Project Proposals

- Due next Monday (13^{th} Feb) 11.59pm ET

- 2 page proposal (more details on "Course Logistics" document on canvas) + References

- Today by 5pm ET, we will post:
  - Project topics and some concrete problems
  - Sample proposals and final reports from past iterations
  - LaTeX and Word templates which you will use to write the proposal

Office Hours and Paper Presentations

- Office hours switch this week
  - Suraj and Jiaqi today
  - Hima on Thursday

- Students signed up for presentations next week should see us in office hours this week
  - Full slide deck (ideally!)
  - An overview of what you plan to present

Rule Based Approaches

Agenda

- Paper 1: Interpretable Rule Lists (Letham et. al.)

- Paper 2: Interpretable Rule Sets (Lakkaraju et. al.)

- Discussion

Interpretable Classifiers Using Rules and Bayesian Analysis

Benjamin Letham, Cynthia Rudin, Tyler McCormick, David Madigan; 2015

Contributions

- Introducing a generative model called Bayesian Rule Lists (BRL)
  - Goal is to output a decision list (if then else-if)
- Novel prior structure to encourage sparsity
- Predictive accuracy on par with top algorithms

Decision List: Example

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

This is “an” accurate and interpretable decision list – possibly one of many such lists

Introduction: BRL

- Produces a posterior distribution over permutations of if.. then.. Else-if.. rules from a large set of pre-mined rules

- Decision lists with high posterior probability tend to be both accurate and interpretable
  - Prior favors concise lists with small number of rules and fewer terms in left hand side

Introduction: BRL

- New type of balance between accuracy, interpretability, and computation

- What about using other similar models?
  - Decision trees (CART)
  - They employ greedy construction methods
  - Not particularly computationally demanding but affects quality of solution – both accuracy and interpretability

Pre-mined Rules

- A major source of practical feasibility: pre-mined rules
  - Reduces model space
  - Complexity of problem depends on number of pre-mined rules

- As long as pre-mined set is expressive, accurate decision list can be found + smaller model space means better generalization (Vapnik, 1995)

Pre-mined Rules: Intuition

Itemsets
{1,2,3,4}
{1,2,4}
{1,2}
{2,3,4}
{2,3}
{3,4}
{2,4}

Item	Support
{1}	3
{2}	6
{3}	4
{4}	5

Item	Support
{1,2}	3
~~{1,3}~~	~~1~~
~~{1,4}~~	~~2~~
{2,3}	3
{2,4}	4
{3,4}	3

Item	Support
{2,3,4}	2

Minimum Support = 3

his is Apriori algorithm. FP-growth is a single pass algorithm (more efficien

Preliminaries: Notation

- Training data $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d \quad y_i \in \{1, \ldots, L\}$

  $\mathbf{x} = (x_1, \ldots, x_n) \quad \mathbf{y} = (y_1, \ldots, y_n)$

- Two labels: stroke or no stroke

Bayesian Decision Lists

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

where $\mathbf{N}_j = (N_{j,1}, \ldots, N_{j,L})$

Preliminaries: Multinomial

- Sampling from a multinomial:

Throw a dice 20 times:

```
>>> np.random.multinomial(20, [1/6.]*6, size=1)
array([[4, 1, 7, 5, 2, 1]])
```

- Parameters are probability values

Preliminaries: Dirichlet

- Dirichlet: sampling over a probability simplex
  - E.g., (0.6, 0.4) is a sample from a Dirichlet distribution;

- K-dimensional Dirichlet has k parameters – any positive number
  - I

$$\Theta \sim \text{Dirichlet}(\alpha_1, \alpha_2, \ldots, \alpha_m)$$

$$P(\theta_1, \theta_2, \ldots, \theta_m) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{m} \theta_k^{\alpha_k - 1}$$

Preliminaries: Dirichlet Prior

- Conjugate prior for multinomial distribution

- Conjugate prior: posterior in the same family as prior

- Prior:

$$(p_1, \ldots, p_k) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$$

- Posterior:

$$(p_1, \ldots, p_k) \Big| (x_1, \ldots, x_k) \sim \text{Dirichlet}(\alpha_1 + x_1, \ldots, \alpha_k + x_k).$$

Bayesian Association Rules

$$a \to y \sim \text{Multinomial}(\boldsymbol{\theta}). \qquad \boldsymbol{\theta}|\boldsymbol{\alpha} \sim \text{Dirichlet}(\boldsymbol{\alpha}).$$

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

$$N = (N_{\cdot,1}, \ldots, N_{\cdot,L})$$

Generative Model

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

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

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

$\mathcal{A}$ is complete collection of pre-mined antecedents

Prior Probabilities

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

Truncated Poisson:

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

Ensures that sampled values are within bounds!

Also, ensures expected value is close tλ
when there are a large number of pre-mined rules

Prior Probabilities

Another Truncated Poisson,

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

$p(a_j|a_{<j},c_j,\mathcal{A})$ is sampled uniformly from available antecedents with appropriate cardinality.

