Efficient Learning of Minimax Risk Classifiers in High Dimensions

Abstract

High-dimensional data is common in multiple areas, such as health care and genomics, where the number of features can be hundreds of thousands. In such scenarios, the large number of features often lead to inefficient learning. Constraint generation methods have recently enabled efficient learning of L1-regularized support vector machines (SVMs). In this paper, we leverage such methods to obtain an efficient learning algorithm for the recently proposed minimax risk classifiers (MRCs). The proposed iterative algorithm also provides a sequence of worst-case error probabilities and performs feature selection. Experiments on multiple high-dimensional datasets show that the proposed algorithm is efficient in high-dimensional scenarios. In addition, the worst-case error probability provides useful information about the classifier performance, and the features selected by the algorithm are competitive with the state-of-the-art.

1 INTRODUCTION

High-dimensional data is common in multiple areas such as health care and genomics. A typical example of a supervised classification task in genomics is to separate healthy patients from cancer patients based on their gene expression. In addition to high-dimensional raw data, learning methods often perform a high-dimensional representation of the input data vector in order to improve the classification performance [Rahimi and Recht, 2008, Gönen and Alpaydın, 2011, Gehler and Nowozin, 2008, Liu et al., 2017]. Learning in such high-dimensional settings often leads to highly complex optimization processes because the number of variables involved in the optimization increases with the number of features [Yuan et al., 2012, Shi et al., 2010].

In addition to high dimensional data, a limited number of samples is common in safety-critical applications such as genomics. In such scenarios, the number of samples (patients) are usually fewer than 100 but the number of features are tens of thousands [Guyon and Elisseeff, 2003, Brown et al., 2012]. In these cases, the conventional performance assessment based on cross-validation can be unreliable [Vabalas et al., 2019, Varoquaux, 2018]. In addition, these cross-validation estimates also increase the computational cost as they require learning multiple classifiers.

Multiple methods have been proposed to improve the learning efficiency in high dimensions (see e.g., Yuan et al. [2012]). These methods are based on several approaches such as coordinate descent [Hsieh et al., 2008, Yu et al., 2011], interior-point method [Koh et al., 2007], and stochastic subgradient [Shalev-Shwartz et al., 2007]. Recently, constraint generation techniques have enabled efficient learning of L1-regularized support vector machines (SVMs) for large number of features [Dedieu et al., 2022]. These techniques obtain improved efficiency for the linear program (LP) corresponding with binary L1-SVMs due to the sparsity in the solution.

In this paper, we present a learning algorithm for the recently proposed minimax risk classifiers (MRCs). The presented algorithm provides efficient learning in high dimensions, obtains worst-case error probabilities that can serve to assess performance, and performs feature selection. Specifically, the main contributions in the paper are as follows.

- We present a learning algorithm for MRCs that utilizes constraint generation methods to significantly improve the efficiency in high dimensions.
- The presented algorithm utilizes a greedy feature selection approach that achieves a fast decrease in worst-case error probability while using a small number of features.
- Our theoretical results show that the proposed algorithm obtains a sequence of classifiers with decreasing worst-case error probabilities that converge to the

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

solution obtained using all the features.

- Using multiple benchmark datasets, we experimentally show that the proposed algorithm can provide efficient learning in high dimensions, alternative error assessment, and effective feature selection.

Notations: For a set $\mathcal{S}$, we denote its complement as $\mathcal{S}^{\mathrm{c}}$ and its cardinality as $|\mathcal{S}|$; bold lowercase and uppercase letters represent vectors and matrices, respectively; vectors and matrices subscripted by a set of indices represent subvectors and submatrices obtained by the rows corresponding to the indices in the set; $\mathbf{I}$ denotes the identity matrix; $\mathbb{1}\{\cdot\}$ denotes the indicator function; $\mathbf{1}$ denotes a vector of ones; $\mathrm{sign}(\cdot)$ denotes the vector given by the signs of the argument components; for a vector $\mathbf{v}$, $|\mathbf{v}|$ and $(\mathbf{v})_+$, denote its component-wise absolute value and positive part, respectively; $\|.\|$ denotes the 1-norm of its arguments; $\otimes$ denotes the Kronecker product; $\preceq$ and $\succeq$ denote vector inequalities; $\mathbb{E}_{\mathrm{p}}\{\,\cdot\,\}$ denotes the expectation of its argument with respect to distribution p; and $\mathbf{e}_i$ denotes the i-th vector in a standard basis.

2 PRELIMINARIES

In this section, we describe the setting addressed in the paper and MRC methods that minimize the worst-case error probability.

2.1 PROBLEM FORMULATION

Supervised classification uses instance-label pairs to determine classification rules that assign labels to instances. We denote by $\mathcal{X}$ and $\mathcal{Y}$ the sets of instances and labels, respectively, with $\mathcal{Y}$ represented by $\{1, 2, \ldots, |\mathcal{Y}|\}$. We denote by $\mathrm{T}(\mathcal{X}, \mathcal{Y})$ the set of all classification rules (both randomized and deterministic) and we denote by $\mathrm{h}(y|x)$ the probability with which rule $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ assigns label $y \in \mathcal{Y}$ to instance $x \in \mathcal{X}$ ($\mathrm{h}(y|x) \in \{0, 1\}$ for deterministic classification rules). In addition, we denote by $\Delta(\mathcal{X} \times \mathcal{Y})$ the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$ and by $\ell(\mathrm{h}, \mathrm{p})$ the expected 0-1 loss of the classification rule $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ with respect to distribution $\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y})$. If $\mathrm{p}^* \in \Delta(\mathcal{X} \times \mathcal{Y})$ is the underlying distribution of the instance-label pairs, then $\ell(\mathrm{h}, \mathrm{p}^*)$ is the classification error probability or classification risk of rule h denoted in the following as $\mathcal{R}(\mathrm{h})$, that is, $\mathcal{R}(\mathrm{h}) = \ell(\mathrm{h}, \mathrm{p}^*)$.

