MIXED-INTEGER SUPPORT VECTOR MACHINE

Wei Guan, Alexander Gray
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Sven Leyffer
Mathematics & Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

ABSTRACT
In this paper, we propose a formulation of a feature selecting support vector machine based on the L0-norm regularization. We explore a perspective relaxation of the optimization problem and solve it using mixed-integer nonlinear programming (MINLP) techniques. Given a training set of labeled data instances, we construct a max-margin classifier that minimizes the hinge loss as well as the cardinality of the weight vector of the separating hyperplane \( \| w \|_0 \), effectively performing feature selection and classification simultaneously in one optimization. We compare this relaxation with the standard SVM, recursive feature elimination (RFE), L1-norm SVM, and two approximated L0-norm SVM methods, and show promising results on real-world datasets in terms of accuracy and sparsity.

Index Terms— Support vector machine, feature selection, L0-norm regularization, mixed-integer nonlinear programming

1. INTRODUCTION
Feature selection is necessary for many classification tasks such as microarray analysis, mass spectrometry analysis, biomedical image analysis, and other modern applications. For example, one of the main goals of microarray analysis is to find biomarkers, i.e., critical genes that are disease related. The data sets from these domains are typically high dimensional, while only a few of the features may be needed or helpful for the learning task. Many of the features may be (i) noisy or irrelevant such that ignoring them would improve the generalization ability of the classifier; (ii) redundant (for example, linear combinations of other features) such that eliminating them would not drastically change the prediction performance. Other advantages of feature selection can include reduced computational cost in classifier learning, and better interpretability.

In this work, we address feature selection in the context of linear support vector machine (SVM) learning [1]. The main goal is to select out an optimal feature subset with as few features as possible while preserving or improving the discriminative ability of the SVM classifier. For a two-class classification problem on a data set \( S = \{ x_i, y_i \}_{i=1}^n \), where \( x_i \in \mathbb{R}^d \) is the feature vector of the \( i \)th data point, and \( y_i \in \{-1, +1\} \) is the corresponding label, we can formulate SVM learning into the following optimization problem:

\[
\min_w C \sum_i L\{f(x_i), y_i\} + \frac{1}{2} \sum_j w_j^2,
\]

where \( L\{f(x), y\} = |1 - y f(x)|_+ \) is the hinge loss, \( f(x) = w x - \gamma \) is the decision function, \( w \) is the weight vector, \( \gamma \) is the bias term. Tuning parameter \( C > 0 \) controls the trade-off between the goodness of data fit, as measured by the hinge loss, and the complexity of model \( f \), as measured by the \( L_2 \)-norm of the weights.

Existing approaches to feature selection for SVMs mainly fall into three categories: filter-based methods, wrapper-based methods, and embedded methods. Filter-based methods adopt feature ranking strategies disjoint from SVM training, such as t-statistics, signal-to-noise ratio, etc. This type of feature selection methods usually compute ranking score of each individual feature according to certain ranking criterion and then select out the \( k \) best features (sorted by ranking scores). Although they might be preferable because of computational efficiency and statistical robustness, individual feature ranking is far from optimal in many cases [2].

In real-world data sets, it’s very common that a feature that is useless by itself can provide a significant prediction performance improvement when combined with some others features. Therefore subset feature selection is desirable in spite of its NP-hardness [3]. Wrapper-based methods assess the relative importance of feature subsets based on their SVM hyperplane parameters or SVM performance on the training data. Representative methods of this approach include recursive feature elimination [4], etc. Current wrapper methodologies usually adopt greedy search strategies, such as forward selection and backward elimination, to avoid exhaustive brute force feature subset search. However, these methods could still be computational expensive.