Likelihood

- Likelihood is the product of multinomial probability mass functions for the observed label counts at each rule

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

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

Marginalize over θ_j, integrate out the intermediate parar θ_j ter

Markov Chain Monte Carlo

- Generate a chain of random samples until convergence

- Each random sample is a stepping stone for the next one (chain)

- New samples do not depend on any samples before the previous one (Markov)

Markov Chain Monte Carlo

- How to go to (optimal) d^* from current d^t
- Move an antecedent to a different position in the list
- Add an antecedent that is not currently in the list
- Remove an antecedent from the list

Metropolis Hastings

- Start with a random decision list
- Choose a move based on “proposal distribution” Q
- After you choose your move, you compute an acceptance probability A
- Generate a random number u
- If u <= A, then accept; otherwise reject

Metropolis Hastings

1. Initialize $x^{(t)}$ for $t = 0$
2. Draw a sample x' from $Q(x' \mid x^{(t)})$
3. Accept the move with probability $A(x' \mid x^{(t)}) = \min(1, \alpha)$ where $\alpha = P(x')/P(x_t)$. If accepted, let $x^{(t+1)} = x'$. Otherwise, let $x^{(t+1)} = x^{(t)}$.
4. Repeat steps 2–3 to draw samples

Proposal Probabilities

- Move chosen uniformly
- Which antecedents and their new position is also chosen uniformly

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

Estimating label of a new observation

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

Match the antecedent by looking at feature values of new observation

Tic-Tac-Toe

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

5 fold cross validation; accuracy computed across 5 folds

Stroke Prediction

- N = 12,586, 14% had stroke
- 6000 times larger than data for CHADS2 score
- Pre-mining: support 10% and max cardinality 2
- 5 fold evaluation

Stroke Prediction

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

Stroke Prediction - AUC

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

Interpretable Decision Sets

Hima Lakkaraju, Stephen Bach, Jure Leskovec; 2016

Contributions

- A framework called Interpretable Decision Sets (IDS) for classification
- Novel objective function + proof of submodularity
- Optimization procedure with optimality guarantees
- Detailed metrics for evaluating interpretability + user studies

Motivation

- Traditional classification models optimize for predictive accuracy
- Very little understanding of the model itself and its predictions
- Model being “readable” is not enough
- Humans should be able to reason about predictions and readily explain the functionality of the model

Decision Sets

If Respiratory-Illness=Yes **and** Smoker=Yes **and** Age$\geq$ 50 **then** Lung Cancer

If Risk-LungCancer=Yes **and** Blood-Pressure$\geq$ 0.3 **then** Lung Cancer

If Risk-Depression=Yes **and** Past-Depression=Yes **then** Depression

If BMI$\geq$ 0.3 **and** Insurance=None **and** Blood-Pressure$\geq$ 0.2 **then** Depression

If Smoker=Yes **and** BMI$\geq$ 0.2 **and** Age$\geq$ 60 **then** Diabetes

If Risk-Diabetes=Yes **and** BMI$\geq$ 0.4 **and** Prob-Infections$\geq$ 0.2 **then** Diabetes

If Doctor-Visits $\geq$ 0.4 **and** Childhood-Obesity=Yes **then** Diabetes

Criteria for Interpretability

- Parsimony: Fewer rules with fewer conditions
  - Cognitive limits of human understanding

- Distinctness: Minimal overlap of rules w.r.t the data points they cover
  - No redundant and contradicting explanations of data points

- Class Coverage: Explain all the classes in the data
  - Rules explaining minority classes are important

Problem Formulation

- **Input:**
  - Set of data points: $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$
  - Set of class labels: $\mathcal{C}$
  - Set of frequent patterns obtained using Apriori algorithm: $\mathcal{S} = \{ s_1, s_2, s_3, \ldots, s_M \}$

    Ex: s_i = Gender = Female and Age > 35 and Smoker = True

- **Output:** An interpretable decision set $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{C}$

Desiderata

- We need to optimize for the following criteria
  - Recall
  - Precision
  - Distinctness
  - Parsimony
  - Class Coverage

- Recall and Precision □ Accurate predications

- Distinctness, Parsimony, and Class Coverage □ Interpretability

Objective Function

- Parsimony
  - Fewer rules: $f_1(\mathcal{R}) = |\mathcal{S}| - \text{size}(\mathcal{R})$

Number of rules

  - Fewer predicates:

$$f_2(\mathcal{R}) = L_{\max} \cdot |\mathcal{S}| - \sum_{r \in \mathcal{R}} \text{length}(r)$$

Number of conditions in the rule

Maximum no. of conditions in any given Input pattern

Total number of input patterns

Objective Function

- Distinctness
  - Intra-class overlap:

$$f_3(\mathcal{R}) = N \cdot |S|^2 - \sum_{\substack{r_i, r_j \in \mathcal{R} \\ i \leq j \\ c_i = c_j}} \text{overlap}(r_i, r_j)$$

