# On $\ell_0$-Regularized Sparse Multiple Kernel Learning for Feature Selection

**Anonymous authors**
**Paper under double-blind review**

## Abstract

This paper studies the direct use of $\ell_0$ regularization in sparse multiple kernel learning (SMKL) problems. We introduce an explicit $\ell_0$ norm on the kernel combination coefficients, leading to min-max mixed-integer optimization formulations, in which kernel selection is handled in the outer problem and a standard SVM is solved in the inner problem. We compare $\ell_0$ and the conventional $\ell_1$-based SMKL formulations. In the case of rank-one linear base kernels, we show that imposing an $\ell_0$ constraint on the kernel coefficients is equivalent to controlling the rank of the combined kernel. Compared to $\ell_1$ regularization, this provides a more direct control of the learned kernel with an interpretation as features selection. Furthermore, we investigate the impact of kernel scaling in $\ell_0$-regularized MKL and show that sparsity control and kernel scaling represent distinct modeling objectives, leading to an $\ell_0$–$\ell_p$ MKL framework that controls sparsity through $\ell_0$ regularization and kernel scaling through $\ell_p$ regularization. Extensive experiments on synthetic and UCI benchmark datasets demonstrate that incorporating $\ell_0$ regularization into SMKL leads to sparser kernels while achieving improved or comparable test accuracy. This contrasts with common knowledge in MKL where less sparse solutions are often observed to perform better.

## 1 Introduction

Multiple Kernel Learning (MKL) is a formulation of the kernel learning problems that learns a linear combination of a predefined dictionary of base positive semidefinite (PSD) kernels through optimization. It was originally proposed by Lanckriet et al. (2004) for support vector machines (SVMs) using semidefinite programming (SDP). MKL has been successfully applied to a wide range of machine learning tasks, including bioinformatics (Nascimento et al., 2016), visual recognition (Bucak et al., 2013), and signal processing (Subrahmanya & Shin, 2009). For an overview of general MKL methods, see (Gönen & Alpaydın, 2011).

Sparse Multiple Kernel Learning (SMKL) refers to a class of MKL methods that promote the selection of a sparse subset of kernels, motivated by the premise that only a subset of kernels is relevant to the learning task. Let $\mathcal{D}_M = \{K_1, \ldots, K_M\}$ denote a predefined dictionary of $M$ base kernels, and let $\mathcal{H}_m, m \in [M]$ denote the $m$-th associated reproducing kernel Hilbert space (RKHS). Let $\theta \in \mathbb{R}_+^M$ denote the kernel combination coefficients. The resulting combined kernel is defined as

$$K(\theta) := \sum_{m=1}^{M} \theta_m K_m. \tag{1}$$

For the combined kernel $K(\theta)$, any function in the associated RKHS can be represented as $f = \sum_{m=1}^{M} f_m$, where $f_m \in \mathcal{H}_m$ for all $m \in [M]$. Given a loss function $\ell$ and a sparsity-inducing regularization $\Omega(\theta)$, a general formulation of SMKL seeks a predictor $\sum_{m=1}^{M} f_m$ that fits the training data while controlling the complexity of the induced RKHS:

$$\min_{\substack{f_m \in \mathcal{H}_m, \\ \theta \geq 0}} \sum_{i=1}^{n} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\theta_m} + \Omega(\theta). \tag{2}$$

The parameter $\lambda$ controls the function norm. A common choice for $\Omega(\theta)$ is a $p$-norm regularizer (Kloft et al., 2011). It is well established in the MKL literature that $p > 1$ generally yields dense kernel combinations, whereas $p \leq 1$ promotes sparsity. More generally, sparsity-inducing mixed-norm regularizers have also been studied (Rakotomamonjy et al., 2011). From both theoretical and practical perspectives, SMKL can be viewed as a structured model selection mechanism that automatically identifies relevant kernels while discarding irrelevant ones, thereby improving interpretability. In the MKL literature, SMKL is typically associated with $\ell_1$-regularized MKL, where sparsity is induced through the convex constraint $\sum_{m=1}^{M} \theta_m = 1$ on the kernel coefficients (Rakotomamonjy et al., 2008; Sonnenburg et al., 2006; Xu et al., 2010; Wang & Tu, 2019). Equivalently, the constraint can be incorporated into the objective through a Lagrangian penalty $\Omega(\theta) = \gamma\|\theta\|_1$, where the parameter $\gamma \geq 0$ controls the sparsity level.

However, the sparsity in $\ell_1$-MKL arises from the simplex structure, which implicitly shrinks some coefficients to zero rather than explicitly modeling sparsity. Instead, $\ell_0$-regularized models are known to achieve superior support recovery in literature (Bertsimas et al., 2016). For a vector $\theta \in \mathbb{R}^p$, the $\ell_0$ pseudo-norm is defined as

$$\|\theta\|_0 := |\{i \in [p] : \theta_i \neq 0\}|\,,$$

i.e., the number of nonzero entries in $\theta$. In general, obtaining an exact solution to an $\ell_0$-regularized optimization problem is NP-hard (Natarajan, 1995) due to the exponentially large combinatorial search space, and exact approaches therefore typically rely on branch-and-bound–type algorithms. More recently, the mixed-integer optimization (MIO) of (Bertsimas et al., 2016) was applied to best subset selection and $\ell_0$-based sparse modeling (Bertsimas & Van Parys, 2020). Their results not only demonstrate that exact sparsity-inducing approaches can outperform $\ell_1$ relaxations in terms of support recovery and estimation accuracy, but also show that, with modern advances in computational optimization, such problems can be solved within reasonable time for statistical tasks of moderate scale.

In this paper, we investigate the role of the $\ell_0$ norm in sparse multiple kernel learning (SMKL), which remains largely unexplored in the MKL literature. We formulate the $\ell_0$-regularized MKL problem as a mixed-integer optimization problem with a nested outer–inner structure, where the outer (wrapper) problem optimizes kernel selection and the inner problem corresponds to a standard SVM. Under a commonly used kernel construction in MKL, where each feature corresponds to a rank-one base kernel, we show that $\ell_0$-MKL imposes an upper bound on the rank of the learned kernel. Motivated by the impact of kernel scaling and the inherent sparsity-inducing property of the $\ell_1$ norm, we further propose two extended formulations that impose $\ell_1$- and $\ell_2$-norm constraints on the kernel weights. These formulations provide complementary perspectives for disentangling sparsity and kernel weighting effects, and for understanding the role of the $\ell_0$ norm in kernel selection. Empirically, we solve the resulting mixed-integer problems to optimality on moderate-scale synthetic and real datasets. The results show that, compared with continuous $\ell_1$-MKL, $\ell_0$-MKL achieves higher sparsity without sacrificing predictive accuracy, while improving the recovery of informative kernels.

