Learning Interpretable Classification Rules with Boolean Compressed Sensing

Dmitry M. Malioutov, Kush R. Varshney, Amin Emad, and Sanjeeb Dash

Abstract An important problem in the context of supervised machine learning is designing systems which are interpretable by humans. In domains such as law, medicine, and finance that deal with human lives, delegating the decision to a black-box machine-learning model carries significant operational risk, and often legal implications, thus requiring interpretable classifiers. Building on ideas from Boolean compressed sensing, we propose a rule-based classifier which explicitly balances accuracy versus interpretability in a principled optimization formulation. We represent the problem of learning conjunctive clauses or disjunctive clauses as an adaptation of a classical problem from statistics, Boolean group testing, and apply a novel linear programming (LP) relaxation to find solutions. We derive theoretical results for recovering sparse rules which parallel the conditions for exact recovery of sparse signals in the compressed sensing literature. This is an exciting development in interpretable learning where most prior work has focused on heuristic solutions. We also consider a more general class of rule-based classifiers, checklists and scorecards, learned using ideas from threshold group testing. We show competitive classification accuracy using the proposed approach on real-world data sets.

Abbreviations

1Rule:	Boolean compressed sensing-based single rule learner
C5.0:	C5.0 Release 2.06 algorithm with rule set option in SPSS
CART:	Classification and regression trees algorithm in Matlab's classregtree function
CS:	Compressed sensing

D.M. Malioutov (✉) • K.R. Varshney • S. Dash
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
e-mail: dmalioutov@us.ibm.com; krvarshn@us.ibm.com; sanjeebd@us.ibm.com

A. Emad
Institute for Genomic Biology, University of Illinois, Urbana Champaign, Urbana, IL, USA

1218 Thomas M. Siebel Center for Computer Science, University of Illinois, Urbana, IL 61801, USA
e-mail: emad2@illinois.edu

T. Cerquitelli et al. (eds.), *Transparent Data Mining for Big and Small Data*,
Studies in Big Data 11, DOI 10.1007/978-3-319-54024-5_5

DList: Decision lists algorithm
ILPD: Indian liver patient dataset
Ionos: Ionosphere dataset
IP: Integer programming
kNN: The k-nearest neighbor algorithm
Liver: BUPA liver disorders dataset
LP: Linear programming
Parkin: Parkinsons dataset
Pima: Pima Indian diabetes dataset
RuB: Boosting approach rule learner
RuSC: Set covering approach rule learner
SCM: Set covering machine
Sonar: Connectionist bench sonar dataset
SQGT: Semiquantitative group testing
SVM: Support vector machine
TGT: Threshold group testing
Trans: Blood transfusion service center dataset
TrBag: The random forests classifier in Matlab's TreeBagger class
UCI: University of California Irvine
WDBC: Wisconsin diagnostic breast cancer dataset

1 Introduction

A great variety of formulations have been developed for the supervised learning problem, but the most powerful among these, such as kernel support vector machines (SVMs), gradient boosting, random forests, and neural networks are essentially black boxes in the sense that it is difficult for humans to interpret them. In contrast, early heuristic approaches such as decision lists that produce Boolean rule sets [9, 10, 44] and decision trees, which can be distilled into Boolean rule sets [43], have a high level of interpretability and are still widely used by analytics practitioners for this reason despite being less accurate. It has been frequently noted that Boolean rules with a small number of terms are the most well-received, trusted, and adopted outputs by human decision makers [32].

We approach this problem from a new computational lens. The sparse signal recovery problem in the Boolean algebra, now often known as Boolean compressed sensing, has received much recent research interest in the signal processing literature [2, 8, 21, 27, 34, 38, 46]. The problem has close ties to classic nonadaptive group testing [16, 17], and also to the compressed sensing and sparse signal recovery literature [6]. Building upon strong results for convex relaxations in compressed sensing, one advance has been the development of a linear programming (LP) relaxation with exact recovery guarantees for Boolean compressed sensing [34].[1] In this chapter,

[1]Other approaches to approximately solve group testing include greedy methods and loopy belief propagation; see references in [34].

we use the ideas of Boolean compressed sensing to address the problem of learning classification rules based on generalizable Boolean formulas from training data, thus drawing a strong connection between these two heretofore disparate problems. We develop a sparse signal formulation for the supervised learning of Boolean classification rules and develop a solution through LP relaxation.

The primary contribution of this work is showing that the problem of learning sparse conjunctive clause rules and sparse disjunctive clause rules from training samples can be represented as a group testing problem, and that we can apply an LP relaxation that resembles the basis pursuit algorithm to solve it [35]. Despite the fact that learning single clauses is NP-hard, we also establish conditions under which, if the data can be perfectly classified by a sparse Boolean rule, the relaxation recovers it exactly. To the best of our knowledge, this is the first work that combines compressed sensing ideas with classification rule learning to produce optimal sparse (interpretable) rules.

Due to the practical concern of classifier interpretability for adoption and impact [24, 49], there has been a renewed interest in rule learning that attempts to retain the interpretability advantages of rules, but changes the training procedures to be driven by optimizing an objective rather than being heuristic in nature [3, 14, 23, 28, 31, 45]. Set covering machines (SCM) formulate rule learning with an optimization objective similar to ours, but find solutions using a greedy heuristic rather than the LP relaxation that we propose [39]. Local analysis of data [5] also considers an optimization formulation to find compact rules, but they treat positive and negative classes separately and do not explicitly balance errors vs. interpretability. Maximum monomial agreement [19] also uses linear programming for learning rules, but they do not encourage sparsity. Two new interpretable rule learning methods which have substantially more complicated optimization formulations appeared after the initial presentation of this work [48, 50, 54].

In addition to considering ordinary classification rules, we show that the connection to sparse recovery can shed light on a related problem of learning checklists and scorecards, which are widely used in medicine, finance and insurance as a simple rubric to quickly make decisions. Such scorecards are typically constructed manually based on domain expert intuition. We show that the problem of automatically learning checklists from data can also be viewed as a version of Boolean sparse recovery, with strong connections to the threshold group-testing problem.

In today's age of big data, machine learning algorithms are often trained in the presence of a large number of features and a large number of samples. In our proposed LP formulation, this results in a large number of variables and a large number of constraints, i.e., columns and rows in the sensing matrix (using the terminology of compressed sensing). We would like to be able to reduce the number of columns and rows before solving the LP for tractability.

The certifiable removal of variables that will not appear in the optimal solution is known as *screening* [11, 20, 33, 51–53, 55–57]; in this chapter, we develop novel screening tests for Boolean compressed sensing and nonadaptive group testing [12].

Specifically, we develop two classes of screening tests: simple screening rules that arise from misclassification error counting arguments, and rules based on obtaining a feasible primal-dual pair of LP solutions.

Additionally, we develop a novel approach to reduce the number of rows [13]. In a sequential setting, we can certify reaching a near-optimal solution while only solving the LP on a small fraction of the available samples. Related work has considered progressive cross-validation error and argues that when this error 'stabilizes' while being exposed to a stream of i.i.d. training samples, then the classifier reaches near-optimal test classification accuracy [4]. The *learning curve* literature in machine learning has considered how the generalization error evolves as a function of the received number of samples [29, 40, 42]. In the context of ordinary (non-Boolean) compressed sensing, [36] has developed sequential tests to establish that a sufficient number of measurements has been obtained to recover the correct sparse signal.

We demonstrate the proposed approach on several real-world data sets and find that the proposed approach has better accuracy than heuristic decision lists and has similar interpretability. The accuracy is in fact on par with less interpretable weighted rule set induction, and not far off from the best non-interpretable classifiers. We find that our screening tests are able to safely eliminate a large fraction of columns, resulting in large decreases in running time. We also find that through our row sampling approach, we are able to obtain accurate near-optimal interpretable classification rules while training on only a small subset of the samples.

The remainder of this chapter is organized as follows. In Sect. 2 we provide a brief review of group testing and formulate supervised binary classification as a group testing problem. In Sect. 3, we present an LP relaxation for rule learning and show that it can recover rules exactly under certain conditions. Section 4 extends the discussion to scorecards. Section 5 develops screenings tests and row sampling. Empirical evaluations on real-world data are given in Sect. 6. We conclude with a summary and discussion in Sect. 7.

2 Boolean Rule Learning as Group Testing

In this section, we formulate the problem of learning sparse AND-clauses such as:

$$\text{height} \leq 6 \text{ feet AND } \text{weight} > 200 \text{ pounds}$$

and OR-clauses such as:

$$\text{Smoke} = \text{True OR } \text{Exercise} = \text{False OR } \text{Blood Pressure} = \text{High}$$

from training data via group testing.

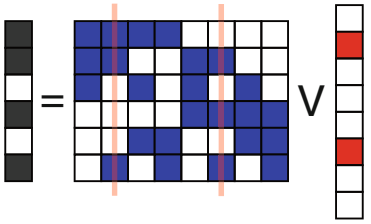

Fig. 1 Illustration of group testing, $\mathbf{y} = \mathbf{A} \vee \mathbf{w}$: measurements $\mathbf{y}$ are Boolean combinations of the sparse unknown vector $\mathbf{w}$. Matrix $\mathbf{A}$ specifies which subjects participate in which pooled test

2.1 The Group Testing Problem

Group testing [16] was developed during World War II, when the US military needed to conduct expensive medical tests on a large number of soldiers. The key idea was that if the test was performed on a group of soldiers simultaneously, by combining their blood samples into a pooled test, the cost could be dramatically reduced. Consider an $m \times n$ Boolean matrix $\mathbf{A}$, where the rows represent different pools (subsets of subjects) and the columns represent the subjects. An entry a_{ij} is one if subject j is part of a pool i and zero otherwise. The true states of the subjects (unknown when conducting the tests) are represented by vector $\mathbf{w} \in \{0,1\}^n$. Group testing, in which the result of a test is the OR of all subjects in a pool, results in a Boolean vector $\mathbf{y} \in \{0,1\}^m$. We summarize the result of all m tests using:

$$\mathbf{y} = \mathbf{A} \vee \mathbf{w}, \tag{1}$$

which represents Boolean matrix-vector multiplication, i.e.,

$$y_i = \bigvee_{j=1}^{n} a_{ij} \wedge w_j. \tag{2}$$

We illustrate this idea in Fig. 1. In the presence of measurement errors,

$$\mathbf{y} = (\mathbf{A} \vee \mathbf{w}) \oplus \mathbf{n}, \tag{3}$$

where $\oplus$ is the XOR operator and $\mathbf{n}$ is a noise vector.