Total number of data points

Number of points that satisfy both the rules

  - Inter-class overlap:

$$f_4(\mathcal{R}) = N \cdot |S|^2 - \sum_{\substack{r_i, r_j \in \mathcal{R} \\ i \leq j \\ c_i \neq c_j}} \text{overlap}(r_i, r_j)$$

Objective Function

- Class Coverage

$$f_5(\mathcal{R}) = \sum_{c' \in \mathcal{C}} 1\left(\exists r = (s, c) \in \mathcal{R} \text{ such that } c = c'\right)$$

Check if there exists some rule corresponding to a given class *c*

Objective Function

- Precision
  - Minimize "incorrect" covers:

$$f_6(\mathcal{R}) = N \cdot |\mathcal{S}| - \sum_{r \in \mathcal{R}} |\text{incorrect-cover}(r)|$$

Given a rule $r = (s,c)$, the no. of data points which satisfy s but do not belong to class c.

Objective Function

- Recall
  - Encourage at least one “correct” cover per data point:

$$f_7(\mathcal{R}) = \sum_{(\mathbf{x},y)\in\mathcal{D}} 1\left(|\{r|(\mathbf{x},y) \in \text{correct-cover}(r)\}| \geq 1\right)$$

Given a rule $r = (s,c)$, the no. of data points which satisfy s and belong to class c.

Objective Function

- Complete objective is

$$\underset{\mathcal{R} \subseteq \mathcal{S} \times \mathcal{C}}{\operatorname{argmax}} \sum_{i=1}^{7} \lambda_i f_i(\mathcal{R})$$

- The intra-class and inter-class overlap terms are non-monotone

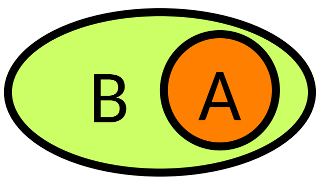
- The parsimony, overlap, and precision terms are non-normal

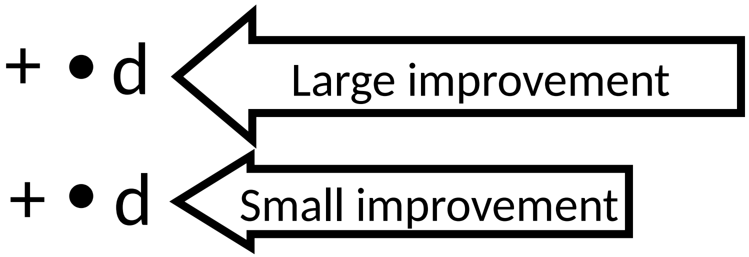
- All the component terms are submodular

Submodularity

Diminishing returns characterization

$$\underbrace{F(A \ \hat{}\ d) - F(A)}_{\text{Gain of adding } \boldsymbol{d} \text{ to a small set}} \geq \underbrace{F(B \ \hat{}\ d) - F(B)}_{\text{Gain of adding } \boldsymbol{d} \text{ to a large set}}$$





A non-negative linear combination of submodular functions is submodular

Objective Function

- Complete objective is

$$\underset{\mathcal{R} \subseteq \mathcal{S} \times \mathcal{C}}{\operatorname{argmax}} \sum_{i=1}^{7} \lambda_i f_i(\mathcal{R})$$

The complete objective is non-negative, non-normal, non-monotone, submodular

non-normal

- All the component terms are submodular

Optimizing the Objective

- Maximizing a non-monotone submodular function is NP-hard

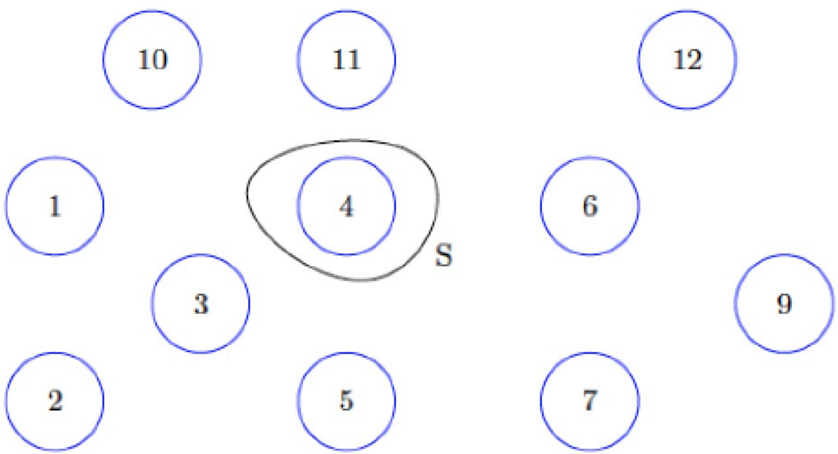
- Smooth local search [SLS] algorithm provides a 2/5 approximation [Feige, Mirrokni, Vondrak FOCS 07; SIAM Comp. J. 11]
  - Will be at least 2/5 of the optimal solution

