SCREENING FOR LEARNING CLASSIFICATION RULES VIA BOOLEAN COMPRESSED SENSING

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

Business Analytics and Mathematical Sciences Department, IBM Thomas J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, USA

ABSTRACT

Convex relaxations for sparse representation problems, which aim to find sparse solutions to systems of equations, have enabled a variety of exciting applications in high-dimensional settings. Yet, with dimensions large enough, even these convex formulations become prohibitively expensive. Screening methods attempt to use duality theory to dramatically reduce the size of the optimization problem through easily computable certificates that many of the variables must be zero in the optimal solution. In this paper we consider learning sparse classification rules via Boolean compressed sensing and develop screening procedures that can significantly reduce the size of the resulting linear program. Boolean compressed sensing deals with systems of Boolean equations (instead of linear equations in traditional compressed sensing); we develop screening methods specifically for this setting. We demonstrate the effectiveness of our screening rules on several real-world classification data sets.

***Index Terms*—** Linear programming duality, rule learning, screening, sparse signal approximation

1. INTRODUCTION

Boolean compressed sensing (CS) attempts to recover a sparse binary vector from a collection of binary measurements which computes disjunctions of subsets of its entries [1–4]. Viewing these measurements as matrix multiplication by a binary sensing matrix in the Boolean algebra, where disjunction and conjunction replace linear algebraic addition and multiplication, establishes a close connection to traditional CS [5]. Recent work has applied the concepts of Boolean CS to supervised classification to learn interpretable decision rules with excellent generalization via a linear programming formulation [6]. Classification rules learned using this proposed method take the form of a conjunctive clause. As an example in a management setting, a rule-based classifier for predicting the voluntary resignation of salespeople is [7]:

- Job Role = Specialty Software Sales Rep; AND
- Base Salary $\leq$ 75,168; AND
- Months Since Promoted $>$ 13; AND
- Months Since Promoted $\leq$ 30; AND
- Quota-Based Compensation = FALSE.

As an example in sports analytics, a rule-based classifier for predicting the winner of a tennis match is:

- Win more than 59% of 4 to 9 shot rallies; AND
- Win more than 78% of points when serving at 30-30 or Deuce; AND
- Serve less than 20% of serves into the body.

Sparsity comes into the rule learning formulation because the five terms in the salesforce example and the three terms in the tennis example are selected from a large dictionary of potential terms. Nonzero entries in a sparse binary vector dictate which terms are included in the decision rule, and all other potential terms, such as 'Job Role = Dealmaker' and 'Base Salary $\leq$ 77,124,' correspond to zeroes in the vector.

For tractability in rule-based classification, continuous-valued features (such as base salary, or percentage of serves into the body) are usually quantized with a small number of thresholds [8–13]. Ideally, one would like to consider all possible thresholds (up to the resolution of the training samples) as candidate terms for the classification rule, but this may require a large number of columns in the sensing matrix and create a large linear program (LP) in the formulation of [6]. However, if there were ways to certifiably know before solving the Boolean compressed sensing LP that certain terms would not appear in the solution, we could remove the corresponding columns in the sensing matrix and solve a much smaller and more tractable LP. Such certifiable removal of columns is known as *screening* in the sparse signal representation literature [14–22].

In this paper, our contribution is to investigate screening for Boolean CS, focused primarily on the rule learning application with continuous features. No previous work on screening has examined the Boolean sparse signal recovery problem [14–22]. As Boolean CS relates to nonadaptive combinatorial group testing, we also note that the screening tests we develop can be applied to the latter problem as well [23, 24].

We develop two classes of screening tests in this work. One class is simple screening rules that arise from misclassification error counting arguments. The other class is based on a feasible primal-dual pair of solutions to the LP. Taken together, the tests eliminate a very large fraction of the Boolean terms in the problem from further consideration. We show performance results on several data sets from the UCI Machine Learning Repository [25], indicating the large fraction of columns that can be safely dropped from the LP and the resulting speedup in running time to solve the LP.