Embedded methods augment the SVM formulation, and seek to learn the SVM classifier as well as the optimal feature subset simultaneously in one optimization. Significant examples of this approach include L1-norm SVM (L1-SVM) [5],
In this paper, we propose a feature-selecting support vector machine derived from the L0-norm SVM formulation. Unlike L1-norm SVM, which performs feature selection as a by-product because of the resulting sparse solution, L0-norm SVM directly minimizes on both hinge loss and cardinality of its weight vector. In the context of regression, where feature selection has been most thoroughly studied, it has been pointed out that though L1 penalization yields sparse solutions, the estimates can be biased since larger penalties are imposed on larger weight coefficients [7]. A recent comparison study of least absolute shrinkage and selection operator (LASSO) regression [8] and forward stepwise regression, which is a greedy surrogate of L0-regularization, further indicated that L1-regularization never outperforms L0-regularization by more than a constant factor, and in some cases, using an L1-norm penalty is much worse than an L0-norm penalty [9]. This comparison analysis also pointed out that “an approximation solution to the right problem can be better than the exact solution to the wrong problem” [9]. Our study follows this guideline.

However, optimizing the L0-norm SVM is a NP-hard problem [3]. Previous work in this direction includes adopting smoothed approximations of the L0-norm [5, 10, 11], using adaptive scaling parameters to control the sparsity [12, 13, 14], and exploring the convex relaxations of the cardinality constraint [15]. In this work, we propose a mixed-integer support vector machine (MISVM) formulation based on the L0-norm SVM, and solve the problem using mixed-integer nonlinear programming (MINLP) techniques. Empirical comparison of our MISVM method with the standard SVM method [1], Recursive Feature Elimination (RFE) method [4], L1-norm SVM method, FSV method [5], and Weston’s feature selection method [12], demonstrates either sparser solutions with roughly identical classification performance, or an increase in classification performance with similar or sparser representations.

In the next section, we briefly summarize the SVM learning problems, and several widely-used SVM related feature selection methods. In section 3, we describe the mixed-integer SVM problem and its convex relaxation formulated as mixed-integer quadratic problems. Section 4 presents the comparison of the methods on four data sets from the UCI repository and two recently published metabolomic mass spectrometry ovarian cancer datasets. Finally, in section 5, we conclude and provide several directions for future work.

2. RELATED WORK

In this section, we briefly describe several commonly used SVM related feature selection methods.

2.1. SVM Recursive Feature Elimination

SVM Recursive feature elimination (RFE) method [4] is one of most commonly used feature selection methods. At each iteration of the RFE algorithm, a new SVM classifier is trained based on the current remaining features.

Next, each feature is scored according to the sensitivity of the objective function with respect to the feature, which is defined as

$$dJ(k) = \frac{1}{2} \alpha^T H \alpha - \frac{1}{2} \alpha^T H(-k) \alpha,$$

where $H$ is the kernel matrix on the feature vector under the current remaining feature set, $H_{ij} = y_i y_j K(x_i, x_j)$, $H(-k)$ is the kernel matrix on the feature vector with the $k$th feature removed from the current feature set. In the case of linear SVM, $dJ(k) = \frac{1}{2} w^2_k$.

Then the feature with the lowest score is removed from the remaining feature set. The RFE process is repeated until the desired number of features remains or the best training accuracy achieved.

2.2. L1-norm SVM

Bradley and Mangasarian proposed the L1-norm SVM (L1-SVM) method [5], which solves the following optimization problem.

$$\min \| w \|_1 + \varepsilon \| \xi \|_1$$
$$\text{s.t. } Y(Xw - \gamma \epsilon_m) + \xi \geq \epsilon_m$$
$$\xi \geq 0.$$ (3)

L1-SVM method performs feature selection as a by-product of the resulting sparse solution. Because of its computational efficiency and the empirically sparse solutions, L1-SVM method have been applied to various problems in computation biology and many other domains.

Decomposing weight vector $w$ with $w = p - q$, $|w| = p + q$ and $p, q \geq 0$, Equation (3) is then equivalent to the following linear programming problem,

$$\min e_n^T (p + q) + c e_n^T \xi$$
$$\text{s.t. } Yxp - Yxq - \gamma Y \epsilon_m + \xi \geq \epsilon_m,$$
$$p, q \geq 0,$$
$$\xi \geq 0.$$ (4)

Mangasarian proposed a fast algorithm that solves the corresponding asymptotic exterior penalty problem of the dual problem through Newton’s method [16]. We use this algorithm in our empirical study.