The remainder of the paper is organized as follows. In Section 2 we introduce the notation used throughout the paper. In Section 3, we review the SMKL literature. In Section 4, we establish connections between $\ell_0$-regularized MKL, low-rank kernel learning, and feature selection, and show that, under the widely used rank-one kernel construction, $\ell_0$ regularization can be interpreted as an explicit rank constraint on the learned kernel. In Section 5, we present the $\ell_0$-regularized MKL formulation and introduce two extended mixed-integer programming formulations to further investigate kernel sparsity and kernel scaling. In Section 6, we evaluate the three proposed $\ell_0$-MKL models, together with $\ell_1$-MKL as a baseline, in terms of support recovery, predictive accuracy, and computational efficiency. Finally, Section 7 concludes the paper.

## 2 Notation and preliminaries

Let $[n]$ denote the index set $\{1, \ldots, n\}$, and let $e_m \in \mathbb{R}^M$ denote the $m$-th canonical basis vector, i.e., the vector whose $m$-th component is equal to 1 and all other components are equal to 0. Let $I_n \in \mathbb{R}^{n\times n}$ denote the identity matrix. Throughout this paper, we consider a binary classification dataset consisting of $n$ samples and $M$ features. The data matrix is denoted by $X \in \mathbb{R}^{n\times M}$, where the $i$-th row $x_i^\top \in \mathbb{R}^M$ represents the $i$-th sample, and $x_{im}$ denotes the value of the $m$-th feature of sample $i$. The corresponding

label vector is denoted by $y \in \{-1, +1\}^n$, and $Y = \mathrm{diag}(y_1, \ldots, y_n)$ denotes the diagonal matrix of sample labels. For each feature $m \in [M]$, the vector of values of the $m$-th feature across all samples is denoted by $X_{:m} \in \mathbb{R}^n$. Using the canonical basis vector $e_m$, we have

$$X_{:m} = X e_m.$$

Let $\mathcal{D}_M = \{K_1, \ldots, K_M\}$ denote a dictionary of $M$ base kernels. Given kernel coefficients $\theta \in \mathbb{R}_+^M$, the combined kernel is defined as

$$K(\theta) = \sum_{m=1}^{M} \theta_m K_m.$$

The reproducing kernel Hilbert space (RKHS) associated with the base kernel $K_m$ is denoted by $\mathcal{H}_m$ and that associated to the combined kernel is denoted by $\mathcal{H}_{K(\theta)}$. For a kernel matrix $K$, $\mathrm{Tr}(K)$ denotes its trace.

## 3 Related Work on SMKL

Given a general loss function $\ell$, the conventional formulation of $\ell_1$-MKL can be written as follows:

$$\begin{aligned} \min_{\theta_m \geq 0} \min_{f_m \in \mathcal{H}_m} & \sum_{i=1}^{n} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\theta_m} \\ \text{s.t.} \quad & \|\theta\|_1 \leq 1 \end{aligned} \tag{3}$$

with the convention that, when $\theta_m = 0$, $\frac{\|f_m\|_{\mathcal{H}_m}^2}{\theta_m} = 0$ and $f_m = 0$ so that it contributes neither to the predictor nor to the regularization term. The $\theta_m$ appearing in the denominator of the regularization term comes from the characterization of the RKHS associated with the combined kernel. Indeed,

$$\|f\|_{\mathcal{H}_{K(\theta)}}^2 = \left\| \sum_{m=1}^{M} f_m \right\|_{\mathcal{H}_{K(\theta)}}^2 = \left\langle \sum_{m=1}^{M} f_m, \sum_{m=1}^{M} f_m \right\rangle_{\mathcal{H}_{K(\theta)}} = \sum_{m=1}^{M} \frac{\langle f_m, f_m \rangle_{\mathcal{H}_m}}{\theta_m},$$

where the last equality follows from the definition of the inner product in the RKHS induced by $K(\theta)$.

In the seminal work of Lanckriet et al. (2004), the kernel combination coefficients $\theta$ were constrained to be nonnegative in order to ensure that the learned kernel remained positive semidefinite. The additional convex combination constraint was introduced as a form of regularization, controlling the scale of the combined kernel and the induced RKHS.

Under the convex combination constraint, the feasible set can be viewed as the unit $\ell_1$ simplex since the nonnegative kernel coefficients must sum to one. It is well known that $\ell_1$ regularization promotes sparsity (Tibshirani, 1996; Iutzeler & Malick, 2020), and subsequent developments in the MKL literature revealed that the $\ell_1$ constraint not only regularizes the kernel scale but also induces sparse kernel selection. Bach (2008) showed that Problem equation 3 is equivalent to the group LASSO (Yuan & Lin, 2006) formulation equation 4:

$$\min_{f_m \in \mathcal{H}_m} \sum_{i=1}^{n} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \left(\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}\right)^2. \tag{4}$$

This equivalence follows from introducing a Lagrange multiplier for the constraint $\|\theta\|_1 \leq 1$ and analytically eliminating the kernel coefficients $\theta$. The properties of the group LASSO provide a fundamental understanding of the sparsity induced by $\ell_1$-MKL, suggesting that the observed kernel selection behavior arises from group-wise regularization effects that try to make some of the $\|f_m\|_{\mathcal{H}_m}$ small and even null rather than from an explicit sparsity mechanism that directly removes some of the kernel terms.

From the generalized additive model perspective (Bach, 2008), SMKL can be viewed as a variable selection problem, where each variable is associated with a base kernel (Rakotomamonjy et al., 2008; Bach, 2008). The simplest example is the rank-one linear kernel [1]

[1] Following the standard functional notation for kernels, let $k_m(\cdot, \cdot)$ denote the kernel function associated with the kernel matrix $K_m$. The functional version of the feature-based linear kernel is $k_m(x, x') = x_m x'_m$, where $x, x' \in \mathbb{R}^M$ are input vectors, and $x_m$ (resp. $x'_m$) denotes the $m$-th component of $x$ (resp. $x'$).

$$K_m = X_{:m} X_{:m}^\top, \tag{5}$$

where $X_{:m} \in \mathbb{R}^n$ denotes the vector of values of the $m$-th feature across all training samples. Under this construction, kernel selection amounts to selecting a subset of rank-one kernel components. Consequently, the learned kernel can be interpreted as a sparse low-rank kernel representation. In this sense, SMKL also performs feature selection.

Historically, however, sparsity in MKL has largely been viewed as a by-product of kernel-weight regularization property of $\ell_1$-MKL, rather than as a primary objective. The level of sparsity is typically governed indirectly by a convex regularization constraint. As a result, such formulations induce sparsity but do not provide explicit control over the degree of sparsity. As in the LASSO, the group LASSO mechanism underlying $\ell_1$-MKL may lead to biased coefficient estimates and suboptimal recovery of truly informative kernels, thereby motivating the development of more accurate sparse modeling approaches in kernel space.

