-\eta}\lambda}{\Gamma(J+1)}\right)^{M^*}\prod_m^{M^*}\left(\frac{\eta}{2}\right)^{L_m}\Gamma(J-L_m+1). \end{aligned} \tag{34}$$

(34) follows from $K_{v_{m,k}} \geq 2$ *since all attributes have at least two values. Using Lemma 2 we have that*

$$\left(\frac{\eta}{2}\right)^{L_m}\Gamma(J-L_m+1) \leq \max\left\{\left(\frac{\eta}{2}\right)\Gamma(J+1), \left(\frac{\eta}{2}\right)^J\right\}. \tag{35}$$

Combining (34) and (35), and the definition of x, *we have*

$$P(A^*;\theta) \leq p(\emptyset)\frac{x^{M^*}}{\Gamma(M^*+1)}. \tag{36}$$

Now we apply (23) combining with (36) and (25), we get

$$\log\mathcal{L}^* + \log p(\emptyset) + \log\frac{x^{M^*}}{\Gamma(M^*+1)} \geq v^{[t]} \tag{37}$$

Now we want to upper bound $\frac{x^{M^*}}{\Gamma(M^*+1)}$.

If $M^* \leq \lambda$, *the statement of the theorem holds trivially. For the remainder of the proof we consider, if* $M^* > \lambda$, $\frac{x^{M^*}}{\Gamma(M^*+1)}$ *is upper bounded by*

$$\frac{x^{M^*}}{M^*!} \leq \frac{x^\lambda}{\lambda!}\frac{x^{(M^*-\lambda)}}{(\lambda+1)^{(M^*-\lambda)}},$$

where in the denominator we used $\Gamma(M^*+1) = \Gamma(\lambda+1)(\lambda+1)\cdots M^* \geq \Gamma(\lambda+1)(\lambda+1)^{(M^*-\lambda)}$. *So we have*

$$\log\mathcal{L}^* + \log p(\emptyset) + \log\frac{x^\lambda}{\Gamma(\lambda+1)} + (M-\lambda)\log\frac{x}{\lambda+1} \geq v^{[t]}. \tag{38}$$

We have $\frac{x}{\lambda+1} \leq 1$. *To see this, note* $\frac{\Gamma(J)}{\Gamma(J+1)} < 1, e^{-\eta}\left(\frac{\eta}{2}\right) < 1$ *and* $\frac{e^{-\eta}\left(\frac{\eta}{2}\right)^J}{\Gamma(J+1)} < 1$ *for every* η *and* J. *Then solving for* M^* *in (38), using* $\frac{x}{\lambda+1} < 1$ *to determine the direction of the inequality yields:*

$$M^* \leq M^{[t]} = \left\lfloor \lambda + \frac{\log \mathcal{L}^* + \log p(\emptyset) + \log \frac{x^\lambda}{\Gamma(\lambda+1)} - v^{[t]}}{\log \frac{\lambda+1}{x}} \right\rfloor \tag{39}$$

$$= \left\lfloor \frac{\log \frac{(\lambda+1)^\lambda}{\Gamma(\lambda+1)}}{\log \frac{\lambda+1}{x}} + \frac{\log \mathcal{L}^* + \log p(\emptyset) - v^{[t]}}{\log \frac{\lambda+1}{x}} \right\rfloor. \tag{40}$$

Step 2 *Now we prove the lower bound on the support of rules in an optimal set* A^*. *Similar to the proof for Theorem 1, we will show that for a rule set* A, *if any rule* a_z *has support* $supp(z) < C$ *on data* S, *then* $A \notin \arg\min_{A' \in \Lambda_S} F(A')$. *Assume rule* a_z *has support less than C and* $A_{\setminus z}$ *has the* k*-th rule removed from* A. *Assume* A *consists of* M *rules, and the* z*-th rule has length* L_z.