Once the tests have been conducted, the objective is to recover $\mathbf{w}$ from $\mathbf{A}$ and the measured $\mathbf{y}$. The recovery can be stated through the following combinatorial optimization problem:

$$\min \|\mathbf{w}\|_0 \quad \text{such that } \mathbf{y} = \mathbf{A} \vee \mathbf{w}, \mathbf{w} \in \{0,1\}^n. \tag{4}$$

In the presence of noise we use parameter λ to balance sparsity of $\mathbf{w}$ and the errors $\mathbf{n}$:

$$\min_{\mathbf{w}} \ \lambda \|\mathbf{w}\|_0 + \sum n_i \quad \text{such that } \mathbf{y} = (\mathbf{A} \vee \mathbf{w}) \oplus \mathbf{n}, \quad \mathbf{w}, \mathbf{n} \in \{0,1\}^n. \tag{5}$$

2.2 Supervised Classification Rule Formulation

We have described group testing; now we show how the formulation can be adapted to rule-based classification. The problem setup of interest is standard binary supervised classification. We are given m labeled training samples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ where the $\mathbf{x}_i \in \mathscr{X}$ are the features in some discrete or continuous space $\mathscr{X}$ and the $y_i \in \{0, 1\}$ are the Boolean labels. We would like to learn a function $\hat{y}(\cdot) : \mathscr{X} \rightarrow \{0, 1\}$ that will accurately generalize to classify unseen, unlabeled feature vectors drawn from the same distribution as the training samples.

In rule-based classifiers, the clauses are made up of individual Boolean terms, e.g., 'weight > 200.' Such a term can be represented by a function $a(\mathbf{x})$ mapping the feature vector to a boolean number. To represent the full diversity and dimensions of the feature space $\mathscr{X}$, we have many such Boolean terms $a_j(\cdot) : \mathscr{X} \rightarrow \{0, 1\}$, $j = 1, \ldots, n$. For each continuous dimension of $\mathscr{X}$, these terms may be comparisons to several suitably chosen thresholds. Then for each of the training samples, we can calculate the truth value for each of the terms, leading to an $m \times n$ truth table $\mathbf{A}$ with entries $a_{ij} = a_j(\mathbf{x}_i)$.

Writing the true labels of the training set as a vector $\mathbf{y}$, we can write the same expression in the classification problem as in group testing (3): $\mathbf{y} = (\mathbf{A} \vee \mathbf{w}) \oplus \mathbf{n}$. In the classification problem, $\mathbf{w}$ is the binary vector to be learned that indicates the rule. The nonzero coefficients directly specify a Boolean clause classification rule which can be applied to new unseen data. This clause is a disjunctive OR-rule. In most of the rule-based classification literature, however, the learning of AND-clauses is preferred. This is easy to handle using DeMorgan's law. If we complement $\mathbf{y}$ and $\mathbf{A}$ prior to the learning, then we have:

$$\mathbf{y} = \mathbf{A} \wedge \mathbf{w} \quad \Leftrightarrow \quad \mathbf{y}^C = \mathbf{A}^C \vee \mathbf{w}. \tag{6}$$

Hence, our results apply to both OR-rules and AND-rules; we focus on the conjunctive case for the remainder of the chapter.

For interpretability and generalization, we are specifically interested in compact Boolean rules, i.e., we would like $\mathbf{w}$ to be sparse, having few non-zero entries. Therefore, the optimization problem to be solved is the same as for group testing (4). We describe our proposed solution next.

3 LP Relaxation

The group testing problem appears closely related to the compressed sensing (CS) problem from the signal processing literature [6]. Both group testing and compressed sensing involve sparse signal recovery, but group testing uses Boolean algebra instead of the typical real-valued linear algebra encountered in CS. In this section, based on their close connection, we show that suitably modified, efficient LP relaxations from compressed sensing can be used to solve the group testing problem, and hence also the classification rule learning problem.

3.1 Boolean Compressed Sensing-Based Formulation

Compressed sensing attempts to find an unknown high-dimensional but sparse real-valued vector $\mathbf{w}$ from a small collection of random measurements $\mathbf{y} = \mathbf{A}\mathbf{w}$, where $\mathbf{A}$ is a random matrix (e.g., with i.i.d. Gaussian entries). The problem is to find the sparsest solution, $\min \|\mathbf{w}\|_0$ such that $\mathbf{y} = \mathbf{A}\mathbf{w}$. It looks very close to group testing (4), except that $\mathbf{y}, \mathbf{A}$, and $\mathbf{w}$ are real-valued, and $\mathbf{A}\mathbf{w}$ denotes the standard matrix-vector product in linear algebra. In compressed sensing, the most popular technique for getting around the combinatorial ℓ_0 objective is to relax it using the convex ℓ_1-norm. This relaxation, known as basis pursuit, results in the following optimization problem:

$$\min \|\mathbf{w}\|_1 \quad \text{such that } \mathbf{y} = \mathbf{A}\mathbf{w}, \tag{7}$$

where $\mathbf{y}$, $\mathbf{w}$, and $\mathbf{A}$ are all real-valued and the product $\mathbf{A}\mathbf{w}$ is the standard matrix-vector product. This optimization problem (7) is a linear program and can be solved efficiently. It has been shown that under certain conditions on the matrix $\mathbf{A}$ and sparsity of $\mathbf{w}$, the ℓ_0 solution and the ℓ_1 solution are equivalent. The work of [34] extends the basis pursuit idea to Boolean algebras.

The challenge in compressed sensing is with the combinatorial nature of the ℓ_0 objective. Additionally, in the Boolean setting, Eq. (1) is not a set of linear constraints. However, if a vector $\mathbf{w}$ satisfies the constraint that $\mathbf{y} = \mathbf{A} \vee \mathbf{w}$, then it also satisfies the pair of ordinary linear inequalities $\mathbf{A}_{\mathscr{P}}\mathbf{w} \geq \mathbf{1}$ and $\mathbf{A}_{\mathscr{Z}}\mathbf{w} = \mathbf{0}$, where $\mathscr{P} = \{i | y_i = 1\}$ is the set of positive tests, $\mathscr{Z} = \{i | y_i = 0\}$ is the set of negative (or zero) tests, and $\mathbf{A}_{\mathscr{P}}$ and $\mathbf{A}_{\mathscr{Z}}$ are the corresponding subsets of rows of $\mathbf{A}$. We refer to the jth column of $\mathbf{A}, \mathbf{A}_{\mathscr{P}}$ and $\mathbf{A}_{\mathscr{Z}}$ as $\mathbf{a}^j, \mathbf{a}^j_{\mathscr{P}}$ and $\mathbf{a}^j_{\mathscr{Z}}$, respectively. The vectors $\mathbf{1}$ and $\mathbf{0}$ are all ones and all zeroes, respectively. These constraints can be incorporated into an LP. Thus the Boolean ℓ_1 problem is the integer program (IP):

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} w_j \\ \text{s.t.} \quad & w_j \in \{0, 1\}, j = 1, \ldots, n \\ & \mathbf{A}_{\mathscr{P}}\mathbf{w} \geq \mathbf{1} \\ & \mathbf{A}_{\mathscr{Z}}\mathbf{w} = \mathbf{0}. \end{aligned} \tag{8}$$

Because of the Boolean integer constraint on the weights, the problem (8) is NP-hard. We can further relax the optimization to the following tractable LP[2]:

[2]Instead of using LP, one can find solutions greedily, as is done in the SCM, which gives a $\log(m)$ approximation. The same guarantee holds for LP with randomized rounding. Empirically, LP tends to find sparser solutions.

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} w_j \\ \text{s.t.} \quad & 0 \le w_j \le 1, j = 1, \ldots, n \\ & \mathbf{A}_{\mathscr{P}}\mathbf{w} \ge \mathbf{1} \\ & \mathbf{A}_{\mathscr{Z}}\mathbf{w} = \mathbf{0}. \end{aligned} \tag{9}$$

If non-integer w_j are found, we either simply set them to one, or use randomized rounding. An exact solution to the integer LP can be obtained by branch and bound.[3]

Slack variables may be introduced in the presence of errors, when there may not be any sparse rules producing the labels $\mathbf{y}$ exactly, but there are sparse rules that approximate $\mathbf{y}$ very closely. This is the typical case in the supervised classification problem. The LP is then:

$$\begin{aligned} \min \quad & \lambda \sum_{j=1}^{n} w_j + \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & 0 \le w_j \le 1, j = 1, \ldots, n \\ & \mathbf{A}_{\mathscr{Z}}\mathbf{w} = \boldsymbol{\xi}_{\mathscr{Z}}, \qquad 0 \le \xi_i,\ i \in \mathscr{Z}. \\ & \mathbf{A}_{\mathscr{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathscr{P}} \ge \mathbf{1}, \quad 0 \le \xi_i \le 1,\ i \in \mathscr{P} \end{aligned} \tag{10}$$

The regularization parameter λ trades training error and the sparsity of $\mathbf{w}$.