Instance-label pairs can be represented as real vectors by using a feature mapping $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$. The most common way to define such feature mapping is using multiple features over instances together with one-hot encodings of labels as follows (see e.g., Mohri et al. [2018])

$$\Phi(x, y) = \mathbf{e}_y \otimes \Psi(x) = \big[\mathbb{1}\{y = 1\}\Psi(x)^{\mathrm{T}}, \mathbb{1}\{y = 2\}\Psi(x)^{\mathrm{T}}, \ldots, \mathbb{1}\{y = |\mathcal{Y}|\}\Psi(x)^{\mathrm{T}}\big]^{\mathrm{T}} \tag{1}$$

where the map $\Psi : \mathcal{X} \to \mathbb{R}^d$ represents instances as real vectors of size d. This map can be just the identity $\Psi(x) = x$ or given by a feature representation such as that provided by random Fourier features (RFF) [Rahimi and Recht, 2008], that is

$$\Psi(x) = \big[\cos(\mathbf{u}_1^{\mathrm{T}}\mathbf{x}), \cos(\mathbf{u}_2^{\mathrm{T}}\mathbf{x}), ..., \cos(\mathbf{u}_D^{\mathrm{T}}\mathbf{x}), \sin(\mathbf{u}_1^{\mathrm{T}}\mathbf{x}), \sin(\mathbf{u}_2^{\mathrm{T}}\mathbf{x}), ..., \sin(\mathbf{u}_D^{\mathrm{T}}\mathbf{x})\big]^{\mathrm{T}} \tag{2}$$

for D random Gaussian vectors $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_D \sim N(0, \gamma\mathbf{I})$ with covariance given by the scaling parameter γ (see e.g., Rahimi and Recht [2008]).

In this paper we consider scenarios in which the features' dimensionality (d) is large, and propose efficient learning methods for the recently presented MRCs. MRCs are briefly described in the following.

2.2 MINIMAX RISK CLASSIFIERS

MRCs are classification rules that minimize the worst-case error probability over distributions in an uncertainty set [Mazuelas et al., 2020, 2022a,c]. Specifically, such rules are solutions to the minimax risk problem defined as

$$\min_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \max_{\mathrm{p} \in \mathcal{U}} \ell(\mathrm{h}, \mathrm{p}) \tag{3}$$

where $\mathcal{U}$ is an uncertainty set of distributions determined by expectation estimates as

$$\mathcal{U} = \{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : |\mathbb{E}_{\mathrm{p}}\{\Phi(x, y)\} - \boldsymbol{\tau}| \preceq \boldsymbol{\lambda}\} \tag{4}$$

where $\boldsymbol{\tau}$ denotes the mean vector of expectation estimates corresponding with the feature mapping Φ, and $\boldsymbol{\lambda} \succeq \mathbf{0}$ is a confidence vector that accounts for inaccuracies in the estimate. The mean and confidence vectors can be obtained from the training samples $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$ as [Mazuelas et al., 2020]

$$\boldsymbol{\tau} = \frac{1}{n}\sum_{i=1}^{n} \Phi(x_i, y_i), \ \boldsymbol{\lambda} = \frac{\mathbf{s}}{\sqrt{(n)}} \tag{5}$$

where $\mathbf{s}$ denotes the vector formed by the component-wise sample standard deviations of $\Phi(x_i, y_i)$ for all $i = 1, 2, \ldots, n$.

As described in Mazuelas et al. [2020, 2022c,a] using the expected 0-1 loss ℓ, the MRC rule h solution of (3) is given by a linear combination of the feature mapping with coefficients given by a vector $\boldsymbol{\mu}^* \in \mathbb{R}^m$. Specifically, the MRC rule h given by $\boldsymbol{\mu}^*$ assigns label $\hat{y} \in \mathcal{Y}$ to instance $x \in \mathcal{X}$ with probability

$$\mathrm{h}(\hat{y}|x) = \begin{cases} \big(\Phi(x, \hat{y})^{\mathrm{T}}\boldsymbol{\mu}^* - \varphi(\boldsymbol{\mu}^*)\big)_+ / d_x & \text{if } d_x \neq 0 \\ 1/|\mathcal{Y}| & \text{if } d_x = 0 \end{cases} \tag{6}$$

$$\text{where} \quad \varphi(\boldsymbol{\mu}) = \max_{x\in\mathcal{X},\mathcal{C}\subseteq\mathcal{Y}} \Big(\frac{\sum_{y\in\mathcal{C}} \Phi(x,y)^{\mathrm{T}}\boldsymbol{\mu} - 1}{|\mathcal{C}|} \Big)$$
$$d_x = \sum_{y\in\mathcal{Y}} \big(\Phi(x,y)^{\mathrm{T}}\boldsymbol{\mu} - \varphi(\boldsymbol{\mu}) \big)_+ .$$

Note that the corresponding deterministic classification rule h^{d} assigns a label that maximizes the probability in (6) given by $\arg\max_{y\in\mathcal{Y}} \mathrm{h}(y|x) = \arg\max_{y\in\mathcal{Y}} \Phi(x,y)^{\mathrm{T}}\boldsymbol{\mu}^*$. In addition, the classification risk of such classification rule h^{d} is bounded by twice the error probability of the MRC rule h, that is,

$$\mathcal{R}(\mathrm{h}^{\mathrm{d}}) \leq 2\mathcal{R}(\mathrm{h}) \tag{7}$$

since $1 - \mathrm{h}^{\mathrm{d}}(y|x) \leq 2(1 - \mathrm{h}(y|x))$ for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$.

The vector $\boldsymbol{\mu}^*$ that determines the 0-1 MRC rule corresponding to (3) is obtained by solving the convex optimization problem [Mazuelas et al., 2020, 2022a]

$$\mathrm{R}^* = \min_{\boldsymbol{\mu}} 1 - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\mu} + \varphi(\boldsymbol{\mu}) + \boldsymbol{\lambda}^{\mathrm{T}}|\boldsymbol{\mu}|. \tag{8}$$

The minimum value R^* of (8) equals the minimax risk value of (3). Hence, it is the MRC's worst-case error probability for distributions included in the uncertainty set $\mathcal{U}$, and can be an upper bound on the MRC's classification error.

The convex optimization problem (8) of MRCs is L1-penalized and leads to sparsity in the coefficients $\boldsymbol{\mu}^*$ corresponding to the features in Φ. This sparsity in the coefficients implies that only a subset of features are sufficient to obtain the optimal worst-case error probability for MRCs. Therefore, in practice, efficient learning can be achieved by using the relevant subset of features to solve the MRC optimization problem. This intuition leads to the efficient learning algorithm presented in this following.

3 EFFICIENT LEARNING OF MRCs IN HIGH DIMENSIONS

This section describes the proposed learning algorithm based on constraint generation over an LP formulation of MRCs learning.

3.1 LP FORMULATION FOR 0-1 MRCs

The theorem below presents an LP formulation (primal and dual) for the 0-1 MRC convex optimization problem (8).