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Each node here corresponds to a candidate rule = (pattern, class) tuple



$$f(S) \overset{S=\{4\}}{=} 10$$

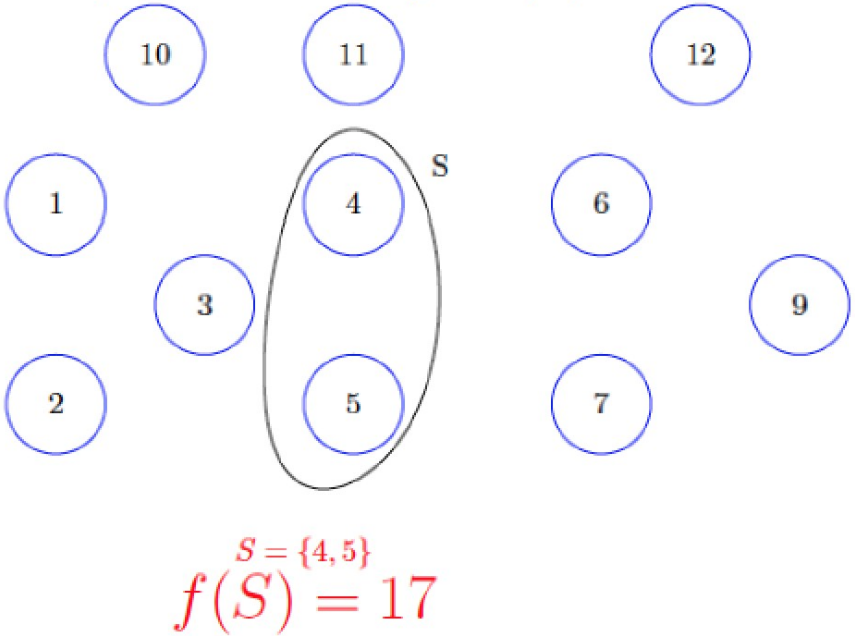
S and S' correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Improving local operation if $f(S') > f(S)$.



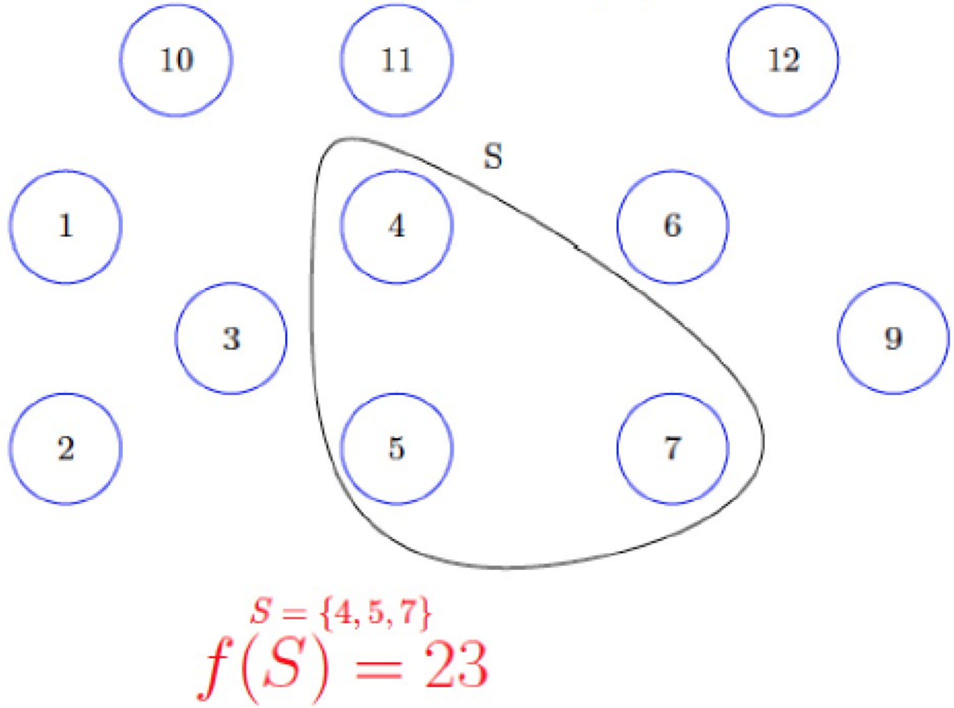
S and S' correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Improving local operation if $f(S') > f(S)$.



