

An ADMM Solver for the MKL- $L_{0/1}$ -SVM

Yijie Shi and Bin Zhu

Abstract—We formulate the Multiple Kernel Learning (abbreviated as MKL) problem for the support vector machine with the infamous $(0, 1)$ -loss function. Some first-order optimality conditions are given and then exploited to develop a fast ADMM solver for the nonconvex and nonsmooth optimization problem. A simple numerical experiment on synthetic planar data shows that our MKL- $L_{0/1}$ -SVM framework could be promising.

I. INTRODUCTION

The support vector machine (SVM) is a classic tool in Machine Learning [1] with numerous applications in Control, see e.g., [2], [3] and the references therein. The idea of SVM dates back to the famous work of Cortes and Vapnik [4]. On p. 281 of that paper, the authors suggested (implicitly) the $(0, 1)$ -loss function, also called $L_{0/1}$ loss in [5], for quantifying the error of classification which essentially counts the number of samples to which the classifier assigns wrong labels. However, they also pointed out that the resulting optimization problem with the $(0, 1)$ loss is *NP-complete, nonsmooth, and nonconvex*, which directed researchers to the path of designing other (easier) loss functions, notably convex ones like the *hinge* loss. Recently in the literature, there is a resurging interest in the original SVM problem with the $(0, 1)$ loss, abbreviated as “ $L_{0/1}$ -SVM”, following theoretical and algorithmic developments for optimization problems with the “ ℓ_0 -norm”, see e.g., [6] and the references therein. In particular, [5] proposed KKT-like optimality conditions for the $L_{0/1}$ -SVM optimization problem and an efficient ADMM solver to obtain an *approximate* solution.

In this work, we draw inspiration from the aforementioned papers and present a *kernelized* version of the theory in which the ambient functional space has a richer structure than the usual Euclidean space. More precisely, we shall formulate the $L_{0/1}$ -SVM problem in the context of *Multiple Kernel Learning* [7] and describe a first-order optimality theory as well as a numerical procedure for the optimization problem via the ADMM. Obviously, the MKL framework can offer much more flexibility than the single-kernel formulation by letting the optimization algorithm determine the best combination of different kernel functions. In this sense, our results represent a substantial generalization of the work in [5] while maintaining the core features of $L_{0/1}$ -SVM.

This work was supported in part by Shenzhen Science and Technology Program (Grant No. 202206193000001-20220817184157001), the Fundamental Research Funds for the Central Universities, and the “Hundred-Talent Program” of Sun Yat-sen University.

The authors are with School of Intelligent Systems Engineering, Sun Yat-sen University, Gongchang Road 66, 518107 Shenzhen, China. Emails: shiyj27@mail2.sysu.edu.cn (Y. Shi), zhub26@mail.sysu.edu.cn (B. Zhu).

The remainder of this paper is organized as follows. Section II reviews the classic $L_{0/1}$ -SVM in the single-kernel case, while Section III discusses the MKL framework. Section IV establishes the optimality theory for the MKL- $L_{0/1}$ -SVM problem, and in Section V we propose an ADMM algorithm to solve the optimization problem. Finally, numerical experiments and concluding remarks are provided in Sections VI and VII, respectively.

Notation

\mathbb{R}_+ denotes the set of nonnegative reals, and $\mathbb{R}_+^n := \mathbb{R}_+ \times \dots \times \mathbb{R}_+$ the n -fold Cartesian product. $\mathbb{N}_m := \{1, 2, \dots, m\}$ is a finite index set for the data points and \mathbb{N}_L for the kernels. Throughout the paper, the summation variables $i \in \mathbb{N}_m$ is reserved for the data index, and $\ell \in \mathbb{N}_L$ for the kernel index. We write \sum_i and \sum_ℓ in place of $\sum_{i=1}^m$ and $\sum_{\ell=1}^L$ to simplify the notation.

II. PROBLEM FORMULATION: THE SINGLE-KERNEL CASE

Given the data set $\{(\mathbf{x}_i, y_i) : i \in \mathbb{N}_m\}$ where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$ is the label, the binary classification task aims to predict the correct label y for each vector \mathbf{x} , seen or unseen. To this end, the SVM first lifts the problem to a *reproducing kernel Hilbert space*¹ (RKHS) \mathbb{H} , in general infinite-dimensional and equipped with a *positive definite* kernel function, say $\kappa : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, via the feature mapping:

$$\mathbf{x} \mapsto \phi(\mathbf{x}) := \kappa(\cdot, \mathbf{x}) \in \mathbb{H}, \quad (1)$$

and then considers discriminant (or decision) functions of the form

$$\tilde{f}(\mathbf{x}) = b + \langle w, \phi(\mathbf{x}) \rangle_{\mathbb{H}} = b + w(\mathbf{x}), \quad (2)$$

where $b \in \mathbb{R}$, $w \in \mathbb{H}$, $\langle \cdot, \cdot \rangle_{\mathbb{H}}$ the inner product associated to the RKHS \mathbb{H} , and the second equality is due to the so-called *reproducing property*. Note that such a discriminant function is in general nonlinear in \mathbf{x} , but is indeed linear with respect to $\phi(\mathbf{x})$ in the feature space \mathbb{H} . The label of \mathbf{x} is assigned via $y(\mathbf{x}) = \text{sign}[\tilde{f}(\mathbf{x})]$ where $\text{sign}(\cdot)$ is the sign function which gives $+1$ for a positive number, -1 for a negative number, and left undefined at zero.

In order to estimate the unknown quantities b and w in (2), one sets up the unconstrained optimization problem:

$$\min_{\substack{w \in \mathbb{H}, b \in \mathbb{R}, \\ \tilde{f}(\cdot) = w(\cdot) + b}} \frac{1}{2} \|w\|_{\mathbb{H}}^2 + C \sum_i \mathcal{L}(y_i, \tilde{f}(\mathbf{x}_i)), \quad (3)$$

¹The theory of RKHS goes back to [8] and many more, see e.g., [9]. It has been used in the SVM as early as [4].

where $\|w\|_{\mathbb{H}}^2 = \langle w, w \rangle_{\mathbb{H}}$ is the squared norm of w induced by the inner product, $\mathcal{L}(\cdot, \cdot)$ is a suitable loss function, and $C > 0$ is a regularization parameter balancing the two parts in the objective function. In the classic case where \mathbb{H} can be identified as \mathbb{R}^n itself, $\|w\|_{\mathbb{H}}$ reduces to the Euclidean norm $\|\mathbf{w}\|$ with $\mathbf{w} = [w_1, \dots, w_n]^T$. Moreover, the quantity $1/\|\mathbf{w}\|$ can be interpreted as the width of the *margin* between the decision hyperplane (corresponding to the equation $\mathbf{w}^T \mathbf{x} + b = 0$) and the nearest points in each class, so that minimizing $\|\mathbf{w}\|^2$ is equivalent to maximizing the margin width, an intuitive measure of robustness of the classifier. As for the loss function, we adopt the most natural choice:

$$\mathcal{L}_{0/1}(y, \tilde{f}(\mathbf{x})) := H(1 - y\tilde{f}(\mathbf{x})) \quad (4)$$

where H is the (Heaviside) unit step function

$$H(t) = \begin{cases} 1, & t > 0 \\ 0, & t \leq 0. \end{cases} \quad (5)$$

In order to understand the loss function, notice that in the case where two classes of points are *linearly separable*, one can always identify a subset of decision hyperplanes such that $y_i f(\mathbf{x}_i) \geq 1$ for all $i \in \mathbb{N}_m$ [10]. In the *linearly inseparable* case, however, the inequality can be violated by some data points and such violations are in turn penalized since the loss function now reads as

$$\mathcal{L}_{0/1}(y, \tilde{f}(\mathbf{x})) = \begin{cases} 1, & \text{if } 1 - y\tilde{f}(\mathbf{x}) > 0 \\ 0, & \text{if } 1 - y\tilde{f}(\mathbf{x}) \leq 0. \end{cases} \quad (6)$$

It is this latter case that will be the focus of this paper.