2. RULE LEARNING FORMULATION

In this section, we briefly describe how to learn classification rules via Boolean CS [6] before turning to various screening tests in the sequel. We are given m labeled training samples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ where the $\mathbf{x}_i \in \mathcal{X}$ are the features and the $y_i \in \{0, 1\}$ are the Boolean labels. We would like to learn a function $\hat{y}(\cdot) : \mathcal{X} \to \{0, 1\}$ that will accurately generalize to classify unlabeled feature vectors drawn from the same distribution as the training samples. We represent individual Boolean terms, such as 'Months Since Promoted $>$ 13,' where a continuous feature is

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

tested against a set of thresholds, by functions $a_j(\cdot) : \mathcal{X} \to \{0,1\}$, $j = 1, \ldots, n$. 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 $\mathbf{y} \in \{0,1\}^m$, we have:

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

where $\mathbf{w} \in \{0,1\}^n$ is the sparse vector to be learned that indicates which terms are included in the decision rule, and $\mathbf{n}$ is noise that flips some values through the exclusive disjunction operation. The notation $\mathbf{y} = \mathbf{A} \vee \mathbf{w}$ is shorthand for:

$$y_i = \bigvee_{j=1}^{n} a_{ij} \wedge w_j, \quad i = 1, \ldots, m. \tag{2}$$

As in the standard sparse signal recovery problem, we would like to find $\mathbf{w}$ satisfying (1) while keeping $\|\mathbf{w}\|_0$ and the noise $\mathbf{n}$ small. Expressing the Boolean constraint $\mathbf{y} = \mathbf{A} \vee \mathbf{w}$ through ordinary linear inequalities, relaxing the ℓ_0 problem to the ℓ_1 problem, relaxing the binary constraint on the vector $\mathbf{w}$ to $0 \leq \mathbf{w} \leq 1$, and introducing slack variables to account for noise, the rule learning problem is captured in the following LP [6]:

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} w_j + \lambda \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & 0 \leq w_j \leq 1,\ j = 1, \ldots, n \\ & 0 \leq \xi_i \leq 1,\ i \in \mathcal{P} \\ & 0 \leq \xi_i,\ i \in \mathcal{Z} \\ & \mathbf{A}_{\mathcal{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathcal{P}} \geq \mathbf{1} \\ & \mathbf{A}_{\mathcal{Z}}\mathbf{w} = \boldsymbol{\xi}_{\mathcal{Z}}, \end{aligned} \tag{3}$$

where the regularization parameter λ trades training error and the sparsity of $\mathbf{w}$, $\mathbf{1}$ is the vector of all ones of appropriate dimension, $\mathcal{P}$ indexes the set of positive training samples, $\mathcal{Z}$ indexes the set of zero-valued training samples, $\mathbf{A}_{\mathcal{P}}$ and $\mathbf{A}_{\mathcal{Z}}$ are the corresponding rows of $\mathbf{A}$, and $\boldsymbol{\xi}_{\mathcal{P}}$ and $\boldsymbol{\xi}_{\mathcal{Z}}$ are the corresponding slack variables.

Let $p = |\mathcal{P}|$ and assume without loss of generality that $\mathbf{A}_{\mathcal{P}}$ consists of the first p rows of $\mathbf{A}$, i.e., $\mathcal{P} = \{1, \ldots, p\}$ and $\mathcal{Z} = \{p+1, \ldots, m\}$. Also, let $\mathbf{a}_i$ stand for the ith row of $\mathbf{A}$ and let $\mathbf{a}^j$ stand for the jth column of $\mathbf{A}$ for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. Furthermore, let $\mathbf{a}^j_{\mathcal{P}}$ and $\mathbf{a}^j_{\mathcal{Z}}$ consist of the components of $\mathbf{a}^j$ corresponding respectively to the index sets $\mathcal{P}$ and $\mathcal{Z}$.

The learned decision rule $\hat{y}(\mathbf{x})$, which is a conjunction of Boolean terms, is obtained from the LP solution $\hat{\mathbf{w}}$ [6]. For fractional solutions, one can use randomized rounding to recover binary solutions, or branch and bound to solve the integer program directly. We denote classification errors as follows. False alarms on the training set are those samples such that $\hat{y}(\mathbf{x}_i) = 1$ where $y_i = 0$, missed detections on the training set are the samples $\hat{y}(\mathbf{x}_i) = 0$ where $y_i = 1$. For simplicity, false alarms and missed detections are assumed to be equally costly in this work, but all our screening tests can be directly extended to the unequally weighted case.

3. SCREENING TESTS

The formulation in (3) produces a very large LP, which can be solved for moderate data sizes, but becomes challenging for large datasets and large numbers of quantization thresholds on continuous features. Our aim here 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.