A number of methods have been proposed to solve $\ell_1$-MKL efficiently. Sonnenburg et al. (2006) introduced a cutting-plane method, where the outer problem is formulated over the simplex and iteratively refined by adding valid cuts derived from solutions of the inner problem. This leads to a semi-infinite linear programming (SILP) formulation, which has been shown to scale well to large datasets. Subsequently, Rakotomamonjy et al. (2008) introduced the SimpleMKL algorithm, which employs a gradient-based update strategy to jointly optimize the SVM dual variables and kernel weights within a unified framework. Beyond $\ell_1$-MKL, another line of research investigates improved convex relaxations motivated by $\ell_0$ formulations (Bertsimas et al., 2025).

## 4 $\ell_0$-Regularized SMKL as Low-Rank Kernel Learning and Feature Selection

Following the general SMKL framework in which regularization is imposed on the kernel coefficients as stated in Equation equation 2, an $\ell_0$-regularized MKL formulation is obtained by imposing a cardinality constraint on the kernel coefficients:

$$\begin{aligned} \min_{\substack{f_m \in \mathcal{H}_m \\ \theta \geq 0}} \quad & \sum_{i=1}^{n} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\theta_m} \\ \text{s.t.} \quad & \|\theta\|_0 \leq s. \end{aligned} \tag{6}$$

An alternative formulation replaces the cardinality constraint with a Lagrangian penalty term. This yields the following unconstrained formulation with a sparsity parameter $\gamma$:

$$\min_{\substack{f_m \in \mathcal{H}_m \\ \theta \geq 0}} \sum_{i=1}^{n} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i)\right) + \lambda \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{\theta_m} + \gamma \|\theta\|_0. \tag{7}$$

Although both formulations are widely used, the cardinality-constrained formulation provides direct control over the number of selected kernels, whereas the penalized formulation controls sparsity indirectly through the regularization parameter. Due to the underlying discrete structure, $\ell_0$-based formulations are generally non-convex and computationally challenging, especially when aiming to solve them to optimality.

From the MKL perspective, under the commonly adopted rank-one kernel construction, $\ell_0$ regularization imposed on the kernel coefficients $\theta$ is equivalent to an explicit low-rank regularization. The following proposition establishes this connection.

**Proposition 1.** *Suppose that each base kernel $K_m$ is a rank-one positive semidefinite matrix. Then the rank of the combined kernel satisfies*

$$\operatorname{rank}\big(K(\theta)\big) \leq \|\theta\|_0. \tag{8}$$

*Proof.* Each $K_m$ is rank one and positive semidefinite; therefore, there exists a vector $u_m \in \mathbb{R}^n$ such that $K_m = u_m u_m^\top$. Let $S = \{m : \theta_m > 0\}$ denote the support of $\theta$. Then

$$K(\theta) = \sum_{m \in S} \theta_m u_m u_m^\top = U_S D U_S^\top,$$

where $U_S = [u_m]_{m\in S} \in \mathbb{R}^{n\times|S|}$ and $D = \mathrm{diag}(\theta_m)_{m\in S}$. Hence,

$$\mathrm{rank}(K(\theta)) \leq \mathrm{rank}(U_S) \leq |S| = \|\theta\|_0.$$

□

Proposition 1 shows that, with rank-one base kernels, $\ell_0$-regularized kernel selection can be interpreted as explicit low-rank regularization on the learned kernel matrix through kernel selection. Specifically, the rank of the learned kernel is bounded by $\|\theta\|_0$, so that imposing the cardinality constraint $\|\theta\|_0 \leq s$ guarantees a kernel representation in a subspace of dimension at most $s$. In MKL, the simplest example is the rank-one linear kernel given by equation 5.

In kernel learning, low-rank kernels have traditionally been obtained through kernel approximation techniques such as the Nyström method (Williams & Seeger, 2000). Such methods typically focus on approximating the Gram matrix and are generally constructed independently of the predictive learning objective. By contrast, the $\ell_0$-regularized Formulations equation 6 and equation 7 express low-rank kernel learning as a supervised optimization problem, where the feasible region comprises kernel combinations satisfying the prescribed sparsity level. The learned low-rank kernel is optimized with respect to the learning objective within the prescribed sparse model class, which creates the possibility of recovering informative kernel components underlying the data.

The rank-one kernel setting can be generalized to a broader class of kernel settings through the introduction of a feature map. The following considerations apply more generally whenever such a feature map is available. Let $\phi : \mathbb{R}^M \to \mathbb{R}^{M'}$ be a feature map transforming $x$ into $\phi(x)$, i.e., $\phi_m : \mathbb{R}^M \to \mathbb{R}$ , $m = 1, \ldots, M'$. Define

$$u_m = \begin{pmatrix} \phi_m(x_1) \\ \phi_m(x_2) \\ \vdots \\ \phi_m(x_n) \end{pmatrix}, \; m = 1, \ldots, M'.$$

The corresponding generalized rank-one kernel matrix [2] $K_m$ can be written as

$$K_m = u_m u_m^\top = \begin{pmatrix} \phi_m(x_1) \\ \vdots \\ \phi_m(x_n) \end{pmatrix} \begin{pmatrix} \phi_m(x_1) & \cdots & \phi_m(x_n) \end{pmatrix}. \tag{9}$$

The feature-map formulation extends the rank-one kernel construction to a broad class of nonlinear feature mappings. Under this generalized construction, kernel selection is equivalent to feature selection in the feature space induced by the feature map, whose dimension $M'$ determines the number of candidate kernels. The feature-wise linear kernel construction equation 5 corresponds to the feature map

$$\phi_m(x_i) = x_{im}, \qquad u_m = Xe_m = X_{:m}. \tag{10}$$

In the linear kernel setting, kernel selection is equivalent to input feature selection.

## 5 $\ell_0$-Regularized SMKL Formulations for SVMs

We now focus on $\ell_0$-regularized kernel selection for SVMs. We begin with the conventional $\ell_0$ formulation based on box constraints (the big-$\mathcal{M}$ method), referred to as the $\ell_0$-$\ell_\infty$ formulation. We then examine the role of box constraints in the MKL setting and discuss their implications for kernel scaling. This analysis motivates a family of $\ell_0$-$\ell_p$-regularized SMKL formulations, in which kernel selection and kernel scale control are modeled separately. Finally, all formulations are reformulated as min–max mixed-integer optimization problems and solved to optimality using a unified outer-approximation (OA) framework. Their empirical properties are compared in Section 6.

[2] The associated kernel function is $k_m(x, x') = \phi_m(x)\phi_m(x')$.