The optimization problem (3) is infinite-dimensional in general due to the ambient space \mathbb{H} . It can however, be reduced to a *finite-dimensional* one via the celebrated *representer theorem* [11]. More precisely, by the *semiparametric* representer theorem [12], any minimizer of (3) must have the form

$$\tilde{f}(\cdot) = \sum_i w_i \kappa(\cdot, \mathbf{x}_i) + b, \quad (7)$$

so that the desired function $w(\cdot)$ is completely characterized by the linear combination of the *kernel sections* $\kappa(\cdot, \mathbf{x}_i)$, and the coefficients in $\mathbf{w} = [w_1, \dots, w_m]^T$ become the new unknowns. After some algebra involving the *kernel trick*, we are left with the following finite-dimensional optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^m, b \in \mathbb{R}} J(\mathbf{w}, b) := \frac{1}{2} \mathbf{w}^T K \mathbf{w} + C \|(1 - A\mathbf{w} - b\mathbf{y})_+\|_0 \quad (8)$$

where,

- $K = K^T$ is the *kernel matrix*

$$\begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix} \in \mathbb{R}^{m \times m} \quad (9)$$

which is *positive semidefinite* by construction,

- $\mathbf{1} \in \mathbb{R}^m$ is a vector whose components are all 1's,
- $\mathbf{y} = [y_1, \dots, y_m]^T$ is the vector of labels,

- the matrix $A = D_{\mathbf{y}} K$ is such that $D_{\mathbf{y}} = \text{diag}(\mathbf{y})$ is the diagonal matrix whose (i, i) entry is y_i ,
- the function $t_+ := \max\{0, t\}$ takes the positive part of the argument when applied to a scalar², and $\mathbf{v}_+ := [(v_1)_+, \dots, (v_m)_+]^T$ represents componentwise application of the scalar function,
- $\|\mathbf{v}\|_0$ is the ℓ_0 -norm³ that counts the number of nonzero components in the vector \mathbf{v} .

Clearly, the composite function $\|\mathbf{v}_+\|_0$ counts the number of (strictly) positive components in \mathbf{v} . For a scalar t , it coincides with the step function in (5).

Remark 1. The above formulation includes the problem investigated in [5] as a special case. To see this, consider the *homogeneous polynomial* kernel

$$\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^d \quad (10)$$

with the degree parameter $d = 1$. Then the discriminant function in (7) becomes

$$\tilde{f}(\mathbf{x}) = \sum_i w_i \mathbf{x}_i^T \mathbf{x} + b = \tilde{\mathbf{w}}^T \mathbf{x} + b, \quad (11)$$

where $\tilde{\mathbf{w}} := \sum_i w_i \mathbf{x}_i \in \mathbb{R}^n$ is identified as the new variable for optimization. Moreover, it is not difficult to verify the relation $\mathbf{w}^T K \mathbf{w} = \tilde{\mathbf{w}}^T \tilde{\mathbf{w}} = \|\tilde{\mathbf{w}}\|^2$, so that the optimization problem in [5] results.

For reasons discussed in Remark 1, in the remaining part of this paper, we shall always assume that the kernel matrix K is *positive definite*, which is indeed true for the *Gaussian* kernel

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right), \quad (12)$$

where $\sigma > 0$ is a parameter (called *hyperparameter*), see [13]. In such a case, the matrix $A = D_{\mathbf{y}} K$ in (8) is also *invertible* since $D_{\mathbf{y}}$ is a diagonal matrix⁴ whose diagonal entries are the labels -1 or 1 .

III. PROBLEM FORMULATION: THE MULTIPLE-KERNEL CASE

In all kernel-based methods, the selection of a suitable kernel and its parameter is a major issue. Usually, this is done via cross-validation which inevitably has an ad-hoc flavor. An active research area to handle such an issue is called Multiple Kernel Learning (MKL), where one employs a set of different kernels and lets the optimization procedure determine the proper combination. One possibility in this direction is to consider the nonlinear modeling function as follows:

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= \sum_{\ell} f_{\ell}(\mathbf{x}) + b \\ &= \sum_{\ell} d_{\ell} \sum_i w_i \kappa_{\ell}(\mathbf{x}, \mathbf{x}_i) + b, \end{aligned} \quad (13)$$

²It is known as the ReLU (Rectified Linear Unit) activation function in the context of artificial neural networks.

³The term ‘‘norm’’ is abused here since strictly speaking, ‘‘ ℓ^p -norms’’ are not *bona fide* norms for $0 \leq p < 1$ due to the violation of the triangle inequality.

⁴In fact, $D_{\mathbf{y}}$ is both *involutory* and *orthogonal*, i.e., $D_{\mathbf{y}}^2 = D_{\mathbf{y}}^T D_{\mathbf{y}} = I$.

where for each $\ell \in \mathbb{N}_L$, f_ℓ lives in a different RKHS \mathbb{H}'_ℓ corresponding to the kernel function $d_\ell \kappa_\ell(\cdot, \cdot)$, the parameters $d_\ell, b, w_i \in \mathbb{R}$, and \mathbf{x}_i comes from the training data⁵. In other words, the decision function \tilde{f} is parametrized by $(\mathbf{w}, \mathbf{d}, b) \in \mathbb{R}^{m+L+1}$. In order to formally state our MKL optimization problem for the $L_{0/1}$ -SVM, we need to borrow the functional space setup from [7].

For each $\ell \in \mathbb{N}_L$, let \mathbb{H}_ℓ be a RKHS of functions on $\mathcal{X} \subset \mathbb{R}^n$ with the kernel $\kappa_\ell(\cdot, \cdot)$ and the inner product $\langle \cdot, \cdot \rangle_{\mathbb{H}_\ell}$. Moreover, take $d_\ell \in \mathbb{R}_+$, and define a Hilbert space $\mathbb{H}'_\ell \subset \mathbb{H}_\ell$ as

$$\mathbb{H}'_\ell := \left\{ f \in \mathbb{H}_\ell : \frac{\|f\|_{\mathbb{H}_\ell}}{d_\ell} < \infty \right\} \quad (14)$$

endowed with the inner product

$$\langle f, g \rangle_{\mathbb{H}'_\ell} = \frac{\langle f, g \rangle_{\mathbb{H}_\ell}}{d_\ell}. \quad (15)$$

In this paper, we use the convention that $x/0 = 0$ if $x = 0$ and ∞ otherwise. This means that, if $d_\ell = 0$ then a function $f \in \mathbb{H}_\ell$ belongs to the Hilbert space \mathbb{H}'_ℓ only if $f = 0$. In such a case, \mathbb{H}'_ℓ becomes a singleton containing only the null element of \mathbb{H}_ℓ . Within this framework, \mathbb{H}'_ℓ is a RKHS with the kernel $\kappa'_\ell(\mathbf{x}, \mathbf{y}) = d_\ell \kappa_\ell(\mathbf{x}, \mathbf{y})$ since

$$\begin{aligned} \forall f \in \mathbb{H}'_\ell \subset \mathbb{H}_\ell, \quad f(\mathbf{x}) &= \langle f(\cdot), \kappa_\ell(\mathbf{x}, \cdot) \rangle_{\mathbb{H}_\ell} \\ &= \frac{1}{d_\ell} \langle f(\cdot), d_\ell \kappa_\ell(\mathbf{x}, \cdot) \rangle_{\mathbb{H}_\ell} \\ &= \langle f(\cdot), d_\ell \kappa_\ell(\mathbf{x}, \cdot) \rangle_{\mathbb{H}'_\ell}. \end{aligned} \quad (16)$$