3.1. Simple Screening Tests

We first observe that a positive entry in $\mathbf{a}^j_{\mathcal{Z}}$ 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 $\mathbf{a}^j_{\mathcal{P}}$. The first screening test is simply to remove columns in which $\|\mathbf{a}^j_{\mathcal{Z}}\|_0 \geq \|\mathbf{a}^j_{\mathcal{P}}\|_0$.

An additional test compares pairs of columns j and j' for different threshold values of the same continuous feature dimension of $\mathcal{X}$. 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 $\mathbf{A}_{\mathcal{Z}}$ and the difference in the number of positive entries between columns of $\mathbf{A}_{\mathcal{P}}$, we never select column j instead of column j' if $\|\mathbf{a}^{j'}_{\mathcal{P}}\|_0 - \|\mathbf{a}^j_{\mathcal{P}}\|_0 > \|\mathbf{a}^{j'}_{\mathcal{Z}}\|_0 - \|\mathbf{a}^j_{\mathcal{Z}}\|_0$ 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 $\mathcal{X}$, the same test can be conducted for any two columns j and j' across different features.

3.2. Tests Based on a Feasible Primal-Dual Pair

Now we use LP duality theory to provide further screening tests when a primal-dual feasible pair is available for the LP and mention cost-effective ways to provide such primal dual pairs. Specifically, we first reformulate (3) 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 in the optimal solution cannot be nonzero 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 from which we can base the screening. Note that this test explicitly assumes we want integral solutions to (3); the columns removed would not be present in an optimal binary solution, but could be present in an optimal fractional solution.

We start off by giving a reformulation of the LP in (3), i.e., we consider an LP with the same set of optimal solutions as the one in (3). 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 (3) without the upper bound constraints such that $\bar{\xi}_i > 1$ for some $i \in \mathcal{P}$. 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 (3) 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 $i \in \mathcal{Z}$ in the objective function by $\mathbf{a}_i\mathbf{w}$ because of the constraints $\mathbf{a}_i\mathbf{w} = \xi_i$ for $i \in \mathcal{Z}$. If we subsequently divide the objective function by λ,

we get the following equivalent LP to (3):

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} \left(\frac{1}{\lambda} + \|\mathbf{a}_{\mathcal{Z}}^{j}\|_1 \right) w_j + \sum_{i=1}^{p} \xi_i \qquad (4) \\ \text{s.t.} \quad & 0 \le w_j,\ j = 1, \ldots, n \\ & 0 \le \xi_i,\ i = 1, \ldots, p \\ & \mathbf{A}_{\mathcal{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathcal{P}} \ge \mathbf{1}. \end{aligned}$$

The optimal solutions are the same as in (3), and the optimal solution values are the same up to multiplication by λ.

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

$$\begin{aligned} \max \quad & \sum_{i=1}^{p} \mu_i \qquad (5) \\ \text{s.t.} \quad & 0 \le \mu_i \le 1,\ i = 1, \ldots, p \\ & \boldsymbol{\mu}\mathbf{A}_{\mathcal{P}} \le \frac{1}{\lambda}\mathbf{1}_n + \mathbf{1}^T\mathbf{A}_{\mathcal{Z}}. \end{aligned}$$

Suppose $\bar{\boldsymbol{\mu}}$ is a feasible solution to (5). Then clearly $\sum_{i=1}^{p} \bar{\mu}_i$ yields a lower bound on the optimal solution value of (4).

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

Now consider the situation where we fix w_1 (say) to 1. Let $\mathbf{A}'$ stand for 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 $\mathbf{A}_{\mathcal{P}}\mathbf{w} + \boldsymbol{\xi}_{\mathcal{P}} \ge \mathbf{1}$ in (4) become $\mathbf{A}'_{\mathcal{P}}\mathbf{w}' + \boldsymbol{\xi}_{\mathcal{P}} \ge \mathbf{1} - \mathbf{a}_{\mathcal{P}}^{1}$. Therefore, for all $i \in \mathcal{S}(1)$, the corresponding constraint is now $(\mathbf{A}'_{\mathcal{P}})_i\mathbf{w}' + \xi_i \ge 0$ which is a redundant constraint as $\mathbf{A}'_{\mathcal{P}} \ge 0$ and $\mathbf{w}', \xi_i \ge 0$. The only remaining nonredundant constraints correspond to the indices in $\mathcal{N}(1)$. Then the value of (4) with w_1 set to 1 becomes