3.2 Recovery Guarantees

We now use tools from combinatorial group testing [16, 17] to establish results for exact recovery and recovery with small error probability in AND-clause learning via LP relaxation. First, we introduce definitions from group testing.

Definition 1 A matrix $\mathbf{A}$ is *K-separating* if Boolean sums of sets of K columns are all distinct.

Definition 2 A matrix $\mathbf{A}$ is *K-disjunct* if the union (boolean sum) of any K columns does not contain any other column.

Any K-disjunct matrix is also K-separating. The K-separating property for $\mathbf{A}$ is sufficient to allow exact recovery of $\mathbf{w}$ with up to K nonzero entries, but in general, requires searching over all K-subsets out of n [16]. The property of K-

[3] Surprisingly, for many practical datasets the LP formulation obtains integral solutions, or requires a small number of branch and bound steps.

disjunctness, which can be viewed as a Boolean analog of spark [15], is a more restrictive condition that allows a dramatic simplification of the search: a simple algorithm that considers rows where $y_i = 0$ and sets all w_j where $a_{ij} = 1$ to zero and the remaining w_j to one, is guaranteed to recover the correct solution. For non-disjunct matrices this simple algorithm finds feasible but suboptimal solutions.

We recall a lemma on exact recovery by the LP relaxation.

Lemma 1 ([34]) *Suppose there exists a* $\mathbf{w}^*$ *with K nonzero entries and* $\mathbf{y} = \mathbf{A} \vee \mathbf{w}^*$. *If the matrix* $\mathbf{A}$ *is K-disjunct, then LP solution* $\hat{\mathbf{w}}$ *in* (9) *recovers* $\mathbf{w}^*$, *i.e.,* $\hat{\mathbf{w}} = \mathbf{w}^*$.

This lemma was presented in the group testing context. To apply it to rule learning, we start with classification problems with binary features in which case, the matrix $\mathbf{A}$ simply contains the feature values.[4] A simple corollary of Lemma 1 is that if $\mathbf{A}$ is K-disjunct and there is an underlying error-free K-term AND-rule, then we can recover the rule exactly via (9).

A critical question is when can we expect our features to yield a K-disjunct matrix?

Lemma 2 *Suppose that for each subset of* $K + 1$ *features, we find at least one example of each one of the* 2^{K+1} *possible binary* $(K + 1)$*-patterns among our m samples. Then the matrix* $\mathbf{A}$ *is K-disjunct.*

Proof Note that there are 2^{K+1} possible binary patterns for K features. Suppose that on the contrary the matrix is not K-disjunct. Without loss of generality, K-disjunctness fails for the first K columns covering the $(K + 1)$-st one. Namely, columns $\mathbf{a}_1, \ldots, \mathbf{a}_{K+1}$ satisfy $\mathbf{a}_{K+1} \subset \cup_{k=1}^{K} \mathbf{a}_k$. This is clearly impossible, since by our assumption the pattern $(0, 0, \ldots, 0, 1)$ for our $K + 1$ variables is among our m samples. □

To interpret the lemma: if features are not strongly correlated, then for any fixed K, for large enough m we will eventually obtain all possible binary patterns. Using a simple union bound, for the case of uncorrelated equiprobable binary features, the probability that at least one of the K-subsets exhibits a non-represented pattern is bounded above by $\binom{n}{K} 2^K (1 - (1/2)^K)^m$. Clearly as $m \to \infty$ this bound approaches zero: with enough samples $\mathbf{A}$ is K-disjunct.

These results also carry over to approximate disjunctness [35] (also known as a weakly-separating design) to develop less restrictive conditions when we allow a small probability of recovery error [37, 41].

In the case of classification with continuous features, we discretize feature dimension x_j using thresholds $\theta_{j,1} \leq \theta_{j,2} \leq \cdots \leq \theta_{j,D}$ such that the columns of $\mathbf{A}$ corresponding to x_j are the outputs of Boolean indicator functions $I_{x_j \leq \theta_{j,1}}(\mathbf{x}), \ldots, I_{x_j \leq \theta_{j,D}}(\mathbf{x}), I_{x_j > \theta_{j,1}}(\mathbf{x}), \ldots, I_{x_j > \theta_{j,D}}(\mathbf{x})$. This matrix is not disjunct

[4] In general it will contain the features and their complements as columns. However, with enough data, one of the two choices will be removed by zero-row elimination beforehand.

because, e.g., $I_{x_j>\theta_{j,1}}(\mathbf{x}) \geq I_{x_j>\theta_{j,2}}(\mathbf{x})$. However, without loss of generality, for each feature we can remove all but one of the corresponding columns of $\mathbf{A}$ as discussed in Sect. 5.1. Through this reduction we are left with a simple classification problem with binary features; hence the results in Lemma 1 apply to continuous features.

3.3 Multi-Category Classification

Extending the proposed rule learner from binary classification to M-ary classification is straightforward through one-vs-all and all-vs-all constructions, as well as a Venn diagram-style approach in which we expand the $\mathbf{y}$ and $\mathbf{w}$ vectors to be matrices $\mathbf{Y} = \left[\mathbf{y}^1 \cdots \mathbf{y}^{\lceil \log_2 M \rceil}\right]$ and $\mathbf{W} = \left[\mathbf{w}^1 \cdots \mathbf{w}^{\lceil \log_2 M \rceil}\right]$. We briefly explain how this can be done for $M = 4$ classes with labels $c_1, \ldots, c_4$. We first construct the label matrix with value 1 in $\mathbf{y}^1$ if a sample has label c_1 or c_2 and zero otherwise. Similarly, an element of $\mathbf{y}^2$ takes value 1 if a sample has label c_1 or c_3 and zero otherwise. The problem of interest then becomes $\mathbf{Y} \approx \mathbf{A} \vee \mathbf{W}$. The constraints in the LP relaxation become $\mathbf{A}_{\mathscr{P}_1}\mathbf{w}^1 \geq \mathbf{1}$, $\mathbf{A}_{\mathscr{P}_2}\mathbf{w}^2 \geq \mathbf{1}$, $\mathbf{A}_{\mathscr{Z}_1}\mathbf{w}^1 = \mathbf{0}$, and $\mathbf{A}_{\mathscr{Z}_2}\mathbf{w}^2 = \mathbf{0}$, where $\mathscr{P}_1 = \{i|y_{1,i} = 1\}$, $\mathscr{P}_2 = \{i|y_{2,i} = 1\}$, $\mathscr{Z}_1 = \{i|y_{1,i} = 0\}$, and $\mathscr{Z}_2 = \{i|y_{2,i} = 0\}$. From a solution $\mathbf{w}^1$, $\mathbf{w}^2$, we will have two corresponding AND-rules which we denote r_1, r_2. A new sample is classified as c_1 if $r_1 \wedge r_2$, as c_2 if $r_1 \wedge \neg r_2$, as c_3 if $\neg r_1 \wedge r_2$, and as c_4 if $\neg r_1 \wedge \neg r_2$.

4 Learning Scorecards Using Threshold Group Testing

We now build upon our approach to learn sparse Boolean AND or OR rules and develop a method for learning interpretable scorecards using sparse signal representation techniques. In the face of high complexity, checklists and other simple scorecards can significantly improve people's performance on decision-making tasks [26]. An example of such a tool in medicine, the *clinical prediction rule*, is a simple decision-making rubric that helps physicians estimate the likelihood of a patient having or developing a particular condition in the future [1]. An example of a clinical prediction rule for estimating the risk of stroke, known as the CHADS_2 score, is shown in Table 1 [25]. The health worker determines which of the five diagnostic indicators a patient exhibits and adds the corresponding points together. The higher the total point value is, the greater the likelihood the patient will develop a stroke. This rule was manually crafted by health workers and notably contains few conditions with small integer point values and is extremely interpretable by people.

Recent machine learning research has attempted to learn clinical prediction rules that generalize accurately from large-scale electronic health record data rather than relying on manual development [31, 48]. The key aspect of the problem is maintaining the simplicity and interpretability of the learned rule to be similar to

Table 1 $CHADS_2$ clinical prediction rule for estimating risk of stroke

Condition	Points
Congestive heart failure	1
Hypertension	1
Age $\geq$ 75	1
Diabetes mellitus	1
Prior stroke, transient ischemic attack, or thromboembolism	2

the hand-crafted version, in order to enable trust and adoption by users in the health care industry. We again employ sparsity as a proxy for interpretability of the rules.

In the previous sections, the form of the classifier we considered was a sparse AND-rule or OR-rule whereas here, we would like to find a sparse set of conditions or features with small integer coefficients that are added together to produce a score. Such a model is between the "1-of-N" and "N-of-N" forms implied by OR-rules and AND-rules, with N active variables. With unit weights on the columns of A it can be viewed as an M-of-N rule. For learning Boolean rules, our Boolean compressed sensing formulation had a close connection with the group testing problem whereas here, the connection is to the *semiquantitative* group testing (SQGT) problem, or more precisely to its special case of threshold group testing (TGT) [21].

4.1 Threshold Group Testing

We start by describing the TGT model and then show how to formulate the interpretable rule learning problem as TGT. Let n, m, and d denote the total number of subjects, the number of tests, and the number of defectives, respectively. As we defined in Sect. 2.1, $\mathbf{A} \in \{0,1\}^{m\times n}$ is a binary matrix representing the assignment of subjects to each test, and $\mathbf{y} \in \{0,1\}^m$ is the binary vector representing the error-free results of the tests. Let $\mathscr{D}_t$ be the true set of defectives and binary vector $\mathbf{w}_t \in \{0,1\}^n$ represent which subject is a defective. In the TGT model, one has

$$\mathbf{y} = f_\eta(\mathbf{A}\mathbf{w}_t), \tag{11}$$

where $f_\eta(\cdot)$ is a quantizing function with threshold η, such that $f_\eta(x) = 0$ if $x < \eta$ and $f_\eta(x) = 1$ if $x \geq \eta$. The goal is to recover the unknown vector $\mathbf{w}$ given the test matrix $\mathbf{A}$ and the vector of test results $\mathbf{y}$.