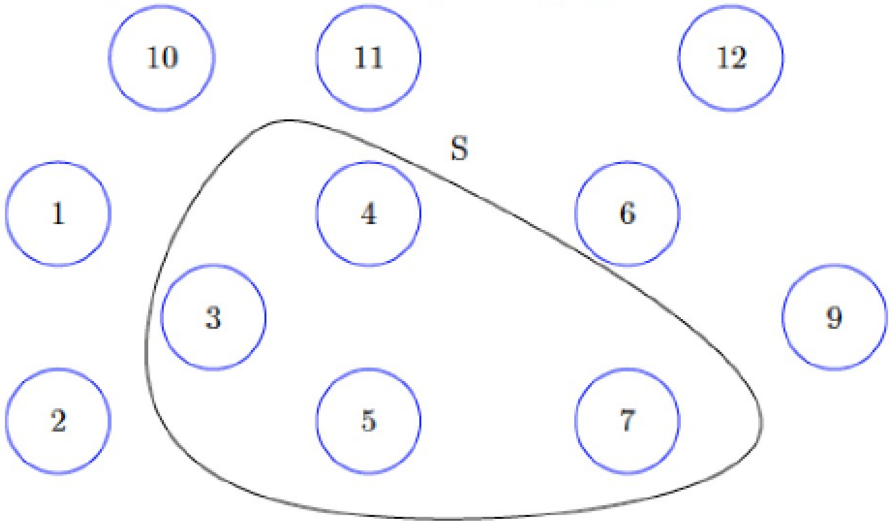
S and S’ correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Improving local operation if $f(S') > f(S)$.



$S = \{4, 5, 7, 3\}$

$f(S) = 27$

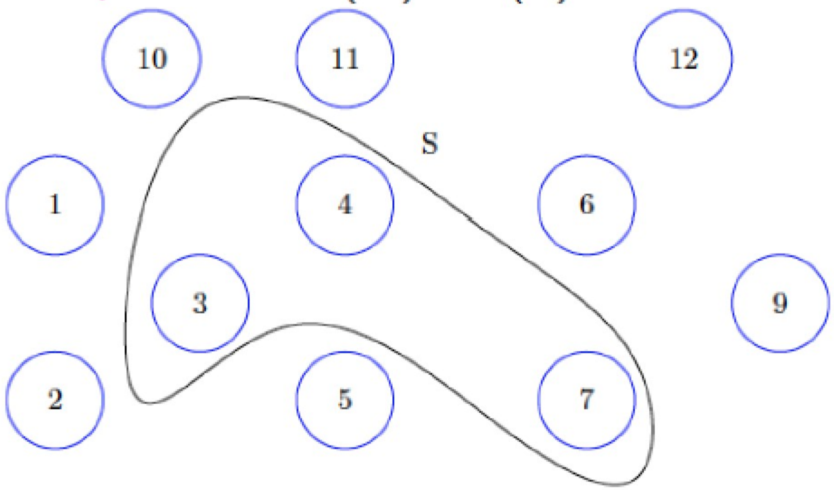
S and S' correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \setminus \{v\}$.

Improving local operation if $f(S') > f(S)$.



$S = \{4, 7, 3\}$

$f(S) = 30$

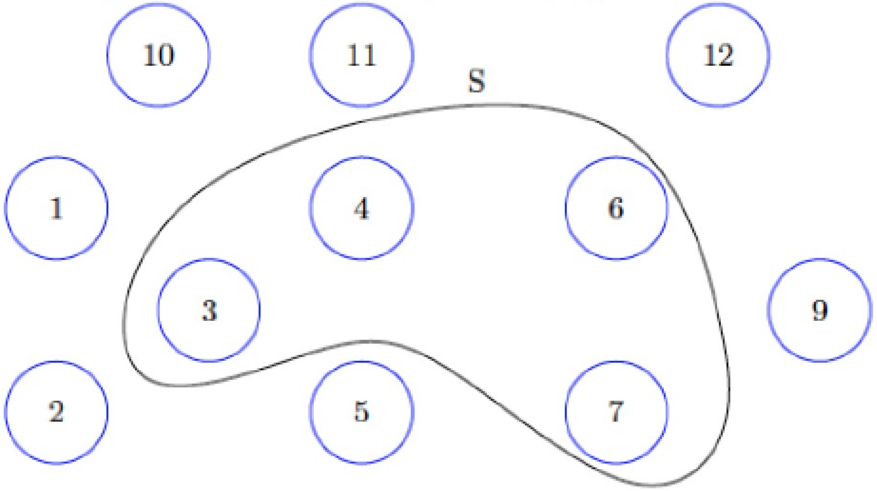
S and S' correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Improving local operation if $f(S') > f(S)$.



