PAPER

Mixed-Integer Linear Optimization Formulations for Feature Subset Selection in Kernel SVM Classification

Ryuta TAMURA [†,††a)], **Yuichi TAKANO** [†††], *Nonmembers*, *and* **Ryuhei MIYASHIRO** [††††], *Member*

SUMMARY We study the mixed-integer optimization (MIO) approach to feature subset selection in nonlinear kernel support vector machines (SVMs) for binary classification. To measure the performance of subset selection, we use the distance between two classes (DBTC) in a high-dimensional feature space based on the Gaussian kernel function. However, DBTC to be maximized as an objective function is nonlinear, nonconvex and nonconcave. Despite the difficulty of linearizing such a nonlinear function in general, our major contribution is to propose a mixed-integer linear optimization (MILO) formulation to maximize DBTC for feature subset selection, and this MILO problem can be solved to optimality using optimization software. We also derive a reduced version of the MILO problem to accelerate our MILO computations. Experimental results show good computational efficiency for our MILO formulation with the reduced problem. Moreover, our method can often outperform the linear-SVM-based MILO formulation and recursive feature elimination in prediction performance, especially when there are relatively few data instances.

key words: *feature subset selection, support vector machine, mixed-integer optimization, kernel–target alignment, machine learning*

1. Introduction

1.1 Background

Support vector machines (SVMs) are a family of sophisticated machine learning methods based on optimal separating hyperplanes. This method was first devised for binary classification by Boser et al. [11] in combination with the kernel method [1] for nonlinear data analyses. Since then, SVMs have attracted considerable attention in various scientific fields because of their solid theoretical foundations and high generalization ability [15], [21], [74]. Kernel methods have been extended to a variety of multivariate analyses (e.g., principal component analysis, cluster analysis, and outlier detection) [64], [65], and they have also been applied to dynamic portfolio selection [67], [68].

Feature subset selection involves selecting a subset of relevant features used in machine learning models. Such selection helps to understand the causality between predictor features and response classes, and it reduces the data collection/storage costs and the computational load of training machine learning models. Moreover, the prediction performance can be improved because overfitting is mitigated by eliminating redundant features. Because of these benefits, algorithms for subset selection have been extensively studied [17], [30], [48], [49]. These algorithms can be categorized into filter, wrapper, and embedded methods. Filter methods (e.g., Fisher score [32] and relief [13], [42]) rank features according to evaluation criteria before the training phase. Wrapper methods (e.g., recursive feature elimination [32]) search for better subsets of features through repeated training of subset models. Embedded methods (e.g., L_1-regularized estimation [34]) provide a subset of features as a result of the training process.

Mixed-integer optimization (MIO; formerly known as mixed-integer programming, or MIP) is the study of how to formulate and solve mathematical optimization problems involving real- and integer-valued decision variables [83]. Since many optimization problems can be formulated as MIO problems [82], MIO has been studied extensively, mainly in the field of operations research. Although solving MIO problems is NP-hard in general, recent advancement of optimization algorithms and computer hardware allows us to tackle large-scale MIO problems [41]. In particular, MILO (mixed-integer linear optimization) problems, namely MIO problems consisting of linear functions of decision variables, are rather tractable. Recent records [27] show that several MILO problems with millions of decision variables are solvable within an hour. In contrast, integer nonlinear optimization (INLO) problems, which consist of integer decision variables and contain nonlinear functions of them, are still quite difficult to solve. INLO problems even with a few hundreds of decision variables cannot be exactly solved by state-of-the-art optimization software [55]. Some linearization/transformation techniques for reducing INLO to MILO problems have been developed, but such techniques have only limited application.

1.2 Related Work

We address MIO approaches to feature subset selection for kernel SVM classification. First proposed for linear regression in the 1970s [2], this approach has recently moved into the spotlight with advances in optimization algorithms and computer hardware [6], [19], [33], [43], [73]. Compared with many heuristic optimization algorithms, the MIO ap-

Manuscript received April 15, 2023.
Manuscript revised August 9, 2023.
Manuscript publicized February 8, 2024.

[†]Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184-8588 Japan.
[††]October Sky Co., Ltd., Fuchu-shi, 183-0055 Japan.
[†††]Institute of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305-8573 Japan.
[††††]Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184-8588 Japan.
a) E-mail: r.tamura.cbc@gmail.com

DOI: 10.1587/transfun.2023EAP1043

Copyright © 2024 The Institute of Electronics, Information and Communication Engineers

proach has the advantage of selecting the best subset of features with respect to given criterion functions [56], [57], [60], [69]. MIO methods for subset selection have been extended to logistic regression [7], [63], ordinal regression [58], [62], count regression [61], dimensionality reduction [4], [79], and elimination of multicollinearity [5], [8], [70], [71]. MIO-based high-performance algorithms have also been designed for subset selection [9], [10], [22], [35], [36], [44].

Several prior studies have dealt with feature subset selection in linear SVM classification. A typical approach involves approximating the L_0-regularization term (or the cardinality constraint) for subset selection by the concave exponential function [12], the L_1-regularization term [12], [80], [84], and convex relaxations [16], [26], and the L_0-regularization term can be handled more accurately by DC (difference of convex functions) algorithms [24], [46]. Meanwhile, Maldonado et al. [51] proposed exact MIO formulations for subset selection in linear SVM classification, and Labbé et al. [45] applied a heuristic kernel search algorithm to the MIO problem. Lagrangian relaxation [25] and generalized Benders decomposition [3] have been used to handle large-scale MIO problems. The MIO formulations were extended to robust cost-effective subset selection in linear SVM classification [47]. However, since these algorithms are focused on linear classification, they cannot be applied to nonlinear classification based on the kernel method.

The feature scaling approach has been studied intensively for feature subset selection in kernel SVM classification [18], [28], [50], [53], [81]. This approach introduces feature weights in a kernel function and updates them iteratively in the gradient descent direction. Other algorithms for subset selection in kernel SVM classification include the filter method based on local kernel gradients [37], local search algorithms [52], [54], [77], and metaheuristic algorithms [38].

Several performance measures for kernel SVM classifiers have been used in feature subset selection. Chapelle et al. [18] designed gradient descent methods for minimizing various performance bounds on generalization errors, Neumann et al. [59] proposed DC algorithms to maximize the kernel–target alignment [20], Wang [77] considered the kernel class separability in subset selection, and Jiménez-Cordero et al. [40] used optimization software to obtain good-quality solutions to their nonlinear min-max optimization problem. To our knowledge, however, no prior studies have developed an exact algorithm to compute the best subset of features in terms of a given performance measure for nonlinear kernel SVM classification.

1.3 Contribution

The goal of this paper is to establish a practicable MIO approach to selecting the best subset of features for nonlinear kernel SVM classification. In line with Neumann et al. [59], we use the kernel–target alignment [20], namely the distance between two classes (DBTC) [66] in a high-dimensional feature space, as an objective function for subset selection. The kernel–target alignment has many applications in various kernel-based machine learning algorithms [78].

First, we introduce an INLO formulation for feature subset selection to maximize DBTC based on the Gaussian kernel function. However, its objective function is nonlinear, nonconvex and nonconcave, and thus, it is very difficult to linearize this function in general. Our major contribution is to reformulate the problem as a MILO problem, which can be solved to optimality using optimization software. To our knowledge, we are the first to transform this subset selection problem into a MILO problem for exactly maximizing DBTC based on the Gaussian kernel function. Our additional contribution is to derive a reduced version of the MILO problem to accelerate our MILO computations.

We assess the efficacy of our method through computational experiments using real-world and synthetic datasets. With the real-world datasets, our MILO formulations produce clear computational advantages over the INLO formulation. In addition, the problem reduction offers highly accelerated MILO computations and allows us to find better quality solutions than does the DC algorithm [59]. With the synthetic datasets, our method often outperforms the linear-SVM-based MILO formulation [51] and recursive feature elimination [32] in terms of accuracy for both classification and subset selection, especially when there are relatively few data instances.

1.4 Organization and Notation

In Sect. 2, we introduce DBTC as a performance measure for nonlinear kernel SVM classification. In Sect. 3, we present our MIO formulations to maximize DBTC for feature subset selection. We report the computational results in Sect. 4 and conclude in Sect. 5. A list of abbreviations is given in Appendix.

Throughout this paper, we denote the set of consecutive integers ranging from 1 to n as $[n] := \{1, 2, \ldots, n\}$. We write a p-dimensional column vector as $\boldsymbol{x} := (x_j)_{j\in[p]} \in \mathbb{R}^p$, and an $m \times n$ matrix as $\boldsymbol{X} := (x_{ij})_{(i,j)\in[m]\times[n]} \in \mathbb{R}^{m\times n}$.