We relate $P(A_{\setminus z}; \theta)$ *with* $P(A; \theta)$. *We multiply* $P(A_{\setminus z}; \theta)$ *with 1 in disguise to relate it to* $P(A; \theta)$:

$$P(A_{\setminus z}) = \frac{M^*\Gamma(J+1)}{\lambda e^{-\eta}\eta^{L_z}\Gamma(J-L_z+1)\prod_k^{L_z}\frac{1}{K_{v_{z,k}}}} P(A)$$

$$\geq \frac{M^*\Gamma(J+1)}{\lambda e^{-\eta}\left(\frac{\eta}{2}\right)^{L_z}\Gamma(J-L_z+1)} P(A) \tag{41}$$

$$= \frac{M^*\Gamma(J+1)}{\lambda e^{-\eta} g(L_z; \eta, J)} P(A) \tag{42}$$

$$\geq \frac{M^* P(A)}{x} \tag{43}$$

where (41) follows that $K_{v_{m,k}} \geq 2$ *since all attributes have at least two values, (42) follows the definition of* $g(l)$ *in Lemma 2 and (43) uses the upper bound in Lemma 2 and. Then combining (22) with (42), the joint probability of* S *and* $A_{\setminus z}$ *is lower bounded by*

$$P(S, A_{\setminus z}) = P(A_{\setminus z})P(S|A_{\setminus z})$$
$$\geq \frac{M^*\Upsilon^{supp(z)}}{x} P(S, A).$$

In order to get $P(S, A_{\setminus z}) \geq P(S, A)$, *we need*

$$\frac{M^*\Upsilon^{supp(z)}}{x} \geq 1,$$

i.e.,

$$\Upsilon^{supp(z)} \geq \frac{x}{M^*} \geq \frac{x}{M^{[t]}},$$

We have $\Upsilon \leq 1$, *thus*

$$supp(z) \leq \frac{\log \frac{x}{M^{[t]}}}{\log \Upsilon}. \tag{44}$$

Therefore, for any rule z in a MAP model A^*,

$$supp(z) \geq \left\lceil \frac{\log \frac{x}{M^{[t]}}}{\log \Upsilon} \right\rceil . \tag{45}$$

Appendix B. Mobile Advertisement data sets

The attributes of this data set include:

1. User attributes
    - Gender: male, female
    - Age: below 21, 21 to 25, 26 to 30, etc.
    - Marital Status: single, married partner, unmarried partner, or widowed
    - Number of children: 0, 1, or more than 1
    - Education: high school, bachelors degree, associates degree, or graduate degree
    - Occupation: architecture & engineering, business & financial, etc.
    - Annual income: less than $12500, $12500 - $24999, $25000 - $37499, etc.
    - Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she buys takeaway food: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she goes to a coffee house: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she eats at a restaurant with average expense less than $20 per person: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
    - Number of times that he/she goes to a bar: 0, less than 1, 1 to 3, 4 to 8 or greater than 8
2. Contextual attributes
    - Driving destination: home, work, or no urgent destination
    - Location of user, coupon and destination: we provide a map to show the geographical location of the user, destination, and the venue, and we mark the distance between each two places with time of driving. The user can see whether the venue is in the same direction as the destination.
    - Weather: sunny, rainy, or snowy
    - Temperature: 30F^{o}, 55F^{o}, or 80F^{o}
    - Time: 10AM, 2PM, or 6PM
    - Passenger: alone, partner, kid(s), or friend(s)
3. Coupon attributes
    - time before it expires: 2 hours or one day

All coupons provide a 20% discount. The survey was divided into different parts, so that Turkers without children would never see a scenario where their “kids" were in the vehicle. Figure 12 and 13 show two examples of scenarios in the survey.