$S = \{4, 7, 3, 6\}$
$f(S) = 33$

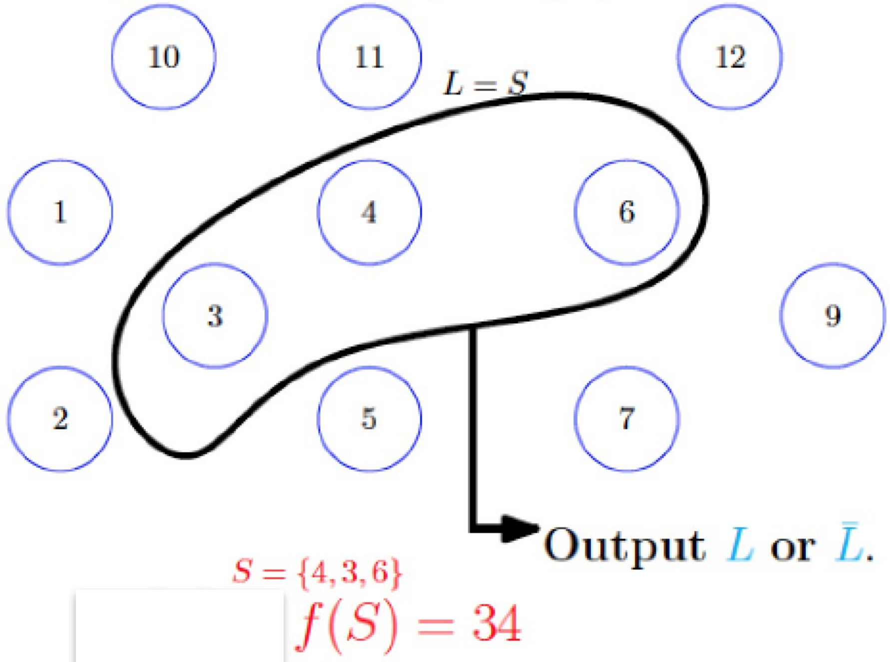
S and S' correspond to the intermediate solution sets

Submodular Maximization: Local Search

Local Operation:

- Add v: $S' = S \cup \{v\}$.
- Remove v: $S' = S \backslash \{v\}$.

Improving local operation if $f(S') > f(S)$.



S and S' correspond to the intermediate solution sets

Local Search

- ~1/3 approximation
  - At least 1/3 of optimal solution

- we use a slightly different version of this algorithm
  - Smooth local search
  - 2/5 approximation

Smooth Local Search

Initialization

Marginal gain

If marginal gain >
threshold,
Add element

If marginal gain <
threshold,
Add element

Algorithm 1 Smooth Local Search (SLS)

1: **Input:** Objective f, domain $X = \mathcal{S} \times \mathcal{C}$, parameters δ and δ'
2:
3: $A = \emptyset$
4: $\texttt{OPT} = f(\Phi_X(X, 0))$
5: **for** each element $x \in X$ **do**
6: Estimate $\mathbb{E}[f(\Phi_X(A, \delta) \cup x)] - \mathbb{E}[f(\Phi_X(A, \delta) \setminus x)]$ within an error of $\frac{1}{|X|^2}\texttt{OPT}$
7: Call this estimate $\tilde{\omega}_{A,\delta}(x)$
8: **end for**
9: **for** each element $x \in X \setminus A$ such that $\tilde{\omega}_{A,\delta}(x) > \frac{2}{|X|^2}\texttt{OPT}$ **do**
10: $A = A \cup x$
11: Goto Line 5
12: **end for**
13: **for** each element $x \in A$ such that $\tilde{\omega}_{A,\delta}(x) < \frac{-2}{|X|^2}\texttt{OPT}$ **do**
14: $A = A \setminus x$
15: Goto Line 5
16: **end for**
17: **return** $\Phi_X(A, \delta')$

Evaluation: Datasets

Dataset	# of datapoints	Features	Classes
Bail Outcomes	86K	Gender, age, current offense details, past criminal record	No Risk, Failure to Appear, New Criminal Activity
Student Performance	21K	Gender, age, grades, `absence rates & tardiness behavior through grades 6 to 8, suspension/withdrawal history	Graduated on Time, Delayed Graduation, Dropped out
Medical Diagnosis	150K	Current ailments, age, BMI, gender, smoking habits, medical history, family history	Asthma, Diabetes, Depression, Lung Cancer, Rare Blood Cancer

Evaluating Predictive Performance

Method	AUC Bail Data	AUC Student Data	AUC Medical Data
Our Approach	**69.78**	**75.12**	**61.19**
Bayesian Decision Lists (Letham et. al.)	67.18	72.54	59.18
Classification Based on Association (Liu et. al.)	70.68	76.02	63.03
CN2	71.02	76.36	64.78
Decision Trees	70.08	75.31	63.28
Gradient Boosted Trees	71.23	77.18	64.21
Random Forests	70.87	77.12	63.92

Evaluating Goodness of Rules

- Results on Medical Diagnosis Data

Method	Fraction of Overlap	Fraction of Data Points Uncovered	Avg. Rule Width	Num. Rules	Fraction of Classes Covered
Our Approach	**0.09**	**0.13**	**3.17**	**12**	**1.0**
Bayesian Decision Lists (Letham et. al.)	0.00	0.18	8.46	11	0.67
Classification Based on Association (Liu et. al.)	0.00	0.14	8.60	32	1.00
CN2	0.12	0.14	9.78	38	1.00