2. Distance between Two Classes

We address the task of nonlinear binary classification, which aims at learning a nonlinear function $f : \mathbb{R}^p \to \mathbb{R}$ correctly assigning a binary class label $\hat{y} \in \{+1, -1\}$ to each p-dimensional feature vector $\boldsymbol{x} := (x_j)_{j\in[p]} \in \mathbb{R}^p$ as

$$\begin{cases} f(\boldsymbol{x}) > 0 & \Rightarrow \quad \hat{y} = +1, \\ f(\boldsymbol{x}) < 0 & \Rightarrow \quad \hat{y} = -1. \end{cases}$$

To express such a nonlinear function in kernel SVM classification, we consider a feature map $\boldsymbol{\phi} : \mathbb{R}^p \to \mathcal{X}$, which nonlinearly transforms the original feature vector $\boldsymbol{x}$ into a high-dimensional feature vector $\boldsymbol{\phi}(\boldsymbol{x})$ in a feature space $\mathcal{X}$. A simple example with $\boldsymbol{x} = (x_1, x_2)^\top$ is given by

$$\boldsymbol{\phi}(\boldsymbol{x}) = (x_1, x_2, x_1^2, x_1x_2, x_2^2)^\top.$$

Suppose that we are given a training dataset $\{(\boldsymbol{x}_i, y_i) \mid i \in [n]\}$ containing n data instances, where $\boldsymbol{x}_i := (x_{ij})_{j\in[p]} \in \mathbb{R}^p$ and $y_i \in \{+1, -1\}$ for each $i \in [n]$. For each class $y \in \{+1, -1\}$, we denote the index set of data instances as

$$N(y) := \{i \in [n] \mid y_i = y\}.$$

We also define a vector of class labels divided by each class size as

$$\boldsymbol{\psi} := (\psi_i)_{i\in[n]} := \left(\frac{y_i}{|N(y_i)|}\right)_{i\in[n]} \in \mathbb{R}^n.$$

To measure the class separability in a high-dimensional feature space $\mathcal{X}$, we focus on DBTC [66] (or the kernel–target alignment [20], [59]) given by the squared Euclidean distance between the centroids of positive and negative classes in a feature space:

$$\begin{aligned}
&\left\| \frac{1}{|N(+1)|} \sum_{i\in N(+1)} \boldsymbol{\phi}(\boldsymbol{x}_i) - \frac{1}{|N(-1)|} \sum_{i\in N(-1)} \boldsymbol{\phi}(\boldsymbol{x}_i) \right\|_2^2 \\
&= \left\| \sum_{i=1}^n \psi_i \boldsymbol{\phi}(\boldsymbol{x}_i) \right\|_2^2 = \left(\sum_{i=1}^n \psi_i \boldsymbol{\phi}(\boldsymbol{x}_i) \right)^\top \left(\sum_{h=1}^n \psi_h \boldsymbol{\phi}(\boldsymbol{x}_h) \right) \\
&= \sum_{i=1}^n \sum_{h=1}^n \psi_i \psi_h k(\boldsymbol{x}_i, \boldsymbol{x}_h), \qquad (1)
\end{aligned}$$

where

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$$

is the kernel function, which is the inner product in a feature space. In Sect. 3, we use DBTC (1) as an objective function to be maximized for subset selection.

We consider the Gaussian kernel function defined as

$$\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{x}') &= \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\right) \\
&= \exp\left(-\gamma \sum_{j=1}^p (x_j - x'_j)^2\right), \qquad (2)
\end{aligned}$$

where $\gamma \in \mathbb{R}_+$ is a user-defined scaling parameter. It is known that the Gaussian kernel function (2) corresponds to the inner product in an infinite-dimensional feature space [64].

3. Mixed-Integer Optimization Formulations for Feature Subset Selection

In this section, we present our MIO formulations to maximize DBTC (1) for feature subset selection. We also apply some problem reduction techniques to our MIO formulations.

3.1 Integer Nonlinear Optimization Formulation

For subset selection, we assume that all features are standardized as

$$\sum_{i=1}^n x_{ij} = 0 \quad \text{and} \quad \frac{\sum_{i=1}^n x_{ij}^2}{n} = 1 \qquad (3)$$

for all $j \in [p]$.

Let $\boldsymbol{z} := (z_j)_{j\in[p]} \in \{0,1\}^p$ be a vector composed of binary decision variables for subset selection; that is, $z_j = 1$ if the jth feature is selected, and $z_j = 0$ otherwise. We then consider the following subset-based Gaussian kernel function:

$$k_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{x}') := \exp\left(-\gamma \sum_{j=1}^p z_j (x_j - x'_j)^2\right). \qquad (4)$$

To select the best subset of features for kernel SVM classification, we maximize DBTC (1) based on the subset-based Gaussian kernel function (4). This problem is formulated as the following INLO problem:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^n \sum_{h=1}^n \psi_i \psi_h \exp\left(-\gamma \sum_{j=1}^p z_j (x_{ij} - x_{hj})^2\right) && (5)\\
\text{subject to} \quad & \sum_{j=1}^p z_j \le \theta, && (6)\\
& \boldsymbol{z} \in \{0,1\}^p, && (7)
\end{aligned}$$

where $\theta \in [p]$ is a user-defined subset size parameter.

However, a globally optimal solution to problem (5)–(7) is very difficult to compute because its objective function (5) is nonlinear, nonconvex and nonconcave.

3.2 Mixed-Integer Linear Optimization Formulations

Theorem 1 states that the INLO problem (5)–(7) can be reformulated as the following MILO problem:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^n \sum_{h=1}^n \psi_i \psi_h e_{ihp} && (8)\\
\text{subject to} \quad & \sum_{j=1}^p z_j \le \theta, && (9)\\
& e_{ih0} = 1 \quad (i \in [n], h \in [n]), && (10)\\
& z_j = 0 \Rightarrow \ e_{ihj} = e_{ih,j-1} \\
& \qquad (i \in [n], h \in [n], j \in [p]), && (11)\\
& z_j = 1 \ \Rightarrow \\
& \qquad e_{ihj} = e_{ih,j-1} \cdot \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) \\
& \qquad (i \in [n], h \in [n], j \in [p]), && (12)\\
& \boldsymbol{e} \in \mathbb{R}_+^{n\times n\times(p+1)}, \ \boldsymbol{z} \in \{0,1\}^p, && (13)
\end{aligned}$$

where $\boldsymbol{e} := (e_{ihj})_{(i,h,j)\in[n]\times[n]\times(\{0\}\cup[p])} \in \mathbb{R}_+^{n\times n\times(p+1)}$ is an array of auxiliary nonnegative decision variables for calculating the objective function (5). Here, Eqs. (11) and (12) are logical implications, which can be imposed by using indicator constraints implemented in modern optimization

software.

Theorem 1: Let $(\boldsymbol{e}^*, \boldsymbol{z}^*)$ be an optimal solution to the MILO problem (8)–(13). Then, $\boldsymbol{z}^*$ is also an optimal solution to the INLO problem (5)–(7).

Proof : For each $\boldsymbol{z} \in \{0,1\}^p$, we can construct a feasible solution $(\boldsymbol{e}, \boldsymbol{z})$ to problem (8)–(13) by using Eqs. (11) and (12) recursively from Eq. (10). Therefore, the feasible region of $\boldsymbol{z}$ in problem (8)–(13) is the same as in problem (5)–(7).

Consequently, it is necessary to prove only that, in Eqs. (5) and (8),

$$e_{ihp} = \exp\left(-\gamma \sum_{j=1}^{p} z_j (x_{ij} - x_{hj})^2\right)$$

for all $(i,h) \in [n] \times [n]$. Note that

$$\exp\left(-\gamma z_j (x_{ij} - x_{hj})^2\right) = \begin{cases} \exp(0) = 1 & \text{if } z_j = 0, \\ \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) & \text{if } z_j = 1. \end{cases}$$

Therefore, the constraints (11) and (12) can be integrated into

$$e_{ihj} = e_{ih,j-1} \cdot \exp\left(-\gamma z_j (x_{ij} - x_{hj})^2\right)$$

for all $j \in [p]$. By substituting this equation recursively for $j \in [p]$, we obtain

$$\begin{aligned} e_{ihp} &= e_{ih0} \prod_{j=1}^{p} \exp\left(-\gamma z_j (x_{ij} - x_{hj})^2\right) \\ &= \exp\left(-\gamma \sum_{j=1}^{p} z_j (x_{ij} - x_{hj})^2\right), \quad \because \text{Eq. (10)} \end{aligned}$$

which completes the proof. □

Note also that problem (8)–(13) can be equivalently rewritten without logical implications using well-known formulation techniques [82] as

$$\text{maximize} \quad \sum_{i=1}^{n} \sum_{h=1}^{n} \psi_i \psi_h e_{ihp} \tag{14}$$

$$\text{subject to} \quad \sum_{j=1}^{p} z_j \le \theta, \tag{15}$$

$$e_{ih0} = 1 \quad (i \in [n], h \in [n]), \tag{16}$$

$$-M z_j \le e_{ihj} - e_{ih,j-1} \le M z_j \quad (i \in [n], h \in [n], j \in [p]), \tag{17}$$

$$-M \cdot (1 - z_j) \le e_{ihj} - e_{ih,j-1} \cdot \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) \le M \cdot (1 - z_j) \quad (i \in [n], h \in [n], j \in [p]), \tag{18}$$

$$\boldsymbol{e} \in \mathbb{R}_+^{n \times n \times (p+1)}, \ \boldsymbol{z} \in \{0,1\}^p, \tag{19}$$

where $M \in \mathbb{R}_+$ is a sufficiently large positive constant (e.g., $M = 1$ due to Eqs. (10)–(12)).