Theorem 1. *Let $\mathcal{S}$ be the set of pairs of instances and labels' subsets, i.e., $\mathcal{S} = \{(x,\mathcal{C}) : x \in \mathcal{X}, \mathcal{C} \subseteq \mathcal{Y}, \mathcal{C} \neq \emptyset\}$, and for each $i = 1, 2 \ldots, |\mathcal{S}|$ corresponding to pair $(x,\mathcal{C})$ let*

$$\mathbf{m}_i = \frac{\sum_{y\in\mathcal{C}} \Phi(x,y)}{|\mathcal{C}|}, \ \mathrm{c}_i = \frac{1}{|\mathcal{C}|} - 1. \tag{9}$$

Then, the 0-1 MRC problem (8) *is equivalent to the LP*

$$\begin{aligned} \mathcal{P}: \min_{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\nu} \ & -(\boldsymbol{\tau}-\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\mu}_1 + (\boldsymbol{\tau}+\boldsymbol{\lambda})^{\mathrm{T}}\boldsymbol{\mu}_2 + \nu \\ \text{s.t.} \quad & \mathbf{M}^{\mathrm{T}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \nu\mathbf{1} \preceq \mathbf{c} \\ & \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \succeq 0 \end{aligned} \tag{10}$$

where the i-th column of matrix $\mathbf{M} \in \mathbb{R}^{m\times|\mathcal{S}|}$ is given by $\mathbf{m}_i$, and the i-th component of vector $\mathbf{c} \in \mathbb{R}^{|\mathcal{S}|}$ is given by c_i. In addition, the Lagrange dual of (10) *is*

$$\begin{aligned} \mathcal{D}: \max_{\boldsymbol{\alpha}} \ & -\mathbf{c}^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\tau} - \boldsymbol{\lambda} \preceq \mathbf{M}\boldsymbol{\alpha} \preceq \boldsymbol{\tau} + \boldsymbol{\lambda} \\ & \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} = 1, \ \boldsymbol{\alpha} \succeq 0 \end{aligned} \tag{11}$$

and if $\boldsymbol{\mu}_1^*$, $\boldsymbol{\mu}_2^*$, ν^* is a solution of (10)*, we have that $\boldsymbol{\mu}^* = \boldsymbol{\mu}_1^* - \boldsymbol{\mu}_2^*$ is a solution of* (8)*.*

Proof. In the convex optimization problem (8), we introduce the additional variable $\nu \in \mathbb{R}$ given by $\nu = \varphi(\boldsymbol{\mu}) + 1$. Then, using $\mathbf{M}$ and $\mathbf{c}$ as defined above, we have

$$\nu \quad = \quad \max_{i\in\{1,2,\ldots,|\mathcal{S}|\}} \quad \mathbf{m}_i^{\mathrm{T}}\boldsymbol{\mu} - \mathrm{c}_i. \tag{12}$$

Therefore, the optimization problem (8) becomes

$$\begin{aligned} \min_{\boldsymbol{\mu},\nu} \ & -\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{\lambda}^{\mathrm{T}}|\boldsymbol{\mu}| + \nu \\ \text{s.t.} \quad & \mathbf{M}\boldsymbol{\mu} - \nu\mathbf{1} \preceq \mathbf{c}. \end{aligned} \tag{13}$$

The optimization in (13) can be reformulated as the LP in (10) using the change of variables $\boldsymbol{\mu}_1 = (\boldsymbol{\mu})_+$ and $\boldsymbol{\mu}_2 = (-\boldsymbol{\mu})_+$ which implies $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ and $|\boldsymbol{\mu}| = \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2$. Finally, the dual in (11) is directly obtained from the LP (10) (see e.g., Section 4.2 in Bertsimas and Tsitsiklis [1997]). □

The number of variables of the primal problem $\mathcal{P}$ in (10) (constraints in dual problem $\mathcal{D}$ in (11)) is given by the number of features $m = d|\mathcal{Y}|$. Therefore, the complexity of MRC learning given by such LP formulation is $O(m^3)$ that is not affordable in high-dimensional settings. The next section presents an algorithm that provides efficient learning in high dimensions based on constraint generation methods.

3.2 ALGORITHM FOR EFFICIENT LEARNING

MRC learning can be efficiently carried out in high-dimensional settings because the solution of (10) is a sparse vector. This sparsity is due to the L1-penalization in (8) and the fact that usually only a small subset of features are informative in high-dimensional settings [Ghosh and Cabrera, 2022]. The sparsity of the MRCs' coefficients enables to carry out the learning process considering only a small subset of variables in (10) (respectively constraints in (11)). In the following, we propose an algorithm based

on constraint generation methods that solves the MRC LP problem in (10) by iteratively selecting small subsets of features.

Algorithm 1 Efficient learning algorithm for 0-1 MRCs

Input: $\mathbf{M}$, $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$, initial subset of features $\mathcal{I}$, constraints threshold ϵ, maximum number of iterations $k_{\max}$, and maximum number of features added in each iteration $n_{\max}$
Output: selected features $\mathcal{I}^* \subseteq \{1, 2, \ldots, m\}$, optimal solution $\boldsymbol{\mu}^* \in \mathbb{R}^m$, and optimal worst-case error probability R^*
$\boldsymbol{\mu}^1 = [0, 0, \ldots, 0] \in \mathbb{R}^m$
LPSOLVE$(\mathcal{P}_{\mathcal{I}})$
$\boldsymbol{\mu}_1^1,\ \boldsymbol{\mu}_2^1,\ \nu^1 \leftarrow$ Solution of primal
$\boldsymbol{\alpha}^1 \leftarrow$ Solution of dual
$\mathrm{R}^1 \leftarrow$ Optimal value (worst-case error probability)
$\boldsymbol{\mu}^1{}_{\mathcal{I}} = \boldsymbol{\mu}_1^1 - \boldsymbol{\mu}_2^1$
$\mathcal{J} \leftarrow$ SELECT$(\mathbf{M},\ \boldsymbol{\tau},\ \boldsymbol{\lambda},\ n_{\max},\ \mathcal{I},\ \epsilon,\ \boldsymbol{\alpha}^1)$
$k = 1$
while $\mathcal{J} \setminus \mathcal{I} \neq \emptyset$ and $k \leq k_{\max}$ **do**
  $k = k + 1$
  $\boldsymbol{\mu}_1^{k-1} = (\boldsymbol{\mu}^{k-1}{}_{\mathcal{J}})_+,\ \boldsymbol{\mu}_2^{k-1} = (-\boldsymbol{\mu}^{k-1}{}_{\mathcal{J}})_+$
  LPSOLVE$(\mathcal{P}_{\mathcal{J}},\ \boldsymbol{\mu}_1^{k-1},\ \boldsymbol{\mu}_2^{k-1},\ \nu^{k-1})$
  $\boldsymbol{\mu}_1^k,\ \boldsymbol{\mu}_2^k,\ \nu^k \leftarrow$ Solution of primal
  $\boldsymbol{\alpha}^k \leftarrow$ Solution of dual
  $\mathrm{R}^k \leftarrow$ Optimal value (worst-case error probability)
  $\boldsymbol{\mu}^k = \boldsymbol{\mu}^{k-1}$
  $\boldsymbol{\mu}^k{}_{\mathcal{J}} = \boldsymbol{\mu}_1^k - \boldsymbol{\mu}_2^k$
  $\mathcal{I} \leftarrow \mathcal{J}$
  $\mathcal{J} \leftarrow$ SELECT$(\mathbf{M},\ \boldsymbol{\tau},\ \boldsymbol{\lambda},\ n_{\max},\ \mathcal{I},\ \epsilon,\ \boldsymbol{\alpha}^k)$
end while
$\mathcal{I}^* = \mathcal{I}, \boldsymbol{\mu}^* = \boldsymbol{\mu}^k, \mathrm{R}^* = \mathrm{R}^k$