### 5.1 $\ell_0 - \ell_\infty$ SMKL Formulation for SVMs

In the context of Support Vector Machines (SVMs), the SMKL Formulation equation 2 is applied with the hinge loss:

$$\min_{\substack{f_m \in \mathcal{H}_m, \\ b \in \mathbb{R}, \theta \geq 0}} \sum_{i=1}^{n} \max\left(0, 1 - y_i\big(\sum_{m=1}^{M} f_m(x_i) + b\big)\right) + \lambda \sum_{m=1}^{M} \frac{\|f_m\|^2_{\mathcal{H}_m}}{\theta_m} + \Omega(\theta). \tag{11}$$

Introducing slack variables $\xi_i$, problem equation 11 can be equivalently written as equation 12:

$$\begin{aligned} \min_{\substack{f_m \in \mathcal{H}_m \\ b \in \mathbb{R},\ \xi_i \geq 0 \\ \theta \geq 0}} \quad & \frac{1}{2} \sum_{m=1}^{M} \frac{\|f_m\|^2_{\mathcal{H}_m}}{\theta_m} + C \sum_{i=1}^{n} \xi_i + \Omega(\theta) \\ \text{s.t.} \quad & y_i\Big(\sum_{m=1}^{M} f_m(x_i) + b\Big) \geq 1 - \xi_i, \quad i \in [n], \end{aligned} \tag{12}$$

where the hyperparameter $C$ is related to $\lambda$ by $C = \frac{1}{2\lambda}$. In practice, the optimization is typically formulated in terms of dual variables rather than the functions $f_m$, since for kernelized SVMs the representer theorem yields an equivalent finite-dimensional dual formulation that depends only on the kernel matrices. The corresponding dual formulation of the SMKL problem for SVMs is given by:

$$\begin{aligned} \min_{\theta \geq 0} \left\{ \Omega(\theta) + \max_{\alpha} \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \, \alpha^\top Y K_m Y \alpha \right) \right\} \\ \text{s.t.} \quad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{13}$$

Formulation equation 13 [3] defines a min–max structured problem, where the outer minimization over $\theta$ optimizes kernel weights, and the inner maximization over $\alpha$ is a standard convex quadratic program solvable by off-the-shelf SVM solvers.

An approach for modeling the $\ell_0$ constraint is the big-$\mathcal{M}$ formulation equation 14. This method introduces a binary indicator variable $z_m \in \{0, 1\}$ for each kernel coefficient $\theta_m$ and links the two variables through linear coupling constraints.

$$0 \leq \theta_m \leq \mathcal{M} z_m. \tag{14}$$

As a result, $\theta_m$ can be nonzero only when $z_m = 1$, while $\theta_m$ is forced to be zero whenever $z_m = 0$. Applying the big-$\mathcal{M}$ reformulation equation 14 to equation 13 yields the following $\ell_0$-regularized SMKL formulation:

$$\begin{aligned} \min_{\theta, z} \left\{ \gamma \sum_{m=1}^{M} z_m + \max_{\alpha} \quad \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \, \alpha^\top Y K_m Y \alpha \right) \right\} \\ \text{s.t.} \quad 0 \leq \theta_m \leq \mathcal{M} z_m, \quad z_m \in \{0, 1\} \\ 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{15}$$

where $\gamma > 0$ is a regularization hyperparameter controlling the degree of sparsity. Its selection will be discussed in the experimental section. From a kernel learning perspective, the introduction of discrete variables in the outer problem leads to a nonconvex feasible region. Consequently, the classical convex duality arguments commonly used in continuous MKL formulations (Kloft et al., 2011) no longer directly

[3] The bias term $b$ is recovered from the KKT optimality conditions using any support vector satisfying $0 < \alpha_i < C$. Let $k_\theta(x, x') = \sum_{m=1}^{M} \theta_m k_m(x, x')$ denote the combined kernel function associated with the learned kernel matrix $K(\theta)$. For a new point $x$, the predicted class is given by $\operatorname{sign}\Big(b + \sum_{i=1}^{n} \alpha_i y_i \, k_\theta(x_i, x)\Big)$.

apply. For selected kernels ($z_m = 1$), the big-$\mathcal{M}$ constraints impose the box constraints $0 \leq \theta_m \leq \mathcal{M}$, which can be interpreted as an $\ell_\infty$-type of bound on the kernel coefficients. The choice of the big-$\mathcal{M}$ constant is important and not unique, as it carries an implicit regularization effect through its influence on kernel scaling. The role of the big-$\mathcal{M}$ parameter is discussed further in Section 5.2. For convenience, formulation equation 15 is referred to as the $\ell_0$-$\ell_\infty$ formulation throughout the remainder of the paper.

### 5.2 Kernel Scale Control in Sparse MKL

In the MKL setting, the scale of the learned kernel that touches upon the complexity of the learned model and therefore its regularization. It requires additional consideration.

Let $K$ be a positive semidefinite kernel, let $K' = tK$ denote its scaled version for some scalar $t > 0$, and let $f \in \mathcal{H}_K = \mathcal{H}_{K'}$. Then, by the definition of the inner product in the RKHS associated with the scaled kernel,

$$\|f\|_{K'}^2 = \langle f, f\rangle_{K'} = \frac{1}{t}\langle f, f\rangle_K = \frac{1}{t}\|f\|_K^2. \tag{16}$$

This implies that scaling the kernel by $t$ is equivalent to rescaling the SVM regularization parameter from $C$ to $Ct$. Scaling the kernel changes the effective regularization of the SVM yet it does not change the functional space over which the classifier is sought. Consequently, variations in predictive performance may be caused by differences in kernel scale rather than by the sparsity mechanism itself, making kernel scale an important factor to control when comparing SMKL formulations.

In kernel learning, for a kernel $K$, a commonly used measure of kernel scale is the normalized trace $\frac{1}{n}\operatorname{Tr}(K)$ [4]. For a combined kernel, $\frac{1}{n}\operatorname{Tr}(K(\theta)) = \frac{1}{n}\sum_{m=1}^{M}\theta_m \operatorname{Tr}(K_m)$ . Hence, the kernel scale is a weighted sum of the traces of the base kernels. The feature-wise linear kernels considered in this paper are not necessarily normalized, as

$$\operatorname{Tr}(K_m) = X_{:m}^\top X_{:m} = \sum_{i=1}^{n} x_{im}^2 \ .$$

When the input features are normalized to have the same second-order moment, $\operatorname{Tr}(K_m)$ becomes a constant independent of $m$, and therefore

$$\frac{1}{n}\operatorname{Tr}(K(\theta)) \propto \|\theta\|_1.$$

Thus, the $\ell_1$-norm provides a natural mechanism for controlling the scale of the combined kernel in MKL.