4.2 LP Formulation for Threshold Group Testing

First, we start by considering the simplest model in which there are no errors in the vector of labels, and the threshold η is known in advance. In this case, and given that $\mathbf{w}_t$ is a sparse vector, one can find the sparsest binary vector that satisfies (11). However, this combinatorial problem is not computationally feasible for large datasets.

To overcome this problem, we use a similar relaxation to the one used in Sect. 3.1. We note that a vector $\mathbf{w}$ that satisfies the constraints imposed by the TGT model in (11), must also satisfy the pair of ordinary linear inequalities:

$$\mathbf{A}_{\mathscr{P}}\mathbf{w} \geq \eta\mathbf{1}, \tag{12}$$

$$\mathbf{A}_{\mathscr{Z}}\mathbf{w} < \eta\mathbf{1}, \tag{13}$$

where, same as before, $\mathscr{P} = \{i|\mathbf{y}(i) = 1\}$ and $\mathscr{Z} = \{i|\mathbf{y}(i) = 0\}$ are the sets of positive and negative tests. Using a convex l_1-norm instead of $||\mathbf{w}||_0$, and relaxing the binary constraint we obtain

$$\min \ ||\mathbf{w}||_1 \tag{14}$$

$$\text{s.t. } 0 \leq \mathbf{w}(j) \leq 1, \quad j = 1, \ldots, n \tag{15}$$

$$\mathbf{A}_{\mathscr{P}}\mathbf{w} \geq \eta\mathbf{1}, \tag{16}$$

$$\mathbf{A}_{\mathscr{Z}}\mathbf{w} \leq (\eta - 1)\mathbf{1}, \tag{17}$$

In presence of noise, we introduce slack variables ξ to allow violation of a small subset of the constraints. We also allow the threshold η to not be known a priori, and learn it from data. We propose the following formulation to jointly find the threshold η and the defective items (or our interpretable rules):

$$\begin{aligned}
&\min \ ||\mathbf{w}||_1 + \lambda||\boldsymbol{\xi}||_1 \qquad (18)\\
&\text{s.t. } 0 \leq \mathbf{w}(j) \leq 1, \quad j = 1, \ldots, n\\
&\quad 0 \leq \boldsymbol{\xi}(i) \leq 1, \quad i \in \mathscr{P}\\
&\quad 0 \leq \boldsymbol{\xi}(i), \quad i \in \mathscr{Z}\\
&\quad \mathbf{A}_{\mathscr{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathscr{P}} \geq \eta\mathbf{1},\\
&\quad \mathbf{A}_{\mathscr{Z}}\mathbf{w} \leq \eta\mathbf{1},\\
&\quad 0 \leq \eta \leq n.
\end{aligned}$$

4.3 Learning Scorecards with Threshold Group Testing

Our goal is to learn an interpretable function $\hat{y}(\cdot) : \mathscr{X} \rightarrow \{0,1\}$, given m labeled training samples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$. To formulate this problem as TGT, we form the vector of test results according to $\mathbf{y}(i) = y_i$, $i = 1, 2, \ldots, m$; also, we form the test matrix $\mathbf{A}$ according to $\mathbf{A}(i,j) = a_j(\mathbf{x}_i)$, where $a_j(\cdot) : \mathscr{X} \rightarrow \{0,1\}$, $j = 1, \ldots, n$, are simple Boolean terms (e.g. age $\geq$ 75). Furthermore, we assume that at most d simple Boolean terms govern the relationship between the labels and the features, i.e. $|\mathscr{D}_t| \leq d$, and this sparse set of terms are encoded in the unknown sparse vector $\mathbf{w}_t$. In addition, we assume that this relationship has the form of a "M-of-N" rule table; in other words, $y_i = 1$ if at least M terms of N are satisfied and $y_i = 0$, otherwise. Therefore, by setting $\eta = M$ and $d = N$, we can write this relationship as (11). Consequently, in order to find the set of interpretable rules corresponding to a "M-of-N" rule table, we need to recover the sparse vector $\mathbf{w}_t$ given $\mathbf{A}$ and $\mathbf{y}$.

4.4 Theoretical Guarantees for TGT

We now summarize results on recovery of sparse rules in the threshold group testing formulation, which generalize our results in Sect. 3.2. We start by defining a generalization of binary d-disjunct codes [30] which is studied under different names in the literature such as cover-free families (e.g. see [7, 18, 47]).

Definition 1 A matrix $\mathbf{A} \in [2]^{m \times n}$ is a (d, η)-disjunct matrix if for any two disjoint sets of column-indices $\mathscr{C}_z$ and $\mathscr{C}_o$,[5] where $|\mathscr{C}_z| = d - \eta + 1$, $|\mathscr{C}_o| = \eta$, and $\mathscr{C}_o \cap \mathscr{C}_z = \varnothing$, there exists at least one row indexed by r such that

$$\mathbf{A}(r,j) = 1 \quad \forall j \in \mathscr{C}_o,$$
$$\mathbf{A}(r,j) = 0 \quad \forall j \in \mathscr{C}_z.$$

Using (d, η)-disjunctness, the following theorem can be established using results in [21].

Theorem 1 *Let* $\mathbf{A}$ *be a* (d, η)*-disjunct binary matrix. The LP formulation* (14)–(17) *will uniquely identify the true set of defectives, i.e.* $\hat{\mathbf{w}} = \mathbf{w}_t$*, as long as* $\eta \leq |\mathscr{D}_t| \leq d$.

The main idea in proving this theorem is that we introduce a reversible transformation that converts the TGT model into another model resembling the Boolean compressed sensing formulation. Given this new formulation, we prove that the LP relaxation can uniquely identify the defectives, hence recovering the sparse set of rules.

[5] Here, the subscript "z" stands for zero and "o" stands for one.

5 Screening and Row Sampling

The formulation (10) produces an LP that becomes challenging for data sets with large numbers of features and large numbers of thresholds on continuous features. Our aim in this section is to provide computationally inexpensive pre-computations which allow us to eliminate the majority of the columns in the $\mathbf{A}$ matrix by providing a certificate that they cannot be part of the optimal solution to the Boolean ℓ_1-IP in (8). The LP also becomes challenging for large numbers of training samples. We develop an approach to only solve the LP for a subset of rows of $\mathbf{A}$ and obtain a near-optimal solution to the full problem.

We first discuss screening approaches that simply examine the nonzero patterns of $\mathbf{A}$ and $\mathbf{y}$, and then discuss screening tests that use a feasible primal-dual pair for the LP to screen even more aggressively. The tests can be applied separately or sequentially; in the end, it is the union of the screened columns that is of importance. Finally, we put forth a row sampling approach also based on feasible primal-dual pairs.

5.1 Simple Screening Tests

We first observe that a positive entry in column $\mathbf{a}^j_{\mathscr{Z}}$ corresponds to a false alarm error if the column is active (i.e., if the corresponding $w_j = 1$). The potential benefit of including column j is upper bounded by the number of positive entries in $\mathbf{a}^j_{\mathscr{P}}$. The first screening test is simply to remove columns in which $\|\mathbf{a}^j_{\mathscr{Z}}\|_0 \geq \|\mathbf{a}^j_{\mathscr{P}}\|_0$.

An additional test compares pairs of columns j and j' for different threshold values of the same continuous feature dimension of $\mathscr{X}$. We note that such columns form nested subsets in the sense of sets of nonzero entries. If θ_j and $\theta_{j'}$ are the thresholds defining $a_j(\cdot)$ and $a_{j'}(\cdot)$ with $\theta_j < \theta_{j'}$, then $\{k \mid x_k < \theta_j\} \subset \{k \mid x_k < \theta_{j'}\}$. Looking at the difference in the number of positive entries between columns of $\mathbf{A}_{\mathscr{Z}}$ and the difference in the number of positive entries between columns of $\mathbf{A}_{\mathscr{P}}$, we never select column j instead of column j' if $\|\mathbf{a}^{j'}_{\mathscr{P}}\|_0 - \|\mathbf{a}^j_{\mathscr{P}}\|_0 > \|\mathbf{a}^{j'}_{\mathscr{Z}}\|_0 - \|\mathbf{a}^j_{\mathscr{Z}}\|_0$ by a similar argument as before.

We consider two variations of this pairwise relative cost-redundancy test: first, only comparing pairs of columns such that $j' = j+1$ when the columns are arranged by sorted threshold values, and second, comparing all pairs of columns for the same continuous feature, which has higher computational cost but can yield a greater fraction of columns screened. Although most applicable to columns corresponding to different threshold values of the same continuous feature of $\mathscr{X}$, the same test can be conducted for any two columns j and j' across different features.

5.2 Screening Tests Based on a Feasible Primal-Dual Pair

We also introduce another kind of screening tests based on LP duality theory, which can further reduce the number of columns when a primal-dual feasible pair is available. We describe a cost-effective way to provide such primal dual pairs. Specifically, we first reformulate (10) along with the requirement that $\mathbf{w}$ is a Boolean vector as a minimum weight set cover problem. Then, if we have a feasible binary primal solution available, we can produce certificates that w_j cannot be nonzero in the optimal solution as follows. If by setting $w_j = 1$ and recomputing the dual, the dual objective function value exceeds the primal objective function value, then any solution with $w_j = 1$ is strictly inferior to the feasible binary primal solution that we started with and we can remove column $\mathbf{a}^j$. Thus a key step is finding feasible binary primal and dual solutions on which we can base the screening. Note that this test explicitly assumes that we want integral solutions to (10); the columns removed would not be present in an optimal binary solution, but could be present in an optimal fractional solution. For the sake of readability, we postpone the detailed derivations of LP-duality based screening tests to Appendix 2.