Define $\mathbb{F} := \mathbb{H}'_1 \times \mathbb{H}'_2 \times \dots \times \mathbb{H}'_L$ as the Cartesian product of the RKHSs $\{\mathbb{H}'_\ell\}$, which is itself a Hilbert space with the inner product

$$\langle (f_1, \dots, f_L), (g_1, \dots, g_L) \rangle_{\mathbb{F}} = \sum_{\ell} \langle f_\ell, g_\ell \rangle_{\mathbb{H}'_\ell}. \quad (17)$$

Let $\mathbb{H} := \bigoplus_{\ell=1}^L \mathbb{H}'_\ell$ be the *direct sum* of the RKHSs $\{\mathbb{H}'_\ell\}$, which is also a RKHS with the kernel function

$$\kappa(\mathbf{x}, \mathbf{y}) = \sum_{\ell} d_\ell \kappa_\ell(\mathbf{x}, \mathbf{y}), \quad (18)$$

see [8]. Moreover, the squared norm of $f \in \mathbb{H}$ is known as

$$\|f\|_{\mathbb{H}}^2 = \min \left\{ \sum_{\ell} \|f_\ell\|_{\mathbb{H}'_\ell}^2 = \sum_{\ell} \frac{1}{d_\ell} \|f_\ell\|_{\mathbb{H}_\ell}^2 : f = \sum_{\ell} f_\ell \text{ such that } f_\ell \in \mathbb{H}'_\ell \right\}. \quad (19)$$

The vector $\mathbf{d} = (d_1, \dots, d_L) \in \mathbb{R}_+^L$ is seen as a tunable parameter for the linear combination of kernels $\{\kappa_\ell\}$ in (18).

⁵Different kernels share the same weight vector \mathbf{w} due to the *representer theorem* applied in the direct sum space introduced before (18).

A typical MKL task can be formulated as

$$\min_{\substack{\mathbf{f}=(f_1, \dots, f_L) \in \mathbb{F} \\ \mathbf{d} \in \mathbb{R}^L, b \in \mathbb{R}}} \frac{1}{2} \sum_{\ell} \frac{1}{d_\ell} \|f_\ell\|_{\mathbb{H}_\ell}^2 + C \sum_i \mathcal{L}_{0/1}(y_i, \tilde{f}(\mathbf{x}_i)) \quad (20a)$$

$$\text{s.t.} \quad d_\ell \geq 0, \ell \in \mathbb{N}_L \quad (20a)$$

$$\sum_{\ell} d_\ell = 1 \quad (20b)$$

$$\tilde{f}(\cdot) = \sum_{\ell} f_\ell(\cdot) + b$$

where $C > 0$ is a regularization parameter. For our SVM task, the first (regularization) term in the objective function is chosen so due to its convexity (see [7, Appendix A.1]), which makes the problem tractable.

IV. OPTIMALITY THEORY

In this section, we give some theoretical results on the existence of an optimal solution to (20), and the KKT-like first-order optimality conditions.

Assumption 1. *Given the data points $\{\mathbf{x}_i : i \in \mathbb{N}_m\}$, each $m \times m$ kernel matrix K_ℓ , whose (i, j) entry is $\kappa_\ell(\mathbf{x}_i, \mathbf{x}_j)$, is positive definite for $\ell \in \mathbb{N}_L$.*

The main results are given in the next two subsections. Due to the page limit of the conference paper, the complete proofs are deferred to the journal version [14].

A. Existence of a minimizer

The existence theorem is provided below.

Theorem 1. *Assume that the intercept b takes value from a closed interval $\mathcal{I} := [-M, M]$ where $M > 0$ is a sufficiently large number. Then the optimization problem (20) has a global minimizer and the set of all global minimizers is bounded.*

B. Characterization of global and local minimizers

The last equality constraint in (20) can be safely eliminated by a substitution into the objective function. Next, define a new variable $\mathbf{u} \in \mathbb{R}^m$ by letting $u_i = 1 - y_i(f(\mathbf{x}_i) + b)$ where $f = \sum_{\ell} f_\ell$. We can then rewrite (20) in the following way:

$$\min_{\substack{\mathbf{f} \in \mathbb{F}, \mathbf{d} \in \mathbb{R}^L \\ b \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^m}} \frac{1}{2} \sum_{\ell} \frac{1}{d_\ell} \|f_\ell\|_{\mathbb{H}_\ell}^2 + C \|\mathbf{u}_+\|_0 \quad (21a)$$

$$\text{s.t.} \quad (20a) \text{ and } (20b)$$

$$u_i + y_i(f(\mathbf{x}_i) + b) = 1, \quad i \in \mathbb{N}_m, \quad (21b)$$

where the last equality constraint is obviously *affine* in the “variables” $(\mathbf{f}, b, \mathbf{u})$.

Before stating the optimality conditions, we need a generalized definition of a stationary point in nonlinear programming.

Definition 1 (P-stationary point of (21)). Fix a regularization parameter $C > 0$. We call $(\mathbf{f}^*, \mathbf{d}^*, b^*, \mathbf{u}^*)$ a proximal stationary (abbreviated as P-stationary) point of (21) if there

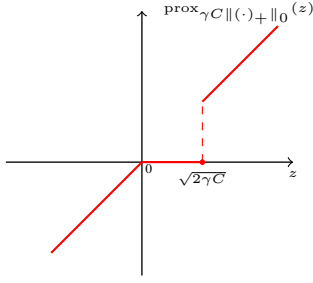


Fig. 1: The $L_{0/1}$ proximal operator on the real line.

exists a vector $(\boldsymbol{\theta}^*, \alpha^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^{L+1+m}$ and a number $\gamma > 0$ such that

$$d_\ell^* \geq 0, \ell \in \mathbb{N}_L \quad (22a)$$

$$\sum_{\ell} d_\ell^* = 1 \quad (22b)$$

$$u_i^* + y_i (f^*(\mathbf{x}_i) + b^*) = 1, i \in \mathbb{N}_m \quad (22c)$$

$$\theta_\ell^* \geq 0, \ell \in \mathbb{N}_L \quad (22d)$$

$$\theta_\ell^* d_\ell^* = 0, \ell \in \mathbb{N}_L \quad (22e)$$

$$\forall \ell \in \mathbb{N}_L, \frac{1}{d_\ell^*} f_\ell^*(\cdot) = - \sum_i \lambda_i^* y_i \kappa_\ell(\cdot, \mathbf{x}_i) \quad (22f)$$

$$-\frac{1}{2(d_\ell^*)^2} \|f_\ell^*\|_{\mathbb{H}_\ell}^2 + \alpha^* - \theta_\ell^* = 0, \ell \in \mathbb{N}_L \quad (22g)$$

$$\mathbf{y}^\top \boldsymbol{\lambda}^* = 0 \quad (22h)$$

$$\text{prox}_{\gamma C \|\cdot\|_+}(\mathbf{u}^* - \gamma \boldsymbol{\lambda}^*) = \mathbf{u}^*, \quad (22i)$$

where the proximal operator is defined as

$$\text{prox}_{\gamma C \|\cdot\|_+}(\mathbf{z}) := \underset{\mathbf{v} \in \mathbb{R}^m}{\text{argmin}} C \|\mathbf{v}_+\|_0 + \frac{1}{2\gamma} \|\mathbf{v} - \mathbf{z}\|^2. \quad (23)$$