$$\begin{aligned} \left(\frac{1}{\lambda} + \|\mathbf{a}_{\mathcal{Z}}^{1}\|_1 \right) + \min \quad & \sum_{j=2}^{n} \left(\frac{1}{\lambda} + \|\mathbf{a}_{\mathcal{Z}}^{j}\|_1 \right) w_j + \sum_{i=1}^{p} \xi_i \qquad (6) \\ \text{s.t.} \quad & 0 \le w_j,\ j = 2, \ldots, n \\ & 0 \le \xi_i,\ i \in \mathcal{N}(1) \\ & \mathbf{A}'_{\mathcal{N}(1)}\mathbf{w}' + \boldsymbol{\xi}_{\mathcal{N}(1)} \ge \mathbf{1}. \end{aligned}$$

This LP clearly has the same form as the LP in (4). Furthermore, given any feasible solution $\bar{\boldsymbol{\mu}}$ of (5), $\bar{\boldsymbol{\mu}}_{\mathcal{N}(1)}$ defines a feasible dual solution of (6) as

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

Therefore $\sum_{i \in \mathcal{N}(n)} \bar{\mu}_i$ is a lower bound on the optimal solution value of the LP in (6) and therefore

$$\frac{1}{\lambda} + ||\mathbf{a}_{\mathcal{Z}}^{1}||_1 + \sum_{i \in \mathcal{N}(1)} \bar{\mu}_i \qquad (7)$$

is a lower bound on the optimal solution value of (4) with w_1 set to 1. In particular, if $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ is a feasible *integral* solution to (4) with objective function value $(\sum_{i=1}^{n} \bar{w}_i)/\lambda + \sum_{i=1}^{p} \bar{\xi}_i$, and if (7) is greater than this value, than no optimal integral solution of (4) 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.

3.3. Obtaining Feasible Primal-Dual Pairs

We use a simple greedy heuristic to find a feasible solution $\bar{\boldsymbol{\mu}}$ to (5), where every nonzero component of $\bar{\boldsymbol{\mu}}$ is 1. In other words, $\bar{\boldsymbol{\mu}}$ corresponds to a subset $\mathcal{R}$ of the row indices $\{1, \ldots, p\}$ of $\mathbf{A}_{\mathcal{P}}$ such that $\sum_{i \in \mathcal{R}} (\mathbf{A}_{\mathcal{P}})_i \le \mathbf{1}^T\mathbf{A}_{\mathcal{Z}}$; after all $\bar{\boldsymbol{\mu}}\mathbf{A}_{\mathcal{P}} \le \mathbf{1}^T\mathbf{A}_{\mathcal{Z}}$ with $\bar{\boldsymbol{\mu}}$ a binary vector implies that $\bar{\boldsymbol{\mu}}$ is feasible for (5). We initialize $\mathcal{R}$ to $\emptyset$ and then simply go through the rows of $\mathbf{A}_{\mathcal{P}}$ in some fixed order (increasing from 1 to p), and for a row k, if

$$\sum_{i \in \mathcal{R}} (\mathbf{A}_{\mathcal{P}})_i + (\mathbf{A}_{\mathcal{P}})_k \le \mathbf{1}^T\mathbf{A}_{\mathcal{Z}},$$

we set $\mathcal{R}$ to $\mathcal{R} \cup \{k\}$. The heuristic needs only a single pass through the matrix $\mathbf{A}_{\mathcal{P}}$, and is thus very fast. Furthermore, the computation of the bounds in (7) for each variable w_j (using the fixed dual solution $\bar{\boldsymbol{\mu}}$) can be executed with another pass through the matrix $\mathbf{A}_{\mathcal{P}}$.

We also use a greedy heuristic to compute a feasible integral solution to (4). Let $\mathbf{e}_j$ stand for the unit vector with a one in the j component and zeroes elsewhere. Note that for any assignment of $\mathbf{w}$ to $\bar{\mathbf{w}}$, where $\bar{\mathbf{w}}$ is a binary vector, $\bar{\boldsymbol{\xi}} = \max\{\mathbf{0}, \mathbf{1} - \mathbf{A}_{\mathcal{P}}\bar{\mathbf{w}}\}$, where the maximum is taken component-wise, is the least cost assignment of values to $\boldsymbol{\xi}$ so that $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ is a feasible integral solution of (4).