3.3 Problem Reduction

For problem reduction, we introduce the following index sets of instance pairs:

$$\begin{aligned} H &:= \{(i,h) \in [n] \times [n] \mid i < h\}, \\ H_+ &:= \{(i,h) \in [n] \times [n] \mid i < h, \ \psi_i \psi_h > 0\}, \\ H_- &:= \{(i,h) \in [n] \times [n] \mid i < h, \ \psi_i \psi_h < 0\}. \end{aligned}$$

Then, Theorem 2 proves that the MILO problem (14)–(19) can be reduced to the following MILO problem:

$$\text{maximize} \quad \sum_{(i,h) \in H} \psi_i \psi_h e_{ihp} \tag{20}$$

$$\text{subject to} \quad \sum_{j=1}^{p} z_j \le \theta, \tag{21}$$

$$e_{ih0} = 1 \quad ((i,h) \in H), \tag{22}$$

$$-M_{ihj} z_j \le e_{ihj} - e_{ih,j-1} \quad ((i,h) \in H_-, j \in [p]), \tag{23}$$

$$e_{ihj} - e_{ih,j-1} \le 0 \quad ((i,h) \in H_+, j \in [p]), \tag{24}$$

$$0 \le e_{ihj} - e_{ih,j-1} \cdot \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) \quad ((i,h) \in H_-, j \in [p]), \tag{25}$$

$$e_{ihj} - e_{ih,j-1} \cdot \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) \le M_{ihj} \cdot (1 - z_j) \quad ((i,h) \in H_+, j \in [p]), \tag{26}$$

$$\boldsymbol{e} \in \mathbb{R}_+^{|H| \times (p+1)}, \ \boldsymbol{z} \in \{0,1\}^p, \tag{27}$$

where

$$M_{ihj} := 1 - \exp\left(-\gamma (x_{ij} - x_{hj})^2\right) \quad (i \in [n], h \in [n], j \in [p]). \tag{28}$$

Theorem 2: Let $(\boldsymbol{e}^*, \boldsymbol{z}^*)$ be an optimal solution to the reduced MILO problem (20)–(27). Then, $\boldsymbol{z}^*$ is also an optimal solution to the INLO problem (5)–(7).

Proof : Because of Theorem 1, it is necessary to prove only that problem (14)–(19), which is equivalent to problem (8)–(13), can be reformulated as problem (20)–(27).

We begin by focusing on the objective function (14). Note that Eq. (5) can be decomposed as

$$\begin{aligned} &\sum_{i=1}^{n} \sum_{h=1}^{n} \psi_i \psi_h \exp\left(-\gamma \sum_{j=1}^{p} z_j (x_{ij} - x_{hj})^2\right) \\ &= \sum_{i=1}^{n} \psi_i^2 + 2 \sum_{(i,h) \in H} \psi_i \psi_h \exp\left(-\gamma \sum_{j=1}^{p} z_j (x_{ij} - x_{hj})^2\right). \end{aligned}$$

This implies that the objective function (14) can be replaced with Eq. (20). Accordingly, the unnecessary decision variables (i.e., e_{ihj} for $(i,h) \notin H$) and the corresponding subset of constraints (16)–(18) can be deleted from the problem.

Next, we consider constraints (17) and (18). It is clear from Eqs. (10)–(12) that

$$0 \le e_{ih,j-1} \cdot \exp\left(-\gamma(x_{ij} - x_{hj})^2\right) \le e_{ihj} \le e_{ih,j-1} \le 1.$$

Therefore, it follows that

$$\begin{aligned} -M_{ihj} &\le -e_{ih,j-1} \cdot \underbrace{\left(1 - \exp\left(-\gamma(x_{ij} - x_{hj})^2\right)\right)}_{M_{ihj}} \\ &\le e_{ihj} - e_{ih,j-1} \le 0, \\ 0 &\le e_{ihj} - e_{ih,j-1} \cdot \exp\left(-\gamma(x_{ij} - x_{hj})^2\right) \\ &\le e_{ih,j-1} \cdot \underbrace{\left(1 - \exp\left(-\gamma(x_{ij} - x_{hj})^2\right)\right)}_{M_{ihj}} \le M_{ihj}. \end{aligned}$$

This implies that the feasible region remains the same even if the constraints (17) and (18) are tightened as

$$-M_{ihj} z_j \le e_{ihj} - e_{ih,j-1} \le 0, \tag{29}$$

$$0 \le e_{ihj} - e_{ih,j-1} \cdot \exp\left(-\gamma(x_{ij} - x_{hj})^2\right) \le M_{ihj} \cdot (1 - z_j). \tag{30}$$

When $\psi_i \psi_h > 0$, the left inequalities in Eqs. (29) and (30) are redundant because e_{ihj} is maximized by the objective function (14). Similarly, when $\psi_i \psi_h < 0$, the right inequalities in Eqs. (29) and (30) are redundant. As a result, constraints (23)–(26) are obtained, thus completing the proof. □

We conclude this section by highlighting the differences between the MILO problem (14)–(19) and its reduced version (20)–(27). The number of continuous decision variables is reduced from $(p+1)n^2$ (Eq. (19)) to $(p+1)n(n-1)/2$ (Eq. (27)), and the number of inequality constraints is reduced from $4pn^2$ (Eqs. (17) and (18)) to $pn(n-1)$ (Eqs. (23)–(26)). Also, the big-M values are equal (e.g., $M = 1$) in Eqs. (17) and (18), whereas they are set to the smaller values (28) in Eqs. (23) and (26).

4. Computational Experiments

In this section, we report the results of computations to evaluate the efficacy of our method for feature subset selection in kernel SVM classification. First, we confirm the optimization performance of our MIO formulations using real-world datasets, and then we examine the prediction performance of our method for subset selection using synthetic datasets.

All computations were performed on a Windows computer with two Intel Xeon E5-2620v4 CPUs (2.10 GHz) and 128 GB of memory using a single thread.

Table 1 Real-world datasets.

Name	n	p	Original dataset [23]
Hepatitis	138	15	Hepatitis
Zoo	101	16	Zoo
Parkinsons	195	22	Parkinsons
Soybean	47	45	Soybean (Small)

4.1 Experimental Design for Real-World Datasets

We downloaded four real-world datasets for classification tasks from the UCI Machine Learning Repository [23]. Table 1 lists the datasets, where n and p are the numbers of data instances and candidate features, respectively. Categorical variables with two categories were treated as dummy variables, and those with more than two categories were transformed into sets of dummy variables. In the Zoo and Parkinsons datasets, the names of data instances were deleted. In the Hepatitis dataset, we removed four variables containing more than 10 missing values, and then data instances containing missing values. In the Soybean dataset, variables with the same value in all data instances were eliminated. The Zoo and Soybean datasets have multiple response classes, so the positive label (i.e., $y_i = +1$) was given to classes 1 and 2 in the Zoo dataset and to classes D1 and D4 in the Soybean dataset, and the negative label (i.e., $y_i = -1$) was given to the other classes.

Although it is difficult to apply our method to large-sized datasets, our method will be effective in small-sized datasets, for instance, for medical applications. The Parkinsons dataset is composed of a range of biomedical voice measurements; each column is a particular voice measure, each row corresponds to one of 195 voice recordings from subjects, and the main aim of this dataset is to discriminate healthy people from those with Parkinson's disease [23]. In this situation, highly accurate discrimination is required from a small number of measurements of a small number of subjects, and thus, our method is expected to work well for such medical applications.

We compare the optimization performance of the following methods for maximizing DBTC through feature subset selection for kernel SVM classification:

INLO-K: INLO formulation (5)–(7);

MILO-K: MILO formulation (14)–(19) with $M = 1$;

RMILO-K: reduced MILO formulation (20)–(27);

DCA-K: DC algorithm [59].