Ablation Study

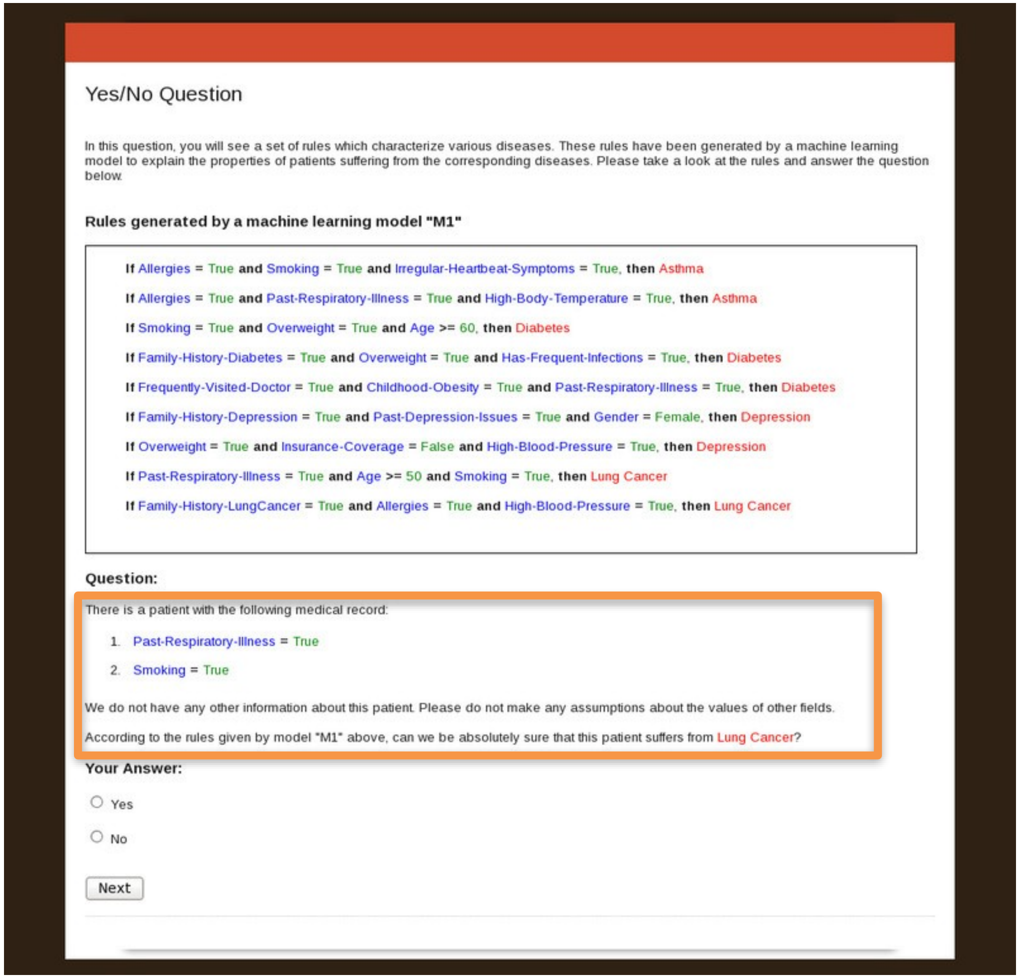
- Results on Medical Diagnosis Data

	AUC	Frac. Overlap	Frac. Uncov.	Rule Length	Num. Rules	Frac. Classes
Full IDS	61.19	0.09	0.13	3.17	12	1.00
No Prec.	51.26	0.09	0.19	3.19	12	1.00
No Recall	53.38	0.10	0.14	3.18	11	1.00
No Overlap	61.02	0.16	0.14	3.54	11	1.00
No Conc.	63.64	0.04	0.13	6.88	14	1.00
No Class	59.28	0.01	0.15	3.09	10	0.83