2.3. Approximation Methods to L0-norm SVM

Current research work on the NP-hard L0-norm SVM optimization [3] mainly includes adopting smoothed approximations of the L0-norm and using adaptive scaling parameters...
to control the sparsity. We select representative method from each category for comparison study.

Feature Selection concaVe (FSV) method [5] approximates the $L_0$-norm pe-nalty, $\sum_{j=1}^{d} |w_j|^0$, with a smooth function $\sum_{j=1}^{d} (1 - \exp^{-\alpha |w_j|})$ and thus convert the $L_0$-norm SVM into the following optimization problem

$$
\min_{w, r, y, z, v} \quad (1 - \lambda) \left( c^T \frac{y}{n} + c^T \frac{z}{m} \right) + \lambda T \left( e - \exp^{-\alpha |w|} \right)
$$

s.t. $$
-Aw + e\gamma + e \leq y
$$

$$
Bw - e\gamma + e \leq z
$$

$$
y \geq 0, z \geq 0
$$

where $A, B$ represents the data matrix of the positive training data points and that of the negative training data points, respectively.

Equation (5) is then solved using Successive Linearization Algorithm (SLA), which iteratively update $(w, \gamma, y, z, v)$ through minimizing a generalization error bound, $R^2 W^2$.

$$
R^2 (\beta, \pi) = \min \beta^T \Pi \Pi^T \beta - \sum_{i=1}^{n} \beta_i x_i^T \Pi x_i
$$

s.t. $$
\beta \geq 0
$$

$$
W^2 (\alpha, \pi) = \min \frac{1}{2} \alpha^T Y \Pi \Pi^T Y \alpha - e^T \alpha
$$

s.t. $$
\alpha \geq 0
$$

where $R^2 (\beta, \pi)$ minimizes the radius of the smallest sphere, centered at the origin, that contains all the data points; $W^2 (\alpha, \pi)$ maximizes the margin distance of the learned classifier, matrix $\Pi = \text{diag} \{ \pi \}$ and $\pi \in \{0, 1\}^d$ denote the scaling variables.

The above problem is further relaxed to $\pi \in R_+^d$ in the algorithm. At $t$th iteration, the algorithm first optimizes $R^2 (\beta, \pi^{(t-1)})$ to get $\alpha^t$, and $W^2 (\alpha, \pi^{(t-1)})$ to obtain $\beta^t$. Second, it updates $\pi$ with gradient of $R^2 (\beta^t, \pi) W^2 (\alpha^t, \pi)$. Third, it sets the smallest nonzero $\pi_k$ to zero, i.e. discards the corresponding feature. The above procedure repeats until only $d$ features left. We denote this method as R2W2.

3. MIXED-INTEGER SVM

In this paper, we consider the following $L_0$-norm SVM formulation:

$$
\min_{w, r, y, z, v} \quad ||w||_0 + c \cdot \xi
$$

s.t. $$
Y (Xw - \gamma e_m) + \xi \geq e_m, \xi \geq 0
$$

where $v$, $z$, and $y$ are further relaxed to $z_j$, which satisfies $z_j \in \{0, 1\}$, and $w_j > 0$ and the perspective constraints $w_j^2 \leq z_j u_j$, where $u_j$ is the squared upper bound of the weight element $w_j$. These define a convex hull of $w_j^2 = z_j u_j$, which is the equality we want to enforce. The proposed mixed-integer SVM can then be formulated as the following mixed-integer quadratically constrained quadratic program.

$$
\min z, u, w, \gamma, \xi \quad ae^T z + \frac{1}{2} e^T u + ce^T m \xi
$$

s.t. $$
Y (Xw - \gamma e_m) + \xi \geq e_m, \xi \geq 0
$$

$$
\xi \geq 0
$$

where vector $z = [z_1, \cdots, z_n]^T$, $u = [u_1, \cdots, u_n]^T$, and constants $a, c > 0$ adjust the trade-off between the cardinality of the weight vector and the hinge loss.