The MILO problems were solved using the optimization software IBM ILOG CPLEX 20.1.0.0 [39], where algorithms for solving relaxed subproblems on each node were set to the interior-point method instead of the dual simplex method. To increase numerical stability, the big-M values for RMILO-K were set as

$$M_{ihj} = \min\{1 - \exp\left(-\gamma(x_{ij} - x_{hj})^2\right) + 0.1, 1.0\}$$

$$(i \in [n], h \in [n], j \in [p]). \tag{31}$$

The INLO problems, which cannot be handled by CPLEX because of nonlinearity, were solved by the optimization software Gurobi Optimizer 9.5.0 [29] using the general constraint EXP function. The DC algorithm [59] was implemented in the Python programming language with the optimization software Ipopt 3.14 [75]. Here, the initial solution was set as $z_j^0 := 0.5$ for $j \in [p]$, and the algorithm was terminated at the kth iteration when $|z_j^{k+1} - z_j^k| \leq 10^{-3}$ for all $j \in [p]$. The weight $\lambda \in \{0, 0.1, \ldots, 0.9\}$ of the penalty term was tuned such that the objective function (5) would be maximized subject to the subset size constraint (6).

Many algorithms have been proposed for tuning SVM hyperparameters [76]. Based on the sigest method [14], we estimated an appropriate value of the scaling parameter γ in the subset-based Gaussian kernel function (4) as follows:

$$\hat{\gamma} := \frac{1}{\text{median of } \left\{(\theta/p) \cdot \sum_{j=1}^{p} (x_{ij} - x_{hj})^2 \mid (i,h) \in H\right\}}. \tag{32}$$

We then set $\gamma = \beta\hat{\gamma}$ with the scaling factor $\beta \in \{0.25, 1.00, 4.00\}$.

The following column labels are used in Tables 2–9:

ObjVal: value of the objective function (5);

OptGap: absolute difference between lower and upper bounds on the optimal objective value divided by the lower bound;

$|\hat{S}|$: subset size of selected features;

Time: computation time in seconds.

A computation was terminated if it did not complete within 10000 s; in those cases, the best feasible solution found within 10000 s was taken as the result. The total computation time required for tuning the weight parameter $\lambda \in \{0, 0.1, \ldots, 0.9\}$ was measured in the DC algorithm.

4.2 Results for Real-World Datasets

Tables 2–5 give the computational results of the four methods for maximizing DBTC for the real-world datasets. First, we focus on the results for the Hepatitis dataset (Table 2). The INLO formulation (INLO-K) always reached the time limit of 10000 s, and therefore its ObjVal values were often very small. In contrast, our reduced MILO formulation (RMILO-K) solved all the problem instances completely within the time limit. Moreover, RMILO-K finished computations much sooner than did the original MILO formulation (MILO-K); for example, the computation times of MILO-K and RMILO-K for $(\theta, \beta) = (3, 0.25)$ were 9498.3 s and 716.6 s, respectively. The DC algorithm (DCA-K) was faster than RMILO-K, whereas better ObjVal values were often attained by RMILO-K.

Table 2 Results for Hepatitis dataset: $(n, p) = (138, 15)$.

θ	β	Method	ObjVal	OptGap	$\lvert\hat{S}\rvert$	Time
3	0.25	INLO-K	0.180	151.1%	2	>10000.0
		MILO-K	0.236	0.0%	3	9498.3
		RMILO-K	0.236	0.0%	3	716.6
		DCA-K	0.180	—	2	110.0
	1.00	INLO-K	0.389	190.6%	2	>10000.0
		MILO-K	0.389	171.6%	2	>10000.0
		RMILO-K	0.411	0.0%	3	2168.6
		DCA-K	0.410	—	3	271.8
	4.00	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.489	0.0%	2	7127.1
		RMILO-K	0.489	0.0%	2	1872.9
		DCA-K	0.458	—	1	971.4
5	0.25	INLO-K	0.000	624.8%	0	>10000.0
		MILO-K	0.215	0.0%	5	7787.1
		RMILO-K	0.215	0.0%	5	760.5
		DCA-K	0.200	—	4	107.3
	1.00	INLO-K	0.344	165.7%	4	>10000.0
		MILO-K	0.360	229.3%	3	>10000.0
		RMILO-K	0.392	0.0%	4	5135.0
		DCA-K	0.392	—	4	246.6
	4.00	INLO-K	0.406	278.8%	1	>10000.0
		MILO-K	0.463	162.7%	2	>10000.0
		RMILO-K	0.463	0.0%	2	2476.5
		DCA-K	0.463	—	2	1000.7

Table 3 Results for Zoo dataset: $(n, p) = (101, 16)$.

θ	β	Method	ObjVal	OptGap	$\lvert\hat{S}\rvert$	Time
3	0.25	INLO-K	0.303	0.0%	3	1624.7
		MILO-K	0.303	0.0%	3	342.7
		RMILO-K	0.303	0.0%	3	55.5
		DCA-K	0.229	—	2	49.3
	1.00	INLO-K	0.916	0.0%	3	1690.1
		MILO-K	0.916	0.0%	3	388.4
		RMILO-K	0.916	0.0%	3	81.5
		DCA-K	0.916	—	3	79.4
	4.00	INLO-K	1.445	0.0%	2	9489.9
		MILO-K	1.445	0.0%	2	315.3
		RMILO-K	1.445	0.0%	2	29.7
		DCA-K	1.445	—	2	118.2
5	0.25	INLO-K	0.278	1.6%	5	>10000.0
		MILO-K	0.278	0.0%	5	379.1
		RMILO-K	0.278	0.0%	5	89.1
		DCA-K	0.245	—	4	48.4
	1.00	INLO-K	0.657	32.5%	5	>10000.0
		MILO-K	0.726	0.0%	5	852.9
		RMILO-K	0.726	0.0%	5	291.2
		DCA-K	0.726	—	5	155.9
	4.00	INLO-K	1.333	8.7%	3	>10000.0
		MILO-K	1.333	0.0%	3	503.3
		RMILO-K	1.333	0.0%	3	62.1
		DCA-K	1.333	—	3	60.6

For the Zoo dataset (Table 3), RMILO-K was still much faster than the other MIO formulations, whereas the differences in ObjVal among the three MIO formulations were relatively small. For the Parkinsons dataset (Table 4), al-

Table 4 Results for Parkinsons dataset: $(n, p) = (195, 22)$.

θ	β	Method	ObjVal	OptGap	$\lvert\hat{S}\rvert$	Time
3	0.25	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.000	>1000.0%	0	>10000.0
		RMILO-K	0.284	81.2%	3	>10000.0
		DCA-K	0.272	—	2	1175.1
	1.00	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.000	>1000.0%	0	>10000.0
		RMILO-K	0.316	206.3%	2	>10000.0
		DCA-K	0.444	—	2	1358.3
	4.00	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.154	>1000.0%	3	>10000.0
		RMILO-K	0.000	>1000.0%	0	>10000.0
		DCA-K	0.457	—	1	3943.2
5	0.25	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.127	965.5%	3	>10000.0
		RMILO-K	0.251	88.6%	5	>10000.0
		DCA-K	0.261	—	5	645.5
	1.00	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.160	>1000.0%	5	>10000.0
		RMILO-K	0.276	268.9%	2	>10000.0
		DCA-K	0.400	—	3	1630.2
	4.00	INLO-K	0.000	>1000.0%	0	>10000.0
		MILO-K	0.158	>1000.0%	5	>10000.0
		RMILO-K	0.000	>1000.0%	0	>10000.0
		DCA-K	0.459	—	1	1703.2

Table 5 Results for Soybean dataset: $(n, p) = (47, 45)$.

θ	β	Method	ObjVal	OptGap	$\lvert\hat{S}\rvert$	Time
3	0.25	INLO-K	0.316	20.1%	3	>10000.0
		MILO-K	0.316	22.7%	3	>10000.0
		RMILO-K	0.316	0.0%	3	346.0
		DCA-K	0.091	—	1	78.8
	1.00	INLO-K	0.137	>1000.0%	3	>10000.0
		MILO-K	0.926	0.0%	3	6669.9
		RMILO-K	0.926	0.0%	3	618.2
		DCA-K	0.762	—	3	98.4
	4.00	INLO-K	1.419	26.3%	3	>10000.0
		MILO-K	1.451	0.0%	3	7055.2
		RMILO-K	1.451	0.0%	3	144.0
		DCA-K	1.451	—	3	135.6
5	0.25	INLO-K	0.300	15.1%	5	>10000.0
		MILO-K	0.267	547.6%	5	>10000.0
		RMILO-K	0.300	0.0%	5	2302.6
		DCA-K	0.127	—	2	56.6
	1.00	INLO-K	0.711	60.6%	5	>10000.0
		MILO-K	0.737	167.1%	5	>10000.0
		RMILO-K	0.870	0.0%	5	5054.1
		DCA-K	0.698	—	4	56.8
	4.00	INLO-K	0.966	84.8%	4	>10000.0
		MILO-K	1.391	20.9%	4	>10000.0
		RMILO-K	1.391	0.0%	4	414.7
		DCA-K	1.302	—	4	162.3

though the three MIO formulations failed to finish computations within the time limit, RMILO-K attained the largest ObjVal and smallest OptGap values in these formulations for $\beta \in \{0.25, 1.00\}$. For the Soybean dataset (Table 5), INLO-K and MILO-K often reached the time limit, whereas RMILO-K solved all the problem instances to optimality. DCA-K was faster than the MIO formulations for these three datasets, and its ObjVal values were often the best for the Parkinsons dataset; however, DCA-K failed to give better ObjVal values than did RMILO-K for the Zoo and Soybean datasets. These results show that our MILO formulations produce clear computational advantages over the INLO formulation, and the problem reduction offers highly accelerated MILO computations and thus good-quality solutions within the time limit.