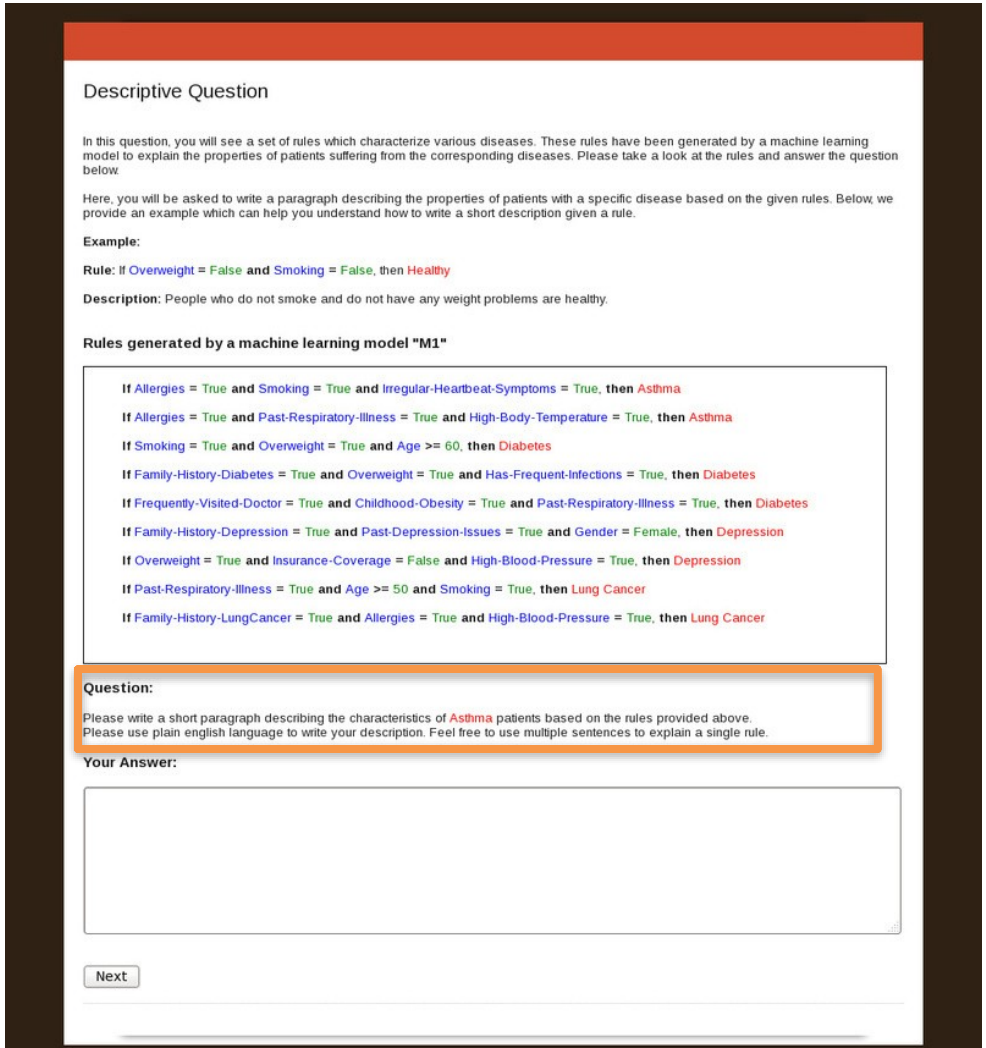
Evaluating Interpretability: User Study

- Compared our interpretable decision sets to Bayesian Decision Lists (Letham et. al.)
- Each user is randomly assigned one of the two models
- 10 objective and 2 descriptive questions per user

Interface for Objective Questions



Interface for Descriptive Questions



Descriptive Question

In this question, you will see a set of rules which characterize various diseases. These rules have been generated by a machine learning model to explain the properties of patients suffering from the corresponding diseases. Please take a look at the rules and answer the question below.

Here, you will be asked to write a paragraph describing the properties of patients with a specific disease based on the given rules. Below, we provide an example which can help you understand how to write a short description given a rule.

Example:

Rule: If Overweight = False **and** Smoking = False, then Healthy

Description: People who do not smoke and do not have any weight problems are healthy.

Rules generated by a machine learning model "M1"

If Allergies = True **and** Smoking = True **and** Irregular-Heartbeat-Symptoms = True, **then** Asthma

If Allergies = True **and** Past-Respiratory-Illness = True **and** High-Body-Temperature = True, **then** Asthma

If Smoking = True **and** Overweight = True **and** Age >= 60, **then** Diabetes

If Family-History-Diabetes = True **and** Overweight = True **and** Has-Frequent-Infections = True, **then** Diabetes

If Frequently-Visited-Doctor = True **and** Childhood-Obesity = True **and** Past-Respiratory-Illness = True, **then** Diabetes

If Family-History-Depression = True **and** Past-Depression-Issues = True **and** Gender = Female, **then** Depression

If Overweight = True **and** Insurance-Coverage = False **and** High-Blood-Pressure = True, **then** Depression

If Past-Respiratory-Illness = True **and** Age >= 50 **and** Smoking = True, **then** Lung Cancer

If Family-History-LungCancer = True **and** Allergies = True **and** High-Blood-Pressure = True, **then** Lung Cancer

Question:

Please write a short paragraph describing the characteristics of Asthma patients based on the rules provided above.
Please use plain english language to write your description. Feel free to use multiple sentences to explain a single rule.

Your Answer:

Next

User Study Results

Task	Metrics	Our Approach	Bayesian Decision Lists
Descriptive	Human Accuracy	0.81	0.17
	Avg. Time Spent (secs.)	113.4	396.86
	Avg. # of Words	31.11	120.57
Objective	Human Accuracy	0.97	0.82
	Avg. Time Spent (secs.)	28.18	36.34

Objective Questions: 17% more accurate, 22% faster;
Descriptive Questions: 74% fewer words, 71% faster.