5.3 Row Sampling

The previous section was concerned with removing columns from $\mathbf{A}$ whereas this section is concerned with removing rows. Suppose that we have a large number $\bar{m}$ of samples available, and we believe that we can learn a near-optimal interpretable classifier from a much smaller subset of $m \ll \bar{m}$ samples. We proceed to develop a certificate which shows that when m is large enough, the solution of (10) on the smaller subset achieves a near optimal solution on the full data set.

To compare the solutions of LPs defined with a different number of samples, we compare their "scaled" optimal objective values, i.e., we divide the objective value by the number of samples (which is equal to m, the number of rows in $\mathbf{A}$). Therefore, we compute and compare error rates rather than raw errors. Let $(\hat{\mathbf{w}}^m, \boldsymbol{\xi}^m)$ and $(\hat{\mathbf{w}}^{\bar{m}}, \boldsymbol{\xi}^{\bar{m}})$ be the optimal solutions for the small LP with m samples and large LP with $\bar{m}$ samples, respectively, with corresponding scaled optimal objective values f_m and $f_{\bar{m}}$. Also, matrices corresponding to the small LP are $\mathbf{A}$, $\mathbf{A}_{\mathscr{P}}$, $\mathbf{A}_{\mathscr{Z}}$ and to the large LP are $\bar{\mathbf{A}}$, $\bar{\mathbf{A}}_{\mathscr{P}}$, $\bar{\mathbf{A}}_{\mathscr{Z}}$. The first m rows of $\bar{\mathbf{A}}$ are $\mathbf{A}$ and the first p entries of $\bar{\mathbf{A}}_{\mathscr{P}}$ are $\mathbf{A}_{\mathscr{P}}$. By definition $\mathbf{A}$ is a submatrix of $\bar{\mathbf{A}}$. Therefore, $f_m \to f_{\bar{m}}$ as $m \to \bar{m}$ and we would like to bound $|f_m - f_{\bar{m}}|$ without solving the large LP.

We show how to extend the primal and the dual solutions of the small LP and obtain both a lower and an upper bound on the scaled optimal objective value of the large LP. To create a feasible primal solution for the large LP we can extend the vector $\hat{\mathbf{w}}^m$ from the small LP by computing the associated errors on the large LP: $\boldsymbol{\xi}^{\bar{m}}_{\mathscr{Z}} = \mathbf{A}_{\mathscr{Z}}\hat{\mathbf{w}}^m$ and

$$\boldsymbol{\xi}_{\mathscr{P}}^{\bar{m}} = \begin{cases} 0 & \text{if } \mathbf{A}_{\mathscr{P}}\hat{\mathbf{w}}^m \geq 1 \\ 1 & \text{otherwise.} \end{cases}$$

This pair $(\hat{\mathbf{w}}^m, \boldsymbol{\xi}^{\bar{m}})$ is feasible for the large LP and the scaled objective value provides an upper bound on $f_{\bar{m}}$. In a similar manner, the solution to the small IP can be extended to a feasible solution of the large IP, thereby giving an upper bound to the optimal IP solution; note that $f_{\bar{m}}$ gives a lower bound.

To find a lower bound on $f_{\bar{m}}$ we extend the dual solution of the small LP to give a feasible (but generally sub-optimal) dual solution of the large LP. We describe the details in Appendix 3.

The discussion so far has been in the batch setting where all training samples are available at the outset; the only goal is to reduce the computations in solving the linear program. We may also be in an online setting where we can request additional i.i.d. samples and would like to declare that we are close to a solution that will not change much with additional samples. This may be accomplished by computing expected upper and lower bounds on the objective value of the large LP as described in [13].

6 Empirical Results

In this section, we evaluate our proposed rule-learner experimentally. After discussing implementation details, and showing a small example for illustration, we then evaluate the performance of our rule-learner on a range of common machine learning datasets comparing it in terms of both accuracy and interpretability to the most popular classification methods. We then confirm the dramatic computational advantages of column-screening and row-sampling on data-sets with a large number of rows and columns.

6.1 Implementation Notes

As discussed in Sect. 3.1, continuous features are approached using indicator functions on thresholds in both directions of comparison; in particular we use 10 quantile-based thresholds per continuous feature dimension. To solve the LP (10), we use IBM CPLEX version 12.4 on a single processor of a 2.33 GHz Intel Xeon-based Linux machine. We find the optimal binary solution via branch and bound. For most of the examples here, the LP itself produces integral solutions. We set the regularization parameter $\lambda = 1/1000$ and do not optimize it in this work. In addition to our single-rule learner, we also consider rule-set learners that we have described in [35]. The set covering approach finds incorrectly classified examples after learning the previous rule, and learns the next rule on these examples only. The

Table 2 Tenfold cross-validation test error on various data sets

	1RULE	RUSC	RUB	DLIST	C5.0	CART	TRBAG	KNN	DISCR	SVM
ILPD	0.2985	0.2985	0.2796	0.3654	0.3053	0.3362	0.2950	0.3019	0.3636	0.3002
IONOS	0.0741	0.0712	0.0798	0.1994	0.0741	0.0997	0.0655	0.1368	0.1425	0.0541
LIVER	0.4609	0.4029	0.3942	0.4522	0.3652	0.3768	0.3101	0.3101	0.3768	0.3217
PARKIN	0.1744	0.1538	0.1590	0.2513	0.1641	0.1282	0.0821	0.1641	0.1641	0.1436
PIMA	0.2617	0.2539	0.2526	0.3138	0.2487	0.2891	0.2305	0.2969	0.2370	0.2344
SONAR	0.3702	0.3137	0.3413	0.3846	0.2500	0.2837	0.1490	0.2260	0.2452	0.1442
TRANS	0.2406	0.2406	0.2420	0.3543	0.2166	0.2701	0.2540	0.2286	0.3369	0.2353
WDBC	0.0703	0.0562	0.0562	0.0967	0.0650	0.0808	0.0422	0.0685	0.0404	0.0228

boosting approach creates a classifier that is a linear combination of our single-rule learners, by emphasizing samples that were incorrectly classified in the previous round. We do not attempt to optimize the number of rounds of set-covering or boosting, but leave it at $T = 5$ having interpretability in mind.

6.2 Illustrative Example

We illustrate the types of sparse interpretable rules that are obtained using the proposed rule learner on the Iris data set. We consider the binary problem of classifying iris versicolor from the other two species, setosa and virginica. Of the four features, sepal length, sepal width, petal length, and petal width, the rule that is learned involves only two features and three Boolean expressions:

- petal length ≤ 5.350 cm; AND
- petal width ≤ 1.700 cm; AND
- petal width > 0.875 cm.

6.3 Classification Performance Comparisons

As an empirical study, we consider several interpretable classifiers: the proposed Boolean compressed sensing-based single rule learner (1Rule), the set covering approach to extend the proposed rule learner (RuSC), the boosting approach to extend the proposed rule learner (RuB), the decision lists algorithm in SPSS (DList), the C5.0 Release 2.06 algorithm with rule set option in SPSS (C5.0), and the classification and regression trees algorithm in Matlab's classregtree function (CART).[6]

[6]We use IBM SPSS Modeler 14.1 and Matlab R2009a with default settings.

Table 3 Tenfold average number of conjunctive clauses in rule set

	1RULE	RUSC	RUB	DLIST	C5.0
ILPD	1.0	1.2	5.0	3.7	11.7
IONOS	1.0	4.1	5.0	3.7	8.4
LIVER	1.0	3.5	5.0	1.1	15.3
PARKIN	1.0	3.1	5.0	1.2	7.3
PIMA	1.0	2.3	5.0	5.0	12.0
SONAR	1.0	3.9	5.0	1.0	10.4
TRANS	1.0	1.2	5.0	2.3	4.3
WDBC	1.0	4.1	5.0	3.2	7.4

We also consider several classifiers that are not interpretable: the random forests classifier in Matlab's TreeBagger class (TrBag), the *k*-nearest neighbor algorithm in SPSS (kNN), discriminant analysis of the Matlab function classify, and SVMs with radial basis function kernel in SPSS (SVM).

The data sets to which we apply these classification algorithms come from the UCI repository [22]. They are all binary classification data sets with real-valued features. (We have not considered data sets with categorical-valued features in this study to allow comparisons to a broader set of classifiers; in fact, classification of categorical-valued features is a setting in which rule-based approaches excel.) The specific data sets are: Indian liver patient dataset (ILPD), Ionosphere (Ionos), BUPA liver disorders (Liver), Parkinsons (Parkin), Pima Indian diabetes (Pima), connectionist bench sonar (Sonar), blood transfusion service center (Trans), and breast cancer Wisconsin diagnostic (WDBC).

Table 2 gives tenfold cross-validation test errors for the various classifiers. Table 3 gives the average number of rules across the tenfolds needed by the different rule-based classifiers to achieve those error rates.

It can be noted that our rule sets have better accuracy than decision lists on all data sets and our single rule has better accuracy than decision lists in all but one instance. On about half of the data sets, our set covering rule set has fewer rules than decision lists. Taking the number of rules as an indication of interpretability, we see that our set covering rule set has about the same level of interpretability as decision lists but with better classification accuracy. (We did not optimize the number of rules in boosting.) Even our single rule, which is very interpretable, typically has better accuracy than decision lists with more rules.

Compared to the C5.0 rule set, our proposed rule sets are much more interpretable because they have many fewer rules on average across the data sets considered. The accuracy of C5.0 and our rule sets is on par, as each approach has better accuracy on half of the data sets. The best performing algorithms in terms of accuracy are SVMs and random forests, but we see generally quite competitive accuracy with the advantage of interpretability by the proposed approach. On the ILPD data set, our boosting approach has the best accuracy among all ten classifiers considered.