Next, we examine how the two user-defined parameters (θ, β) affected the MILO computations. The computation time of RMILO-K was longer when θ was large; this is reasonable because the number of feasible subsets of features increases with θ. Also, the computation time of RMILO-K was often longest with $\beta = 1.00$; this implies that solving MILO problems is computationally expensive when the scaling parameter γ is tuned appropriately by Eq. (32).

4.3 Experimental Design for Synthetic Datasets

We prepared synthetic datasets based on the MADELON dataset [31] from the NIPS 2003 Feature Selection Challenge. Specifically, we supposed that there were θ^* relevant features and $p - \theta^*$ irrelevant features. The relevant features were generated using the NDCC (normally distributed clusters on cubes) data generator [72], which is designed to create datasets for nonlinear binary classification. The expansion factor "exp" is used in the NDCC data generator to stretch the covariance matrix of multivariate normal distributions; as the expansion factor increases, two classes overlap and become difficult to discriminate. The irrelevant features were drawn randomly from the standard normal distribution. All these features were standardized as in Eq. (3).

We used n data instances as a training dataset for each combination $(n, p, \exp, \theta^*)$ of parameter values. For this training dataset, we selected a subset $\hat{S}$ of features. The accuracy of subset selection is measured by the F1 score, which is the harmonic average of Recall $:= |S^* \cap \hat{S}|/|S^*|$ and Precision $:= |S^* \cap \hat{S}|/|\hat{S}|$ as follows:

$$\textbf{SetF1} := \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}},$$

where S^* is the set of relevant features.

By means of the training dataset, we trained SVM classifiers with the selected subset $\hat{S}$ of features. We then evaluated the prediction performance by applying the trained classifier to a testing dataset consisting of sufficiently many data instances. Let $\hat{y}_i(\hat{S})$ be the class label predicted for the ith data instance. The classification accuracy for the testing dataset is calculated as

$$\textbf{ClsAcc} := \frac{|\{i \in \tilde{N} \mid y_i = \hat{y}_i(\hat{S})\}|}{|\tilde{N}|},$$

where $\tilde{N}$ is the index set of testing data instances. We re-

peated this process 10 times and give average values in Tables 6–9.

We compare the prediction performance of the following methods for feature subset selection:

MILO-L: MILO formulation (MILP2 [51]) for linear SVM classification;

RFE-K: recursive feature elimination [32] for kernel SVM classification;

RMILO-K: our reduced MILO formulation (20)–(27) for kernel SVM classification.

The MILO formulation (denoted by MILP2 in Maldonado et al. [51]) was proposed for subset selection in linear SVM classification. The recursive feature elimination was implemented using the `caret` package in the R programming language. The MILO problems were solved using the optimization software IBM ILOG CPLEX 20.1.0.0 [39], where the interior-point method was used to solve relaxed subproblems. The big-M values for RMILO-K were set as in Eq. (31). For a selected subset $\hat{S}$ of features, SVM classifiers were trained using the `sklearn.svm.LinearSVC` function (MILO-L) and the `sklearn.svm.SVC` function (RFE-K and RMILO-K) in the Python programming language. We set the misclassification penalty parameter as $C = 1$, which performed well for our synthetic datasets. We also used the scaling parameter $\gamma = \hat{\gamma}$, which was tuned by Eq. (32) for the subset-based Gaussian kernel function (4).

4.4 Results for Synthetic Datasets

Tables 6–9 show the computational results of the three methods for feature subset selection for the synthetic datasets. Recall that the tables show average values over 10 repetitions, with standard errors of the ClsAcc and SetF1 values in parentheses, where the best ClsAcc and SetF1 values for each problem instance $(n, p, \exp, \theta^*)$ are given in bold. Note also that where the tables show ">10000.0" in the column labeled "Time," the computation reached the time limit of 10000 s at least once out of 10 repetitions.

Table 6 gives the results for the expansion factor exp = 25 and the subset size $\theta = \theta^* = 3$. When $n \in \{25, 50\}$, our kernel-based MILO method (RMILO-K) achieved good accuracy for both classification (ClsAcc) and subset selection (SetF1). When $n = 100$, the kernel-based recursive feature elimination (RFE-K) performed relatively well. On the whole, the linear-SVM-based MILO method (MILO-L) performed badly.

Table 7 gives the results for the expansion factor exp = 25 and the subset size $\theta = \theta^* = 5$. When $n = 25$, MILO-L provided the best accuracy for both classification and subset selection. When $n = 50$, RMILO-K maintained good accuracy for both classification and subset selection. When $n = 100$, RFE-K and MILO-L had the best accuracy for classification and subset selection, respectively. However, note that RMILO-K selected nearly half the features selected by MILO-L. Accordingly, it is also the case that RMILO-K delivered overall good performance with relatively few features.

Table 8 gives the results for the expansion factor exp = 100 and the subset size $\theta = \theta^* = 3$. In this case, MILO-L outperformed the other kernel-based methods in terms of the classification accuracy. In other words, this dataset was compatible with linear SVM classifiers. On the other hand, the accuracy for classification and subset selection was higher for RMILO-K than for RFE-K overall.

Table 6 Results for the synthetic dataset (exp = 25 and $\theta = \theta^* = 3$).

n	p	Method	ClsAcc	SetF1	OptGap	$\lvert\hat{S}\rvert$	Time
25	10	MILO-L	0.810 (±0.023)	0.633 (±0.060)	0.0%	3.0	0.1
		RFE-K	0.773 (±0.051)	0.583 (±0.051)	—	2.1	12.1
		RMILO-K	**0.895** (±0.020)	**0.793** (±0.029)	0.0%	2.3	10.3
	20	MILO-L	0.773 (±0.022)	0.467 (±0.054)	0.0%	3.0	0.2
		RFE-K	0.737 (±0.050)	0.573 (±0.047)	—	1.8	12.3
		RMILO-K	**0.874** (±0.021)	**0.727** (±0.049)	0.0%	2.5	155.1
	30	MILO-L	0.766 (±0.022)	0.467 (±0.054)	0.0%	3.0	0.4
		RFE-K	0.715 (±0.045)	0.560 (±0.047)	—	1.6	12.3
		RMILO-K	**0.859** (±0.024)	**0.687** (±0.067)	0.0%	2.6	826.1
50	10	MILO-L	0.867 (±0.007)	**0.833** (±0.056)	0.0%	3.0	0.1
		RFE-K	0.898 (±0.019)	0.667 (±0.055)	—	2.3	13.0
		RMILO-K	**0.935** (±0.005)	0.820 (±0.020)	0.0%	2.1	46.8
	20	MILO-L	0.849 (±0.010)	0.700 (±0.078)	0.0%	3.0	0.2
		RFE-K	0.905 (±0.014)	0.690 (±0.052)	—	2.1	13.2
		RMILO-K	**0.931** (±0.004)	**0.807** (±0.025)	0.0%	2.2	785.2
	30	MILO-L	0.837 (±0.011)	0.633 (±0.078)	0.0%	3.0	0.4
		RFE-K	0.849 (±0.036)	0.647 (±0.062)	—	1.8	13.4
		RMILO-K	**0.935** (±0.006)	**0.827** (±0.032)	5.3%	2.3	>10000.0
100	10	MILO-L	0.866 (±0.006)	0.833 (±0.056)	0.0%	3.0	0.1
		RFE-K	**0.944** (±0.009)	**0.880** (±0.033)	—	2.4	13.0
		RMILO-K	0.934 (±0.010)	0.810 (±0.043)	0.0%	2.1	378.4
	20	MILO-L	0.865 (±0.006)	**0.833** (±0.056)	0.0%	3.0	0.4
		RFE-K	0.922 (±0.018)	0.830 (±0.047)	—	2.2	13.2
		RMILO-K	**0.933** (±0.010)	0.810 (±0.043)	0.0%	2.1	4660.3
	30	MILO-L	0.866 (±0.006)	**0.833** (±0.056)	0.0%	3.0	0.4
		RFE-K	**0.922** (±0.018)	0.830 (±0.047)	—	2.2	13.4
		RMILO-K	0.906 (±0.011)	0.650 (±0.050)	130.9%	1.5	>10000.0