The above equation tries to minimize the $L_0$-norm penalization $\sum z_j$, the $L_2$-norm upper bound $\sum u_j$, and the hinge loss $\sum \xi_i$. The first type of constraints $Y (Xw - \gamma e_m) + \xi \geq e_m, \xi \geq 0$ regulates the classification error for each training instance. The second type of constraints $w_j^2 \leq u_j z_j$ enforce that $w_j = 0$ when $z_j = 0$, and $w_j = w_j^2$ at optimal.

During optimization, the constraints $z_j \in \{0, 1\}$ are further relaxed into $z_j \in [0, 1]$. However, solving this problem directly with the existing MINLP tools such as Bonmin [18] or MINLP [19] fails. The experiments of optimizing this problem over even small datasets resulted in either infeasible states or unsatisfying solutions with all indicator variable setting to zero. We believe that the failure of the nonlinear solvers is due to a failure of constraint qualification on the conic constraints $w_j^2 \leq z_j u_j$. For example, whenever $z_j = 0$ during the tree-search or in the solution of continuous subproblems in Bonmin, the relaxation contains a constraint $w_j^2 \leq 0$, which violates Slater’s constraint qualification [20]. While it is in principle straightforward to remedy this situation by pre-processing the constraint $w_j^2 \leq 0$ and replacing it by $w_j = 0$, current nonlinear solvers do not perform this operation. The errors that we observe from the nonlinear solvers are consistent with a failure of a constraint qualification.

To remedy this adverse situation, we thus relax the conic constraints $w_j^2 \leq u_j z_j$ in (Eqn. 9) into big-M constraints $w_j \leq M z_j$, where $M$ is a fixed large number ($M$ was set...
for a given training data set and parameter value \( c \). This results in a mixed-integer quadratic problem (\( P_1 \)).

\[
\min_{z, w, \gamma, \xi} \quad a z^T z + \frac{1}{2} w^T w + ce_m^T \xi \\
\text{s.t.} \quad Y(Xw - \gamma e_m) + \xi \geq e_m; \xi \geq 0 \\
|w_j| - M\xi_j \leq 0 \\
z \in \{0, 1\}^n, \xi \geq 0
\]

(\( P_1 \))

### 4. RESULTS AND DISCUSSION

In this section, we present the comparison study of our proposed MI-SVM method, with the standard SVM method, SVM-RFE, L1-SVM, FSV and Weston’s R2W2 methods.

We used the LibSVM packages [21] and the L1SVM code from [22, 16]. We implemented RFE, FSV, and R2W2 methods in MATLAB. Due to numerical reasons, for FSV method, the elements of the optimal weight vector that have small relative magnitude, i.e. \( \frac{|w_j|}{\max(|w_i|)} < 10^{-4} \), are set to zero.

For MI-SVM method, we used CPLEX12.1 [23] to solve the mixed-integer quadratic problem, Eqn. (\( P_1 \)). Since CPLEX tool has the similar numerical issues, we can apply the same threshold rule as the FSV method. We denote this approach as MI-SVM\(^1\). Furthermore, since we also obtain the optimal indicator variable assignment after solving Eqn. (\( P_1 \)), we would apply standard SVM on the sub data that only contains the features with non-zero indicator variables, and then obtain the final weight vector. We denoted this approach as MI-SVM\(^2\).

#### 4.1. Parameter Tuning

To estimate the generalization ability of each method, we need to tune the parameter \( c \) of the standard SVM method and the RFE method, parameters \( \delta, c \) of L1-norm SVM, parameter \( \lambda \) for FSV method, and parameters \( a, c \) of the MI-SVM methods for the performance evaluation. The candidate parameter values used for the tuning experiments were

- \( c \in \{2^{-7}, \cdots, 2^{-1}, 1, 2^1, \cdots, 2^7\} \),
- \( \delta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\} \),
- \( \lambda \in \{0.05, 0.1, 0.15, \cdots, 0.85, 0.9, 0.05\} \),
- \( a \in \{2^{-3}, \cdots, 2^{-1}, 1, 2^1, \cdots, 2^3\} \).