6.4 Examples of Learned Rules

We now illustrate small rules learned with our approach on the 'WDBC' Breast cancer classification dataset. First we use the LP relaxation in (10) with $\lambda = 2.0$ to find a compact AND-rule. The resulting rule has 5 active clauses, and the resulting train error-rate is 0.065. The clauses are as follows:

- mean texture > 14.13 and
- mean concave points > 0.05 and
- standard error radius > 0.19 and
- standard error area > 13.48 and
- worst area > 475.18.

Next we consider an M-of-N rule for the same data-set, and use the same number of clauses. We note that the LP relaxation in fact produces fractional solutions for this dataset, but only a small proportion of variables is fractional, and the problem can be solved reasonably efficiently using IBM CPLEX in about 8 s. The resulting error-rate is 0.028, an improvement over the AND-rule (N-of-N rule) with the same number of clauses. The rule finds 5 clauses, of which at least 3 need to be active for the classifier to return a positive label:

- mean texture > 16.58
- worst perimeter > 120.26
- worst area > 724.48
- worst smoothness > 0.12
- worst concave points > 0.18.

6.5 Screening Results

In this section, we examine the empirical performance of the screening tests on several data sets from the UCI Machine Learning Repository [22] which have many continuous-valued features: ionosphere ($m = 351$), banknote authentication ($m = 1372$), MAGIC gamma telescope ($m = 19{,}020$), and gas sensor array drift ($m = 13{,}910$). The first three are naturally binary classification problems, whereas the fourth is originally a six class problem that we have converted into a binary problem. The classification problem in ionosphere is to classify whether there is structure in the ionosphere based on radar data from Goose Bay, Labrador, in banknote is to classify genuine and forged banknotes from statistics of wavelet-transformed images, in MAGIC is a signal detection task from measurements of the Cherenkov gamma telescope, and in gas is to classify pure gaseous substances using measurements from different chemical sensors. In gas, we map the class labels ammonia, acetaldehyde and acetone to the binary class 0 and ethylene, ethanol and toluene to the binary class 1.

Table 4 Screening results for all pairs column comparison and enhanced primal and dual heuristics

Data set	Features	Thresholds	Columns	Columns screened by simple tests	Columns screened by duality test	Total columns screened	Fraction columns screened
Ionosphere	33	10	642	596	638	638	0.994
		20	1282	1210	1271	1271	0.991
		50	3202	3102	2984	3133	0.978
		100	6402	6269	5968	6310	0.986
Banknote	4	10	80	40	71	71	0.888
		20	160	92	142	142	0.888
		50	400	259	350	355	0.888
		100	800	548	700	712	0.890
MAGIC	10	10	200	188	183	188	0.940
		20	400	375	369	377	0.943
		50	1000	945	916	945	0.945
		100	2000	1892	1799	1892	0.946
Gas	128	10	2560	1875	2242	2256	0.881
		20	5120	3943	4684	4758	0.929
		50	12,800	10,235	11,378	11,824	0.924
		100	25,600	20,935	22,814	23,893	0.933

Table 4 gives the results with the enhanced version of the screening tests described in Sect. 5 that compares all pairs of columns in the pairwise test. The tables shows results for the four data sets with four different numbers of thresholds per feature dimension. We construct the $a_j(\mathbf{x})$ functions by quantile-based thresholds, and consider both directions of Boolean functions, e.g., 'weight ≤ 100' as well as 'weight > 100.' The results show the number of columns screened by the simple tests alone, the number of columns screened by the duality-based test alone, and their union in total columns screened. The tests may also be run sequentially, but for brevity we do not discuss this here.

The first thing to note in the tables is that our screening tests dramatically reduce the number of columns in the LP, ranging from removing 90% to over 99% of columns of the matrix $\mathbf{A}$. The fraction of columns screened is fairly stable across the number of thresholds within a data set, but tends to slightly improve with more thresholds. The simple tests and duality-based tests tend to have a good deal of overlap, but there is no pattern with one being a superset of the other.

The implications for running time are presented in Table 5, where we focus on the largest data set, gas. The first column shows the full LP without any screening. We compare that to the total time for screening and solving the reduced LP for the basic and enhanced screening tests. We can see that screening dramatically reduces the total solution time for the LP. Enhanced screening, while requiring more computation, does compensate the LP time and significantly reduces the total running time. With 100 thresholds we solve a very large binary integer problem with 25,600 variables to optimality in under 15 s.

Table 5 Gas data set running times in seconds for screening, solving the LP, and the total of the two: (a) basic tests, and (b) enhanced tests

Thr.	Full LP	(a) Scr.	(a) LP	(a) Tot.	(b) Scr.	(b) LP	(b) Tot.
10	18.58	0.34	2.47	2.81	0.74	1.38	2.12
20	39.52	0.73	3.96	4.69	1.53	1.29	2.82
50	103.46	2.01	12.12	14.13	4.03	3.56	7.59
100	215.57	4.28	24.30	28.58	8.86	5.90	14.76

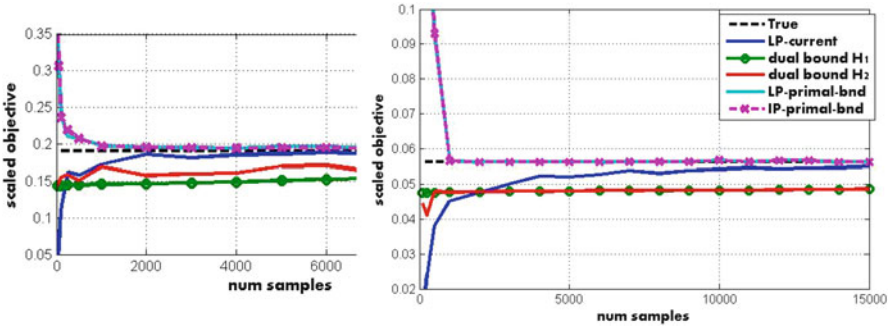

Fig. 2 Illustration of upper and lower bounds on the rule-learning LP and IP objective values for the UCI Adult (**a**) and Census Income (**b**) classification datasets. We obtain tight bounds using only a small fraction of the data.

6.6 Row Sampling Results

In this section, we apply our row-sampling bounds from Sect. 5.3 to two large-scale binary classification tasks from the UCI Machine Learning Repository [22] with a large number of rows. We consider the "Adult" and the "Census Income" datasets, which come with 32,560 and 199,522 training samples respectively. After converting categorical and continuous features to binary (using 10 thresholds) the 101 features in "Adult" dataset and the 354 features in the "Census income" dataset produce 310 and 812 columns in the corresponding **A**-matrix representations.

The results for both datasets are illustrated in Fig. 2. We plot our various bounds as a function of m: we show the objective value of the full LP (constant dashed line), and of the small LP, the upper bounds on both the LP and IP solutions for the full dataset, and the two dual bounds. We can see that the objective value of the small LP and both the LP and IP upper bounds quickly approach the objective value of the full LP (after about 2000 samples out of 32,560 total for the Adult dataset, and after 5000 samples out of 199,522 total for the bigger "Census income"). The dual bounds improve with time, albeit slower than the upper bounds. The second dual extension approach provides a much tighter lower bound for the "Adult" dataset in plot (a), but only a very modest gain for "Census Income" in plot (b). Remarkably,

for both UCI examples the LP and IP solutions for the small LP are either the same or very close, allowing quick integral solution via branch and bound. The same holds for the LP and IP upper bounds.

7 Conclusion

In this chapter, we have developed a new approach for learning decision rules based on compressed sensing ideas. The approach leads to a powerful rule-based classification system whose outputs are easy for human users to trust and draw insight from. In contrast to typical rule learners, the proposed approach is not heuristic. We prove theoretical results showing that exact rule recovery is possible through a convex relaxation of the combinatorial optimization problem under certain conditions. We also extend the framework to a more general classification problem of learning checklists and scorecards, using M-of-N rule formulation. We extend the LP formulation, and the theoretical results using ideas from Threshold Group Testing.

For large scale classification problems, we have developed novel screening tests and row sampling approaches. One class of screening tests is based on counting false alarm and missed detection errors whereas the other class is based on duality theory. In contrast to Lasso screening, which makes use of strong duality, the proposed tests consider the integer nature of the Boolean compressed sensing problem to check if the dual value is less than or equal to the optimal integer value. We developed LP duality-based techniques to guarantee a near-optimal solution after training the classifier only on a small subset of the available samples in both batch and online settings.

In our empirical evaluation we have shown that the proposed algorithm is practical and leads to a classifier that has a better trade-off of accuracy with interpretability than existing classification approaches. It produces better accuracy than existing interpretable classifiers, and much better interpretability than the powerful black-box classifiers such as SVMs and random forests while typically paying only minor cost in accuracy. Furthermore, our experimental evaluation confirms the significant gains in computational complexity of the proposed screening and row-sampling approaches.

Appendix 1: Dual Linear Program

We now derive the dual LP, which we use in Sect. 5. We start off by giving a reformulation of the LP in (10), i.e., we consider an LP with the same set of optimal solutions as the one in (10). First note that the upper bounds of 1 on the variables ξ_i are redundant. Let $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ be a feasible solution of (10) without the upper bound constraints such that $\bar{\xi}_i > 1$ for some $i \in \mathscr{P}$. Reducing $\bar{\xi}_i$ to 1 yields a

feasible solution (as $\mathbf{a}_i\bar{\mathbf{w}} + \bar{\xi}_i \geq 1$—the only inequality ξ_i participates in besides the bound constraints—is still satisfied). The new feasible solution has lower objective function value than before, as ξ_i has a positive coefficient in the objective function (which is to be minimized). One can similarly argue that in every optimal solution of (10) without the upper bound constraints, we have $w_j \leq 1$ (for $j = 1, \ldots, n$). Finally, observe that we can substitute ξ_i for $i \in \mathscr{Z}$ in the objective function by $\mathbf{a}_i\mathbf{w}$ because of the constraints $\mathbf{a}_i\mathbf{w} = \xi_i$ for $i \in \mathscr{Z}$. We thus get the following LP equivalent to (10):

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} \left(\lambda + \|\mathbf{a}^j_{\mathscr{Z}}\|_1\right) w_j + \sum_{i=1}^{p} \xi_i \\ \text{s.t.} \quad & 0 \leq w_j,\ j = 1, \ldots, n \\ & 0 \leq \xi_i,\ i = 1, \ldots, p \\ & \mathbf{A}_{\mathscr{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathscr{P}} \geq \mathbf{1}. \end{aligned} \tag{19}$$

The optimal solutions and optimal objective values are the same as in (10).