Table 7 Results for the synthetic dataset (exp = 25 and $\theta = \theta^* = 5$).

n	p	Method	ClsAcc	SetF1	OptGap	$\lvert\hat{S}\rvert$	Time
25	10	MILO-L	**0.865** (±0.013)	**0.660** (±0.060)	0.0%	5.0	0.0
		RFE-K	0.843 (±0.009)	0.451 (±0.054)	—	1.8	17.6
		RMILO-K	0.848 (±0.012)	0.632 (±0.038)	0.0%	2.5	13.2
	20	MILO-L	**0.853** (±0.018)	**0.600** (±0.060)	0.0%	5.0	0.1
		RFE-K	0.831 (±0.009)	0.415 (±0.032)	—	2.1	17.7
		RMILO-K	0.833 (±0.014)	0.584 (±0.053)	0.0%	2.8	581.2
	30	MILO-L	**0.852** (±0.020)	**0.640** (±0.050)	0.0%	5.0	0.3
		RFE-K	0.839 (±0.009)	0.385 (±0.028)	—	1.6	17.8
		RMILO-K	0.829 (±0.014)	0.577 (±0.054)	0.0%	2.9	3322.4
50	10	MILO-L	0.872 (±0.008)	**0.720** (±0.044)	0.0%	5.0	<0.1
		RFE-K	**0.885** (±0.008)	0.618 (±0.069)	—	2.7	18.9
		RMILO-K	0.875 (±0.011)	0.661 (±0.030)	0.0%	2.5	72.3
	20	MILO-L	0.854 (±0.008)	0.640 (±0.040)	0.0%	5.0	0.2
		RFE-K	0.860 (±0.007)	0.466 (±0.048)	—	2.0	19.1
		RMILO-K	**0.873** (±0.011)	**0.661** (±0.030)	0.0%	2.5	3472.5
	30	MILO-L	0.849 (±0.007)	0.600 (±0.030)	0.0%	5.0	0.7
		RFE-K	0.860 (±0.007)	0.466 (±0.048)	—	2.0	19.3
		RMILO-K	**0.871** (±0.011)	**0.643** (±0.029)	74.6%	2.4	>10000.0
100	10	MILO-L	0.892 (±0.004)	**0.880** (±0.033)	0.0%	5.0	0.1
		RFE-K	**0.896** (±0.010)	0.643 (±0.062)	—	2.8	18.9
		RMILO-K	0.886 (±0.010)	0.643 (±0.029)	0.0%	2.4	524.5
	20	MILO-L	0.880 (±0.006)	**0.780** (±0.047)	0.0%	5.0	0.4
		RFE-K	**0.890** (±0.010)	0.557 (±0.063)	—	2.2	19.1
		RMILO-K	0.879 (±0.009)	0.625 (±0.027)	72.0%	2.3	>10000.0
	30	MILO-L	0.872 (±0.007)	**0.640** (±0.027)	0.0%	5.0	1.2
		RFE-K	**0.887** (±0.009)	0.538 (±0.053)	—	2.9	19.3
		RMILO-K	0.864 (±0.003)	0.540 (±0.024)	187.5%	2.0	>10000.0

Table 8 Results for the synthetic dataset (exp = 100 and $\theta = \theta^* = 3$).

n	p	Method	ClsAcc	SetF1	OptGap	$\lvert\hat{S}\rvert$	Time
25	10	MILO-L	**0.777** (±0.020)	0.800 (±0.074)	0.0%	3.0	0.1
		RFE-K	0.645 (±0.029)	0.520 (±0.085)	—	1.8	12.1
		RMILO-K	0.718 (±0.024)	**0.813** (±0.052)	0.0%	2.9	14.3
	20	MILO-L	**0.748** (±0.021)	**0.700** (±0.060)	0.0%	3.0	0.2
		RFE-K	0.611 (±0.025)	0.440 (±0.086)	—	1.7	11.7
		RMILO-K	0.677 (±0.037)	0.680 (±0.100)	0.0%	2.9	280.5
	30	MILO-L	**0.708** (±0.029)	**0.567** (±0.051)	0.0%	3.0	0.4
		RFE-K	0.607 (±0.025)	0.410 (±0.077)	—	1.3	11.8
		RMILO-K	0.645 (±0.043)	0.533 (±0.113)	0.0%	3.0	2027.2
50	10	MILO-L	**0.800** (±0.012)	0.833 (±0.056)	0.0%	3.0	0.1
		RFE-K	0.720 (±0.019)	0.687 (±0.044)	—	2.6	13.1
		RMILO-K	0.757 (±0.025)	**0.853** (±0.066)	0.0%	2.6	88.3
	20	MILO-L	**0.774** (±0.023)	0.767 (±0.071)	0.0%	3.0	0.2
		RFE-K	0.707 (±0.022)	0.640 (±0.055)	—	2.7	13.3
		RMILO-K	0.746 (±0.025)	**0.820** (±0.066)	0.0%	2.6	1807.0
	30	MILO-L	**0.757** (±0.021)	**0.700** (±0.060)	0.0%	3.0	0.4
		RFE-K	0.699 (±0.023)	0.637 (±0.059)	—	2.4	13.5
		RMILO-K	0.699 (±0.020)	0.653 (±0.057)	226.9%	2.6	>10000.0
100	10	MILO-L	**0.816** (±0.008)	0.867 (±0.054)	0.0%	3.0	0.1
		RFE-K	0.793 (±0.009)	0.820 (±0.043)	—	2.6	13.1
		RMILO-K	0.808 (±0.006)	**0.920** (±0.033)	0.0%	2.6	645.2
	20	MILO-L	**0.810** (±0.008)	0.833 (±0.056)	0.0%	3.0	0.5
		RFE-K	0.785 (±0.008)	0.780 (±0.032)	—	2.4	13.3
		RMILO-K	0.802 (±0.008)	**0.887** (±0.040)	134.8%	2.6	>10000.0
	30	MILO-L	**0.807** (±0.008)	**0.800** (±0.054)	0.0%	3.0	0.3
		RFE-K	0.774 (±0.008)	0.733 (±0.022)	—	2.5	13.5
		RMILO-K	0.726 (±0.028)	0.687 (±0.072)	709.0%	2.1	>10000.0

Table 9 Results for the synthetic dataset (exp = 100 and $\theta = \theta^* = 5$).

n	p	Method	ClsAcc	SetF1	OptGap	$\lvert\hat{S}\rvert$	Time
25	10	MILO-L	0.766 (±0.012)	0.560 (±0.040)	0.0%	5.0	0.1
		RFE-K	0.780 (±0.013)	0.508 (±0.053)	—	2.6	17.8
		RMILO-K	**0.793** (±0.017)	**0.565** (±0.039)	0.0%	2.7	19.7
	20	MILO-L	0.709 (±0.023)	0.400 (±0.042)	0.0%	5.0	0.2
		RFE-K	0.733 (±0.031)	0.349 (±0.050)	—	1.7	18.8
		RMILO-K	**0.764** (±0.019)	**0.505** (±0.054)	0.0%	3.6	2484.8
	30	MILO-L	0.670 (±0.029)	0.320 (±0.044)	0.0%	5.0	0.3
		RFE-K	0.725 (±0.023)	0.375 (±0.039)	—	2.5	18.3
		RMILO-K	**0.750** (±0.024)	**0.458** (±0.044)	44.5%	3.7	>10000.0
50	10	MILO-L	0.793 (±0.008)	**0.700** (±0.061)	0.0%	5.0	<0.1
		RFE-K	0.809 (±0.007)	0.535 (±0.047)	—	3.0	19.2
		RMILO-K	**0.822** (±0.005)	0.582 (±0.020)	0.0%	2.2	138.7
	20	MILO-L	0.780 (±0.007)	**0.620** (±0.070)	0.0%	5.0	0.2
		RFE-K	0.805 (±0.008)	0.498 (±0.040)	—	2.6	19.4
		RMILO-K	**0.820** (±0.006)	0.575 (±0.022)	178.3%	2.3	>10000.0
	30	MILO-L	0.770 (±0.008)	0.520 (±0.053)	0.0%	5.0	1.1
		RFE-K	0.810 (±0.009)	0.513 (±0.038)	—	2.3	19.3
		RMILO-K	**0.822** (±0.005)	**0.557** (±0.010)	295.3%	2.2	>10000.0
100	10	MILO-L	0.808 (±0.004)	**0.740** (±0.043)	0.0%	5.0	0.1
		RFE-K	**0.836** (±0.009)	0.639 (±0.049)	—	3.0	19.2
		RMILO-K	0.827 (±0.007)	0.571 (±0.000)	0.0%	2.0	708.6
	20	MILO-L	0.808 (±0.005)	**0.720** (±0.053)	0.0%	5.0	0.6
		RFE-K	**0.836** (±0.007)	0.552 (±0.028)	—	3.0	19.5
		RMILO-K	0.826 (±0.007)	0.571 (±0.000)	195.2%	2.0	>10000.0
	30	MILO-L	0.804 (±0.005)	0.540 (±0.052)	0.0%	5.0	0.9
		RFE-K	**0.830** (±0.007)	0.518 (±0.030)	—	3.6	19.3
		RMILO-K	0.826 (±0.007)	**0.571** (±0.000)	336.5%	2.0	>10000.0

Table 9 gives the results for the expansion factor exp = 100 and the subset size $\theta = \theta^* = 5$. When $n = 25$, RMILO-K achieved the best accuracy for both classification and subset selection. In addition, RMILO-K and RFE-K attained good classification accuracy when $n = 50$ and $n = 100$, respectively. As for the subset selection accuracy when $n \in \{50, 100\}$, although MILO-L had the overall best performance, RMILO-K with fewer features outperformed RFE-K on the whole.