In the $\ell_0$-$\ell_\infty$ SMKL formulation equation 15, although the $\mathcal{M}$-controlled box constraints provide a convenient mechanism for modeling sparsity, they offer little control over kernel scaling. Consider Formulation equation 15. For any selected kernel, i.e., any kernel with $\theta_m > 0$, for fixed dual variables $\alpha$, the inner objective

$$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{m=1}^{M}\theta_m \alpha^\top Y K_m Y \alpha \tag{17}$$

is linear in $\theta$. Since increasing the value of any selected $\theta_m$ improves equation 17, while each $\theta_m$ is bounded above by $\mathcal{M}$, the effective kernel scale is jointly controlled by $\mathcal{M}$ and the sparsity level, since $\|\theta\|_1 \leq \mathcal{M}\|\theta\|_0$. Because $\mathcal{M}$ only affects the active components of $\theta$, its choice lacks a clear interpretation from a kernel learning perspective and complicates fair comparisons across different SMKL formulations. $\ell_1$-MKL inherently provides kernel scale control, as the $\ell_1$ regularizer typically serves a dual purpose: promoting sparsity in kernel selection while simultaneously controlling the scale of the kernel combination. This motivates the introduction of additional regularization mechanisms to achieve explicit kernel scale control in $\ell_0$-regularized SMKL formulations.

[4] A commonly used measure of kernel scale is the normalized trace $\frac{1}{n}\operatorname{Tr}(K)$, which equals the average squared norm of the feature map or, from the Gaussian process perspective, the average prior variance at the data points. For normalized kernels (e.g., correlation or Gaussian kernels), $\operatorname{Tr}(K_m) = n$, so that $\frac{1}{n}\operatorname{Tr}(K) = \sum_{m=1}^{M}\theta_m$.

### 5.3 $\ell_0$-$\ell_p$ Regularized SMKL Formulations for SVMs

We aim to study the sparsity-inducing effect of $\ell_0$ regularization while explicitly controlling kernel scale through a separate regularization mechanism. Building on the SMKL formulation equation 13, we propose mixed-norm SMKL models that combine an $\ell_0$ sparsity term with an $\ell_p$-norm regularization on $\theta$ to regulate the scale of the learned kernel.

A natural extension of the $\ell_0$-regularized formulation is to incorporate the convex constraint $\|\theta\|_1 \leq 1$, yielding the following $\ell_0$-$\ell_1$ SMKL model equation 18:

$$\begin{aligned} \min_{\theta,z} \left\{ \gamma \sum_{m=1}^{M} z_m + \max_{\alpha} \quad \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \, \alpha^\top Y K_m Y \alpha \right) \right\} \\ \text{s.t.} \quad \|\theta\|_1 \leq 1, \\ 0 \leq \theta_m \leq \mathcal{M} z_m, \quad z_m \in \{0,1\}, \\ 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{18}$$

Alternatively, the $\ell_1$ regularization on $\theta$ can be imposed through a Lagrangian penalty. In $\ell_0$-$\ell_1$ SMKL Formulation equation 19, the parameter $\mu_1$ explicitly controls the kernel scale, while $\gamma$ controls sparsity.

$$\begin{aligned} \min_{\theta,z} \left\{ \gamma \sum_{m=1}^{M} z_m + \mu_1 \|\theta\|_1 + \max_{\alpha} \quad \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \, \alpha^\top Y K_m Y \alpha \right) \right\} \\ \text{s.t.} \quad 0 \leq \theta_m \leq \mathcal{M} z_m, \quad z_m \in \{0,1\}, \\ 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{19}$$

However, from the perspective of sparsity, incorporating an $\ell_1$ norm into an $\ell_0$-regularized formulation does not facilitate interpretation. In continuous optimization, the $\ell_1$ norm is well known to induce sparsity through shrinkage. Although the introduction of the $\ell_0$ term fundamentally changes the nature of the problem from continuous to discrete, the sparsity-inducing effect of the $\ell_1$ norm remains present. Consequently, it becomes difficult to disentangle whether the observed sparsity should be attributed to the explicit kernel selection induced by $\ell_0$ regularization or to the shrinkage effect of the $\ell_1$ regularization.

To address this issue, we focus on the $\ell_0$-$\ell_2$ SMKL Formulation equation 20, which provides a cleaner setting for disentangling sparsity and kernel scaling effects. Unlike $\ell_1$ regularization, $\ell_2$ regularization does not induce sparsity because the minimum of an $\ell_2$ norm is achieved by distributing $\theta$ mass over all the features. Yet, $\|\theta\|_2$ controls the scale of the kernel coefficients since $\|\theta\|_1/\sqrt{M} \leq \|\theta\|_2$. Therefore, any sparsity exhibited by the $\ell_0$-$\ell_2$ formulation can be unambiguously attributed to the effect of $\ell_0$ regularization.

$$\begin{aligned} \min_{\theta,z} \left\{ \gamma \sum_{m=1}^{M} z_m + \mu_2 \|\theta\|_2^2 + \max_{\alpha} \quad \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \, \alpha^\top Y K_m Y \alpha \right) \right\} \\ \text{s.t.} \quad 0 \leq \theta_m \leq \mathcal{M} z_m, \quad z_m \in \{0,1\}, \\ 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{20}$$

In practice, the $\ell_0$-$\ell_2$ formulation does not provide direct control over a prescribed kernel scale; for example, achieving a unit kernel scale $\sum_{m=1}^{M} \theta_m = 1$ may require additional parameter tuning.

Notably, the proposed $\ell_0$-$\ell_p$ regularized SMKL formulations share a structural similarity with the Elastic Net (Zou & Hastie, 2005), in that sparsity and coefficient scaling are controlled by separate regularization mechanisms.

In practice, we impose the bound $0 \leq \theta_m \leq 1$ for all formulations, which is equivalent to setting $\mathcal{M} = 1$ in all $\ell_0$-regularized models and yields the unit box constraint $\|\theta\|_\infty \leq 1$. This choice provides a common

normalization and places the kernel coefficients on a scale comparable to that of the unit-$\ell_1$ constraint used in classical $\ell_1$-MKL. The $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$, and $\ell_0$-$\ell_2$ formulations can all be solved using modern mixed-integer optimization solvers. The $\ell_0$-$\ell_\infty$ and $\ell_0$-$\ell_1$ formulations give rise to a mixed-integer linear programming (MILP) outer problem, whereas the $\ell_0$-$\ell_2$ formulation leads to a mixed-integer quadratic programming (MIQP) outer problem. In general, solving mixed-integer optimization problems is computationally challenging due to the combinatorial nature of the discrete decision variables. We now present an Outer Approximation (OA) algorithm for solving such mixed-integer problems.

### 5.4 Outer Approximation Algorithm

We apply the Outer Approximation (OA) Algorithm 1 (Duran & Grossmann, 1986) for solving the $\ell_0$ regularized SMKL problems for SVMs. The outer problem is a mixed-integer program, while the inner problem is a standard SVM. Optimality is achieved via a cutting-plane scheme. The detailed formulations of the master problem, subproblem, and cutting-plane generation procedure, as well as the convergence proof of the proposed algorithm, are presented in the Appendix A.