Writing $\mathbf{A}_{\mathscr{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathscr{P}}$ as $\mathbf{A}_{\mathscr{P}}\mathbf{w} + \mathbf{I}\boldsymbol{\xi}_{\mathscr{P}}$, where $\mathbf{I}$ is the $p \times p$ identity matrix, $||\mathbf{a}^j_{\mathscr{Z}}||_1$ as $\mathbf{1}^T\mathbf{a}^j_{\mathscr{Z}}$, and letting $\boldsymbol{\mu}$ be a row vector of p dual variables, one can see that the dual is:

$$\begin{aligned} \max \quad & \sum_{i=1}^{p} \mu_i \\ \text{s.t.} \quad & 0 \leq \mu_i \leq 1,\ i = 1, \ldots, p \\ & \boldsymbol{\mu}^T\mathbf{A}_{\mathscr{P}} \leq \lambda\mathbf{1}_n + \mathbf{1}^T\mathbf{A}_{\mathscr{Z}}. \end{aligned} \tag{20}$$

Suppose $\bar{\boldsymbol{\mu}}$ is a feasible solution to (20). Then clearly $\sum_{i=1}^{p} \bar{\mu}_i$ yields a lower bound on the optimal solution value of (19).

Appendix 2: Derivation of Screening Tests

Let $\mathscr{S}(j)$ stand for the support of $\mathbf{a}^j_{\mathscr{P}}$. Furthermore, let $\mathscr{N}(j)$ stand for the support of $\mathbf{1} - \mathbf{a}^j_{\mathscr{P}}$, i.e, it is the set of indices from $\mathscr{P}$ such that the corresponding components of $\mathbf{a}^j_{\mathscr{P}}$ are zero.

Now consider the situation where we fix w_1 (say) to 1. Let $\mathbf{A}'$ stand for the submatrix of $\mathbf{A}$ consisting of the last $n-1$ columns. Let $\mathbf{w}'$ stand for the vector of variables $w_2, \ldots, w_n$. Then the constraints $\mathbf{A}_{\mathscr{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathscr{P}} \geq \mathbf{1}$ in (19) become $\mathbf{A}'_{\mathscr{P}}\mathbf{w}' + \boldsymbol{\xi}_{\mathscr{P}} \geq \mathbf{1} - \mathbf{a}^1_{\mathscr{P}}$. Therefore, for all $i \in \mathscr{S}(1)$, the corresponding constraint is now $(\mathbf{A}'_{\mathscr{P}})_i\mathbf{w}' + \xi_i \geq 0$ which is a redundant constraint as $\mathbf{A}'_{\mathscr{P}} \geq 0$ and $\mathbf{w}', \xi_i \geq 0$.

The only remaining non-redundant constraints correspond to the indices in $\mathcal{N}(1)$. Then the value of (19) with w_1 set to 1 becomes

$$\begin{aligned} \left(\lambda + \|\mathbf{a}^1_{\mathscr{Z}}\|_1\right) + \min \quad & \sum_{j=2}^{n} \left(\lambda + \|\mathbf{a}^j_{\mathscr{Z}}\|_1\right) w_j + \sum_{i \in \mathcal{N}(1)} \xi_i \\ \text{s.t.} \quad & 0 \le w_j,\ j = 2, \ldots, n \\ & 0 \le \xi_i,\ i \in \mathcal{N}(1) \\ & \mathbf{A}'_{\mathcal{N}(1)} \mathbf{w}' + \boldsymbol{\xi}_{\mathcal{N}(1)} \ge \mathbf{1}. \end{aligned} \tag{21}$$

This LP clearly has the same form as the LP in (19). Furthermore, given any feasible solution $\bar{\boldsymbol{\mu}}$ of (20), $\bar{\boldsymbol{\mu}}_{\mathcal{N}(1)}$ defines a feasible dual solution of (21) as

$$\begin{aligned} & \bar{\boldsymbol{\mu}}^T \mathbf{A}_{\mathscr{P}} \le \lambda \mathbf{1}_n + \mathbf{1}^T \mathbf{A}_{\mathscr{Z}} \\ \Rightarrow\ & \bar{\boldsymbol{\mu}}^T_{\mathscr{S}(1)} \mathbf{A}'_{\mathscr{S}(1)} + \bar{\boldsymbol{\mu}}^T_{\mathcal{N}(1)} \mathbf{A}'_{\mathcal{N}(1)} \le \lambda \mathbf{1}_{n-1} + \mathbf{1}^T \mathbf{A}'_{\mathscr{Z}} \\ \Rightarrow\ & \bar{\boldsymbol{\mu}}^T_{\mathcal{N}(1)} \mathbf{A}'_{\mathcal{N}(1)} \le \lambda \mathbf{1}_{n-1} + \mathbf{1}^T \mathbf{A}'_{\mathscr{Z}}. \end{aligned}$$

Therefore $\sum_{i \in \mathcal{N}(n)} \bar{\mu}_i$ is a lower bound on the optimal solution value of the LP in (21) and therefore

$$\lambda + ||\mathbf{a}^1_{\mathscr{Z}}||_1 + \sum_{i \in \mathcal{N}(1)} \bar{\mu}_i \tag{22}$$

is a lower bound on the optimal solution value of (19) with w_1 set to 1. In particular, if $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ is a feasible *integral* solution to (19) with objective function value $\lambda(\sum_{i=1}^{n} \bar{w}_i) + \sum_{i=1}^{p} \bar{\xi}_i$, and if (22) is greater than this value, than no optimal integral solution of (19) can have $w_1 = 1$. Therefore $w_1 = 0$ in any optimal solution, and we can simply drop the column corresponding to w_1 from the LP.

In order to use the screening results in this section we need to obtain a feasible primal and a feasible dual solution. Some useful heuristics to obtain such a pair are described in [12].

Appendix 3: Extending the Dual Solution for Row-Sampling

Suppose that $\hat{\boldsymbol{\mu}}^p$ is the optimal dual solution to the small LP in Sect. 5.3. Note that the number of variables in the dual for the large LP increases from p to $\bar{p}$ and the bound on the second constraint grows from $\lambda \mathbf{1}_n + \mathbf{1}^T \mathbf{A}_{\mathscr{Z}}$ to $\lambda \mathbf{1}_n + \mathbf{1}^T \bar{\mathbf{A}}_{\mathscr{Z}}$.

We use a greedy heuristic to extend $\hat{\boldsymbol{\mu}}^p$ to a feasible dual solution $\bar{\boldsymbol{\mu}}_{\bar{p}}$ of the large LP. We set $\bar{\mu}_j = \hat{\mu}_j$ for $j = 1, .., p$. We extend the remaining entries $\bar{\mu}_j$ for $j = (p+1), .., \bar{p}$ by setting a subset of its entries to 1 while satisfying the dual

feasibility constraint. In other words the extension of $\bar{\boldsymbol{\mu}}$ corresponds to a subset $\mathscr{R}$ of the row indices $\{p+1,\ldots,\bar{p}\}$ of $\bar{\mathbf{A}}_{\mathscr{P}}$ such that $\hat{\boldsymbol{\mu}}_p^T\mathbf{A}_{\mathscr{P}} + \sum_{i\in\mathscr{R}}(\bar{\mathbf{A}}_{\mathscr{P}})_i \leq \mathbf{1}^T\bar{\mathbf{A}}_{\mathscr{Z}}$. Having $\bar{\boldsymbol{\mu}}^T\mathbf{A}_{\mathscr{P}} \leq \mathbf{1}^T\mathbf{A}_{\mathscr{Z}}$ with $\bar{\boldsymbol{\mu}}$ extended by a binary vector implies that $\bar{\boldsymbol{\mu}}$ is feasible for (20). We initialize $\mathscr{R}$ to $\emptyset$ and then simply go through the unseen rows of $\bar{\mathbf{A}}_{\mathscr{P}}$ in some fixed order (increasing from $p+1$ to $\bar{p}$), and for a row k, if

$$\hat{\boldsymbol{\mu}}_p^T\mathbf{A}_{\mathscr{P}} + \sum_{i\in\mathscr{R}}(\bar{\mathbf{A}}_{\mathscr{P}})_i + (\bar{\mathbf{A}}_{\mathscr{P}})_k \leq \mathbf{1}^T\bar{\mathbf{A}}_{\mathscr{Z}},$$

we set $\mathscr{R}$ to $\mathscr{R}\cup\{k\}$. The heuristic (we call it H1) needs only a single pass through the matrix $\bar{\mathbf{A}}_{\mathscr{P}}$, and is thus very fast.

This heuristic, however, does not use the optimal solution $\hat{\mathbf{w}}^m$ in any way. Suppose $\hat{\mathbf{w}}^m$ were an optimal solution of the large LP. Then complementary slackness would imply that if $(\bar{\mathbf{A}}_{\mathscr{P}})_i\hat{\mathbf{w}}^m > 1$, then in any optimal dual solution $\boldsymbol{\mu}$, $\mu_i = 0$. Thus, assuming $\hat{\mathbf{w}}^m$ is close to an optimal solution for the large LP, we modify heuristic H1 to obtain heuristic H2, by simply setting $\bar{\mu}_i = 0$ whenever $(\bar{\mathbf{A}}_{\mathscr{P}})_i\hat{\mathbf{w}}^m > 1$, while keeping the remaining steps unchanged.

