Bayesian Or's of And's for Interpretable Classification with Application to Context Aware Recommender Systems

Tong Wang, Cynthia Rudin
MIT
100 Main St
Cambridge MA 02142
tongwang,rudin@mit.com

Finale Doshi-Velez
Department of Computer Science
29 Oxford Street
Cambridge MA 02138
finale@seas.harvard.edu

Yimin Liu, Erica Klampfl, Perry MacNeille
Ford Motor Company
2101 Village Rd
Dearborn, MI, 48124
yliu,eklampfl,pmacneil@ford.com

ABSTRACT

We present a machine learning algorithm for building classifiers that are comprised of a small number of disjunctions of conjunctions (*or*'s of *and*'s). An example of a classifier of this form is as follows: If X satisfies (conditions A1, A2, and A3) OR (conditions B1 and B2) OR (conditions C1 and C2), then we predict that Y=1, ELSE predict Y=0. Models of this form have the advantage of being interpretable to human experts, since they produce a set of conditions that concisely describe a specific class. In our Bayesian model, there are prior parameters that the user can set in order for the model to have a desired size and shape to conform with a domain-specific definition of interpretability. Our method has a major advantage over classical associative classification methods in that it is not greedy. We present an approximate MAP inference technique involving association rule mining and simulated annealing, which allows the method to scale nicely. Our interest in developing this model is to use it to create a predictive model of user behavior with respect to in-vehicle context-aware advertising. This is part of an effort to create the *connected vehicle*, where context data might be collected from the vehicle in order to benefit the driver and passengers. We present several predictive models of user behavior based on data collected from Mechanical Turk; these models are accurate, yet interpretable. We also quantify the effect on prediction accuracy of having contextual attributes.

Keywords

association rules, interpretable classifier, Bayesian modeling

1. INTRODUCTION

Our goal is to construct predictive classification models that consist of a small number of disjunctions of conjunctions, that is, the classifiers are *or*'s of *and*'s. These are logical models that have the advantage of being interpretable to human experts. Interpretability of a predictive model is a fundamentally desirable quality, particularly for knowledge discovery problems. For instance, consider the study of human behavior, where our goal is to understand how people make decisions. Consider the possibility, for instance, that there are simple rules characterizing how humans react to a coupon for a local business in certain contexts. If we can identify these simple rules, we can better understand consumer behavior and target advertisements more effectively to consumers. Our goal is not just to predict what the decisions a person will make; we want to explain *why* we believe the person will make that decision.

Machine learning methods often produce black box models where the relationship between variables is extremely complicated (e.g., neural networks), or where the number of variables is so large that humans cannot easily see the input-output relationships (humans can handle about 7±2 cognitive entities at once [40]). On the other hand, it is possible that the space of good predictive models is large enough to include much sparser models that are interpretable [24]. There are a number of reviews on interpretability in predictive modeling [3, 18, 25, 37, 38, 47], and the form of model we consider (disjunctions of conjunctions) has some recent precedent in the literature [19, 21, 22, 36] as a form of model that is natural for modeling consumer behavior, and beyond that, as a form of model that is interpretable generally to human experts.

Here is an example of a classifier that is a disjunction of conjunctions (an *or*-of-*and*'s):

if X satisfies (A1 AND A2 AND A3)
OR (B1 AND B2)
OR (C1 AND C2) **then**
  Predict $Y = 1$
else
  Predict $Y = 0$
end if

We will call conjunctions such as (A1 AND A2 AND A3) *patterns* and the disjunction of conjunctions a *pattern set*. We refer to an individual element of a pattern, such as A1, as a *attribute-value pair* or an *item*.

We take a Bayesian approach to the construction of *or*-of-*and* classifiers. The parameters of the Bayesian model allow us to focus on interpretable classifiers by providing expected *pattern lengths* (number of items in a pattern) and *pattern set size* (number of patterns in a pattern set). Our approximate inference technique uses a combination of association rule mining and simulated annealing to approximate

WOODSTOCK '97 El Paso, Texas USA

the global optima; it is not a greedy method. Our key insight is that accurate models can often be composed of sufficiently frequent patterns; thus itemset mining can be used to create the building blocks for our approximation. Our approach contrasts with methods that either solve a full computationally hard problem or find a (possibly severe) relaxation of the optimization problem [21, 22]. It also contrasts heavily with associative classification techniques that mine for frequent patterns and combine them heuristically, without optimizing for accuracy of the final model [11, 12, 32, 34, 39, 46, 61], as well as greedy approaches that add patterns one at a time (e.g., [19, 36]).

Our applied interest involves the *connected vehicle*, where in the future, one's car would record information (e.g., time, date, terrain, route, weather, passenger information, etc.) and provide useful services. In particular, we aim to understand user response to personalized advertisements that are chosen based on the user, the advertisement, and the context. Such systems are called *context-aware recommender systems* (see surveys [1, 2, 6, 57] and references therein). One major challenge in the design of recommender systems, reviewed in [57], is the *interaction challenge*: users typically wish to know *why* certain recommendations were made and why they change. Our work addresses precisely this challenge: we provide patterns in data that describe when a recommendation will be accepted.

2. RELATED WORK

The models we are studying have different names in different fields. They are called "disjunctions of conjunctions" or "non-compensatory decision rules" in marketing, "classification rules" in data mining, "disjunctive normal forms" (DNF) in artificial intelligence. Learning logical models of this form has an extensive history. Valiant [55] showed that DNFs could be learned in polynomial time in the PAC (probably approximately correct) setting, and recent work has improved those bounds via polynomial threshold functions [28] and Fourier analysis [17]. However, these theoretical approaches often require unrealistic modeling assumptions (such as noiseless observations) and are not designed to scale to realistic problems.

In parallel, the data-mining literature has developed approaches to building logical models. Associative classification methods (e.g., [11,12,32,34,39,46,61]) mine for frequent patterns in the data and combine them to build classifiers, generally in a heuristic way, where rules are ranked by an interestingness criteria and the top several rules are used. Some of these methods, like CBA (Classification Based on Associations), CPAR (Classification based on Predictive Association Rules) and CMAR (Classification based on Multiple Association Rules) [11, 12, 32, 61] still suffer from a huge number of rules and do not yield interpretable classifiers.

Another class of approaches aim to construct DNF models by greedily adding the conjunction that explains the most of the remaining data [19, 20, 36, 44, 59].

Work on or "inductive logic programming" aims to find optimized conjunctions. They are generally used together as a set of disjunctions of conjunctions, even though the disjunction would not be optimized, just the individual conjunctions.

There are few recent techniques that do aim to fully learn DNF models [21, 22]. Both of these works present integer programming approaches for solving the full problems, and also present relaxations for computational efficiency.

Note that all methods for combining conjunctions are robust to outliers and naturally handle missing data, with no imputation needed for missing attribute values (and as a result, can sometimes outperform outperform traditional convex optimization-based methods such as support vector machines or Lasso). Generally, the pre-mined frequent itemsets are then combined heuristically, without optimizing for accuracy of the final model.