**Algorithm 1** $\ell_0$-regularized SMKL for SVMs

**Require:** tolerance $\varepsilon > 0$
Initialize the set of cuts $\mathcal{A} \leftarrow \emptyset$, the iteration counter $k \leftarrow 0$, $LB \leftarrow -\infty$, $UB \leftarrow +\infty$
**while** $UB - LB > \varepsilon$ **do**
  Solve master MIP over $(\theta, z, \eta)$ using cuts $\mathcal{A}$, obtain $(\theta^k, z^k, \eta^k)$
  $LB \leftarrow \eta^k + \gamma \sum_m z_m^k$
  Solve SVM dual $\alpha^k = \arg\max_{\alpha \in \mathcal{Y}} \left( \sum_i \alpha_i - \frac{1}{2} \sum_m \theta_m^k \alpha^\top Y K_m Y \alpha \right)$
  Compute $h(\theta^k, \alpha^k) = \sum_i \alpha_i^k - \frac{1}{2} \sum_m \theta_m^k \alpha^{k\top} Y K_m Y \alpha^k$
  $UB \leftarrow \min(UB,\ h(\theta^k, \alpha^k) + \gamma \sum z_m^k)$
  **if** $UB - LB \leq \varepsilon$ **then**
    **break**
  **else**
    Add cut induced by $\alpha^k$ to $\mathcal{A}$
  **end if**
  $k \leftarrow k + 1$
**end while**

Algorithm 1 provides a general solution framework for solving $\ell_0$ regularized SMKL formulations for SVMs. The overall computational complexity of the OA algorithm is dominated by solving a sequence of mixed-integer programs (MIPs). In each iteration, the master problem is a MIP, which is typically solved via branch-and-bound methods with exponential worst-case complexity in the number of binary variables. The subproblem corresponds to solving a standard SVM, whose complexity is typically polynomial in the number of samples $n$. Let $T$ denote the number of outer iterations. The total computational cost can thus be expressed as $\mathcal{O}(T \cdot (C_{\text{MIP}} + C_{\text{SVM}}))$, where $C_{\text{MIP}}$ and $C_{\text{SVM}}$ denote the computational costs of solving the mixed-integer master problem and the SVM subproblem, respectively. In practice, the MIP component dominates the overall runtime.

Next, we instantiate this framework for the $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$, and compare their properties with those of $\ell_1$-based methods. Note that, in Formulation equation 18, if the $\ell_0$ regularization is removed and the $\ell_1$ norm is adopted (i.e., $p = 1$), the model reduces to the SILP-based implementation of $\ell_1$-MKL (Sonnenburg et al., 2006), which we use as a baseline for comparisons.

## 6 Computational Experiments

We aim to study the role of $\ell_0$ regularization in the MKL setting, in comparison with the $\ell_1$-MKL baseline. We conduct experiments on four models – $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$, $\ell_0$-$\ell_2$, and $\ell_1$ (SILP) – on both synthetic and real-world datasets for binary classification with SVMs.

### 6.1 Experimental Setup

#### 6.1.1 Computational Environment

All experiments were conducted on a machine equipped with an Intel(R) Xeon(R) CPU E5-2670 @ 2.60 GHz. Each task was allocated 16 GB of RAM and 12 CPU cores. The mathematical models are implemented in Python 3.10. Mixed-integer optimization problems are solved using Gurobi 11 (Gurobi Optimization, LLC, 2026), while SVM subproblems are handled using LIBSVM.

#### 6.1.2 Synthetic Data

To evaluate our algorithm in the SMKL setting, where only a small subset of kernels is assumed to be involved in the label-generating process, we generate synthetic data with redundant feature information. Specifically, we construct datasets of size $n \times M$, where $n$ denotes the number of samples and $M$ the number of features, for binary classification with SVMs. Let `ssr` denote the support size ratio, which controls the proportion of informative kernels, and define $s = \max(1, \mathrm{round}(M \cdot \texttt{ssr}))$ as the number of informative features. A set of informative features $\mathcal{S} \subset \{1, \ldots, M\}$ with $|\mathcal{S}| = s$ is randomly selected as the informative support, and the remaining indices form the degraded and redundant set $\mathcal{R} = \{1, \ldots, M\} \setminus \mathcal{S}$. For each informative feature $m \in \mathcal{S}$, we generate an independent Gaussian feature vector

$$X_{:m} \sim \mathcal{N}(0, I_n),$$

where $n$ is the number of samples. For each redundant feature $j \in \mathcal{R}$, an informative feature $m \in \mathcal{S}$ is selected uniformly at random as its source feature. The degraded and redundant feature (we will just say redundant henceforth) is then generated as a noisy copy of this source feature:

$$X_{:j} = \sqrt{\mathrm{corr}}\, X_{:m} + \sqrt{1 - \mathrm{corr}}\, \epsilon_n,$$

where $\epsilon_n \sim \mathcal{N}(0, I_n)$ is an independent Gaussian noise vector. After constructing all informative and redundant features, an additional Gaussian perturbation is added to the entire feature matrix to achieve a prescribed signal-to-noise ratio (SNR = 2), followed by feature standardization. Labels are generated from a linear latent-score model defined only on the informative features. A sparse ground-truth weight vector $w_{\mathrm{true}} \in \mathbb{R}^M$ is generated with support restricted to $\mathcal{S}$, and the latent score is computed as $s_i = x_i^\top w_{\mathrm{true}} + \hat{\epsilon}_i$, where $\hat{\epsilon}_i \sim \mathcal{N}(0, \sigma^2_{\mathrm{label}})$ is an independent Gaussian label noise term. Binary labels are then obtained by thresholding the latent score via $y_i = \mathrm{sign}(s_i)$.

#### 6.1.3 Kernel Dictionary Construction

Given an input data matrix $X \in \mathbb{R}^{n \times M}$, where $n$ is the number of samples and $M$ is the number of features, we construct a dictionary of linear base kernels by associating each feature dimension with one kernel. Specifically, let $X_{:m} \in \mathbb{R}^n$ denote the vector corresponding to the $m$-th feature. The associated linear kernel matrix is defined as

$$K_m = X_{:m} X_{:m}^\top, \quad m = 1, \ldots, M,$$

resulting in a dictionary of $M$ rank-one kernels. Under this construction, the feature-level structure directly induces a kernel-level structure. In particular, the informative features correspond to *informative (true) kernels*, while the redundant features give rise to *redundant kernels* in the dictionary.

#### 6.1.4 Evaluation Metrics

To evaluate the ability of different methods to recover informative kernels, we report three metrics: precision, test classification accuracy, and true mass.