Acknowledgements The authors thank Vijay S. Iyengar, Benjamin Letham, Cynthia Rudin, Viswanath Nagarajan, Karthikeyan Natesan Ramamurthy, Mikhail Malyutov and Venkatesh Saligrama for valuable discussions.

References

1. Adams, S.T., Leveson, S.H.: Clinical prediction rules. Br. Med. J. **344**, d8312 (2012)
2. Atia, G.K., Saligrama, V.: Boolean compressed sensing and noisy group testing. IEEE Trans. Inf. Theory **58**(3), 1880–1901 (2012)
3. Bertsimas, D., Chang, A., Rudin, C.: An integer optimization approach to associative classification. In: Advances in Neural Information Processing Systems 25, pp. 269–277 (2012)
4. Blum, A., Kalai, A., Langford, J.: Beating the hold-out: bounds for k-fold and progressive cross-validation. In: Proceedings of the Conference on Computational Learning Theory, Santa Cruz, CA, pp. 203–208 (1999)
5. Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I.: An implementation of logical analysis of data. IEEE Trans. Knowl. Data Eng. **12**(2), 292–306 (2000)
6. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
7. Chen, H.B., Fu, H.L.: Nonadaptive algorithms for threshold group testing. Discret. Appl. Math. **157**, 1581–1585 (2009)
8. Cheraghchi, M., Hormati, A., Karbasi, A., Vetterli, M.: Compressed sensing with probabilistic measurements: a group testing solution. In: Proceedings of the Annual Allerton Conference on Communication Control and Computing, Allerton, IL, pp. 30–35 (2009)
9. Clark, P., Niblett, T.: The CN2 induction algorithm. Mach. Learn. **3**(4), 261–283 (1989)
10. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the International Conference on Machine Learning, Tahoe City, CA, pp. 115–123 (1995)
11. Dai, L., Pelckmans, K.: An ellipsoid based, two-stage screening test for BPDN. In: Proceedings of the European Signal Processing Conference, Bucharest, Romania, pp. 654–658 (2012)

12. Dash, S., Malioutov, D.M., Varshney, K.R.: Screening for learning classification rules via Boolean compressed sensing. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Florence, Italy, pp. 3360–3364 (2014)
13. Dash, S., Malioutov, D.M., Varshney, K.R.: Learning interpretable classification rules using sequential row sampling. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Brisbane, Australia (2015)
14. Dembczyński, K., Kotłowski, W., Słowiński, R.: ENDER: a statistical framework for boosting decision rules. Data Min. Knowl. Disc. **21**(1), 52–90 (2010)
15. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. Proc. Natl. Acad. Sci. **100**(5), 2197–2202 (2003)
16. Du, D.Z., Hwang, F.K.: Pooling Designs and Nonadaptive Group Testing: Important Tools for DNA Sequencing. World Scientific, Singapore (2006)
17. Dyachkov, A.G., Rykov, V.V.: A survey of superimposed code theory. Prob. Control. Inf. **12**(4), 229–242 (1983)
18. Dyachkov, A.G., Vilenkin, P.A., Macula, A.J., Torney, D.C.: Families of finite sets in which no intersection of l sets is covered by the union of s others. J. Combin. Theory **99**, 195–218 (2002)
19. Eckstein, J., Goldberg, N.: An improved branch-and-bound method for maximum monomial agreement. INFORMS J. Comput. **24**(2), 328–341 (2012)
20. El Ghaoui, L., Viallon, V., Rabbani, T.: Safe feature elimination in sparse supervised learning. Pac. J. Optim. **8**(4), 667–698 (2012)
21. Emad, A., Milenkovic, O.: Semiquantitative group testing. IEEE Trans. Inf. Theory **60**(8), 4614–4636 (2014)
22. Frank, A., Asuncion, A.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2010)
23. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Ann. Appl. Stat. **2**(3), 916–954 (2008)
24. Fry, C.: Closing the gap between analytics and action. INFORMS Analytics Mag. **4**(6), 4–5 (2011)
25. Gage, B.F., Waterman, A.D., Shannon, W., Boechler, M., Rich, M.W., Radford, M.J.: Validation of clinical classification schemes for predicting stroke. J. Am. Med. Assoc. **258**(22), 2864–2870 (2001)
26. Gawande, A.: The Checklist Manifesto: How To Get Things Right. Metropolitan Books, New York (2009)
27. Gilbert, A.C., Iwen, M.A., Strauss, M.J.: Group testing and sparse signal recovery. In: Conference Record - Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, pp. 1059–1063 (2008)
28. Jawanpuria, P., Nath, J.S., Ramakrishnan, G.: Efficient rule ensemble learning using hierarchical kernels. In: Proceedings of the International Conference on Machine Learning, Bellevue, WA, pp. 161–168 (2011)
29. John, G.H., Langley, P.: Static versus dynamic sampling for data mining. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Portland, OR, pp. 367–370 (1996)
30. Kautz, W., Singleton, R.: Nonrandom binary superimposed codes. IEEE Trans. Inf. Theory **10**(4), 363–377 (1964)
31. Letham, B., Rudin, C., McCormick, T.H., Madigan, D.: Building interpretable classifiers with rules using Bayesian analysis. Tech. Rep. 609, Department of Statistics, University of Washington (2012)
32. Liu, J., Li, M.: Finding cancer biomarkers from mass spectrometry data by decision lists. J. Comput. Biol. **12**(7), 971–979 (2005)
33. Liu, J., Zhao, Z., Wang, J., Ye, J.: Safe screening with variational inequalities and its application to lasso. In: Proceedings of the International Conference on Machine Learning, Beijing, China, pp. 289–297 (2014)

34. Malioutov, D., Malyutov, M.: Boolean compressed sensing: LP relaxation for group testing. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, pp. 3305–3308 (2012)
35. Malioutov, D.M., Varshney, K.R.: Exact rule learning via Boolean compressed sensing. In: Proceedings of the International Conference on Machine Learning, Atlanta, GA, pp. 765–773 (2013)
36. Malioutov, D.M., Sanghavi, S.R., Willsky, A.S.: Sequential compressed sensing. IEEE J. Spec. Top. Signal Proc. **4**(2), 435–444 (2010)
37. Malyutov, M.: The separating property of random matrices. Math. Notes **23**(1), 84–91 (1978)
38. Malyutov, M.: Search for sparse active inputs: a review. In: Aydinian, H., Cicalese, F., Deppe, C. (eds.) Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede, pp. 609–647. Springer, Berlin/Germany (2013)
39. Marchand, M., Shawe-Taylor, J.: The set covering machine. J. Mach. Learn. Res. **3**, 723–746 (2002)
40. Maron, O., Moore, A.W.: Hoeffding races: accelerating model selection search for classification and function approximation. Adv. Neural Inf. Proces. Syst. **6**, 59–66 (1993)
41. Mazumdar, A.: On almost disjunct matrices for group testing. In: Proceedings of the International Symposium on Algorithms and Computation, Taipei, Taiwan, pp. 649–658 (2012)
42. Provost, F., Jensen, D., Oates, T.: Efficient progressive sampling. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, pp. 23–32 (1999)
43. Quinlan, J.R.: Simplifying decision trees. Int. J. Man Mach. Stud. **27**(3), 221–234 (1987)
44. Rivest, R.L.: Learning decision lists. Mach. Learn. **2**(3), 229–246 (1987)
45. Rückert, U., Kramer, S.: Margin-based first-order rule learning. Mach. Learn. **70**(2–3), 189–206 (2008)
46. Sejdinovic, D., Johnson, O.: Note on noisy group testing: asymptotic bounds and belief propagation reconstruction. In: Proceedings of the Annual Allerton Conference on Communication Control and Computing, Allerton, IL, pp. 998–1003 (2010)
47. Stinson, D.R., Wei, R.: Generalized cover-free families. Discret. Math. **279**, 463–477 (2004)
48. Ustun, B., Rudin, C.: Methods and models for interpretable linear classification. Available at http://arxiv.org/pdf/1405.4047 (2014)
49. Wagstaff, K.L.: Machine learning that matters. In: Proceedings of the International Conference on Machine Learning, Edinburgh, United Kingdom, pp. 529–536 (2012)
50. Wang, F., Rudin, C.: Falling rule lists. Available at http://arxiv.org/pdf/1411.5899 (2014)
51. Wang, J., Zhou, J., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. Adv. Neural Inf. Proces. Syst. **26**, 1070–1078 (2013)
52. Wang, Y., Xiang, Z.J., Ramadge, P.J.: Lasso screening with a small regularization parameter. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, pp. 3342–3346 (2013)
53. Wang, Y., Xiang, Z.J., Ramadge, P.J.: Tradeoffs in improved screening of lasso problems. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, pp. 3297–3301 (2013)
54. Wang, T., Rudin, C., Doshi, F., Liu, Y., Klampfl, E., MacNeille, P.: Bayesian or's of and's for interpretable classification with application to context aware recommender systems. Available at http://arxiv.org/abs/1504.07614 (2015)
55. Wu, H., Ramadge, P.J.: The 2-codeword screening test for lasso problems. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, pp. 3307–3311 (2013)
56. Xiang, Z.J., Ramadge, P.J.: Fast lasso screening tests based on correlations. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, pp. 2137–2140 (2012)
57. Xiang, Z.J., Xu, H., Ramadge, P.J.: Learning Sparse Representations of High Dimensional Data on Large Scale Dictionaries. Advances in Neural Information Processing Systems, vol. 24, pp. 900–908. MIT Press, Cambridge, MA (2011)