According to [5], for a scalar z the proximal operator in (22) can be evaluated in a closed form:

$$\text{prox}_{\gamma C \|\cdot\|_+}(z) = \begin{cases} 0, & 0 < z \leq \sqrt{2\gamma C} \\ z, & z > \sqrt{2\gamma C} \text{ or } z \leq 0, \end{cases} \quad (24)$$

see Fig. 1. For a vector $\mathbf{z} \in \mathbb{R}^m$, the proximal operator in (23) is evaluated by applying the scalar version (24) to each component of \mathbf{z} , namely

$$[\text{prox}_{\gamma C \|\cdot\|_+}(\mathbf{z})]_i = \text{prox}_{\gamma C \|\cdot\|_+}(z_i), \quad (25)$$

because the objective function on the right-hand side of (23) can be decomposed as

$$\sum_i C \|(v_i)_+\|_0 + \frac{1}{2\gamma} (v_i - z_i)^2.$$

Formula (25) is called “ $L_{0/1}$ proximal operator” in [5].

The components of the vector $(\boldsymbol{\theta}^*, \alpha^*, \boldsymbol{\lambda}^*)$ in Definition 1 can be interpreted as the *Lagrange multipliers* as it appeared in a smooth SVM problem and played a similar role in the optimality conditions [4], although a direct dual analysis here can be difficult due the presence of the nonsmooth nonconvex function $\|\cdot\|_+$. The set of equations (22) are understood as the *KKT-like* optimality conditions for the

optimization problem (21), where (22a), (22b), and (22c) are the primal constraints, (22d) the dual constraints, (22e) the complementary slackness, and (22f), (22g), (22h), and (22i) are the stationarity conditions of the Lagrangian with respect to the primal variables. Notice that the only nonsmooth term presented is $\|\mathbf{u}_+\|_0$, and the corresponding stationarity condition (22i) with respect to \mathbf{u} is given by the proximal operator (23).

The following theorem connects the optimality conditions for (21) to P-stationary points.

Theorem 2. *The global and local minimizers of (21) admit the following characterizations:*

1) *A global minimizer is a P-stationary point with $0 < \gamma < C_1$, where the positive number*

$$C_1 = \min \{ \lambda_{\min}(\mathcal{K}(\mathbf{d})) : \mathbf{d} \text{ satisfies (20a) and (20b)} \}$$

in which $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of a matrix and $\mathcal{K}(\mathbf{d}) := \sum_{\ell} d_\ell K_\ell$ with K_ℓ the kernel matrix corresponding to $\kappa_\ell(\cdot, \cdot)$.

2) *Any P-stationary point (with $\gamma > 0$) is also a local minimizer of (21).*

V. ALGORITHM DESIGN

In this section, we take advantages of the Alternating Direction Method of Multipliers (ADMM) [15] and working sets (active sets) to devise a first-order algorithm for our MKL- $L_{0/1}$ -SVM optimization problem. More precisely, we aim to obtain a P-stationary point (Definition 1) and hence a local minimizer of (21) by Theorem 2. The unique issue is that each $f_\ell \in \mathbb{H}_\ell$ could be an infinite-dimensional object. Appealing to the KKT-like conditions (22), the next lemma says that f_ℓ admits another finite-dimensional representation which is suitable for numerical computation.

Lemma 1. *Under Assumption 1, each f_ℓ is completely represented by its values at the data points which are collected into a vector*

$$\mathbf{v}_{f_\ell} := [f_\ell(\mathbf{x}_1) \quad f_\ell(\mathbf{x}_2) \quad \cdots \quad f_\ell(\mathbf{x}_m)]^\top. \quad (26)$$

Moreover, we have $\|f_\ell\|_{\mathbb{H}_\ell}^2 = \mathbf{v}_{f_\ell}^\top K_\ell^{-1} \mathbf{v}_{f_\ell}$.

Proof. According to (22f), we see that optimal f_ℓ is restricted to the linear span of the m kernel sections $\{\kappa_\ell(\cdot, \mathbf{x}_i)\}$. Let the coefficients of linear combination be $\{\tilde{w}_i\}$. Then it holds that $\mathbf{v}_{f_\ell} = K_\ell \tilde{\mathbf{w}}$ and $\|f_\ell\|_{\mathbb{H}_\ell}^2 = \tilde{\mathbf{w}}^\top K_\ell \tilde{\mathbf{w}} = \mathbf{v}_{f_\ell}^\top K_\ell^{-1} \mathbf{v}_{f_\ell}$ where K_ℓ is invertible by assumption. \square

Now we can introduce the working set (with respect to the data) and related *support vectors* along the lines of [5, Subsec. 4.1]. Let $(\mathbf{f}^*, \mathbf{d}^*, b^*, \mathbf{u}^*)$ be a P-stationary point of (21). Then by Definition 1, there exists a Lagrange multiplier $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and a scalar $\gamma > 0$ such that (22i) holds. Define a set

$$T_* := \left\{ i \in \mathbb{N}_m : u_i^* - \gamma \lambda_i^* \in (0, \sqrt{2\gamma C}] \right\}, \quad (27)$$

and its complement $\bar{T}_* := \mathbb{N}_m \setminus T_*$. For a vector $\mathbf{z} \in \mathbb{R}^m$ and an index set $T \subset \mathbb{N}_m$ with cardinality $|T|$, we write \mathbf{z}_T for

the $|T|$ -dimensional subvector of \mathbf{z} whose components are indexed in T . Then it follows from (22i), (25), and (24) that

$$\begin{bmatrix} \mathbf{u}_{T_*}^* \\ \mathbf{u}_{\bar{T}_*}^* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{u}^* - \gamma \boldsymbol{\lambda}^*)_{\bar{T}_*} \end{bmatrix}. \quad (28)$$

Consequently, we have $\boldsymbol{\lambda}_{T_*}^* = \mathbf{0}$ and the working set (27) can be equivalently written as

$$T_* := \left\{ i \in \mathbb{N}_m : \lambda_i^* \in [-\sqrt{2C/\gamma}, 0) \right\}. \quad (29)$$

In plain words, the nonzero components of $\boldsymbol{\lambda}^*$ are indexed only in T_* with values in the interval $[-\sqrt{2C/\gamma}, 0)$. This brings significant sparsification of the decision function since by (22f) we have

$$\frac{1}{d_\ell^*} f_\ell^*(\cdot) = - \sum_{i \in T_*} \lambda_i^* y_i \kappa_\ell(\cdot, \mathbf{x}_i), \quad \ell \in \mathbb{N}_L. \quad (30)$$

This familiar formula calls for the following comments:

- The vectors $\{\mathbf{x}_i : i \in T_*\}$ correspond to nonzero Lagrange multipliers $\{\lambda_i^*\}$ just like standard support vectors in [4]. They are called $L_{0/1}$ -support vectors in [5] since they are selected by the proximal operator (23).
- Moreover, the condition (22c) implies that

$$y_i (f^*(\mathbf{x}_i) + b^*) = 1 \quad \text{for } i \in T_* \quad (31)$$

since $\mathbf{u}_{T_*}^* = \mathbf{0}$ by (28). This means that any $L_{0/1}$ -support vector \mathbf{x}_i satisfies the equation $f^*(\mathbf{x}) + b^* = \pm 1$ which gives two hyperplanes in the RKHS \mathbb{H} , called *support hyperplanes*. It is well known that such a property is guaranteed for linearly separable datasets, and may not hold for linearly inseparable datasets with penalty functions other than the $(0, 1)$ -type. Such an observation explains why the $L_{0/1}$ -SVM can (in principle) have fewer support vectors than other SVM models.