The proposed learning algorithm obtains the optimal solution $\boldsymbol{\mu}^*$ to the original problem by iteratively solving a sequence of subproblems (see details in Algorithm 1). These subproblems correspond with small subsets of features selected by a constraint generation method (SELECT function detailed in Algorithm 2).

The subproblem of (10) corresponding to the subset of features $\mathcal{J} \subseteq \{1, 2, \ldots, m\}$ is defined as

$$\begin{aligned} \mathcal{P}_{\mathcal{J}}: \min_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \nu} \quad & -(\boldsymbol{\tau}_{\mathcal{J}} - \boldsymbol{\lambda}_{\mathcal{J}})^{\mathrm{T}} \boldsymbol{\mu}_1 + (\boldsymbol{\tau}_{\mathcal{J}} + \boldsymbol{\lambda}_{\mathcal{J}})^{\mathrm{T}} \boldsymbol{\mu}_2 + \nu \\ \text{s.t.} \quad & (\mathbf{M}_{\mathcal{J}})^{\mathrm{T}} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \nu \mathbf{1} \preceq \mathbf{c} \\ & \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \succeq 0 \end{aligned} \tag{14}$$

where $\boldsymbol{\tau}_{\mathcal{J}}$ and $\boldsymbol{\lambda}_{\mathcal{J}}$ denote the subvectors of $\boldsymbol{\tau}$ and $\boldsymbol{\lambda}$ corresponding to the $\mathcal{J}$ components, and $\mathbf{M}_{\mathcal{J}}$ denotes the submatrix of $\mathbf{M}$ corresponding to the $\mathcal{J}$ columns. In addition, the dual of (14) is

$$\begin{aligned} \mathcal{D}_{\mathcal{J}}: \quad \max_{\boldsymbol{\alpha}} \quad & -\mathbf{c}^{\mathrm{T}} \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\tau}_{\mathcal{J}} - \boldsymbol{\lambda}_{\mathcal{J}} \preceq \mathbf{M}_{\mathcal{J}} \boldsymbol{\alpha} \preceq \boldsymbol{\tau}_{\mathcal{J}} + \boldsymbol{\lambda}_{\mathcal{J}} \\ & \mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \succeq 0. \end{aligned} \tag{15}$$

At each iteration k of Algorithm 1, the LPSOLVE function solves such subproblems and obtains the primal solution $\boldsymbol{\mu}_1^k, \boldsymbol{\mu}_2^k, \nu^k$ and dual solution $\boldsymbol{\alpha}^k$ along with the worst-case error probability R^k. The dual solution is used by the SELECT function to obtain a subsequent subset of features while the primal solution can be used to warm-start the optimization in the next iteration. This iterative process ends when the SELECT function returns a subset of features that does not contain any new feature.

Notice that at each iteration k, the algorithm obtains coefficients $\boldsymbol{\mu}^k$, features subset $\mathcal{J}$, and worst-case error probability R^k. In Section 3.5, we show that such coefficients $\boldsymbol{\mu}^k$ provide an MRC with worst-case error probability R^k corresponding with the uncertainty set given by features $\mathcal{J}$. In addition, we show that such worst-case error probability monotonically decreases with the number of iterations and converges to the optimal corresponding with all the features.

3.3 GREEDY FEATURE SELECTION

The SELECT function determines the subset of features for the next iteration based on the previous subset and the current solution. Specifically, this greedy selection process adds features corresponding to the most violated constraints in the dual and removes features corresponding to the constraints satisfied with a positive slack. Such a process aims to achieve the fastest decrease in worst-case error probability while using the smallest number of features. As described in the literature for constraint generation methods [Bertsimas and Tsitsiklis, 1997, Desrosiers and Lübbecke, 2005], this type of selection process can be implemented in several alternative ways such as adding any subset of violated constraints or not removing constraints.

The implementation details for function SELECT are provided in Algorithm 2. The vector $\mathbf{v}$ quantifies the violations in the dual constraints. In particular, the features corresponding to the negative values of vector $\mathbf{v}$ (over-satisfied constraints) are removed, and the features corresponding to the $n_{\max}$ largest positive values of vector $\mathbf{v}$ are selected. Moreover, the selection of features is also restricted by hyperparameter $\epsilon \geq 0$ that represents the minimum violation for dual constraints.