Our method is also a special case of another form of interpretable modeling called M-of-N rules [13, 18, 42, 52, 54], in particular when M=1. In an M-of-N rules model, an example is classified as positive if at least M criteria among N are satisfied. If M=1, the model becomes a disjunction of conditions. If M=N, then the model is a single conjunction. (In these models, one rule generally refers to one feature/condition, whereas in our model, each pattern can have multiple conditions in each disjunction.)

In-vehicle context-aware recommender systems for coupons are different than, for instance, recommendation systems for in-vehicle context-aware music recommendations (on which there is a growing interest, see [7, 26, 51, 58]). 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 context is entirely different, as well as the commitment that would be made by the user in accepting the recommendation. The setup of in-vehicle recommendation systems are also different than, for instance, mobile-tourism guides [41, 49, 53, 56] 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, Hong, and Cho [43] who also consider Bayesian predictive models for context aware recommender systems to restaurants. They also consider demographic and context-based attributes. Lee at al. [30] 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 advertising targeted at users in vehicles, which means the contextual information they considered did not include a user's destination, relative locations of recommended restaurants to the destination, passenger(s) in the vehicle, etc.

3. A BAYESIAN OR-OF-AND MODEL

We work with standard classification data. The data matrix $\mathbf{X}$ consists of N observations (rows) described by T attributes (columns), either categorical or numerical. $\mathcal{D}^+$ is the class of observations with positive labels, and the observations with negative labels are $\mathcal{D}^-$. Our goal is to find the best set of patterns that describe $\mathcal{D}^+$ and discriminate it from $\mathcal{D}^-$.

Figure 1 shows an example of a pattern set. Each pattern is a yellow patch that covers a particular area. (An observation obeys the conjunction if it lies within all of the boundaries of the patch.) In Figure 1, the white oval in the middle indicates the positive class. Our goal is to find a set of patterns that covers mostly the positive class, but little of the negative class.

Let us present a probabilistic model for selecting patterns for our classifier. Taking a Bayesian approach allows us to flexibly incorporate users' expectations on the number of patterns in our pattern set and the number of conditions in

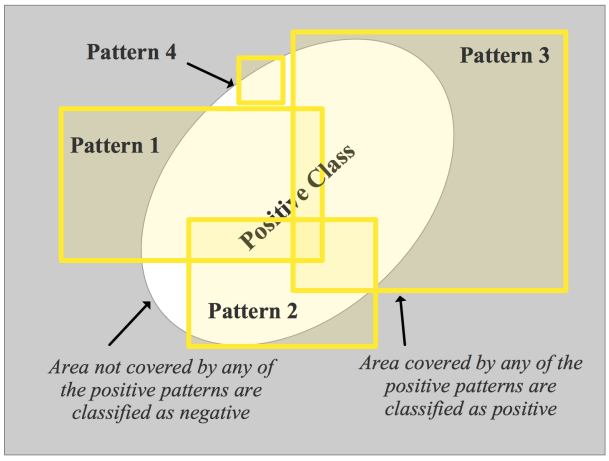

Figure 1: Illustration of disjunction of conjunctions

a pattern. In this way, the user can guide the model toward more domain-interpretable solutions by specifying a desired balance between the size and number of patterns—without committing to any particular value for these parameters.

As with all Bayesian approaches, our model has two parts. The prior encourages interpretability by encouraging the classifier to use relatively few, short patterns. The likelihood ensures that the model still explains the data well, that is, has good classification performance. We detail both of these parts below.

3.1 Prior

Let R denote the set of all patterns. Each pattern is a conjunction of attribute-value pairs, such as "$X_{\cdot,3}$=red and $X_{\cdot,7}$=square". Let R_l be the set of all patterns with length l; thus, $\cup_l R_l = R$. In our model, interpretability of a pattern is determined by its length, so that the *a priori* probability that a pattern r of length l is selected for a pattern set depends only on l. We use a beta prior on the probability p_l for the inclusion of a pattern $r \in R_l$:

$$p_l \sim \text{Beta}(\alpha_l, \beta_l). \tag{1}$$

The parameters $\{\alpha_l, \beta_l | l \in \{1, ..., L\}\}$ on the priors control the expected number of patterns of each length in the pattern set. Specifically, let $\hat{R}_l \subseteq R_l$ denote the set of patterns r selected from R_l, and $M_l = |\hat{R}_l|$ be the number patterns of length l in the pattern set. We then have $E[M_l] = |R_l|E[p_l] = |R_l|\frac{\alpha_l}{\alpha_l+\beta_l}$. Therefore, if we favor short patterns, we could simply increase $\frac{\alpha_l}{\alpha_l+\beta_l}$ for smaller l and decrease the ratio for bigger l.

A pattern set $\hat{R}_l$ is a collection of M_l patterns independently selected from R_l, $\hat{R}_l \subseteq R_l$. We integrate out the probability p_l to get the probability of $\hat{R}_l$:

$$P(\hat{R}_l|\alpha_l, \beta_l) = \int_{p_l} p_l^{M_l} \text{Beta}(p_l; \alpha_l, \beta_l) d(p_l)$$
$$\propto \frac{\Gamma(|R_l|+1)}{\Gamma(M_l+1)\Gamma(|R_l|-M_l+1)} \times \frac{\Gamma(M_l+\alpha_l)\Gamma(|R_l|-M_l+\beta_l)}{\Gamma(|R_l|+\alpha_l+\beta_l)}$$

where the first line follows because each pattern is selected independently and the second line follows from integrating over the beta prior on p_l. The BOA classifier is represented by $\hat{R} = \cup_{l\in\{1,...\}}\hat{R}_l$, and thus the probability of a pattern set $\hat{R}$ that incorporates patterns of different lengths is:

$$P(\hat{R}|\{\alpha_l, \beta_l\}_l) = \prod_l P(\hat{R}_l|\alpha_l, \beta_l). \tag{2}$$

3.2 Likelihood

Let $\mathcal{D}_n$ denote the n-th observation, z_n denote the classification outcome for $\mathcal{D}_n$ using $\hat{R}$, and y_n denote the observed outcome. We introduce likelihood parameter ρ_+ to govern the probability that an observation is a real positive class case when it satisfies the pattern set, and ρ_- as the probability that $y = 1$ when it does not satisfy the pattern set.

If $z_n = 1$,

$$y_n = \begin{cases} 1 \text{ with probability } \rho_+ \\ 0 \text{ with probability } 1-\rho_+. \end{cases} \tag{3}$$

If $z_n = 0$,

$$y_n = \begin{cases} 1 \text{ with probability } \rho_- \\ 0 \text{ with probability } 1-\rho_-. \end{cases} \tag{4}$$

The likelihood of data given a pattern set $\hat{R}$ and parameters ρ_+, ρ_- is thus:

$$P(\mathcal{D}|\hat{R}, \rho_+, \rho_-) = \tag{5}$$
$$\prod_n \rho_1^{z_n y_n}(1-\rho_1)^{z_n(1-y_n)}(1-\rho_0)^{(1-z_n)(1-y_n)}\rho_0^{(1-z_n)y_n},$$

where the four components in formula (5) represent four classification outcomes: true positive, false positive, true negative, and false negative.