Next we give the framework of ADMM for (21) which is now viewed as a finite-dimensional problem. In order to handle the inequality constraints (20a), we employ the indicator function (in the sense of Convex Analysis) of the nonnegative orthant \mathbb{R}_+^L , namely $g(\mathbf{z}) = 0$ for $\mathbf{z} \in \mathbb{R}_+^L$ and is equal to $+\infty$ otherwise, and convert (21) to the form:

$$\min_{\substack{\mathbf{f} \in \mathbb{F}, \mathbf{d} \in \mathbb{R}^L \\ b \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^m \\ \mathbf{z} \in \mathbb{R}^L}} \frac{1}{2} \sum_{\ell} \frac{1}{d_\ell} \|f_\ell\|_{\mathbb{H}_\ell}^2 + C \|\mathbf{u}_+\|_0 + g(\mathbf{z}) \quad (32a)$$

$$\text{s.t.} \quad \mathbf{d} = \mathbf{z} \quad (32b)$$

(20b) and (21b),

see [15, Section 5]. Obviously we have $g(\mathbf{z}) = \sum_{\ell} g_\ell(z_\ell)$ where each g_ℓ is the respective indicator function for the nonnegative semiaxis $z_\ell \geq 0$. The *augmented Lagrangian* of (32) is given by

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{f}, \mathbf{d}, b, \mathbf{u}, \mathbf{z}; \boldsymbol{\lambda}, \boldsymbol{\theta}, \alpha) &= \frac{1}{2} \sum_{\ell} \frac{1}{d_\ell} \|f_\ell\|_{\mathbb{H}_\ell}^2 + C \|\mathbf{u}_+\|_0 \\ &+ g(\mathbf{z}) + \boldsymbol{\lambda}^\top \mathbf{r} + \frac{\rho_1}{2} \|\mathbf{r}\|^2 + \boldsymbol{\theta}^\top (\mathbf{d} - \mathbf{z}) + \frac{\rho_2}{2} \|\mathbf{d} - \mathbf{z}\|^2 \\ &+ \alpha (\mathbf{1}^\top \mathbf{d} - 1) + \frac{\rho_3}{2} (\mathbf{1}^\top \mathbf{d} - 1)^2 \end{aligned} \quad (33)$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m)$, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_L)$, and α are the Lagrangian multipliers, $\mathbf{r} := \mathbf{u} + D_{\mathbf{y}} \mathbf{v}_f + b\mathbf{y} - \mathbf{1}$ is the residual vector, and $\boldsymbol{\rho} = (\rho_1, \rho_2, \rho_3)$ contains three positive penalty parameters. We have also written $f = \sum_{\ell} f_\ell$ in the style of (21b) to simplify the notation.

Given the k -th iterate $(\mathbf{f}^k, \mathbf{d}^k, b^k, \mathbf{u}^k, \mathbf{z}^k; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k)$, the framework to update each variable is as follows:

$$\begin{aligned} \mathbf{u}^{k+1} &= \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{f}^k, \mathbf{d}^k, b^k, \mathbf{u}, \mathbf{z}^k; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k) \\ \mathbf{f}^{k+1} &= \underset{\mathbf{f} \in \mathbb{F}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{f}, \mathbf{d}^k, b^k, \mathbf{u}^{k+1}, \mathbf{z}^k; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k) \\ b^{k+1} &= \underset{b \in \mathbb{R}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{f}^{k+1}, \mathbf{d}^k, b, \mathbf{u}^{k+1}, \mathbf{z}^k; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k) \\ \mathbf{z}^{k+1} &= \underset{\mathbf{z} \in \mathbb{R}^L}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{f}^{k+1}, \mathbf{d}^k, b^{k+1}, \mathbf{u}^{k+1}, \mathbf{z}; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k) \\ \mathbf{d}^{k+1} &= \underset{\mathbf{d} \in \mathbb{R}^L}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{f}^{k+1}, \mathbf{d}, b^{k+1}, \mathbf{u}^{k+1}, \mathbf{z}^{k+1}; \boldsymbol{\lambda}^k, \boldsymbol{\theta}^k, \alpha^k) \\ \boldsymbol{\theta}^{k+1} &= \boldsymbol{\theta}^k + \rho_2 (\mathbf{d}^{k+1} - \mathbf{z}^{k+1}) \\ \alpha^{k+1} &= \alpha^k + \rho_3 (\mathbf{1}^\top \mathbf{d}^{k+1} - 1) \\ \lambda_i^{k+1} &= \lambda_i^k + \rho_1 [u_i^{k+1} + y_i (f^{k+1}(\mathbf{x}_i) + b^{k+1}) - 1], \quad i \in \mathbb{N}_m. \end{aligned} \quad (34)$$

Next, we describe how to solve each subproblem above.

- 1) **Updating \mathbf{u}^{k+1} .** For each component of \mathbf{u} , the u_i -subproblem in (34) admits a separation of variables and can be solved along the following lines:

$$\begin{aligned} u_i^{k+1} &= \underset{u_i \in \mathbb{R}}{\operatorname{argmin}} C \|u_+\|_0 + \sum_i \lambda_i^k u_i \\ &+ \frac{\rho_1}{2} \sum_i [u_i + y_i (f^k(\mathbf{x}_i) + b^k) - 1]^2 \\ &= \underset{u_i \in \mathbb{R}}{\operatorname{argmin}} C \|(u_i)_+\|_0 + \frac{\rho_1}{2} (u_i - s_i^k)^2 \\ &= \operatorname{prox}_{\frac{C}{\rho_1} \|\cdot\|_+} (s_i^k), \end{aligned} \quad (35)$$

where $s_i^k = 1 - y_i (f^k(\mathbf{x}_i) + b^k) - \lambda_i^k / \rho_1$ and the proximal operator given in (24). The corresponding vector can be written compactly as

$$\mathbf{s}^k = \mathbf{1} - D_{\mathbf{y}} \mathbf{v}_{f^k} - b^k \mathbf{y} - \boldsymbol{\lambda}^k / \rho_1 \quad (36)$$

with \mathbf{v}_{f^k} defined similarly to (26). Define a working set T_k at the k -th step by

$$T_k := \left\{ i \in \mathbb{N}_m : s_i^k \in (0, \sqrt{2C/\rho_1}] \right\}. \quad (37)$$

Then (35) can equivalently be written as

$$\mathbf{u}_{T_k}^{k+1} = \mathbf{0}, \quad \mathbf{u}_{\bar{T}_k}^{k+1} = \mathbf{s}_{\bar{T}_k}^k. \quad (38)$$

- 2) **Updating \mathbf{f}^{k+1} .** The \mathbf{f} -subproblem in (34) is

$$\begin{aligned} \mathbf{f}^{k+1} &= \underset{\mathbf{f} \in \mathbb{F}}{\operatorname{argmin}} \frac{1}{2} \sum_{\ell} \frac{1}{d_\ell^k} \|f_\ell\|_{\mathbb{H}_\ell}^2 + \\ &+ \sum_i \lambda_i^k [u_i^{k+1} + y_i (f(\mathbf{x}_i) + b^k) - 1] \\ &+ \frac{\rho_1}{2} \sum_i [u_i^{k+1} + y_i (f(\mathbf{x}_i) + b^k) - 1]^2. \end{aligned} \quad (39)$$