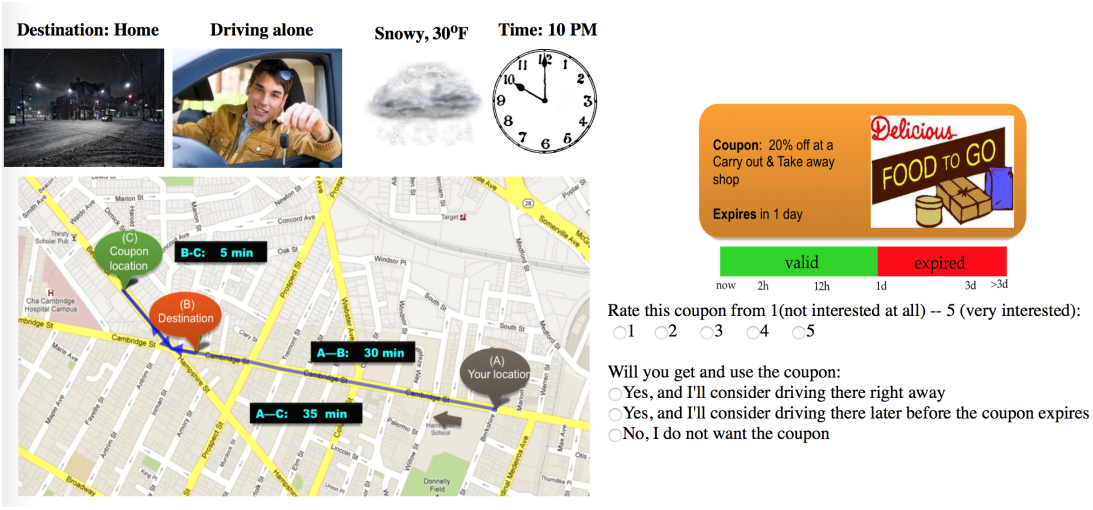


Figure 12: Example 1 of scenario in the survey

Figure 13: Example 2 of scenario in the survey

References

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005. ISSN 1041-4347.

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Proc of the 2008 ACM Conf on Rec Systems*, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM.

Hiva Allahyari and Niklas Lavesson. User-oriented assessment of classification model understandability. In *SCAI*, pages 11–19, 2011.

E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin. Learning certifiably optimal rule lists for categorical data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, 2017.

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

Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, and Roland Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies*, pages 89–100. Springer, 2011.

Elena Baralis, Luca Cagliero, Tania Cerquitelli, Paolo Garza, and Marco Marchetti. Cas-mine: providing personalized services in context-aware applications by means of generalized rules. *Knowledge and information systems*, 28(2):283–310, 2011.

Christian Borgelt. An implementation of the fp-growth algorithm. In *Proc of the 1st interl workshop on open source data mining: frequent pattern mining implementations*, pages 1–5. ACM, 2005.

Endre Boros, Peter L Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz, and Ilya Muchnik. An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering*, 12(2):292–306, 2000.

Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD Record*, volume 26 (2), pages 265–276. ACM, 1997.

Guoqing Chen, Hongyan Liu, Lan Yu, Qiang Wei, and Xing Zhang. A new approach to classification based on association rule mining. *Decision Support Systems*, 42(2):674–689, 2006.

Hong Cheng, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, pages 716–725. IEEE, 2007.

P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

William Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

Yves Crama, Peter L Hammer, and Toshihide Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 16(1):299–325, 1988.

Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. Caep: Classification by aggregating emerging patterns. In *International Conference on Discovery Science*, pages 30–42. Springer, 1999.

Vitaly Feldman. Hardness of approximate two-level logic minimization and pac learning with membership queries. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 363–372. ACM, 2006.

Vitaly Feldman. Learning DNF expressions from fourier spectrum. *CoRR*, abs/1203.0594, 2012.

Eibe Frank and Ian H. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 144–151, 1998.

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

Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

Brian R. Gaines and Paul Compton. Induction of ripple-down rules applied to modeling large databases. *Journal of Intelligent Information Systems*, 5(3):211–228, 1995.

Siong Thye Goh and Cynthia Rudin. Box drawings for learning with imbalanced data. In *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

John R Hauser, Olivier Toubia, Theodoros Evgeniou, Rene Befurt, and Daria Dzyabura. Disjunctions of conjunctions, cognitive simplicity, and consideration sets. *Jour of Marketing Research*, 47(3):485–496, 2010.

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

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

Chii-Ruey Hwang. Simulated annealing: theory and applications. *Acta Applicandae Mathematicae*, 12(1):108–111, 1988.