We place beta priors over the classification error probabilities ρ_+ and ρ_-:

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

Here, $\alpha_+, \beta_+, \alpha_-, \beta_-$ should be chosen such that $E[\rho_+]$ is larger than $E[\rho_-]$ which means the positive class is correctly characterized by the pattern set.

Integrating out the priors on the classification error probabilities ρ_+ and ρ_- from the likelihood in (5), we get

$$P(\mathcal{D}|R, \alpha_+, \beta_+, \alpha_-, \beta_-) \propto$$
$$\frac{\Gamma(\sum_n z_n + 1)}{\Gamma(\sum_n z_n y_n + 1)\Gamma(\sum_n z_n(1-y_n)+1)}$$
$$\times \frac{\Gamma(\sum_n z_n y_n + \alpha_+)\Gamma(\sum_n z_n(1-y_n)+\beta_+)}{\Gamma(\sum_n z_n + \alpha_+ + \beta_+)}$$
$$\times \frac{\Gamma(\sum_n (1-z_n)+1)}{\Gamma(\sum_n (1-z_n)y_n + 1)\Gamma(\sum_n (1-z_n)(1-y_n)+1)}$$
$$\times \frac{\Gamma(\sum_n (1-z_n)y_n + \alpha_-)\Gamma(\sum_n (1-z_n)(1-y_n)+\beta_-)}{\Gamma(\sum_n (1-z_n) + \alpha_- + \beta_-)}. \tag{6}$$

According to the two outcomes of y_n and z_n, the training data are divided into true positives (TP = $\sum_n z_n y_n$), false positives (FP = $\sum_n z_n(1-y_n)$), true negatives (TN = $\sum_n (1-z_n)(1-y_n)$) and false negatives (FN = $\sum_n (1-$

$z_n)y_n)$. The above likelihood can be rewritten as:

$$P(\mathcal{D}|R,\alpha_+,\beta_+,\alpha_-,\beta_-) \propto \frac{\Gamma(\mathrm{TP}+\mathrm{FP}+1)}{\Gamma(\mathrm{TP}+1)\Gamma(\mathrm{FP}+1)} \times \frac{\Gamma(\mathrm{TP}+\alpha_+)\Gamma(\mathrm{FP}+\beta_+)}{\Gamma(\mathrm{TP}+\mathrm{FP}+\alpha_++\beta_+)} \times \frac{\Gamma(\mathrm{FN}+\mathrm{TN}+1)}{\Gamma(\mathrm{FN}+1)\Gamma(\mathrm{TN}+1)} \times \frac{\Gamma(\mathrm{FN}+\alpha_-)\Gamma(\mathrm{TN}+\beta_-)}{\Gamma(\mathrm{FN}+\mathrm{TN}+\alpha_-+\beta_-)}. \quad (7)$$

4. APPROXIMATE MAP INFERENCE

In this section, we describe a procedure for approximately solving for the maximum *a posteriori* or MAP solution to the BOA model. Inference in the BOA model is challenging because finding the best model involves a search over exponentially many possible sets of patterns: the number of patterns increases exponentially with the number of attribute-value pairs, and the number of sets of patterns increases exponentially with the number of patterns. To efficiently search for the MAP solution, we first eliminate poor patterns via pattern mining. Then we optimize over the remaining patterns through a simulated annealing approach with moves designed to quickly explore promising solutions.

4.1 Narrowing the Search Space via Pattern Mining

We note that any reasonably accurate sparse classifier should have high accuracy for all patterns it contains. Rather than considering all possible conjunctions (exponential in the number of attributes), we use only the pre-mined conjunctions. As long as enough high-quality patterns are included, we can still discover an approximate MAP solution corresponding to an accurate classifier. The approximation to frequent itemsets is a statistical approximation, rather than a greedy approximation. (By contrast, decision tree algorithms such as CART or C4.5 [9, 45] rely heavily on greedy splitting and pruning.) Our assumption provides a dramatic reduction in computation.

We remark also that building an optimal disjunction of conjunctions is an optimal subset selection problem, which is a much easier problem computationally than optimizing fully over decision trees. (See also [31, 46] for more details.)

We first convert each row in $\mathcal{D}^+$ into a set of items (conditions). An item is an attribute-value pair, where the value could be a category for categorical attributes, or a numerical value or range of values for numerical attributes. We consider both positive associations (e.g., X_{jl}='blue') and negative associations (X_{jl}=not 'green') as items. (The importance of negative items is stressed, for instance, by [10, 50,60].) We then mine for frequent patterns within the set of positive observations $\mathcal{D}^+$ using an established frequent pattern mining algorithm. In our implementation, we use the FP-growth algorithm [8], which can in practice be replaced with any desired frequent pattern-mining method. We set only a minimum support and a maximum length of pattern, and the frequent pattern mining algorithm generates a list of patterns (conjunctions of items). Frequent pattern mining algorithms all return the same results since they all perform a type of breadth-first-search for all sufficiently frequent patterns. There are many existing algorithms in the data mining literature that discuss how to handle discretization of continuous attributes [15, 16, 48], and other possible types of patterns one might be interested in mining.

Even when we restrict the length of patterns and the minimum support, the number of patterns generated by FP-growth could still be too large to handle. (For example, almost a million patterns are generated for one of the advertisement datasets we are interested in). Therefore, we may wish to use a second criterion besides support to screen for the most potentially useful conjunctions. We first filter out patterns 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 patterns, similarly to other works [11, 12]. For a pattern r, the information gain is

$$\mathrm{InfoGain}(\mathcal{D}|r) = H(\mathcal{D}) - H(\mathcal{D}|r)$$

where $H(\mathcal{D})$ is the entropy of the data and $H(\mathcal{D}|r)$ is the conditional entropy for data that obey conjunction r. Given a dataset $\mathcal{D}$, entropy $H(\mathcal{D})$ is constant; therefore our screening technique chooses the M patterns that have the smallest $H(\mathcal{D}|r)$, where M is user-defined.

We illustrate the effect of screening on one of our advertisement data sets that is discussed later. We mined all patterns with minimum support 5% and maximum length 3. For each pattern, we calculated its true positive rate and false positive rate on the training data, and plotted it as a dot in Figure 2. The top M patterns according to the information gain criteria are colored in red, and the rest are in blue. As shown in the figure, information gain indeed selected good patterns as they are closer to the upper left corner in ROC space.