To solve (39), we again adopt a componentwise strategy, that is, for each $d_\ell > 0$ we update each f_ℓ separately

according to the stationarity condition. By Lemma 1, we only need to update the function values at all the inputs $\{\mathbf{x}_i\}$, namely the vector in (26). More precisely, notice that by the reproducing property, the Fréchet derivative of $f_\ell(\mathbf{x}_i) = \langle f_\ell, \kappa_\ell(\cdot, \mathbf{x}_i) \rangle$ with respect to f_ℓ can be identified as $\kappa_\ell(\cdot, \mathbf{x}_i)$. Then the stationary-point equation for f_ℓ can be written as

$$0 = \frac{1}{d_\ell^k} f_\ell(\cdot) + \sum_i \left\{ \lambda_i^k y_i \kappa_\ell(\cdot, \mathbf{x}_i) + \rho_1 y_i \kappa_\ell(\cdot, \mathbf{x}_i) [u_i^{k+1} + y_i(f(\mathbf{x}_i) + b^k) - 1] \right\} \quad (40)$$

which holds for any \mathbf{x} , and in particular for all \mathbf{x}_i . Notice that here $f = f_\ell + \sum_{t \neq \ell} f_t^k$. In vector form, we have the equation

$$\left(\frac{1}{d_\ell^k} I + \rho_1 K_\ell \right) \mathbf{v}_{f_\ell} = -\rho_1 K_\ell D_{\mathbf{y}} \left(\mathbf{u}^{k+1} + D_{\mathbf{y}} \sum_{t \neq \ell} \mathbf{v}_{f_t^k} + b^k \mathbf{y} - \mathbf{1} + \boldsymbol{\lambda}^k / \rho_1 \right). \quad (41)$$

which can be readily solved since the coefficient matrix before \mathbf{v}_{f_ℓ} is positive definite. On the other hand, for each $d_\ell = 0$ we keep $f_\ell \equiv 0$.

- 3) **Updating b^{k+1} .** The b -subproblem can be written as

$$b^{k+1} = \operatorname{argmin}_{b \in \mathbb{R}} \sum_i \lambda_i^k [u_i^{k+1} + y_i(f^{k+1}(\mathbf{x}_i) + b) - 1] + \frac{\rho_1}{2} \sum_i [u_i^{k+1} + y_i(f^{k+1}(\mathbf{x}_i) + b) - 1]^2. \quad (42)$$

The stationary-point equation is

$$0 = \sum_i \lambda_i^k y_i + \rho_1 \sum_i y_i [u_i^{k+1} + y_i(f^{k+1}(\mathbf{x}_i) + b) - 1], \quad (43)$$

which is solved by

$$b^{k+1} = \frac{1}{m} \left[\mathbf{y}^\top (\mathbf{1} - \mathbf{u}^{k+1} - \boldsymbol{\lambda}^k / \rho_1) - \mathbf{1}^\top \mathbf{v}_{f^{k+1}} \right]. \quad (44)$$

- 4) **Updating \mathbf{z}^{k+1} .** The subproblem for each component of \mathbf{z}^{k+1} in (34) also admits a separation of variables and we carry out the update as follows:

$$\begin{aligned} z_\ell^{k+1} &= \operatorname{argmin}_{z_\ell \in \mathbb{R}} g(\mathbf{z}) - (\boldsymbol{\theta}^k)^\top \mathbf{z} + \frac{\rho_2}{2} \|\mathbf{d}^k - \mathbf{z}\|^2 \\ &= \operatorname{argmin}_{z_\ell \in \mathbb{R}} g_\ell(z_\ell) - \theta_\ell^k z_\ell + \frac{\rho_2}{2} (d_\ell^k - z_\ell)^2 \\ &= (d_\ell^k + \theta_\ell^k / \rho_2)_+. \end{aligned} \quad (45)$$

where the function $(\cdot)_+$ takes the positive part of the argument. Inspired by this expression, we can define another working set

$$S_k := \{\ell \in \mathbb{N}_L : d_\ell^k + \theta_\ell^k / \rho_2 > 0\} \quad (46)$$

for the selection of kernels and $\bar{S}_k = \mathbb{N}_L \setminus S_k$. Then an equivalent update formula for \mathbf{z} is

$$\mathbf{z}_{S_k}^{k+1} = (\mathbf{d}^k + \boldsymbol{\theta}^k / \rho_2)_{S_k}, \quad \mathbf{z}_{\bar{S}_k}^{k+1} = \mathbf{0}. \quad (47)$$

Notice that this working set is less complicated than T_k in (37) since the function $(\cdot)_+$ is continuous, unlike the proximal mapping.

- 5) **Updating \mathbf{d}^{k+1} .** Again we adopt a componentwise strategy for the update of d_ℓ^{k+1} in (34):

$$\begin{aligned} d_\ell^{k+1} &= \operatorname{argmin}_{d_\ell \in \mathbb{R}} \frac{1}{2d_\ell} \|f_\ell^{k+1}\|_{\mathbb{H}_\ell}^2 + \theta_\ell^k (d_\ell - z_\ell^{k+1}) \\ &\quad + \frac{\rho_2}{2} (d_\ell - z_\ell^{k+1})^2 + \alpha^k d_\ell + \frac{\rho_3}{2} \left(d_\ell + \sum_{t \neq \ell} d_t^k - 1 \right)^2 \end{aligned} \quad (48)$$

where $\|f_\ell\|_{\mathbb{H}_\ell}^2 = \mathbf{v}_{f_\ell}^\top K_\ell^{-1} \mathbf{v}_{f_\ell}$ (see Lemma 1), and d_t^k are held fixed for $t \neq \ell$. The stationary-point equation for d_ℓ is just

$$\begin{aligned} 0 &= -\frac{1}{2d_\ell^2} \mathbf{v}_{f_\ell^{k+1}}^\top K_\ell^{-1} \mathbf{v}_{f_\ell^{k+1}} + \theta_\ell^k + \rho_2 (d_\ell - z_\ell^{k+1}) \\ &\quad + \alpha^k + \rho_3 \left(d_\ell + \sum_{t \neq \ell} d_t^k - 1 \right). \end{aligned} \quad (49)$$

This is a cubic polynomial equation (after multiplying both sides by d_ℓ^2) which can be solved numerically. Moreover, since the coefficients are real, there must be at least one real root. When $\|f_\ell^{k+1}\|_{\mathbb{H}_\ell}^2 > 0$, the objective function in (48) is strictly convex in the positive semi-axis $d_\ell > 0$ and one can show without difficulty that a local minimizer, which must be a stationary point, exists. Therefore, we conclude that (49) must have a *positive real root*.

However, if instead we search the minimum of (48) in all of \mathbb{R} , a pathology happens when $\|f_\ell^{k+1}\|_{\mathbb{H}_\ell}^2 > 0$ and d_ℓ tends to zero from the left. In that case, the objective function tends to $-\infty$ and $+\infty$ on two sides of zero. As an ad-hoc recipe, we take the positive real root⁶ of (49) as d_ℓ^{k+1} for $\ell \in S_k$. For $\ell \in \bar{S}_k$, on the other hand, we let

$$\mathbf{d}_{\bar{S}_k}^{k+1} = \mathbf{0} \quad (50)$$

in accordance with the second formula in (47), because in the end (if the algorithm converges) we will have the equality (32b).