These results show that our MILO formulation delivers good prediction performance, especially when there are relatively few data instances. One of the main reasons for this is that DBTC is a performance measure that is robust against small datasets. Also, our MILO formulation can outperform recursive feature elimination in terms of the subset selection accuracy.

5. Conclusion

This paper dealt with feature subset selection for nonlinear kernel SVM classification. First, we introduced the INLO formulation for computing the best subset of features based on DBTC, which is the distance between the centroids of two response classes in a high-dimensional feature space. Next, we reformulated the problem as a MILO problem and then devised some problem reduction techniques to solve the problem more efficiently.

In computational experiments conducted using real-world and synthetic datasets, our MILO problems were solved in much shorter times than was the original INLO problem, and the computational efficiency was improved by our reduced MILO formulation. Our method often found better quality solutions than did the DC algorithm [59]. Moreover, our method often attained better classification accuracy than did the linear-SVM-based MILO formulation [51] and recursive feature elimination [32], especially when there were relatively few data instances.

It is known that feature subset selection for maximizing DBTC (i.e., the kernel–target alignment) leads to nonconvex optimization [59]. To our knowledge, we are the first to transform this subset selection problem into a MILO problem, which can be solved to optimality using optimization software. Note that if we try to solve the original nonconvex optimization problem exactly, then we cannot avoid numerical errors caused by its nonlinear objective function. In contrast, our method offers globally optimal solutions to small-sized problems without such numerical errors, and the obtained optimal solutions can be used to evaluate the solution quality of other algorithms. We also expect our formulation techniques to be applicable to other nonconvex optimization problems whose structures are similar to that of our problem.

A limitation of our method is the computational inefficiency of dealing with large-sized datasets. In fact, our method often failed to complete MILO computations within 10000 s for large-sized datasets in the computational experiments. Another limitation is that our method is specialized for maximizing the performance measure DBTC based on the Gaussian kernel function. Additionally, the robustness of our method against outliers should be tested. In view of these limitations, we can provide the following future research directions:

- accelerating the MILO computation by implementing a specialized branch-and-bound algorithm;
- developing a heuristic algorithm for finding high-quality solutions efficiently for large-sized datasets;

- devising tractable MIO formulations for other performance measures or other kernel functions;
- evaluating the prediction performance of our method for datasets containing outliers.

Acknowledgments

The authors would like to thank Yoshitsugu Yamamoto for giving valuable comments on our manuscript. This work was partially supported by JSPS KAKENHI Grant Numbers JP21K04526 and JP21K04527.

References

[1] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer, "Theoretical foundations of potential function method in pattern recognition," Automation and Remote Control, vol.25, no.6, pp.917–936 1964.
[2] T.S. Arthanari and Y. Dodge, Mathematical Programming in Statistics, Wiley, 1981.
[3] H. Aytug, "Feature selection for support vector machines using generalized Benders decomposition," European Journal of Operational Research, vol.244, no.1, pp.210–218, 2015.
[4] L. Berk and D. Bertsimas, "Certifiably optimal sparse principal component analysis," Math. Prog. Comp., vol.11, no.3, pp.381–420, 2019.
[5] D. Bertsimas and A. King, "An algorithmic approach to linear regression,"Operations Research, vol.64, no.1, pp.2–16, 2016.
[6] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," Ann. Statist., vol.44, no.2, pp.813–852, 2016.
[7] D. Bertsimas and A. King, "Logistic regression: From art to science," Statist. Sci., vol.32, no.3, pp.367–384, 2017.
[8] D. Bertsimas and M.L. Li, "Scalable holistic linear regression," Operations Research Letters, vol.48, no.3, pp.203–208, 2020.
[9] D. Bertsimas, J. Pauphilet, and B. Van Parys, "Sparse regression: Scalable algorithms and empirical performance," Statist. Sci., vol.35, no.4, pp.555–578, 2020.
[10] D. Bertsimas, J. Pauphilet, and B. Van Parys, "Sparse classification: A scalable discrete optimization perspective," Mach. Learn., vol.110, no.11, pp.3177–3209, 2021.
[11] B.E. Boser, I.M. Guyon, and V.N. Vapnik, "A training algorithm for optimal margin classifiers," Proc. Fifth Annual Workshop on Computational Learning Theory, pp.144–152, July 1992.
[12] P.S. Bradley and O.L. Mangasarian, "Feature selection via concave minimization and support vector machines," Proc. Fifteenth International Conference on Machine Learning, pp.82–90, July 1998.
[13] B. Cao, D. Shen, J.T. Sun, Q. Yang, and Z. Chen, "Feature selection in a kernel space," Proc. 24th International Conference on Machine Learning, pp.121–128, June 2007.
[14] B. Caputo, K. Sim, F. Furesjo, and A. Smola, "Appearance-based object recognition using SVMs: Which kernel should I use?," Proc. NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, Whistler, vol.2002, Dec. 2002.
[15] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol.408, pp.189–215, 2020.
[16] A.B. Chan, N. Vasconcelos, and G.R. Lanckriet, "Direct convex relaxations of sparse SVM," Proc. 24th International Conference on Machine Learning, pp.145–153, June 2007.
[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers & Electrical Engineering, vol.40, no.1, pp.16–28, 2014.
[18] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," Machine Learning, vol.46, no.1, pp.131–159, 2002.
[19] A. Cozad, N.V. Sahinidis, and D.C. Miller, "Learning surrogate models for simulation-based optimization," AIChE J., vol.60, no.6, pp.2211–2227, 2014.
[20] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel target alignment," Innovations in Machine Learning, pp.205–256, Springer, Berlin, Heidelberg, 2006.
[21] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
[22] A. Dedieu, H. Hazimeh, and R. Mazumder, "Learning sparse classifiers: Continuous and mixed integer optimization perspectives," Journal of Machine Learning Research, vol.22, no.135, pp.1–47, 2021.
[23] D. Dua and C. Graff, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, University of California, School of Information and Computer Science, Irvine, CA, 2019.
[24] M. Gaudioso, E. Gorgone, and J.B. Hiriart-Urruty, "Feature selection in SVM via polyhedral k-norm," Optim. Lett., vol.14, no.1, pp.19–36, 2020.
[25] M. Gaudioso, E. Gorgone, M. Labbé, and A.M. Rodríguez-Chía, "Lagrangian relaxation for SVM feature selection," Computers & Operations Research, vol.87, pp.137–145, 2017.
[26] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," European Journal of Operational Research, vol.265, no.3, pp.993–1004, 2018.
[27] A. Gleixner and J. Krüger, MIPLIB 2017 — The mixed integer programming library, https://miplib.zib.de/, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2022.
[28] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," Proc. 15th International Conference on Neural Information Processing Systems, pp.569–576, Jan. 2002.
[29] Gurobi Optimization, Gurobi Optimizer Reference Manual, version 9.5, Gurobi Optimization, 2021.
[30] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol.3, pp.1157–1182, March 2003.
[31] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," Advances in Neural Information Processing Systems 17, 2004.
[32] I. Guyon, S. Gunn, M. Nikravesh, and L.A. Zadeh, eds., Feature Extraction: Foundations and Applications, STUDFUZZ, vol.207, Springer, 2008.
[33] T. Hastie, R. Tibshirani, and R.J. Tibshirani, "Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons," Statist. Sci., vol.35, no.4, pp.579–592, 2020.
[34] T. Hastie, R. Tibshirani, and M. Wainwright, Statistical Learning with Sparsity: The Lasso and Generalizations, Chapman and Hall/CRC, 2019.
[35] H. Hazimeh and R. Mazumder, "Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms," Operations Research, vol.68, no.5, pp.1517–1537, 2020.
[36] H. Hazimeh, R. Mazumder, and A. Saab, "Sparse regression at scale: Branch-and-bound rooted in first-order optimization," Math. Program., vol.196. no.1-2, pp.347–388, 2022.
[37] L. Hermes and J.M. Buhmann, "Feature selection for support vector machines," Proc. 15th International Conference on Pattern Recognition, ICPR-2000, vol.2, pp.712–715, IEEE, Sept. 2000.
[38] C.L. Huang and C.J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," Expert Systems with Applications, vol.31, no.2, pp.231–240, 2006.
[39] IBM, IBM ILOG CPLEX Optimization Studio 20.1.0, https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/,