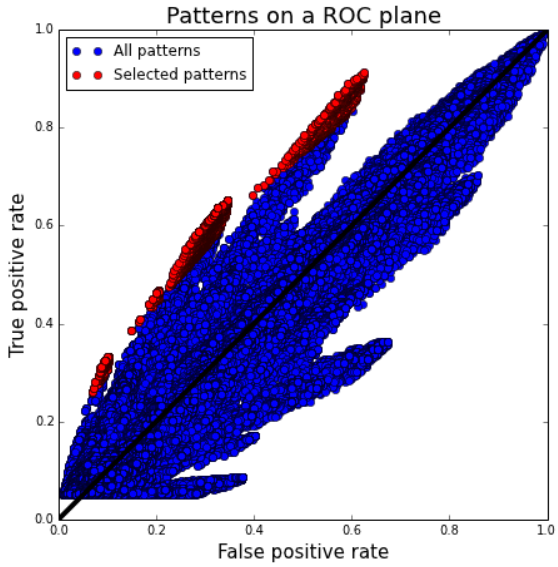

Figure 2: All patterns and selected patterns on a ROC plane

4.2 Stochastic Optimization and Simulated Annealing

We want to maximize the posterior probability:

$$P(\hat{R}|D;\alpha_+,\beta_+,\alpha_-,\beta_-,\{\alpha_l,\beta_l\}_l). \quad (8)$$

Exhaustive evaluation of $\hat{R}$ over all R will not be feasible; if we were to brute force search for the best classifier out of the whole set of possible classifiers, this would involve evaluating all possible subsets of patterns on the training data, and for M candidate patterns, there are 2^M such subsets. We use

simulated annealing [27] to search for a MAP pattern set. At each iteration, the proposed $\hat{R}^*$ is generated from the current $\hat{R}^t$ using one of two options, chosen at random:

1. ADD: Choose a pattern uniformly from R that is not currently in $\hat{R}^t$ and add it to produce $\hat{R}^*$. Tidy $\hat{R}^*$.
2. CUT: Choose a pattern uniformly from $\hat{R}^t$ and remove it.

In action 'Add', tidying $\hat{R}^*$ is useful since sometimes multiple longer patterns can be merged into a shorter pattern without affecting the predictions. For example, two patterns [Gender:Male,Age:<21] and [Gender:Male,Age:>=21] should be merged into [Gender:Male]. The merging does not affect the likelihood since the two patterns are equivalent to the merged shorter pattern, but it does affect the prior since we choose priors that favor shorter patterns and smaller pattern sets. So the latter will have larger posterior probability.

At each iteration, the proposal $\hat{R}_l^*$ is accepted with probability

$$\min\left\{1, \exp\left(-\frac{E(\hat{R}^*) - E(\hat{R}^t)}{T(t)}\right)\right\}$$

where $T(t)$ is the temperature, which follows a cooling schedule:

$$T(t) = \frac{T_0}{\log(1+t)}. \quad (9)$$

We perform the search three times, from three random starting points, and we select the one with the highest MAP.

5. SIMULATION STUDIES

In this section, we present three simulation studies to show that if data are generated from a fixed pattern set, our simulated annealing procedure can recover it with high probability. We also provide analysis for convergence on simulated data sets to show that our model can achieve the optimal solution in a relatively short time.

5.1 Performance variation with different parameters

Given observations with arbitrary features and a collection of patterns on those features, we can construct a binary matrix where the rows represent observations and the columns represent patterns, and the entry is 1 if the pattern matches that observation and 0 otherwise. We need only simulate this binary matrix to represent the observations without losing generality. Each entry is set to 1 independently with probability 0.1. Here are the most important variables in this simulation study:

- M: the number of candidate patterns
- m: the number of patterns in a true pattern set
- N: the number of observations in a data set

The binary matrix representing the data set has size $N \times M$. To compute the prior, we chose all patterns to have same length (and we did not need to write out those patterns by our setup). A true pattern set was generated by randomly selecting m patterns to form the pattern set. We used edit distance between the true pattern set and a generated pattern set as the performance measure. We repeated each experiment in the simulation 100 times and reported the mean performance.

Performance with size of data set, N.

In the first study, we set $m = 5$, $M = 1000$, and varied the size of the data set N. For each sample size $N \in \{100, 500, 1000, 2000, 3000, 4000\}$, we generated 100 independent data sets and pattern sets, and we thus obtained data for 600 recovery problems. For each recovery problem, we then used simulated annealing as described in Section 4.2 with three different starting points. We chose the number of iterations for the simulated annealing runs to be 5000, 10000, and 20000, and we recorded the output of BOA. The edit distance was computed for each of the 100 replicates and the means are plotted in Figure 3. Our results show that as the number of iterations increases, the true pattern sets were recovered with higher probability, as expected. However, the number of observations N did not have a large influence on the result for N approximately greater than 500. The curve is almost flat after $N = 500$. This means that accuracy at $N = 500$ on the recovery problem is similar to the accuracy at $N = 4000$. This result is quite intuitive since simulated annealing searches over the pattern space, and likely finds the same solution once N is sufficiently large.

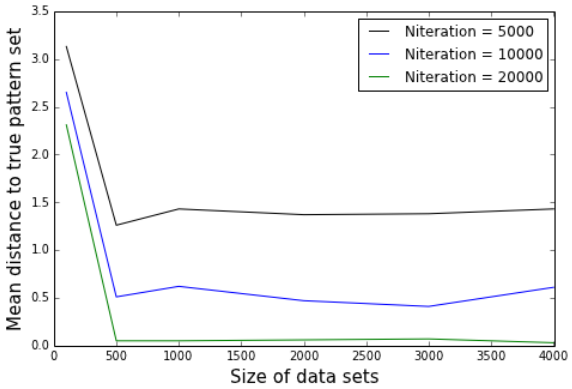

Figure 3: Mean edit distance to the true pattern sets with increasing N

Performance with size of pattern space, M.

In the second study, we set $m = 5, N = 2000$, and varied the size of pattern space. For each size M of patterns in $\{100, 200, 500, 1000, 2000\}$, we repeated the above procedure and plotted the mean over 100 replicates in Figure 4. The number of possible pattern sets of patterns is $\mathcal{O}(2^M)$; therefore as M increases, searching the space becomes difficult for simulated annealing. (This does not mean, however, that prediction performance will suffer; as we increase the number of iterations, the mean edit distance decreases.) We can compensate for larger M by running the simulation for longer times in order to recover the underlying pattern.

Performance with size of true pattern set, m.

In the third study, we set $N = 2000, M = 1000$ and chose the size of the true pattern m within $\{1, 2, 4, 6, 8\}$. Figure 5

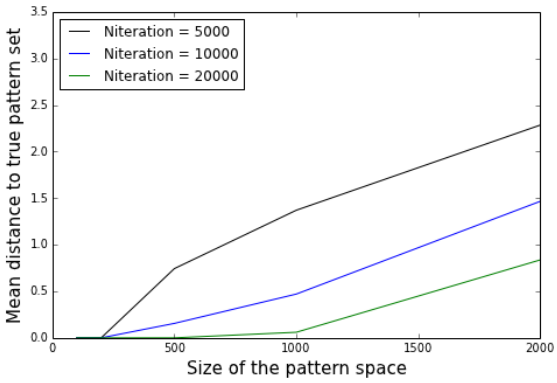

Figure 4: Mean edit distance to the true pattern sets with increasing M

shows that as the number of patterns increases, it becomes harder for the model to recover the true pattern set; however, performance improves over simulated annealing iterations.