Algorithm 2 SELECT (function for greedy feature selection)

Input: $\mathbf{M}$, $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$, $n_{\max}$, current subset of features $\mathcal{I}$, constraints threshold ϵ and dual solution $\boldsymbol{\alpha}$

Output: selected features $\mathcal{J} \subseteq \{1, \ldots, m\}$

$\mathbf{v} = |\mathbf{M}\boldsymbol{\alpha} - \boldsymbol{\tau}| - \boldsymbol{\lambda}$

$\mathcal{J} = \{i : i \in \mathcal{I},\ \mathrm{v}_i = 0\}$

$\mathcal{A} \leftarrow \mathrm{ARGNMAX}(\mathbf{v}, n_{\max})$

$\mathcal{A} = \{i : i \in \mathcal{A},\ \mathrm{v}_i > \epsilon\}$

$\mathcal{J} \leftarrow \mathcal{J} \cup \mathcal{A}$

3.4 COMPUTATIONAL COMPLEXITY

The computational complexity of Algorithm 1 is given by the number of iterations and the complexity per iteration. The number of iterations depend on the maximum number of features selected in each iteration $n_{\max}$ and the hyperparameter ϵ, and the complexity per iteration depends on $n_{\max}$. Decreasing $n_{\max}$ results in a reduced complexity per iteration at the expense of an increased number of iterations. Increasing ϵ can decrease the number of iterations and complexity per iteration at the expense of achieving approximate solutions. For instance, if $n_{\max} = m$ and $\epsilon = 0$ the algorithm finds the exact solution in only one iteration but the complexity of such iteration is large. In Section 4, we show that in practice $\epsilon = 0.0001$ and $n_{\max} \in [50, 500]$ obtains efficient and accurate learning of MRCs

The number of iterations can be further reduced by using an adequate choice for the initial subset of features. This initial subset of features $\mathcal{I}$ can be obtained efficiently using a simple approach for coarse feature selection such as correlation screening [Tibshirani et al., 2012] or few iterations of a first order optimization method [Mazuelas et al., 2022b]. Similarly, the complexity per iteration is reduced by using a warm-start in LPSOLVE. In the proposed algorithm, a warm-start for iteration k is obtained directly from the previous solution $\boldsymbol{\mu}^{k-1}$ of the LP at iteration $k-1$. Note that this warm-start is a basic feasible solution for the LP at iteration k since it is obtained by removing and adding features for which the corresponding coefficients at $k-1$ are zero.

3.5 THEORETICAL ANALYSIS OF THE ALGORITHM

The following shows the theoretical properties of the proposed learning algorithm. In particular, it shows that the algorithm's iterations provide a sequence of MRCs with worst-case error probabilities that converge to the MRC corresponding with all the features.

The following theorem shows that the algorithm provides a decreasing sequence of worst-case error probabilities that can provide an upper bound to the classification error of the corresponding MRCs.

Theorem 2. *Let $\boldsymbol{\mu}^k$ and R^k be the coefficients and worst-case error probability obtained by the proposed algorithm at iteration k. If h^k is the MRC given by $\boldsymbol{\mu}^k$, then we have that*

$$\mathcal{R}(\mathrm{h}^k) \leq \mathrm{R}^k + (|\mathbb{E}_{\mathrm{p}^*}\{\Phi(x,y)_{\mathcal{J}}\} - \boldsymbol{\tau}_{\mathcal{J}}| - \boldsymbol{\lambda}_{\mathcal{J}})^{\mathrm{T}} |\boldsymbol{\mu}^k_{\mathcal{J}}| \quad (16)$$

where $\mathcal{J}$ is the subset of features used at iteration k. In addition, the algorithm provides MRCs with decreasing worst-case error probabilities, that is,

$$\mathrm{R}^{k+1} \leq \mathrm{R}^k,\ k = 1, 2, \ldots \quad (17)$$

Proof. At each intermediate step, $\boldsymbol{\mu}^k$ and R^k are the coefficients and worst-case error probability of the MRC corresponding with uncertainty set

$$\mathcal{U}^k = \{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : |\mathbb{E}_{\mathrm{p}^*}\{\Phi(x,y)_{\mathcal{J}}\} - \boldsymbol{\tau}_{\mathcal{J}}| \preceq \boldsymbol{\lambda}_{\mathcal{J}}\} \quad (18)$$

using Theorem 1 for the subproblem (14). Therefore, inequality (16) is obtained using the bounds for error probabilities for MRCs in Mazuelas et al. [2022b]. The second result is obtained since the warm-start at iteration $k+1$ is a feasible solution with an objective value that equals the worst-case error probability R^k at iteration k. The warm-start is feasible at $k+1$ and has value R^k because it is obtained from the solution at iteration k by removing and adding features corresponding to zero coefficients. □

Inequality (16) bounds the classification error of the MRC classifier at any intermediate iteration. Specifically, the minimax risk R^k at any iteration k corresponding to a feature subset $\mathcal{J}$ directly provides such a bound if the error in the mean vector estimate is not underestimated, i.e., $\boldsymbol{\lambda}_{\mathcal{J}} \succeq |\mathbb{E}_{\mathrm{p}^*}\{\Phi(x,y)_{\mathcal{J}}\} - \boldsymbol{\tau}_{\mathcal{J}}|$. In particular, if $\boldsymbol{\lambda}_{\mathcal{J}}$ is a confidence vector with converge probability $1-\delta$, that is, $\mathbb{P}\{|\mathbb{E}_{\mathrm{p}^*}\{\Phi(x,y)_{\mathcal{J}}\} - \boldsymbol{\tau}_{\mathcal{J}}| \preceq \boldsymbol{\lambda}_{\mathcal{J}}\} \geq 1-\delta$, then

$$\mathcal{R}(\mathrm{h}^k) \leq \mathrm{R}^k \quad (19)$$

with probability at least $1-\delta$. In other cases, R^k still provides approximate bounds as long as the underestimation $|\mathbb{E}_{\mathrm{p}^*}\{\Phi(x,y)_{\mathcal{J}}\} - \boldsymbol{\tau}_{\mathcal{J}}| - \boldsymbol{\lambda}_{\mathcal{J}}$ is not substantial. Similar bounds also hold for the sequence of deterministic classification rules due to inequality in (7).

The next theorem shows the convergence of the sequence of MRCs obtained by the proposed algorithm to the MRC corresponding with all the features.

Theorem 3. *Let $\boldsymbol{\mu}^*$ and R^* be the MRC coefficients and worst-case error probability obtained by solving* (10) *using all the features. If R^k, $k = 1, 2, \ldots$, is the sequence of*

worst-case error probabilities obtained by the proposed algorithm using dual constraints' violation threshold ϵ. Then, there exists $k_0 \geq 1$ such that

$$\mathrm{R}^* \leq \mathrm{R}^k \leq \mathrm{R}^* + \epsilon \|\boldsymbol{\mu}^*\|_1 \tag{20}$$

for any $k \geq k_0$.

Proof. The first inequality follows by noting that the feasible set of (11) corresponding with all the features is contained in the feasible set of any subproblem given a subset of features. In the following, we prove the second inequality.