#### 4.2. Simulation Study

In this experiment, we compare the performance of MI-SVM, L1-SVM methods with the NP-hard L0-SVM optimization.

We use brute force search for the L0-SVM optimization. For a given training data set and parameter value \( c \), we first learned the optimal weights under the standard L2-SVM optimization for each possible feature subset of the training data set; next, we compute the objective value following Equation 10 under the given parameter value \( c \); then, we select out the feature subset with the minimal objective value; finally, we compute the classification accuracy of the SVM model on the corresponding training dataset and testing dataset with the optimal feature subset.

\[
\min c \cdot ||w||_0 + 0.5 \cdot ||w||_2 + \sum_i L(f(x_i), y_i)
\]

Since brute force search is used, we can only conduct the investigation on small data sets. For example, if feature dimension is 16, then we will need to explore 65535 feature subsets for the L0-SVM optimization. We generate the synthetic data sets by modifying the example in [12]. There are 16 features in the data set, only the first two features are relevant. With a probability of 0.7, the first feature \( f_1 \) was drawn from \( y N(3, 1) \) and the second feature \( f_2 \) was drawn from \( N(0, 1) \); otherwise, the first feature was drawn from \( N(0, 1) \) and the second feature as \( y N(3, 1) \). The remaining features \( f_j, j = 3, \cdots, 16 \) are noise and independently generated from \( N(0, 20) \). The probability of label \( y = 1 \) or \(-1\) is equal.

For each method, we repeat the following evaluation procedure 10 times. First, we generate a training data set of size 100, and a testing data set of size 100. Then we evaluate the performance of each method on the training/testing datasets. The final classification and feature selection performance of each method is computed as the average testing error and the average number of selected features over the 10 runs.

We employ a grid search procedure for choosing the right parameters for each method. For each method under each parameter setting, we perform the above evaluation procedure, and the score for this parameter setting is the averaged training error rate over the 10 runs. Then we select the parameter setting with the best score (ties are broken by choosing the one with sparser solutions).

<table>
<thead>
<tr>
<th>Sparisity (Feature #)</th>
<th>L1-SVM</th>
<th>MI-SVM$^1$</th>
<th>L0-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>14</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

As we can see from Table 1, the brute-force L0-SVM optimization method achieved the best performance in both classification and feature selection. Our MI-SVM method were able to discard 2 3 noisy features for each data set. In contrast, the L1-SVM method couldn’t find sparse representations for these synthetic data sets. In addition, MI-SVM method obtained better classification accuracy than L1-SVM method. Overall, the results indicates the advantage of L0-SVM optimization or even its approximation methods (such as MI-SVM method) over the L1-SVM optimization for feature selection/classification task.
4.3. Empirical Study on Real-World Data Sets

In this experiment, we present the comparison study of our proposed MI-SVM method, with the standard SVM method, and other commonly-used sparse SVM methods on six real-world data sets.

4.3.1. Data Sets

We include four widely-used real world biomedical data sets from the UCI repository [24] in our experiments.

- Ionosphere data set consists of 351 instances with 34 features. There are 225 radar returns termed “good” or showing some type of structure in the ionosphere, and 126 radar returns termed “bad”; their signals pass through the ionosphere.

- Wisconsin prognostic breast cancer (wpbc) data set consists of 198 instances with 32 numerical features representing follow-up data of the patients. Two of its variants are used.
  - The first data set (denoted as wpbc24) includes 28 patients who had a cancer recurrence in less than 24 months and 127 patients who didn’t have a cancer recurrence in less than 24 months.
  - The second variant (denoted as wpbc60) contains 41 patients with a cancer recurrence in less than 60 months, and 69 patients which cancer had not recurred in less than 60 months.

- SPECTF heart data set: the training dataset consists of 80 instances with 44 features (40 instances labeled with “1” and “0”, respectively); the testing dataset consists of 187 instances with 172 instances labeled with “1” and 25 labeled with “0”.

- The OvarianCancer data set [25] consists of metabolomic profiles of 37 ovarian cancer patients and 35 benign controls. Each metabolomic profile contains intensity values of the same 592 features extracted from the liquid chromatography/time-of-flight mass spectra of the patient serum.