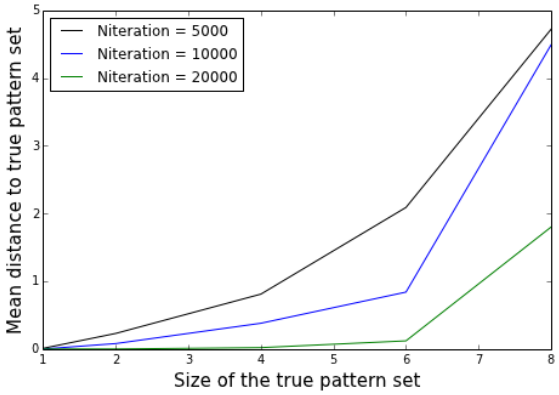

Figure 5: Mean edit distance to the true pattern sets with increasing m

5.2 Convergence analysis

We show how fast our algorithm converges to the optimal solution. We set the size of the data set N to be 2000 and the size of the true pattern set m to be 5. We then ran simulated annealing and recorded the output at steps 100, 500, 1000, 2000, 5000, 10000 and 20000. We repeated this procedure 100 times and plotted the mean and variance of edit distances to true pattern sets in Figure 6, along with running times in seconds. Running times were less than one minute, even for 20000 iterations.

6. APPLICATION TO THE ADVERTISING IN THE CONNECTED VEHICLE

Our goal was to determine the feasibility of an in-vehicle recommender system that would provide coupons to the driver for local businesses. Such systems do not exist presently, but are likely to exist within the next ~ 5 years (see patent applications [33,35]). The coupons would be targeted to the user in his/her particular context, and use of the coupon sys-

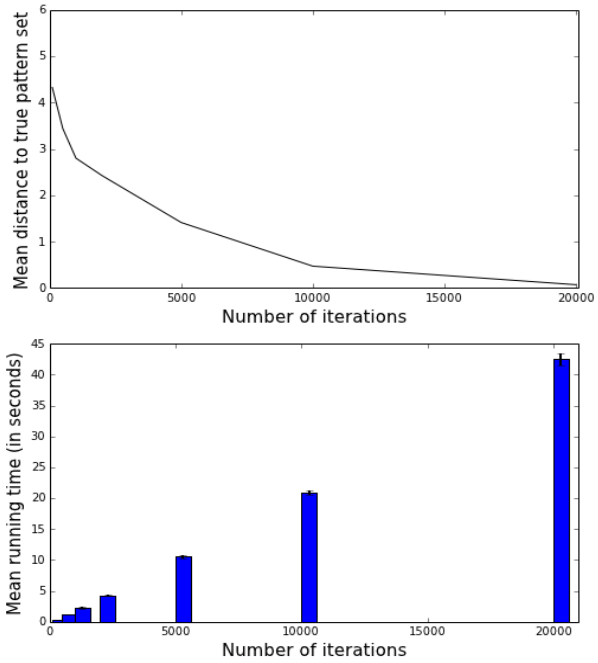

Figure 6: Converence of mean edit distance and running time with number of iterations

tem would be completely optional. The danger of driver distraction is a major consideration in the use of such systems. Currently, the preliminary prototype takes this into account as follows: the prototype substitutes advertisements that the user would hear anyway between, for instance, songs on a music channel, with targeted coupons that are relevant to the user. To accept the coupon, either the driver states (out loud) the word "yes," or s/he could press a button on the steering wheel, which is a small thumb movement. Since such recommender systems do not exist currently in vehicles, the feasibility of a system needs to be explored via surveys. There is quite a lot of work showing that conclusions from surveys often translate into conclusions in real situations [4, 14, 23] (this issue is clearly beyond our scope).

More broadly, our project fits into the goal of the *connected vehicle*, where information connected by sensors inside the car would help to provide useful services. For instance, weight sensors can be placed under the seats to determine how many passengers are in the car and how many of them are children. Voice recorders could determine whether passengers are male or female. The car's historical route information can be used to predict the current destination [29]. In our particular application, this information would be used to recommend local businesses to interested users, but there could potentially be many other uses for these data.

Our data were collected on Amazon Mechanical Turk via a survey that we will describe shortly.[1] We used Turkers with high ratings (95% or above). The conclusions we make are conditioned on the population of Turkers with high scores who chose to complete the survey; this sample can be importance sample-weighted or subsampled to approximate a local area with different population characteristics. Each user provided very detailed demographic information, so it

[1]These data are publicly available here: Placeholder

would be easy to subsample to create a population with different overall demographics. In order to collect high quality data, we used two random questions with easy answers in the survey to determine whether the worker was self-consistent while completing the survey. The Turkers' surveys were accepted only if they provided the correct answers to the two questions. Out of 752 surveys, 652 were accepted by us, which generated a data set containing 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 5 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, and in the second part, we described 20 different driving scenarios to each user along with additional context information and coupon information. We then asked the user if s/he will use the coupon. In the appendix we provided samples of two coupons with their contexts shown to the Turkers.

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: never, less than 1, 1-3, 4-8 or greater than 8
   - Number of times that he/she buys takeaway food: never, less than 1, 1-3, 4-8 or greater than 8
   - Number of times that he/she goes to a coffee house: never, less than 1, 1-3, 4-8 or greater than 8
   - Number of times that he/she eats at a restaurant with average expense less than $20 per person: never, less than 1, 1-3, 4-8 or greater than 8
   - Number of times that he/she goes to a bar: never, less than 1, 1-3, 4-8 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 clearly see whether the venue is in the same direction as the destination.
   - Weather: sunny, rainy, or snowy
   - Temperature: 30F°, 55F°, or 80F°
   - 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.

For categorical attributes, each attribute-value pair was directly coded into an item. Using marital status as an example, 'marital status is single' is 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; each attribute-value pair was converted into two items, each using one side of the range, For example, age is '20 to 25' was converted into (Age:>=20) and (Age:<=25). Then each item is a half-space defined by threshold values.

We will show that BOA does not lose too much accuracy on the mobile advertisement data sets (with respect to the highly complicated black box machine learning methods) even though we restricted the lengths of pattern sets and the number of patterns to yield interpretable models. We compared with other classification algorithms C4.5, CART, random forest, linear lasso, linear ridge, logistic lasso, logistic ridge, and SVM, which span the space of widely used methods that are known for interpretability and/or accuracy. The decision tree methods are representatives of the class of greedy and heuristic methods (e.g., [11, 12, 19, 20, 32, 34, 36, 39, 44, 46, 59, 61]) that yield interpretable models (though in many cases decision trees are often too large to be interpretable). For all experiments, we measured out-of-sample performance using AUC (the Area Under The ROC Curve) from 5-fold testing where the MAP BOA from the training data was used to predict on each test fold.[2] We used the RWeka package in R for the implementations of the competing methods and tuned the hyperparameters using grid search in nested cross validation. Our experimental setup for BOA is as follows, which was not altered throughout the full set of experiments. For rule mining, we set the minimum support to be 5% and set the maximum length of patterns to be 3. We used information gain to select the best 5000 patterns to use in our Bayesian model. We ran simulated annealing for 50000 iterations to obtain a pattern set.