Let $k_0 \geq 1$ be an iteration in which SELECT function in Algorithm 1 does not add any new feature. Such a case occurs after a finite number of iterations due to the properties of constraint generation methods [Bertsimas and Tsitsiklis, 1997]. If $\boldsymbol{\alpha}^{k_0}$ is the dual solution obtained at such iteration k_0 with subset of features $\mathcal{J}$, then we have that

$$\boldsymbol{\tau}_{\mathcal{J}} - \boldsymbol{\lambda}_{\mathcal{J}} \preceq \mathbf{M}_{\mathcal{J}} \boldsymbol{\alpha}^{k_0} \preceq \boldsymbol{\tau}_{\mathcal{J}} + \boldsymbol{\lambda}_{\mathcal{J}} \tag{21}$$

$$\boldsymbol{\tau}_{\mathcal{J}^c} - \boldsymbol{\lambda}_{\mathcal{J}^c} - \epsilon \mathbf{1} \preceq \mathbf{M}_{\mathcal{J}^c} \boldsymbol{\alpha}^{k_0} \preceq \boldsymbol{\tau}_{\mathcal{J}^c} + \boldsymbol{\lambda}_{\mathcal{J}^c} + \epsilon \mathbf{1} \tag{22}$$

On combining (21) and (22), we have

$$\boldsymbol{\tau} - \boldsymbol{\lambda} - \epsilon \mathbf{1} \preceq \mathbf{M} \boldsymbol{\alpha}^{k_0} \preceq \boldsymbol{\tau} + \boldsymbol{\lambda} + \epsilon \mathbf{1} \tag{23}$$

Now, consider the following primal problem corresponding to the dual (11) with constraints as in (23).

$$\begin{aligned} \min_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \nu} \quad & -(\boldsymbol{\tau} - \boldsymbol{\lambda} - \epsilon \mathbf{1})^{\mathrm{T}} \boldsymbol{\mu}_1 + (\boldsymbol{\tau} + \boldsymbol{\lambda} + \epsilon \mathbf{1})^{\mathrm{T}} \boldsymbol{\mu}_2 + \nu \\ \text{s.t.} \quad & \mathbf{M}^{\mathrm{T}} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - \nu \mathbf{1} \preceq \mathbf{c} \\ & \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \succeq 0 \end{aligned} \tag{24}$$

If $\boldsymbol{\mu}_1^*$, $\boldsymbol{\mu}_2^*$, and ν^* correspond to the MRC coefficients obtained by solving (10) using all the features, then it is also a feasible solution to the problem (24) since they have the same feasible set. Therefore, using weak duality (Theorem 4.3 in Bertsimas and Tsitsiklis [1997]), we have that

$$-\mathbf{c}^{\mathrm{T}} \boldsymbol{\alpha}^{k_0} \leq -(\boldsymbol{\tau} - \boldsymbol{\lambda} - \epsilon \mathbf{1})^{\mathrm{T}} \boldsymbol{\mu}_1^* + (\boldsymbol{\tau} + \boldsymbol{\lambda} + \epsilon \mathbf{1})^{\mathrm{T}} \boldsymbol{\mu}_2^* + \nu^* \tag{25}$$

since $\boldsymbol{\alpha}^{k_0}$ is a feasible solution for the dual of (24). Therefore,

$$\mathrm{R}^{k_0} \leq \mathrm{R}^* + \epsilon \mathbf{1} (\boldsymbol{\mu}_1^* + \boldsymbol{\mu}_2^*) = \mathrm{R}^* + \epsilon \|\boldsymbol{\mu}^*\|_1 \tag{26}$$

since $\boldsymbol{\mu}_1^* = (\boldsymbol{\mu}^*)_+$ and $\boldsymbol{\mu}_2^* = (-\boldsymbol{\mu}^*)_+$. Hence, the second inequality holds for any $k \geq k_0$ since $\mathrm{R}^k \leq \mathrm{R}^{k_0}$ due to the monotonic decrease of the worst-case error probability shown in Theorem 2. □

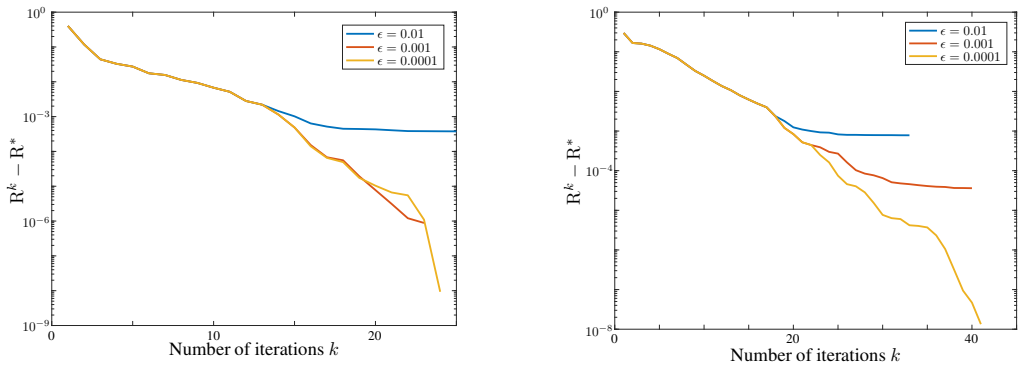
Inequality (20) shows that if $\epsilon = 0$ the algorithm finds the MRC corresponding to all the features. In other cases, it finds an approximate solution whose accuracy depends on the hyperparameter ϵ. Therefore, the hyperparameter ϵ can serve to reduce the complexity of the algorithm while obtaining near optimal solutions. In the next section, we further analyze the effect of hyperparameter ϵ through numerical experiments using multiple real datasets.

4 RESULTS

In this section, we present a set of experimental results that analyze the effect of hyperparamters, quantify the training time improvements in practice, compare the training time with the state-of-the-art. In addition, we show the benefits of having the worst-case error probabilities, and show that the presented algorithm perform efficient feature selection. The proposed Algorithm 1 (MRC-CG) is compared with the LP formulation for MRCs in 10 using all the features (MRC-LP), the constraint generation method for L1-SVMs (SVM-CG) [Dedieu et al., 2022]. In addition, the quality of feature selection is compared with SVM-CG and recursive feature elimination (RFE) for SVMs [Guyon et al., 2002]. The experimental results are obtained using 13 real-world high-dimensional datasets with number of features ranging from 2,000 to 100,000 and number of samples less than 300. Further details about the datasets and the experimental setup along with the implementation details of the proposed algorithm can be found in Appendix A.