The *test classification accuracy* is defined as

$$\frac{1}{n^{(\mathrm{test})}} \sum_{i=1}^{n^{(\mathrm{test})}} \mathbf{1}\left(y_i^{(\mathrm{predicted})} = y_i\right),$$

where $\mathbf{1}(\cdot)$ denotes the indicator function, $n^{(\text{test})}$ denotes the number of test samples and $y_i^{(\text{predicted})}$ and $y_i$ denote the predicted and true labels of the $i$-th test sample, respectively. For training and evaluation, we perform the train–test split at the sample level and construct kernel matrices accordingly. The data are split into training and test sets with a ratio of 4:1. Given a partition of the samples, we extract submatrices $K_m^{(\text{train})}$ and $K_m^{(\text{test})}$ from each $K_m$. The corresponding combined kernels $K_\theta^{(\text{train})}$ and $K_\theta^{(\text{test})}$ are then formed in the same manner. This modular construction ensures consistency across data splits and facilitates kernel-based learning.

The *precision* measures the proportion of selected kernels that are truly informative, defined as

$$\text{Precision} = \frac{|\text{supp}(\theta) \cap \mathcal{S}|}{|\text{supp}(\theta)|},$$

where $\text{supp}(\theta) = \{m : \theta_m > 0\}$. A higher precision indicates that a larger fraction of the selected kernels are informative. Throughout this section, sparsity is defined as $\|\theta\|_0$, i.e., the number of selected kernels.

The *true mass* is defined as the proportion of total weight assigned to informative kernels:

$$\text{TrueMass} = \frac{\sum_{m \in \mathcal{S}} \theta_m}{\sum_{m=1}^{M} \theta_m}$$

A larger true mass indicates that more weight is assigned to the informative kernels.

All experiments are repeated 10 times using different random seeds for train–test splits, and the mean and standard deviation are reported.

### 6.2 Recovery of Informative Kernels

We evaluate four models – $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$ (Formulation equation 19), $\ell_0$-$\ell_2$, and $\ell_1$ (SILP) – in terms of their ability to recover informative kernels on synthetic datasets consisting of 200 samples and 20 features. Because of the different modeling properties of these methods, we adopt the number of selected kernels as a unified criterion to enable a fair comparison across models. The hyperparameters $\mu_1$ for $\ell_0$-$\ell_1$ and $\mu_2$ for $\ell_0$-$\ell_2$ is selected from the range $[1, 10^3]$, while the sparsity penalty $\gamma$ is chosen from $[10^{-2}, 10^2]$. All models are formulated using the Lagrangian penalty form to make them comparable and are evaluated under the same kernel dictionary and data generation process. The SVM regularization parameter is fixed at $C = 100$. For the mixed-integer optimization procedure, the relative optimality gap of the OA algorithm 1 is defined as $\frac{UB-LB}{|UB|+1e^{-8}}$, and is set to 0.01, with a maximum of 1000 iterations.

Several observations regarding sparse pattern recovery (as measured by precision) and test accuracy can be made from Figures 1 and 2. In terms of precision, $\ell_0$-$\ell_1$ performs best, followed by $\ell_0$-$\ell_2$, while $\ell_1$ and $\ell_0$-$\ell_\infty$ achieve comparable lower performance. In particular, for each setting, the $\ell_0$-$\ell_1$ method consistently achieves solutions whose support size is closest to the true number of informative kernels while also attaining the highest precision. This indicates that the proportion of informative kernels within the selected set is maximized by $\ell_0$-$\ell_1$, demonstrating its superior ability to recover the true kernel support. The $\ell_0$-$\ell_2$ method also outperforms $\ell_1$ in terms of precision and sparsity. However, in terms of sparsity too, it remains inferior to $\ell_0$-$\ell_1$.

Within the $\ell_0$ family, augmenting the $\ell_0$ formulation with an additional kernel-scale regularization term consistently improves precision, with both $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$ outperforming $\ell_0$-$\ell_\infty$. We observe that even with a large $\ell_1$ penalty, it remains difficult for $\ell_1$-based MKL to approach the true sparsity level. In contrast, all $\ell_0$-based variants overcome this limitation and achieve both higher precision and lower sparsity than $\ell_1$-based MKL through explicit $\ell_0$ regularization. It can also be seen that the true sparsity level is always approached by superior values in the number of active features i.e., by lower sparsity.

For each value of the signal correlation `corr`, a clear trend can be observed: as the proportion of active kernels, `ssr`, decreases, the overall precision deteriorates. For a fixed `ssr`, the precision is little affected by the value of `corr`.