- 6) **Updating $\boldsymbol{\theta}^{k+1}$.** With the help of the working set (46), the update of $\boldsymbol{\theta}$ in (34) can be simplified as:

$$\boldsymbol{\theta}_{S_k}^{k+1} = \boldsymbol{\theta}_{S_k}^k + \rho_2 (\mathbf{d}^{k+1} - \mathbf{z}^{k+1})_{S_k}, \quad \boldsymbol{\theta}_{\bar{S}_k}^{k+1} = \boldsymbol{\theta}_{\bar{S}_k}^k. \quad (51)$$

- 7) **Updating α^{k+1} .** See (34).
8) **Updating $\boldsymbol{\lambda}^{k+1}$.** Inspired by the property of the working set T_* (see (28) and the next line after it), the update of $\boldsymbol{\lambda}$ in (34) is simplified as follows:

$$\boldsymbol{\lambda}_{T_k}^{k+1} = \boldsymbol{\lambda}_{T_k}^k + \rho_1 \mathbf{r}_{T_k}^{k+1}, \quad \boldsymbol{\lambda}_{\bar{T}_k}^{k+1} = \mathbf{0} \quad (52)$$

⁶If there are multiple positive real roots (possibly due to some numerical issue), we take the one with the largest absolute value.

where the vector $\mathbf{r}^{k+1} = \mathbf{u}^{k+1} + D_{\mathbf{y}} \mathbf{v}_{f^{k+1}} + b^{k+1} \mathbf{y} - \mathbf{1}$. In other words, we remove the components of $\boldsymbol{\lambda}$ which are not in the current working set.

The update steps above are collected into the next algorithm.

Algorithm 1 ADMM for the MKL- $L_{0/1}$ -SVM

- 1: Set $C, \rho_1, \rho_2, \rho_3, \{\kappa_\ell\}, \text{max_iter}$, and $k = 0$.
 - 2: Initialize $(\mathbf{f}^0, \mathbf{d}^0, b^0, \mathbf{u}^0, \mathbf{z}^0; \boldsymbol{\theta}^0, \alpha^0, \boldsymbol{\lambda}^0)$.
 - 3: **while** The terminating condition is not met and $k \leq \text{max_iter}$ **do**
 - 4: Update T_k as in (37).
 - 5: Update \mathbf{u}^{k+1} by (38).
 - 6: Update \mathbf{f}^{k+1} by (40).
 - 7: Update b^{k+1} by (44).
 - 8: Update \mathbf{z}^{k+1} by (47).
 - 9: Update \mathbf{d}^{k+1} by (49).
 - 10: Update $\boldsymbol{\theta}^{k+1}$ by (51).
 - 11: Update α^{k+1} in (34).
 - 12: Update $\boldsymbol{\lambda}^{k+1}$ by (52).
 - 13: Set $k = k + 1$.
 - 14: **end while**
 - 15: **return** the final iterate $(\mathbf{f}^k, \mathbf{d}^k, b^k, \mathbf{u}^k, \mathbf{z}^k; \boldsymbol{\theta}^k, \alpha^k, \boldsymbol{\lambda}^k)$.
-

Unfortunately, we are not able to prove the convergence of Algorithm 1 as it seems very hard in general due to the nonconvexity and nonsmoothness of the optimization problem (21). However, we can give a characterization of the limit point *if* the algorithm converges, see the next result.

Theorem 3. *Suppose that the sequence*

$$\{\Psi^k\} = \{(\mathbf{f}^k, \mathbf{d}^k, b^k, \mathbf{u}^k, \mathbf{z}^k; \boldsymbol{\theta}^k, \alpha^k, \boldsymbol{\lambda}^k)\}$$

generated by the ADMM algorithm above has a limit point $\Psi^ = (\mathbf{f}^*, \mathbf{d}^*, b^*, \mathbf{u}^*, \mathbf{z}^*; \boldsymbol{\theta}^*, \alpha^*, \boldsymbol{\lambda}^*)$. Then $(\mathbf{f}^*, \mathbf{d}^*, b^*, \mathbf{u}^*)$ is a P -stationary point with $\gamma = 1/\rho_1$ and also a local minimizer of the problem (21).*

Remark 2. Similar to the working set T_* in (27) and the associated support vectors, the working set S_k in (46) and its limit set S_* renders *sparsity* in the combination of the kernels $\{\kappa_\ell\}$ for the MKL task, because the constraint (20b) can be interpreted as $\|\mathbf{d}\|_1 = 1$, an equality involving the ℓ_1 -norm, due to the nonnegativity condition (20a). Such an effect of sparsification can also be observed from our numerical example in the next section.

VI. SIMULATION ON SYNTHETIC DATA

In this section, we conduct numerical experiments using Matlab on a Dell laptop workstation with 64 GB of memory and an Intel Core i7 CPU of 2.5 GHz on synthetic data to demonstrate the sparsity and effectiveness of the proposed MKL- $L_{0/1}$ -SVM. The simulation presented here is very simple and by no means extensive. More precisely, we work with two-dimensional (planar) data ($n = 2$) for the purpose of easy visualization using a ten-kernel ($L = 10$) $L_{0/1}$ -SVM. The ten kernel functions are all Gaussian, see (12), with quite

arbitrarily chosen hyperparameters $\{\sigma_\ell\}$ which are listed in Table I. Some specific points are discussed next.

TABLE I: An arbitrary choice of the hyperparameters

σ_1	σ_2	σ_3	σ_4	σ_5
0.1400	0.0995	0.0161	0.0409	0.1561
σ_6	σ_7	σ_8	σ_9	σ_{10}
0.0156	0.1221	0.1175	0.0539	0.1247

(a) Stopping criteria. In the implementation, we terminate Algorithm 1 if the iterate $(\mathbf{f}^k, \mathbf{d}^k, b^k, \mathbf{u}^k, \mathbf{z}^k; \boldsymbol{\theta}^k, \alpha^k, \boldsymbol{\lambda}^k)$ satisfies the condition:

$$\max \{\beta_1^k, \beta_2^k, \beta_3^k, \beta_4^k, \beta_5^k, \beta_6^k, \beta_7^k, \beta_8^k\} < \text{tol}, \quad (53)$$

where the number $\text{tol} > 0$ is the tolerance level, and

$$\begin{aligned} \beta_1^k &:= \|\mathbf{u}^k - \mathbf{u}^{k-1}\|, & \beta_2^k &:= \|\mathbf{f}^k - \mathbf{f}^{k-1}\|, \\ \beta_3^k &:= |b^k - b^{k-1}|, & \beta_4^k &:= \|\mathbf{z}^k - \mathbf{z}^{k-1}\|, \\ \beta_5^k &:= \|\mathbf{d}^k - \mathbf{d}^{k-1}\|, & \beta_6^k &:= \|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1}\|, \\ \beta_7^k &:= |\alpha^k - \alpha^{k-1}|, & \beta_8^k &:= \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}\|. \end{aligned} \quad (54)$$

The condition says, in plain words, that two successive iterates are sufficiently close.

(b) Parameters setting. In Algorithm 1, the parameters C and ρ_1 characterize the working set (37) which is related to the number of support vectors, see (30) and the comments right after it. For simplicity, we have taken $\rho = \rho_1 = \rho_2 = \rho_3$ in the ADMM algorithm. In order to choose the two parameters, the standard 10-fold Cross Validation (CV) is employed on the training data, where C and ρ are selected from $\{2^{-2}, 2^{-1}, \dots, 2^7\}$ and $\{a^{-2}, a^{-1}, \dots, a^7\}$ with $a = \sqrt{2}$, respectively. The parameter combination with the highest CV accuracy is picked out. In addition, we set the maximum number of iterations $\text{max_iter} = 10^3$ in Algorithm 1 and the tolerance level $\text{tol} = 10^{-3}$ in (53).