6.1 Interpretability of results

Our first set of experimental results consider five separate coupon prediction problems, for different types of coupons. The AUC's for BOA and baseline methods for all five problems are reported in Table 1. The BOA classifier, while restricted to produce interpretable models, tends to perform

[2]We do not perform hypothesis tests, as it is now known that they are not valid due to reuse of data over folds.

	Bar	Takeaway Food	Coffee House	Restaurant (<$20)	Restaurant ($20 to $50)
BOA	0.777 (0.010)	0.700 (0.045)	0.786 (0.019)	0.729 (0.014)	0.687 (0.014)
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)
Random Forest	0.798 (0.016)	0.640 (0.036)	0.815 (0.010)	0.700 (0.022)	0.689 (0.010)
Linear Lasso	0.795 (0.014)	0.673 (0.042)	0.786 (0.011)	0.769 (0.024)	0.706 (0.017)
Linear Ridge	0.795 (0.018)	0.671 (0.043)	0.784 (0.012)	0.769 (0.020)	0.706 (0.020)
Logistic Lasso	0.796 (0.014)	0.673 (0.042)	0.787 (0.011)	0.767 (0.024)	0.706 (0.016)
Logistic Ridge	0.793 (0.018)	0.670 (0.042)	0.783 (0.011)	0.768 (0.021)	0.705 (0.020)
SVM	0.842 (0.018)	0.735 (0.031)	0.845 (0.007)	0.799 (0.022)	0.736 (0.022)

Table 1: AUC comparison for mobile advertisement data set, means and standard deviations over folds are reported.

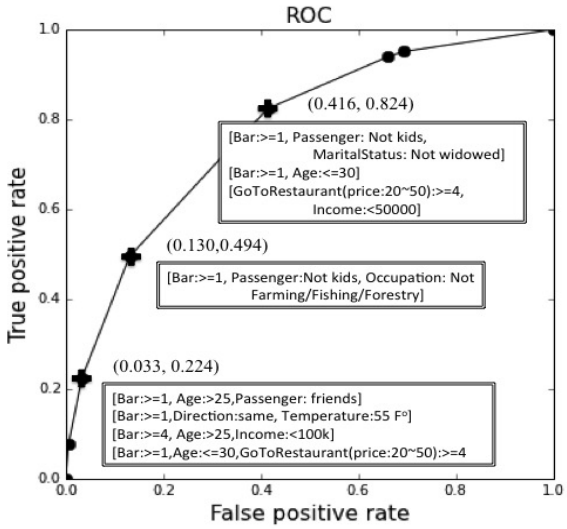

Figure 7: ROC for dataset of coupons for bars

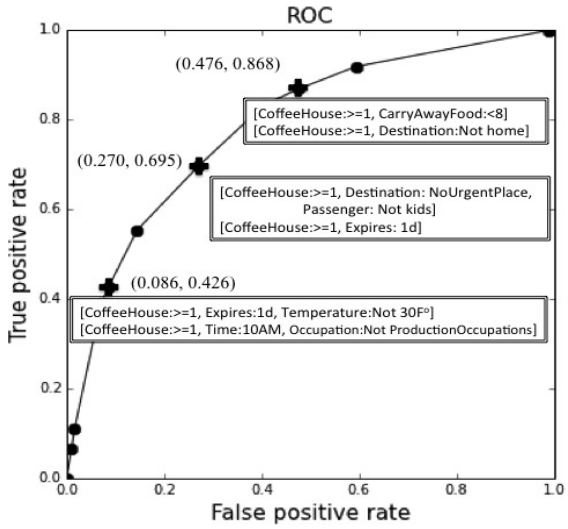

Figure 8: ROC for dataset of coupons for coffee houses

almost as well as the black box machine learning methods, and outperforms the decision tree algorithms. 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.

We show several classifiers produced by BOA in Figure 7 and Figure 8. We varied the hyperparameters $\alpha_+, \beta_+, \alpha_-, \beta_-$ to obtain different sets of patterns, and plotted corresponding points on the curve. Example pattern sets are listed in each box along the curve. For instance, the classifier near the middle of the curve in Figure 7 has one pattern, 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". For another example in Figure 8, the classifier in the middle has two patterns: "If a person visits a coffee house more than once per month, and has no urgent destination to go, and does not have kids, or s/he visits a coffee house more than once per month and the coupon expires in one day, then we predict the person will use the coupon." 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 pattern in the two figures.

We generated 30 pattern sets for each coupon prediction problem by varying hyperparameters α_l, β_l, and thus we obtained 150 BOA models. In Figure 9 we show a histogram of the mean length of pattern sets in the 150 BOA models, and a histogram of the number of patterns in these 150 BOA models.

6.2 Performance without contextual attributes

In order to determine whether contextual information is useful in a online in-vehicle advertisement system, we ran BOA with and without the contextual attributes listed above and compared performance, for all five coupon prediction problems. Results for the 5 types of coupon prediction problems are in Figure 10, showing that the ROC curve for prediction *with* contextual information completely dominates the ROC curve for prediction *without* contextual information, illustrating the benefit of using contextual information in collecting data about the user's context for an in-vehicle