We initialize $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ by setting $\bar{\mathbf{w}}$ to $\mathbf{0}$, and $\bar{\boldsymbol{\xi}}$ appropriately. The discussion in Section 3.1 suggests that it is desirable to set a variable w_j to 1 if for the corresponding column $(\|\mathbf{a}_{\mathcal{Z}}^{j}\|_0 - \|\mathbf{a}_{\mathcal{P}}^{j}\|_0)$ is small. We therefore process the w_j variables in increasing order of $(\|\mathbf{a}_{\mathcal{Z}}^{j}\|_0 - \|\mathbf{a}_{\mathcal{P}}^{j}\|_0)$ values, and set w_j to 1 (and $\bar{\mathbf{w}}$ to $\bar{\mathbf{w}} + \mathbf{e}_j$) if and only if the objective function value decreases (after updating $\bar{\boldsymbol{\xi}}$ appropriately). Finally, we compare the lower bounds in (7) with the objective function value of our computed primal integral solution $(\bar{\mathbf{w}}, \bar{\boldsymbol{\xi}})$ to delete columns.

We also consider enhanced versions of the primal and dual heuristics. In the enhanced primal heuristic, instead of making a single pass through the sorted columns and choosing columns in a greedy fashion, we choose a column to include in our solution by making a complete pass through all unchosen columns and finding the one which reduces the objective function the most when added to the current solution. To avoid taking too much time, we never add more than a fixed number (three in our experiments) columns to the solution. In the enhanced dual heuristic, we simply sort the rows of $\mathbf{A}_{\mathcal{P}}$ by increasing number of nonzeroes before applying the greedy dual heuristic.

4. EMPIRICAL FINDINGS

In this section, we examine the empirical performance of the screening tests on several data sets from the UCI Machine Learning Repository [25], which all have continuous-valued features: ionosphere ($m = 351$), banknote authentication ($m = 1372$), MAGIC gamma telescope ($m = 19020$), and gas sensor array drift ($m = 13910$). 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. We use IBM ILOG CPLEX 12.4 on a single processor of a 2.33 GHz Intel Xeon-based Linux machine to solve the LP and find the optimal binary solution via branch and bound. For most of the examples here, the LP produced integral solutions.

Table 1 and Table 2 give the results with two different variations of the screening tests described in Section 3. The first variation is less expensive computationally and only compares consecutive columns in the $\mathbf{A}$ matrix for the simple pairwise test, and uses the basic primal and dual heuristic. The second variation compares

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	637	637	0.992
		20	1282	1207	1264	1265	0.987
		50	3202	3071	2913	3119	0.974
		100	6402	6204	5870	6287	0.982
Banknote	4	10	80	36	66	67	0.838
		20	160	79	139	141	0.881
		50	400	222	350	354	0.885
		100	800	461	694	711	0.889
MAGIC	10	10	200	188	183	188	0.940
		20	400	375	369	377	0.943
		50	1000	944	901	944	0.944
		100	2000	1888	1763	1888	0.944
Gas	128	10	2560	1857	1977	2050	0.801
		20	5120	3866	3948	4257	0.831
		50	12800	9813	9046	10402	0.813
		100	25600	19827	17888	20911	0.817

Table 1. Screening results for consecutive column comparison and basic 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	12800	10235	11378	11824	0.924
		100	25600	20935	22814	23893	0.933

Table 2. Screening results for all pairs column comparison and enhanced primal and dual heuristics.

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

Table 3. 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.

all pairs of columns in the pairwise test and uses the enhanced versions of the primal and dual heuristics. The tables show 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. 'Base Salary $\leq$ 75,168' as well as 'Base Salary $>$ 75,168.' 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. 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. As expected, the fraction of columns screened by the enhanced tests shown in Table 2 is greater than or equal to the basic tests shown in Table 1; this difference is most significant in the gas data set where the basic tests screen 81% of the columns but the enhanced tests screen 93%, i.e. only 7% of the columns remain after screening. 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 3, 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 seconds.

5. CONCLUSION

In this paper, we have developed two classes of novel screening tests for Boolean CS applied to classification rule learning. One class of 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 [14–22], which makes use of strong duality, the proposed tests take advantage of the integer nature of the Boolean CS problem to check if the dual value is less than or equal to the optimal integer value. We develop both basic and enhanced versions of the tests, resulting in various options that trade screening running time and fraction of columns screened. Results on several real-world classification problems indicate the merit of the proposed method.