IBM, 2020.
[40] A. Jiménez-Cordero, J.M. Morales, and S. Pineda, "A novel embedded min-max approach for feature selection in nonlinear support vector machine classification," European Journal of Operational Research, vol.293, no.1, pp.24–35, 2021.
[41] T. Koch, T. Berthold, J. Pedersen, and C. Vanaret, "Progress in mathematical programming solvers from 2001 to 2020," EURO Journal on Computational Optimization, vol.10, 100031, 2022.
[42] K. Kira and L.A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," Proc. Tenth National Conference on Artificial Intelligence, pp.129–134, July 1992.
[43] H. Konno and R. Yamamoto, "Choosing the best set of variables in regression analysis using integer programming," J. Glob. Optim., vol.44, no.2, pp.273–282, 2009.
[44] K. Kudo, Y. Takano, and R. Nomura, "Stochastic discrete first-order algorithm for feature subset selection," IEICE Trans. Inf. & Syst., vol.E103-D, no.7, pp.1693–1702, July 2020.
[45] M. Labbé, L.I. Martínez-Merino, and A.M. Rodríguez-Chía, "Mixed integer linear programming for feature selection in support vector machine," Discrete Applied Mathematics, vol.261, pp.276–304, 2019.
[46] H.A. Le Thi, H.M. Le, and T.P. Dinh, "Feature selection in machine learning: An exact penalty approach using a difference of convex function algorithm," Mach. Learn., vol.101, no.1, pp.163–186, 2015.
[47] I.G. Lee, Q. Zhang, S.W. Yoon, and D. Won, "A mixed integer linear programming support vector machine for cost-effective feature selection," Knowledge-Based Systems, vol.203, 106145, 2020.
[48] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Computing Surveys, vol.50, no.6, pp.94:1–94:45, 2017.
[49] H. Liu and H. Motoda, eds., Computational Methods of Feature Selection, CRC Press, 2007.
[50] S. Maldonado and J. López, "Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification," Applied Soft Computing, vol.67, pp.94–105, 2018.
[51] S. Maldonado, J. Pérez, R. Weber, and M. Labbé, "Feature selection for support vector machines via mixed integer linear programming," Information Sciences, vol.279, pp.163–175, 2014.
[52] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," Information Sciences, vol.179, no.13, pp.2208–2217, 2009.
[53] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized support vector machines," Information Sciences, vol.181, no.1, pp.115–128, 2011.
[54] O.L. Mangasarian and G. Kou, "Feature selection for nonlinear kernel support vector machines," Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), pp.231–236, IEEE, Oct. 2007.
[55] H. Mittelmann, Decision tree for optimization software, https://plato.asu.edu/bench.html, School of Mathematical and Statistical Sciences, Arizona State University, Arizona, 2023.
[56] R. Miyashiro and Y. Takano, "Subset selection by Mallows' C_p: A mixed integer programming approach," Expert Systems with Applications, vol.42, no.1, pp.325–331, 2015.
[57] R. Miyashiro and Y. Takano, "Mixed integer second-order cone programming formulations for variable selection in linear regression," European Journal of Operational Research, vol.247, no.3, pp.721–731, 2015.
[58] M. Naganuma, Y. Takano, and R. Miyashiro, "Feature subset selection for ordered logit model via tangent-plane-based approximation," IEICE Trans. Inf. & Syst., vol.E102-D, no.5, pp.1046–1053, May 2019.
[59] J. Neumann, C. Schnörr, and G. Steidl, "Combined SVM-based feature selection and classification," Mach. Learn., vol.61, no.1–3, pp.129–150, 2005.
[60] Y.W. Park and D. Klabjan, "Subset selection for multiple linear regression via optimization," J. Glob. Optim., vol.77, no.3, pp.543–574, 2020.
[61] H. Saishu, K. Kudo, and Y. Takano, "Sparse Poisson regression via mixed-integer optimization," PloS ONE, vol.16, no.4, e0249916, 2021.
[62] T. Sato, Y. Takano, and R. Miyashiro, "Piecewise-linear approximation for feature subset selection in a sequential logit model," Journal of the Operations Research Society of Japan, vol.60, no.1, pp.1–14, 2017.
[63] T. Sato, Y. Takano, R. Miyashiro, and A. Yoshise, "Feature subset selection for logistic regression via mixed integer optimization," Comput. Optim. Appl., vol.64, no.3, pp.865–880, 2016.
[64] B. Schölkopf and A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
[65] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[66] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning SVM hyperparameters," IEEE Trans. Neural Netw., vol.21, no.2, pp.305–318, 2010.
[67] Y. Takano and J. Gotoh, "A nonlinear control policy using kernel method for dynamic asset allocation," Journal of the Operations Research Society of Japan, vol.54, no.4, pp.201–218, 2011.
[68] Y. Takano and J. Gotoh, "Multi-period portfolio selection using kernel-based control policy with dimensionality reduction," Expert Systems with Applications, vol.41, no.8, pp.3901–3914, 2014.
[69] Y. Takano and R. Miyashiro, "Best subset selection via cross-validation criterion," TOP, vol.28, no.2, pp.475–488, 2020.
[70] R. Tamura, K. Kobayashi, Y. Takano, R. Miyashiro, K. Nakata, and T. Matsui, "Best subset selection for eliminating multicollinearity," Journal of the Operations Research Society of Japan, vol.60, no.3, pp.321–336, 2017.
[71] R. Tamura, K. Kobayashi, Y. Takano, R. Miyashiro, K. Nakata, and T. Matsui, "Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor," J. Glob. Optim., vol.73, no.2, pp.431–446, 2019.
[72] M.E. Thompson, NDCC: Normally distributed clustered datasets on cubes, https://www.cs.wisc.edu/dmi/svm/ndcc/, Computer Sciences Department, University of Wisconsin, Madison, 2006.
[73] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," Mach. Learn., vol.102, no.3, pp.349–391, 2016.
[74] V. Vapnik, Statistical Learning Theory, Wiley Interscience, 1998.
[75] A. Wächter and L.T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," Math. Program., vol.106, no.1, pp.25–57, 2006.
[76] J. Wainer and P. Fonseca, "How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms," Artif. Intell. Rev., vol.54, pp.4771–4797, 2021.
[77] L. Wang, "Feature selection with kernel class separability," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.9, pp.1534–1546, 2008.
[78] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," Artif. Intell. Rev., vol.43, no.2, 179–192, 2015.
[79] A. Watanabe, R. Tamura, Y. Takano, and R. Miyashiro, "Branch-and-bound algorithm for optimal sparse canonical correlation analysis," Expert Systems with Applications, vol.217, 119530, 2023.
[80] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero norm with linear models and kernel methods," The Journal of Machine Learning Research, vol.3, pp.1439–1461, 2003.
[81] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," Proc. 13th International Conference on Neural Information Processing Systems, pp.647–653, Jan. 2000.
[82] H.P. Williams, Model Building in Mathematical Programming, John Wiley & Sons, 2013.
[83] L.A. Wolsey, Integer Programming, John Wiley & Sons, 2020.
[84] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," Proc. 16th International Conference on Neural Information Processing Systems, pp.49–56, Dec. 2003.

Appendix: List of Abbreviations

DBTC	Distance Between Two Classes
DC	Difference of Convex functions
DCA	Difference of Convex functions Algorithm
INLO	Integer Nonlinear Optimization
MILO	Mixed-Integer Linear Optimization
MIO	Mixed-Integer Optimization
RFE	Recursive Feature Elimination
SVM	Support Vector Machine

Ryuta Tamura received the B.E. and M.E. degrees from Tokyo University of Agriculture and Technology. He is currently a Ph.D. course student at Tokyo University of Agriculture and Technology. His research interests are mathematical programming and combinatorial optimization.

Yuichi Takano is an associate professor at the Institute of Systems and Information Engineering, University of Tsukuba, Japan. He received his Bachelor's degree in Policy and Planning Sciences in 2005, Master's degree in Engineering in 2007, and Doctorate in Engineering in 2010, all from the University of Tsukuba. His primary research interests are mathematical optimization and its application to financial engineering and machine learning.

Ryuhei Miyashiro received the B.E., M.E., and Ph.D. degrees from the University of Tokyo. He is presently an associate professor at Tokyo University of Agriculture and Technology. His research interests include mathematical and combinatorial optimization.