The hyperparameter ϵ: We present results to study the influence of ϵ and illustrate, in practice, the theoretical properties of MRC-CG using datasets "Ovarian" and "Arcene". The worst-case error probability R^* is obtained by solving (10) using all the features and the sequence of worst-case error probabilities R^k is obtained by MRC-CG for increasing number of iterations k. In Figure 1, we show the convergence of R^k to R^* using $\epsilon = \{0.01, 0.001, 0.0001\}$ and $n_{\max} = 100$. In practice, we observe that R^k converges to R^* with differences in the order of 10^{-3} and lower even for $\epsilon = 0.1$. In addition, we observe that R^k is monotonically decreasing (as shown in Theorem 2) and achieves significant convergence in few iterations. We also observe that the smaller values of ϵ lead to higher precision (small error), as shown in Theorem 3. Additional results using more datasets are available in Appendix B. We observe that highly accurate results are obtained in 20 iterations using ϵ in the order of 10^{-4}.

The hyperparameter $n_{\max}$: We present results to study the influence of the hyperparameter $n_{\max}$ on the training time of MRC-CG, and we compare it with MRC-LP using datasets "Prostate_GE", "Arcene", "Ovarian" and "SMK_CAN_187". The parameters of MRC-CG are $\epsilon = 0.0001$ and $k_{\max} = 20$. All the training times



(a) Ovarian (b) Arcene

Figure 1. Monotonic decrease of the worst-case error probability R^k with increasing number of iterations k.

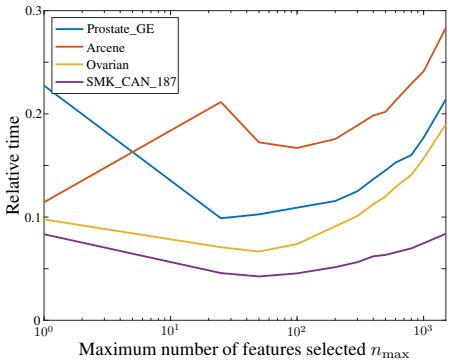


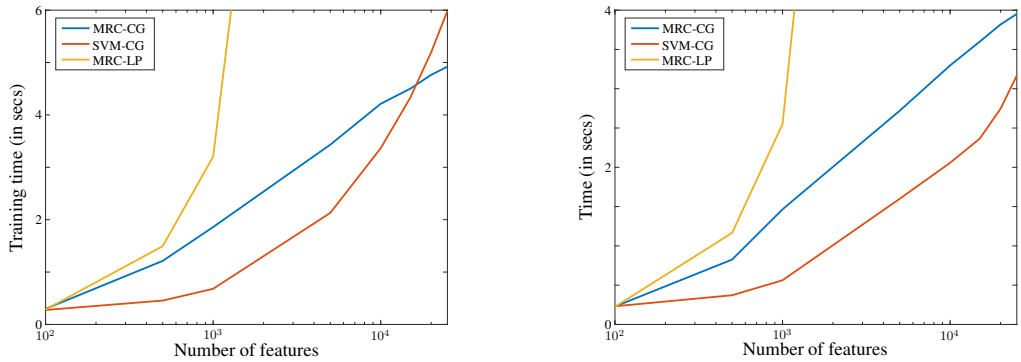
Figure 2: Improved efficiency of MRC-CG over MRC-LP for different values of maximum number of features selected $n_{\max}$

are averaged over 50 random repetitions using 90% of the data. Figure 2 presents the relative time computed as the time taken by MRC-CG over the time taken by MRC-LP for values of $n_{\max}$ in the range of 1 to 1,500. The figure shows that MRC-CG is 10 times faster on average than MRC-LP for $n_{\max} = 100$. In addition, we observe that the MRC-CG is faster than MRC-LP for all the values of $n_{\max}$ and in general, any value in the range of 50 to 500 provides lower training times. Based on these results, in the remainder of the section we take $n_{\max} = 100$, $\epsilon = 0.0001$ and $k_{\max} = 20$.

Scalability with increasing number of features: We present results to compare the scalability of MRC-CG with MRC-LP, and SVM-CG for increasing number of features on the datasets "Ovarian" and "Arcene". The results show the effect of the number of features on the training time of MRC-CG and compare it with MRC-LP and SVM-CG. The number of features range from 100 to 25,000 obtained using RFFs. Figure 3 presents the average training times for MRC-CG, MRC-LP, and SVM-CG over 20 random repetitions. We observe that MRC-CG is faster than MRC-LP for all cases and improves the scaling with the number of features significantly better than MRC-LP. In addition, the training times of MRC-CG are competitive with training times of SVM-CG, especially for large number of features. *Additional results using more datasets are available in Appendix B.*

Comparison using real-world datasets: Table 2 presents results to compare the average training times for MRC-CG and SVM-CG using the 13 high-dimensional datasets. In addition, we present the worst-case error probability R^* given by MRC-CG along with the 10 fold cross-validated error estimate. We observe that MRC-CG is competitive with the SVM-CG in terms of training times. In addition, the worst-case error probability R^* which is obtained using all the data can provide an error assessment for MRC-CG without requiring cross-validation. For instance, the R^* is in the confidence interval of the estimated error in 8 out of 13 datasets, and is an upper bound to the confidence interval of the estimated error in the remaining datasets. Additional results for the error estimates of SVM-CG using 10 fold cross-validation are presented in Appendix B.

Feature selection: We present results in Table 1 to compare the MRC-CG as a feature selection approach with SVM-CG, and RFE. The table shows the 10 fold cross-validated errors for logistic regression (LR) and decision tree (DT) using features selected by MRC-CG, SVM-CG, and RFE. We assess the quality of features selected based on the errors and the number of features selected. Note that the number of features selected by RFE were set to the number features selected by SVM-CG. The errors show that the presented MRC-CG provides competitive feature selection compared SVM-CG and RFE. In addition, the number of features selected MRC-CG and SVM-CG are comparable and much lower than the original number of features.



(a) Ovarian (b) Arcene

Figure 3. Comparison of training times (in secs) of MRC-CG, MRC-LP, and SVM-CG for increasing number of RFF.

Table 1. Comparison of number of features selected and the error estimate obtained for logistic regression (LR) and decision tree (DT) using MRC-CG, SVM-CG, and RFE for feature selection.