Adam R. Klivans and Rocco Servedio. Learning dnf in time $2^{O(n^{1/3})}$. In *Proc of the 33rd Annual ACM Symp on Theory of Computing*, STOC '01, pages 258–265, New York, NY, USA, 2001. ACM.

Bae-Hee Lee, Heung-Nam Kim, Jin-Guk Jung, and Geun-Sik Jo. Location-based service with context data for a restaurant recommendation. In *Database and Expert Systems Applications*, pages 430–438. Springer, 2006.

Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 2015. accepted with minor revision.

Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM 2001*, pages 369–376. IEEE, 2001.

Bing Ma, Wynne Liu, and Yiming Hsu. Integrating classification and association rule mining. In *Proceedings of the 4th International Conf on Knowledge Discovery and Data Mining (KDD)*, 1998.

Dmitry Malioutov and Kush Varshney. Exact rule learning via boolean compressed sensing. In *ICML*, 2013.

David Martens and Bart Baesens. Building acceptable classification models. In *Data Mining*, pages 53–74. Springer, 2010.

David Martens, Jan Vanthienen, Wouter Verbeke, and Bart Baesens. Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793, 2011.

Tyler McCormick, Cynthia Rudin, and David Madigan. A hierarchical model for association rule mining of sequential events: An approach to automated medical symptom prediction. *Annals of Applied Statistics*, 2012.

R.S. Michalski. On the quasi-minimal solution of the general covering problem. In *Proceedings of the Fifth International Symposium on Information Processing*, pages 125–128, 1969.

Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

José M. Noguera, Manuel J. Barranco, Rafael J. Segura, and Luis Martinez. A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215(0):37 – 52, 2012.

Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

Murray M Pollack, Urs E Ruttimann, and Pamela R Getson. Pediatric risk of mortality (prism) score. *Critical care medicine*, 16(11):1110–1116, 1988.

Peter R Rijnbeek and Jan A Kors. Finding a short and accurate decision rule in disjunctive normal form by exhaustive search. *Machine learning*, 80(1):33–62, 2010.

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

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

Robert E. Schapire. Using output codes to boost multiclass learning problems. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, pages 313–321, 1997.

W Schwinger, Chr Grün, B Pröll, W Retschitzegger, and A Schauerhuber. Context-awareness in mobile tourism guides–a comprehensive survey. Technical report, Johannes Kepler University Linz, 2005.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Wei-Guang Teng, Ming-Jyh Hsieh, and Ming-Syan Chen. On the mining of substitution rules for statistically dependent items. In *ICDM 2003*, pages 442–449. IEEE, 2002.

Hung-Wen Tung and Von-Wun Soo. A personalized restaurant recommender agent for mobile e-service. In *IEEE Int Conf on e-Technology, e-Commerce and e-Service, 2004. (EEE)*, pages 259–262, 2004.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782.

Mark Van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive hypermedia and adaptive web-based systems*, pages 235–244. Springer, 2004.

Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2015.

Tong Wang, Cynthia Rudin, F Doshi, Yimin Liu, Erica Klampfl, and Perry MacNeille. In-vehicle coupon recommendation dataset, 2015. URL `www.researchgate.net/publication/\318645502_in-vehicle_coupon_recommendation`.

Tong Wang, Cynthia Rudin, Finale Doshi, Yimin Liu, Erica Klampfl, and Perry MacNeille. Bayesian or's of and's for interpretable classification with application to context aware recommender systems. In *Proceedings of the International Conference on Data Mining (ICDM)*, 2016.

Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 99–108, 2012.

Geoffrey I. Webb. Opus: An efficient admissible algorithm for unordered search. *J. Artif. Intell. Res.*, 3:431–465, 1995.

Xindong Wu, Chengqi Zhang, and Shichao Zhang. Mining both positive and negative association rules. In *Proceedings of the Nineteenth ICML*, pages 658–665. Morgan Kaufmann Publishers Inc., 2002.

Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In *SDM*, volume 3, pages 369–376. SIAM, 2003.