For the starting point, we set $\mathbf{u}^0 = \boldsymbol{\lambda}^0 = \mathbf{0}$, $\boldsymbol{\theta}^0 = \mathbf{0}$, $\mathbf{f}^0 = \mathbf{0}$, $\mathbf{z}^0 = \mathbf{0}$, $\alpha^0 = 0$ and $\mathbf{d}^0 = \frac{1}{L} \mathbf{1}$, $b^0 = 1$ or -1 . The reason for such a choice is explained in the following. Let us call the objective function in (20) $J(\mathbf{f}, \mathbf{d}, b)$. Then we immediately notice that $J(\mathbf{0}, \frac{1}{L} \mathbf{1}, 1) = Cm_-$ and $J(\mathbf{0}, \frac{1}{L} \mathbf{1}, -1) = Cm_+$ where m_+ and m_- denote the numbers of positive and negative components in the label vector \mathbf{y} . Therefore, we should choose $(\mathbf{f}^0, \mathbf{d}^0, b^0)$ such that $J(\mathbf{f}^0, \mathbf{d}^0, b^0) \leq C \min\{m_+, m_-\}$.

(c) Evaluation criteria. To evaluate the classification performance of our $L_{0/1}$ -SVM, we report two criteria: the testing accuracy (TACC) and the number of support vectors (NSV) which is equal to $|T_*|$. Let $\{(\mathbf{x}_j^{\text{test}}, y_j^{\text{test}}) : j = 1, \dots, m_{\text{test}}\}$ be the testing data. The testing accuracy is defined as

$$\text{TACC} := 1 - \frac{1}{2m_{\text{test}}} \sum_{j=1}^{m_{\text{test}}} \left| \text{sign} \left(\sum_{\ell} f_{\ell}^*(\mathbf{x}_j^{\text{test}}) + b^* \right) - y_j^{\text{test}} \right|.$$

Here the quantity $\sum_{\ell} f_{\ell}^*(\mathbf{x}_j^{\text{test}})$ can be computed using (22f). More specifically, for each $\ell \in \mathbb{N}_L$ we can evaluate $f_{\ell}^*(\cdot) =$

$-d_\ell^* \sum_i \lambda_i^* y_i \kappa_\ell(\cdot, \mathbf{x}_i)$ on the test data using the convergent iterate produced by Algorithm 1.

(d) **Simulation result.** The planar data are generated randomly in the four quadrants and then (randomly) split into a training set and a testing set of equal size, i.e., $m = m_{\text{test}} = 100$. More specifically⁷, points in the first and third quadrants are given label 1, while points in the second and fourth quadrants are given label -1 . The parameters C , ρ are determined by the CV procedure described before on the training set. After that, the MKL- $L_{0/1}$ -SVM is optimized on the training set via Algorithm 1, and then the accuracy of the optimal classifier is verified using the testing set. The results are shown in Fig. 2 and Table II which also gives the best parameters selected during the CV. Notice that we have only reported the four d_ℓ^* 's which are significant in size (larger than 10^{-4}). The rest d_ℓ^* 's are set to zero. Additionally, we found that setting d_1^* and d_6^* (which are relatively small among the four) to 0 does not affect TACC. Therefore, our optimization procedure has claimed the following: the optimal linear combination of the ten candidate kernels in our $L_{0/1}$ -SVM involves essentially only *two* kernels, namely the ones with hyperparameters σ_3 and σ_5 . In other words, we have obtained great sparsity in the kernel combination in accordance with Remark 2. We comment at last that the testing accuracy of 90% is obviously not good enough and an improvement is possible via a better selection of the kernels and the hyperparameters.

TABLE II: Simulation results

C	ρ	d_1^*	d_3^*	d_5^*	d_6^*	TACC	NSV
16	4	0.0021	0.4575	0.5290	0.0113	0.90	100

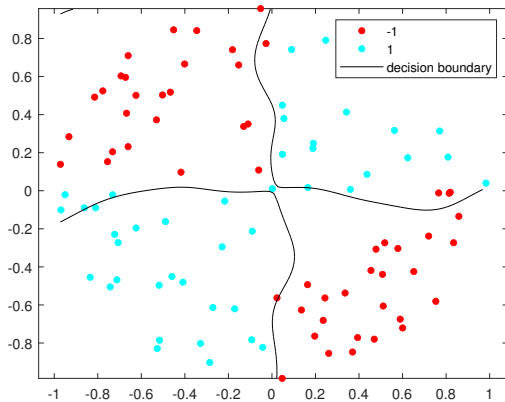


Fig. 2: Scatter diagram for the result of classification on the testing set with the decision boundary which corresponds to all \mathbf{x} (on a regular grid) that satisfy the equation $f^*(\mathbf{x}) + b^* = 0$.

⁷For details about data generation, the reader can refer to the section “Train SVM Classifier Using Custom Kernel” in the online documentation <https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>.

VII. CONCLUSIONS

We have considered a MKL task for the $L_{0/1}$ -SVM in order to select the best possible combination of some given kernel functions while minimizing a regularized $(0, 1)$ -loss function. Despite the nonconvex and nonsmooth nature of the objective function, we have provided a set of KKT-like first-order optimality conditions to characterize global and local minimizers. Numerically, we have developed an efficient ADMM solver to obtain a locally optimal solution to the MKL- $L_{0/1}$ -SVM problem. Preliminary simulation results have shown the effectiveness of our theory and algorithm. Future studies should regard the convergence analysis of Algorithm 1 with reference to [16], [17].

REFERENCES

- [1] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 2nd ed. Academic Press, 2020.
- [2] J. A. Suykens, “Support vector machines: a nonlinear modelling and control perspective,” *European Journal of Control*, vol. 7, no. 2-3, pp. 311–327, 2001.
- [3] S. Iplikci, “Support vector machines-based generalized predictive control,” *International Journal of Robust and Nonlinear Control*, vol. 16, no. 17, pp. 843–862, 2006.
- [4] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [5] H. Wang, Y. Shao, S. Zhou, C. Zhang, and N. Xiu, “Support vector machine classifier via $L_{0/1}$ soft-margin loss,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7253–7265, 2022.
- [6] M. Nikolova, “Description of the minimizers of least squares regularized with ℓ_0 -norm. uniqueness of the global minimizer,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 904–937, 2013.
- [7] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
- [8] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.
- [9] V. I. Paulsen and M. Raghupathi, *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016, vol. 152.
- [10] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer Science & Business Media, 2000.
- [11] G. Kimeldorf and G. Wahba, “Some results on Tchebycheffian spline functions,” *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.
- [12] B. Schölkopf and A. J. Smola, *Learning with Kernels*, ser. Adaptive Computation and Machine Learning. Cambridge: MIT Press, 2001, vol. 4.
- [13] K. Slavakis, P. Bouboulis, and S. Theodoridis, “Online learning in reproducing kernel Hilbert spaces,” in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 1, pp. 883–987.
- [14] B. Zhu and Y. Shi, “MKL- $L_{0/1}$ -SVM,” arXiv preprint: 2308.12016, 2023.
- [15] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [16] T. Lin, S. Ma, and S. Zhang, “On the global linear convergence of the admm with multiblock variables,” *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1478–1497, 2015.
- [17] M. Hong and Z.-Q. Luo, “On the linear convergence of the alternating direction method of multipliers,” *Mathematical Programming*, vol. 162, no. 1-2, pp. 165–199, 2017.