- The DART data set [26] consists of metabolomic profiles of 44 women diagnosed with serous papillary ovarian cancer (stages I-IV) and 50 healthy women or women with benign conditions. The metabolomic profiles are obtained using direct analysis in real-time mass spectrometry technique. Each metabolomic profile contains the intensity values of 20,000 features that are uniformly resampled over m/z range of 60 and 990 from the normalized raw mass spectra.

We also use two recently published metabolomic mass spectrometry cancer data sets in this study.

- The OvarianCancer data set [25] consists of metabolomic profiles of 37 ovarian cancer patients and 35 benign controls. Each metabolomic profile contains intensity values of the same 592 features extracted from the liquid chromatography/time-of-flight mass spectra of the patient serum.

- The DART data set [26] consists of metabolomic profiles of 44 women diagnosed with serous papillary ovarian cancer (stages I-IV) and 50 healthy women or women with benign conditions. The metabolomic profiles are obtained using direct analysis in real-time mass spectrometry technique. Each metabolomic profile contains the intensity values of 20,000 features that are uniformly resampled over m/z range of 60 and 990 from the normalized raw mass spectra.

To reduce the curse of dimensionality, for each data set, we rank the features according to the absolute value of the t-statistics, and then select the 50 features with the smallest absolute value of t-statistics for our study.

4.3.2. Parameter Tuning

We estimate the generalization ability of each method via 10-fold cross validation (10-fold CV), except for SPECTF as its training and testing split are given. We employ the following tuning procedure on each data set: for each parameter setting, we perform a 10-fold CV, and the score for this parameter setting is the averaged training accuracy over cross-validation; while for SPECTF data set, we use the training accuracy as its score. Then we select the parameter setting with the best score (ties are broken by choosing the sparser solutions).

4.3.3. Result Analysis

Table 2 summarizes the feature selection performance, measured by sparsity, that is the number of features selected by each method, while Table 3 describes the classification performance, measured by testing accuracy of each classifier. For the first five data sets, we give the average sparsity and testing accuracy as well as their standard deviations over the 10-fold cross validation. Overall, the experiment results show that MI-SVM methods are able to learn sparser representations with roughly identical or increased classification performance. And MI-SVM\(^2\), the approach of re-learning the standard SVM classifier on the selected feature subset (features with non-zeros indicator variables) had a higher prediction performance than MI-SVM\(^1\), the thresholding approach.

MI-SVM\(^2\) approach outperform the other method with 80.4% testing accuracy and 17.6 sparsity averaged over the six data sets. MI-SVM\(^1\), RFE, R2W2 methods had the second best testing accuracy (77.8% on average) with MI-SVM\(^1\) having the smallest average sparsity (20.8). While L1-norm SVM had the worst evaluation performance (74.3% averaged testing accuracy) with the an average sparsity of 23.38, and FSV method had the second worst performance (76.5%). In datasets such as wpbc24, SPECTF, Ovarian, and DART, the testing accuracy increase significantly when using MI-SVM, which indicates that some of the features in these datasets may be noise or irrelevant. In other data sets like Ionosphere, wpbc60, the accuracy remains roughly the same while sparsity increases, which suggests that these data sets may contain redundant features. In both cases, MI-SVM is able to learn a lower dimensional representation with at least comparable classification performance. The comparison analysis indicates that our MI-SVM method realizes a suitable trade off between the classification errors and the number of selected features. Moreover, the sparse representations learned with the MI-SVM method are generally more predictive than those produced by the L1-norm SVM or other L0-norm SVM approximations such as the FSV method and the R2W2 method.
Table 2. Feature Selection Performance (Sparsity)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM</th>
<th>RFE</th>
<th>L1-SVM</th>
<th>FSV</th>
<th>R2W2</th>
<th>MI-SVM*</th>
<th>MI-SVM**</th>
</tr>
</thead>
<tbody>
<tr>
<td>OvarianCancer</td>
<td>50</td>
<td>40.5</td>
<td>24.1</td>
<td>16.5</td>
<td>49.4</td>
<td>16.5</td>
<td>12.6</td>
</tr>
<tr>
<td>DART</td>
<td>50</td>
<td>13.5</td>
<td>8.1</td>
<td>4.8</td>
<td>12.9</td>
<td>10.3</td>
<td>10</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>33</td>
<td>22.7</td>
<td>29.8</td>
<td>30.3</td>
<td>30.1</td>
<td>30.8</td>
<td>30.7</td>
</tr>
<tr>
<td>wpbc24</td>
<td>32</td>
<td>27.7</td>
<td>24.3</td>
<td>5.4</td>
<td>27</td>
<td>19.1</td>
<td>24.3</td>
</tr>
<tr>
<td>wpbc60</td>
<td>32</td>
<td>23.5</td>
<td>25.2</td>
<td>19.3</td>
<td>21.8</td>
<td>21.2</td>
<td>16.1</td>
</tr>
<tr>
<td>SPECTF</td>
<td>44</td>
<td>14</td>
<td>28</td>
<td>34</td>
<td>21</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>on-average</td>
<td>40.2</td>
<td>23.7</td>
<td>23.3</td>
<td>18.4</td>
<td>27.0</td>
<td>18.3</td>
<td>17.6</td>
</tr>
</tbody>
</table>