Dataset	MRC-CG			SVM-CG			RFE	
	Error		No. of features	Error		No. of features	Error	
	LR	DT		LR	DT		LR	DT
1	0.29 ± 0.10	0.33 ± 0.06	172 ± 3	0.27 ± 0.11	0.41 ± 0.09	150 ± 3	0.28 ± 0.05	0.31 ± 0.12
2	0.20 ± 0.10	0.19 ± 0.10	33 ± 2	0.19 ± 0.10	0.22 ± 0.13	32 ± 1	0.24 ± 0.11	0.20 ± 0.10
5	0.23 ± 0.13	0.17 ± 0.12	76 ± 0	0.14 ± 0.07	0.29 ± 0.11	51 ± 2	0.20 ± 0.11	0.31 ± 0.13
7	0.04 ± 0.06	0.12 ± 0.09	63 ± 1	0.02 ± 0.05	0.15 ± 0.12	37 ± 2	0.05 ± 0.06	0.16 ± 0.18
10	0.00 ± 0.00	0.02 ± 0.02	118 ± 6	0.00 ± 0.00	0.01 ± 0.02	33 ± 2	0.00 ± 0.00	0.03 ± 0.02
11	0.08 ± 0.06	0.16 ± 0.08	77 ± 2	0.10 ± 0.08	0.13 ± 0.10	45 ± 1	0.10 ± 0.10	0.18 ± 0.12
12	0.29 ± 0.12	0.41 ± 0.11	135 ± 3	0.27 ± 0.09	0.37 ± 0.09	127 ± 4	0.33 ± 0.08	0.41 ± 0.11

Table 2. Comparison of training times (in secs) of MRC-CG and SVM-CG along with the worst-case error probabilities R^* and cross-validated error estimate for MRC-CG using multiple high dimensional datasets.

Dataset	MRC-CG			SVM-CG
	R^*	Error	Time (in secs)	Time (in secs)
1	0.20	0.30 ± 0.10	3.56 ± 0.04	8.25 ± 0.54
2	0.24	0.13 ± 0.12	0.20 ± 0.02	0.58 ± 0.07
3	0.16	0.15 ± 0.09	6.84 ± 0.04	3.55 ± 0.35
4	0.07	0.07 ± 0.01	31.4 ± 0.64	220 ± 10.5
5	0.13	0.18 ± 0.17	1.50 ± 0.07	1.36 ± 0.18
6	0.18	0.38 ± 0.20	4.55 ± 0.02	2.58 ± 0.24
7	0.11	0.02 ± 0.05	0.70 ± 0.05	0.37 ± 0.08
8	0.15	0.05 ± 0.05	44.9 ± 0.96	16.6 ± 0.76
9	0.11	0.04 ± 0.06	5.06 ± 0.03	1.45 ± 0.20
10	0.10	0.00 ± 0.00	3.86 ± 0.26	3.35 ± 0.32
11	0.14	0.08 ± 0.07	0.79 ± 0.09	1.00 ± 0.10
12	0.23	0.25 ± 0.11	2.65 ± 0.13	8.9 ± 0.63
13	0.18	0.03 ± 0.03	17.7 ± 0.21	10.5 ± 0.22

5 CONCLUSION

In this paper, we presented a learning algorithm for the recently proposed minimax risk classifiers (MRCs) that is efficient in high-dimensional settings with a large number of features. The presented algorithm utilizes a greedy feature selection approach that iteratively removes and adds features achieving a fast decrease in worst-case error probability and using a small number of features. We prove theoretically that the proposed iterative algorithm obtains a sequence of MRCs with decreasing worst-case error probabilities that converge to the solution obtained using all the features. The numerical results asses the efficiency of the presented algorithm and compare it with state-of-the-art using 13 high-dimensional datasets with a large number of features. The results show that the presented algorithm converges to the solution obtained using all the features after few iterations, and provides a significant efficiency increase, especially in cases with a large number of features. We show that the proposed efficient algorithm is appropriate to deal with the feature selection problem. In addition, the algorithm provides the worst-case probability error which is particularly useful in high-dimensional scenarios with limited number of samples., which suffer from a high variability in scenarios with a limited number of samples.

References

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*, volume 6. Athena Scientific, Belmont, MA, 1997.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13:27–66, 2012.

Antoine Dedieu, Rahul Mazumder, and Haoyue Wang. Solving l1-regularized svms and related linear programs: Revisiting the effectiveness of column and constraint generation. *Journal of Machine Learning Research*, 23: 1–41, 2022.

Jacques Desrosiers and Marco E Lübbecke. *A primer in column generation*. Springer, 2005.

Peter Vincent Gehler and Sebastian Nowozin. Infinite kernel learning. *Max Planck Institute for Biological Cybernetics*, 2008.

Debopriya Ghosh and Javier Cabrera. Enriched random forest for high dimensional genomic data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(5):2817–2828, 2022.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422, 2002.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, 2008.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine learning research*, 8 (Jul):1519–1555, 2007.

Yong Liu, Shizhong Liao, Hailun Lin, Yinliang Yue, and Weiping Wang. Infinite kernel learning: generalization bounds and algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Santiago Mazuelas, Andrea Zanoni, and Aritz Perez. Minimax classification with 0-1 loss and performance guarantees. In *Advances in Neural Information Processing Systems*, pages 302–312, 2020.

Santiago Mazuelas, Mauricio Romero, and Peter Grünwald. Minimax risk classifiers with 0-1 loss. *arXiv preprint*, arXiv:2201.06487, 2022a.

Santiago Mazuelas, Mauricio Romero, and Peter Grünwald. Minimax risk classifiers with 0-1 loss, 2022b.

Santiago Mazuelas, Yuan Shen, and Aritz Pérez. Generalized maximum entropy for supervised classification. *IEEE Transactions on Information Theory*, 68(4):2530–2550, 2022c.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, Cambridge, MA, second edition, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, 2008.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814, 2007.

Jianing Shi, Wotao Yin, Stanley Osher, and Paul Sajda. A fast hybrid algorithm for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 11:713–741, 2010.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, 2012.

Andrius Vabalas, Emma Gowen, Ellen Poliakoff, and Alexander J Casson. Machine learning algorithm validation with a limited sample size. *PloS One*, 14:e0224365, 2019.

Gaël Varoquaux. Cross-validation failure: Small sample sizes lead to large error bars. *Neuroimage*, 180:68–77, 2018.

Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85:41–75, 2011.

Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012.