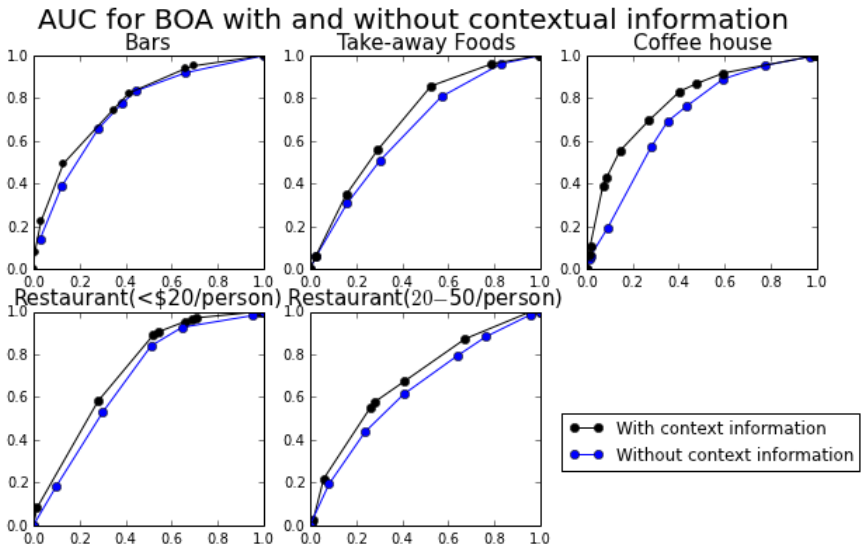

Figure 10: ROC for BOA on data with and without contextual information

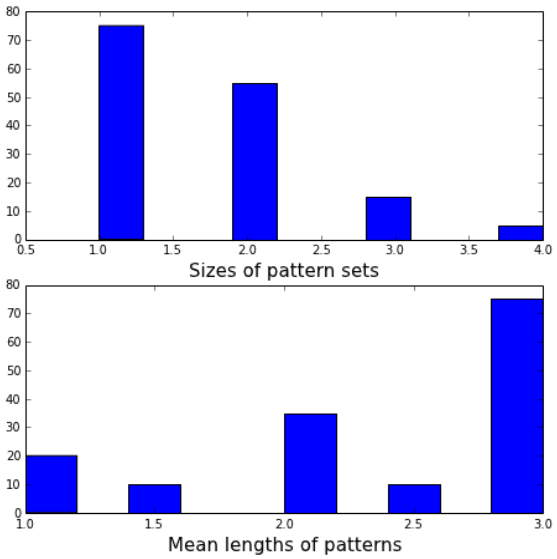

Figure 9: Size of pattern sets and mean length of patterns

recommender system.

7. EXPERIMENTS WITH UCI DATA SETS

We tested BOA on several datasets from the UCI machine learning repository [5], along with baseline algorithms, and Table 2 displays the results. We observed that BOA achieves the best performance on each of the data sets we used. This is not a surprise: most these data sets have an underlying true set of conditions that greedy methods would have difficulty recovering. For example in the tic-tac-toe data set, the positive class can be classified using exactly 8 conditions. BOA has the capability to exactly learn these conditions, whereas the greedy methods that are pervasive throughout the data mining literature (e.g., CART, C4.5) and convexified approximate methods (e.g., SVM) have substantial difficulty with this.

We also added 30% noise to the tic-tac-toe training set and BOA was still able to achieve perfect performance while other methods' performance suffered.

We illustrate BOA's output on the breast cancer dataset. BOA took exactly 2 minutes on a laptop to generate the following pattern set:

if X satisfies (Marginal Adhesion ≥ 3 AND Uniformity of Cell Shape ≥ 3)
OR (Clump Thickness ≥ 7)
OR (Bland Chromatin ≥ 4 AND Uniformity of Cell Size ≥ 1 AND Clump Thickness ≥ 2) **then**
  Predict the tumor is malignant
else
  Predict the tumor is benign.
end if

The out-of-sample accuracy of this model was 0.952, with true positive rate 0.974, and false positive rate 0.060. Or-of-and models could potentially be ideal for medical applications, since they could characterize simple sets of conditions that would place a patient in a high risk category; this may be more useful in some cases than the typical scoring system used in medical calculators.[3]

8. CONCLUSION

We presented an algorithm that produces sparse disjunctions of conjunctions. This method has major benefits over other predictive modeling methods: It is not a black box method, and produces classifiers of a form that is known to be interpretable to human experts. Further, it arises from a principled generative modeling approach. It is not a heuristic or greedy method, like the vast majority of decision tree algorithms and associative classification algorithms, which avoid computationally hard problems by making severe greedy approximations. It does not have problems with robustness to outliers or missing data like the convexified methods (lo-

[3]See mdcalc.com for a list of medical calculators.

	Monk 1	Mushroom	Breast Cancer	Connect4	Tic-tac-toe	Tic-tac-toe (30% noise)
BOA	1.000 (0.000)	1.000 (0.000)	0.990 (0.003)	0.926 (0.002)	1.000 (0.000)	1.000 (0.000)
C4.5	0.906 (0.067)	1.000 (0.000)	0.873 (0.017)	0.867 (0.002)	0.949 (0.016)	0.942 (0.022)
CART	0.826 (0.061)	1.000 (0.000)	0.978 (0.010)	0.703 (0.003)	0.966 (0.011)	0.962 (0.014)
Random Forest	1.000 (0.000)	1.000 (0.000)	0.970 (0.016)	0.940 (0.002)	0.991 (0.003)	0.989 (0.006)
Linear Lasso	0.556 (0.061)	0.995 (0.002)	0.985 (0.005)	0.858 (0.002)	0.986 (0.002)	0.854 (0.019)
Linear Ridge	0.560 (0.078)	0.999 (0.000)	0.987 (0.003)	0.857 (0.002)	0.931 (0.017)	0.820 (0.033)
Logistic Lasso	0.666 (0.084)	0.989 (0.002)	0.988 (0.003)	0.859 (0.002)	0.988 (0.002)	0.860 (0.029)
Logistic Ridge	0.686 (0.103)	0.999 (0.000)	0.988 (0.003)	0.857 (0.002)	0.869 (0.025)	0.805 (0.032)
SVM	0.957 (0.034)	0.999 (0.000)	0.986 (0.005)	0.924 (0.002)	0.993 (0.001)	0.992 (0.002)

Table 2: AUC comparison for some UCI data sets

gistic regression, SVM, Lasso). We used this method in a knowledge discovery framework for investigating the potential usefulness of an in-vehicle recommender system, as part of the *connected vehicle* effort. We applied our machine learning techniques to Mechanical Turk survey data generated by several hundred individuals, and showed that simple patterns based on a user's context can be directly useful in predicting the user's response.

Acknowledgements

We gratefully acknowledge funding provided by Ford and Wistron to C. Rudin.

APPENDIX

A. EXAMPLES OF SCENARIOS IN THE SURVEY

Figure 11: Example 1 of scenario in the survey

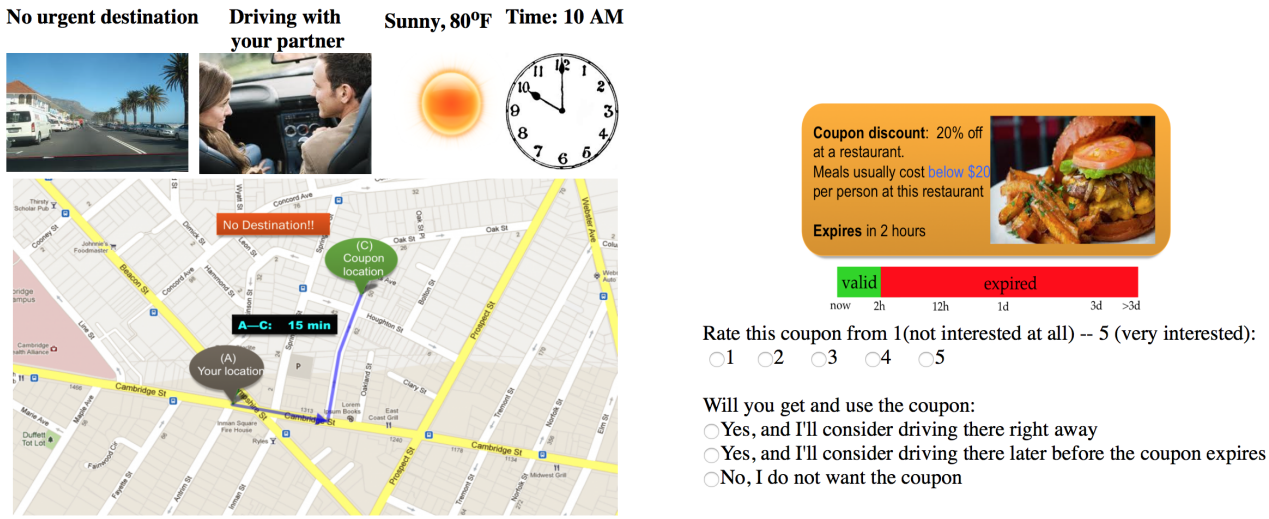

Figure 12: Example 2 of scenario in the survey

B. REFERENCES

[1] G. Adomavicius and A. 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.