Ultimately our goal is to enable learning interpretable classification rules via Boolean CS for very large datasets. In addition to screening, we are investigating reparameterization of the $\mathbf{A}$ matrix to make it sparse and column generation approaches for efficiently solving the LP [26].

6. REFERENCES

[1] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, "Group testing and sparse signal recovery," in *Asilomar Conf. Signals Syst. Comp. Conf. Record*, Pacific Grove, CA, Oct. 2008, pp. 1059–1063.

[2] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1880–1901, Mar. 2012.

[3] D. Malioutov and M. Malyutov, "Boolean compressed sensing: LP relaxation for group testing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3305–3308.

[4] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, Jul. 2012, pp. 1837–1841.

[5] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[6] D. M. Malioutov and K. R. Varshney, "Exact rule learning via Boolean compressed sensing," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, Jun. 2013, pp. 765–773.

[7] K. R. Varshney, J. C. Rasmussen, A. Mojsilović, M. Singh, and J. M. DiMicco, "Interactive visual salesforce analytics," in *Proc. Int. Conf. Inf. Syst.*, Orlando, FL, Dec. 2012.

[8] R. L. Rivest, "Learning decision lists," *Mach. Learn.*, vol. 2, no. 3, pp. 229–246, Nov. 1987.

[9] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, Mar. 1989.

[10] W. W. Cohen, "Fast effective rule induction," in *Proc. Int. Conf. Mach. Learn.*, Tahoe City, CA, Jul. 1995, pp. 115–123.

[11] M. Marchand and J. Shawe-Taylor, "The set covering machine," *J. Mach. Learn. Res.*, vol. 3, pp. 723—746, Dec. 2002.

[12] U. Rückert and S. Kramer, "Margin-based first-order rule learning," *Mach. Learn.*, vol. 70, no. 2–3, pp. 189–206, Mar. 2008.

[13] K. Dembczyński, W. Kotłowski, and R. Słowiński, "ENDER: A statistical framework for boosting decision rules," *Data Min. Knowl. Disc.*, vol. 21, no. 1, pp. 52–90, Jul. 2010.

[14] L. El Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination in sparse supervised learning," *Pac. J. Optim.*, vol. 8, no. 4, pp. 667–698, Oct. 2012.

[15] Z. J. Xiang, H. Xu, and P. J. Ramadge, "Learning sparse representations of high dimensional data on large scale dictionaries," in *Adv. Neural Inf. Process. Syst. 24*. Cambridge, MA: MIT Press, 2011, pp. 900–908.

[16] Z. J. Xiang and P. J. Ramadge, "Fast lasso screening tests based on correlations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 2137–2140.

[17] L. Dai and K. Pelckmans, "An ellipsoid based, two-stage screening test for BPDN," in *Proc. Eur. Signal Process. Conf.*, Bucharest, Romania, Aug. 2012, pp. 654–658.

[18] Y. Wang, Z. J. Xiang, and P. J. Ramadge, "Tradeoffs in improved screening of lasso problems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, May 2013, pp. 3297–3301.

[19] ——, "Lasso screening with a small regularization parameter," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, May 2013, pp. 3342–3346.

[20] H. Wu and P. J. Ramadge, "The 2-codeword screening test for lasso problems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, May 2013, pp. 3307–3311.

[21] J. Liu, Z. Zhao, J. Wang, and J. Ye, "Safe screening with variational inequalities and its applicaiton [sic] to LASSO," available at http://arxiv.org/pdf/1307.7577, Jul. 2013.

[22] J. Wang, J. Zhou, P. Wonka, and J. Ye, "Lasso screening rules via dual polytope projection," in *Adv. Neur. Inf. Process. Syst. 26*, 2013, pp. 1070–1078.

[23] A. G. Dyachkov and V. V. Rykov, "A survey of superimposed code theory," *Probl. Control Inform.*, vol. 12, no. 4, pp. 229–242, 1983.

[24] D.-Z. Du and F. K. Hwang, *Pooling Designs and Nonadaptive Group Testing: Important Tools for DNA Sequencing*. Singapore: World Scientific, 2006.

[25] A. Frank and A. Asuncion, "UCI machine learning repository," available at http://archive.ics.uci.edu/ml, 2010.

[26] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, May–Jun. 1998.