Table 3. Classification Performance (Accuracy)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM</th>
<th>RFE</th>
<th>L1-SVM</th>
<th>FSV</th>
<th>R2W2</th>
<th>MI-SVM*</th>
<th>MI-SVM**</th>
</tr>
</thead>
<tbody>
<tr>
<td>OvarianCancer</td>
<td>66.3</td>
<td>69.1</td>
<td>63.2</td>
<td>67.7</td>
<td>64.8</td>
<td>63.8</td>
<td>76.1</td>
</tr>
<tr>
<td>DART</td>
<td>78.8</td>
<td>91.4</td>
<td>95.7</td>
<td>96.7</td>
<td>91.4</td>
<td>95.7</td>
<td>95.7</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>88.4</td>
<td>87.9</td>
<td>88.7</td>
<td>88.1</td>
<td>88.7</td>
<td>88.7</td>
<td>88.7</td>
</tr>
<tr>
<td>wpbc24</td>
<td>78.8</td>
<td>81.8</td>
<td>81.5</td>
<td>70.8</td>
<td>78.2</td>
<td>81.3</td>
<td>82.1</td>
</tr>
<tr>
<td>wpbc60</td>
<td>66.2</td>
<td>65.6</td>
<td>58.2</td>
<td>61.9</td>
<td>66.5</td>
<td>60.9</td>
<td>63.6</td>
</tr>
<tr>
<td>SPECTF</td>
<td>72.2</td>
<td>75.4</td>
<td>58.8</td>
<td>73.8</td>
<td>73.8</td>
<td>76.5</td>
<td>76.5</td>
</tr>
<tr>
<td>on-average</td>
<td>75.1</td>
<td>77.9</td>
<td>74.3</td>
<td>76.5</td>
<td>77.2</td>
<td>77.8</td>
<td>80.4</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND FUTURE WORK

In this work, we propose a feature-selecting SVM that uses mixed-integer quadratic programming to solve a robust convex relaxation of the L0-norm SVM. Note that this study is in contrast to previous work, which either used a smoothed penalty function that approximates the L0-norm in the objective or used adaptive scale parameters, and then solved through convex optimization techniques. Empirical results show either an increase in sparsity with comparable classification performance, or an increase in classification accuracy compared to other widely-used feature selection approaches: RFE method, L1SVM method, FSV method and Weston’s R2W2 method. We believe our MISVM method demonstrates effective MINLP techniques which have not previous been widely used in machine learning.

Several questions arise from this study. First, is the suggested way to approximate the conic constraints using the big-M method the best? We will investigate other convex functions for approximation. Secondly, the scalability is now constrained by the ability of Cplex to handle high dimensional data sets. We are currently investigating alternative tools and custom approaches to increase performance as well as the computational cost.

6. REFERENCES