[2] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM.

[3] H. Allahyari and N. Lavesson. User-oriented assessment of classification model understandability. In *SCAI*, pages 11–19, 2011.

[4] B. Anckar and D. D'incau. Value creation in mobile commerce: Findings from a consumer survey. *Journal of Information Technology Theory and Application (JITTA)*, 4(1):8, 2002.

[5] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[6] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[7] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies*, pages 89–100. Springer, 2011.

[8] C. Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 1–5. ACM, 2005.

[9] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[10] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD Record*, volume 26 (2), pages 265–276. ACM, 1997.

[11] G. Chen, H. Liu, L. Yu, Q. Wei, and X. Zhang. A new approach to classification based on association rule mining. *Decision Support Systems*, 42(2):674–689, 2006.

[12] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 716–725. IEEE, 2007.

[13] Y. Chevaleyre, F. Koriche, and J.-D. Zucker. Rounding methods for discrete linear classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 651–659, 2013.

[14] A. Deaton. *Economics and consumer behavior*. Cambridge university press, 1980.

[15] J. Dougherty, R. Kohavi, M. Sahami, et al. Supervised and unsupervised discretization of continuous features. In *ICML*, pages 194–202, 1995.

[16] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 1993.

[17] V. Feldman. Learning DNF expressions from fourier spectrum. *CoRR*, abs/1203.0594, 2012.

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

[19] J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

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

[21] S. T. Goh and C. Rudin. Box drawings for learning with imbalanced data. In *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[22] J. R. Hauser, O. Toubia, T. Evgeniou, R. Befurt, and D. Dzyabura. Disjunctions of conjunctions, cognitive simplicity, and consideration sets. *Journal of Marketing Research*, 47(3):485–496, 2010.

[23] B. E. Hayes. *Measuring customer satisfaction and loyalty: survey design, use, and statistical analysis methods*. ASQ Quality Press, 2008.

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

[25] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[26] Y. Kim and G. Choi. Music selecting system and method thereof, February 2013. US Patent 8,370,290.

[27] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[28] A. R. Klivans and R. Servedio. Learning dnf in time $2^{O(n^{1/3})}$. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 258–265, New York, NY, USA, 2001. ACM.

[29] V. Kostov, J. Ozawa, M. Yoshioka, and T. Kudoh. Travel destination prediction using frequent crossing pattern from driving history. In *Proceedings of IEEE Intelligent Transportation Systems (ITSC)*, pages 343–350, September 2005.

[30] B.-H. Lee, H.-N. Kim, J.-G. Jung, and G.-S. Jo. Location-based service with context data for a restaurant recommendation. In *Database and Expert Systems Applications*, pages 430–438. Springer, 2006.

[31] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. An interpretable stroke prediction model using rules and bayesian analysis. In *Proceedings of AAAI Late Breaking Track*, 2013.

[32] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM 2001*, pages 369–376. IEEE, 2001.

[33] Y. Liu, P. MacNeille, and O. Gusikhin. Method and apparatus for advertisement screening, July 2014. US Patent App. 13/744,659.

[34] B. Ma, W. Liu, and Y. Hsu. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD)*, 1998.

[35] P. MacNeille, Y. Liu, J. Marchwicki, S. Burnell, E. Wehrman, O. Gusikhin, and B. Tonshal. Method and apparatus for digital coupon presentation, May 2014. US Patent App. 13/671,987.

[36] D. Malioutov and K. Varshney. Exact rule learning via boolean compressed sensing. In *ICML*, 2013.

[37] D. Martens and B. Baesens. Building acceptable classification models. In *Data Mining*, pages 53–74. Springer, 2010.

[38] D. Martens, J. Vanthienen, W. Verbeke, and B. Baesens. Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793, 2011.

[39] T. McCormick, C. Rudin, and D. Madigan. A hierarchical model for association rule mining of sequential events: An approach to automated medical symptom prediction. *Annals of Applied Statistics*, 2012.

[40] G. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[41] J. M. Noguera, M. J. Barranco, R. J. Segura, and L. Martinez. A mobile 3d-gis hybrid recommender system for tourism. *Information Sciences*, 215(0):37 – 52, 2012.

[42] L. D. Ordóñez, L. Benson III, and L. R. Beach. Testing the compatibility test: How instructions, accountability, and anticipated regret affect prechoice screening of options. *Organizational Behavior and Human Decision Processes*, 78(1):63–80, 1999.

[43] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

[44] M. M. Pollack, U. E. Ruttimann, and P. R. Getson. Pediatric risk of mortality (prism) score. *Critical care medicine*, 16(11):1110–1116, 1988.

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

[46] C. Rudin, B. Letham, and D. Madigan. Learning theory analysis for association rules and sequential event prediction. *Journal of Machine Learning Research*, 14:3384–3436, 2013.

[47] S. Rüping. *Learning interpretable models*. PhD thesis, Universität Dortmund, 2006.

[48] A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 494–502. IEEE, 1998.

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

[50] W.-G. Teng, M.-J. Hsieh, and M.-S. Chen. On the mining of substitution rules for statistically dependent items. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 442–449. IEEE, 2002.

[51] K. A. Theisen, O. Y. Gusikhin, P. R. MacNeille, and D. P. Filev. Intelligent music selection in vehicles, February 2011. US Patent App. 12/539,743.

[52] G. G. Towell and J. W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13(1):71–101, 1993.

[53] H.-W. Tung and V.-W. Soo. A personalized restaurant recommender agent for mobile e-service. In *2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004. (EEE)*, pages 259–262, 2004.

[54] B. Ustun and C. Rudin. Methods and models for interpretable linear classification. Technical report, MIT, 2014.

[55] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, Nov. 1984.

[56] M. Van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive hypermedia and adaptive web-based systems*, pages 235–244. Springer, 2004.

[57] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, 5(4):318–335, 2012.

[58] X. Wang, D. Rosenblum, and Y. Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 99–108, 2012.

[59] C. William et al. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

[60] X. Wu, C. Zhang, and S. Zhang. Mining both positive and negative association rules. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 658–665. Morgan Kaufmann Publishers Inc., 2002.

[61] X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *SDM*, volume 3, pages 369–376. SIAM, 2003.