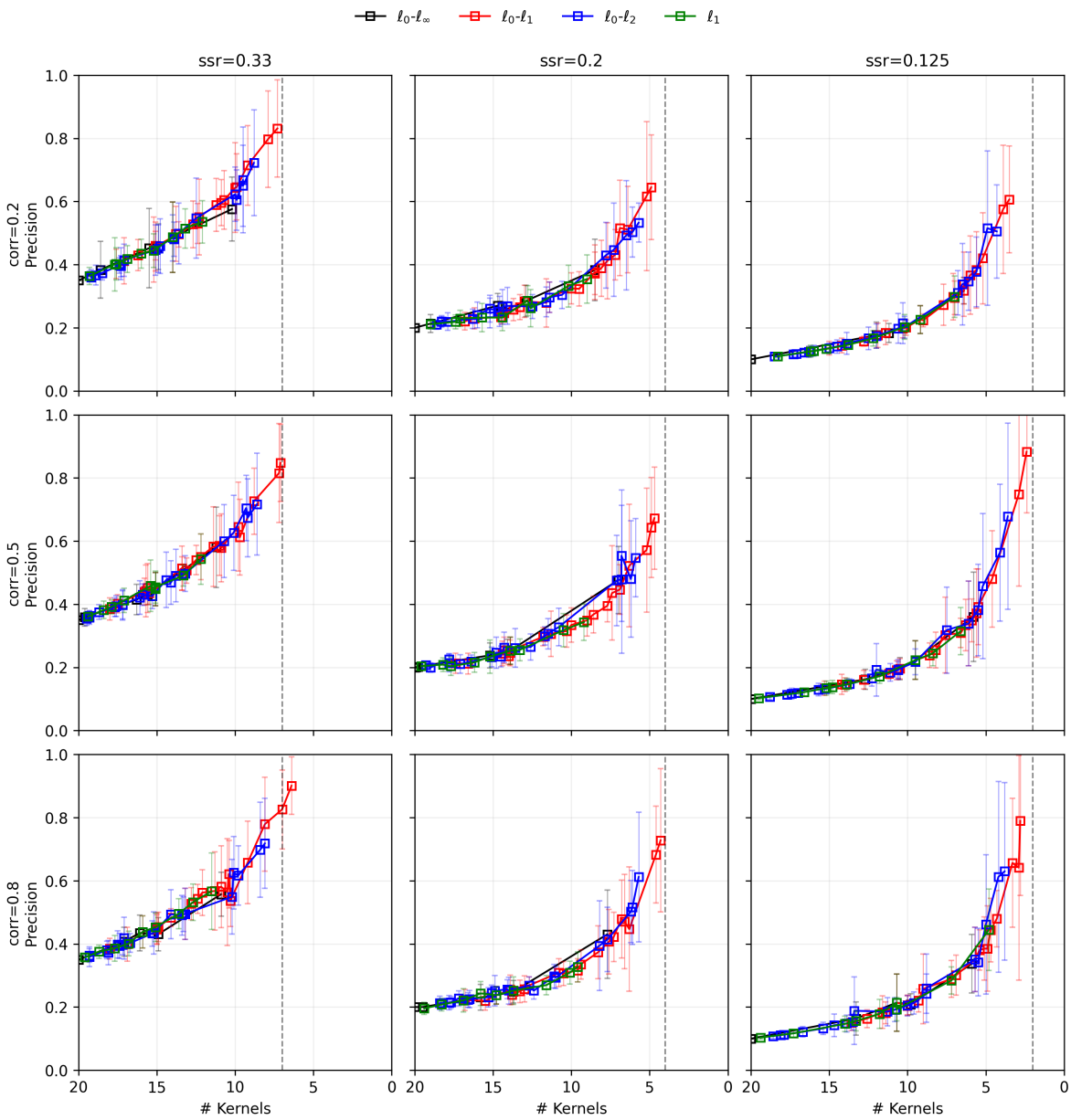


Figure 1: Precision comparison of $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$, $\ell_0$-$\ell_2$, and $\ell_1$ under different settings of the regularization hyperparameters $\mu_1$, $\mu_2$, and $\gamma$. Rows correspond to fixed correlation levels with varying `ssr`, while columns correspond to fixed `ssr` with varying `corr`. All experiments are repeated 10 times using different random train–test splits, and the reported results show the mean and standard deviation over the 10 runs.

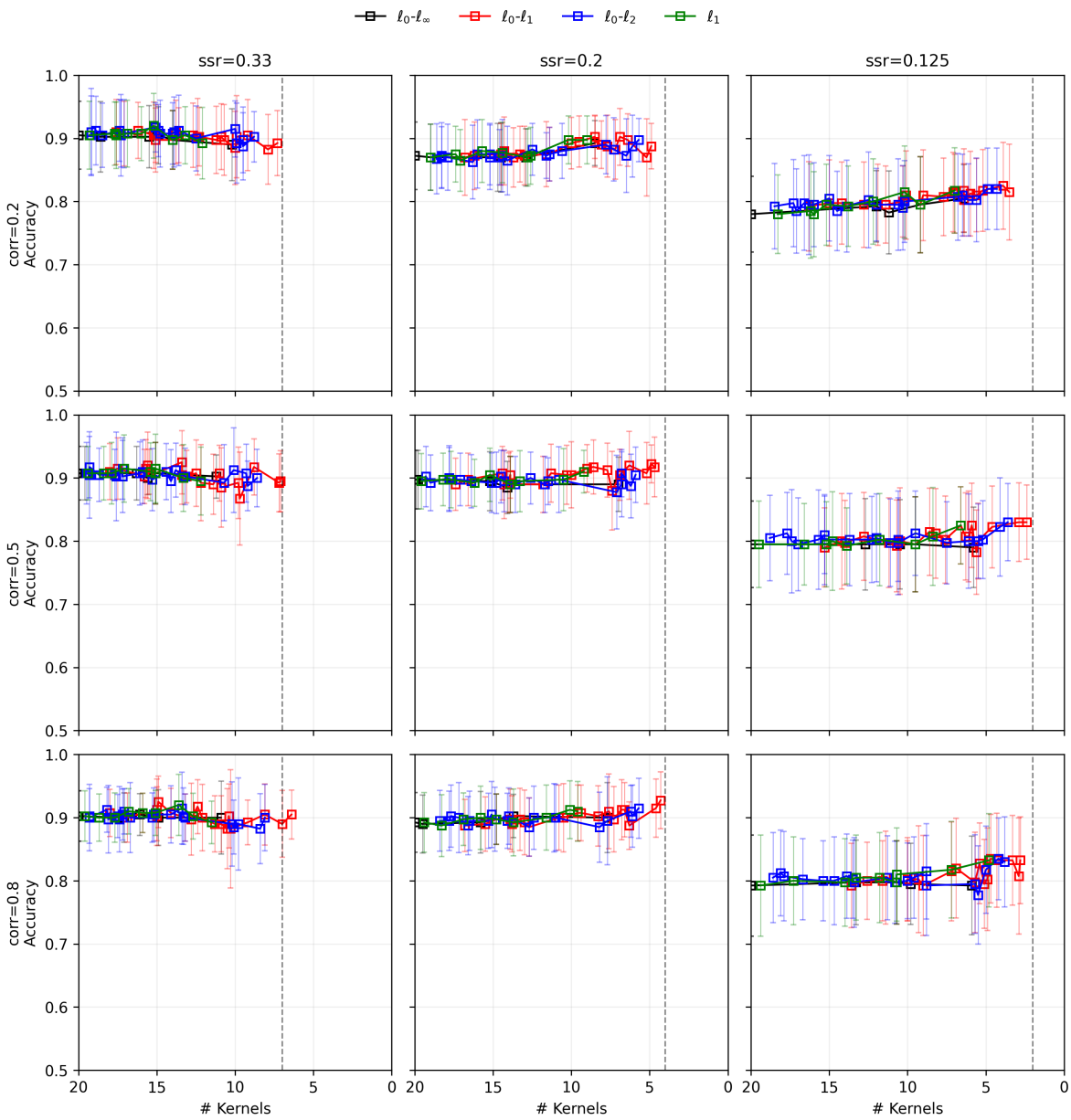


Figure 2: Accuracy comparison of $\ell_0$-$\ell_\infty$, $\ell_0$-$\ell_1$, $\ell_0$-$\ell_2$, and $\ell_1$ under different settings of the regularization hyperparameters $\mu_1$, $\mu_2$, and $\gamma$. Rows correspond to fixed correlation levels with varying `ssr`, while columns correspond to fixed `ssr` with varying `corr`. All experiments are repeated 10 times using different random train–test splits, and the reported results show the mean and standard deviation over the 10 runs.

In terms of classification accuracy, as shown in Figure 2, we observe that the performance differences among the four methods are small provided there are sufficient kernels. In other terms, all methods recover the same information for classification, yet at different sparsity levels. The test accuracy offers a similar trend as the precision with respect to the proportion of informative kernels, `ssr`. As `ssr` decreases, the test accuracy of all methods also decreases. This suggests that when only a small fraction of kernels is informative, recovering the complete informative kernel set becomes more challenging, which may in turn degrade the predictive performance. The test accuracy is relatively insensitive to changes in `corr`.

However, across all settings, methods incorporating $\ell_0$ regularization are able to achieve comparable accuracy while producing significantly sparser solutions that are closer to the true support size. This illustrates that $\ell_0$-based methods can identify more compact kernel combinations without sacrificing predictive performance.

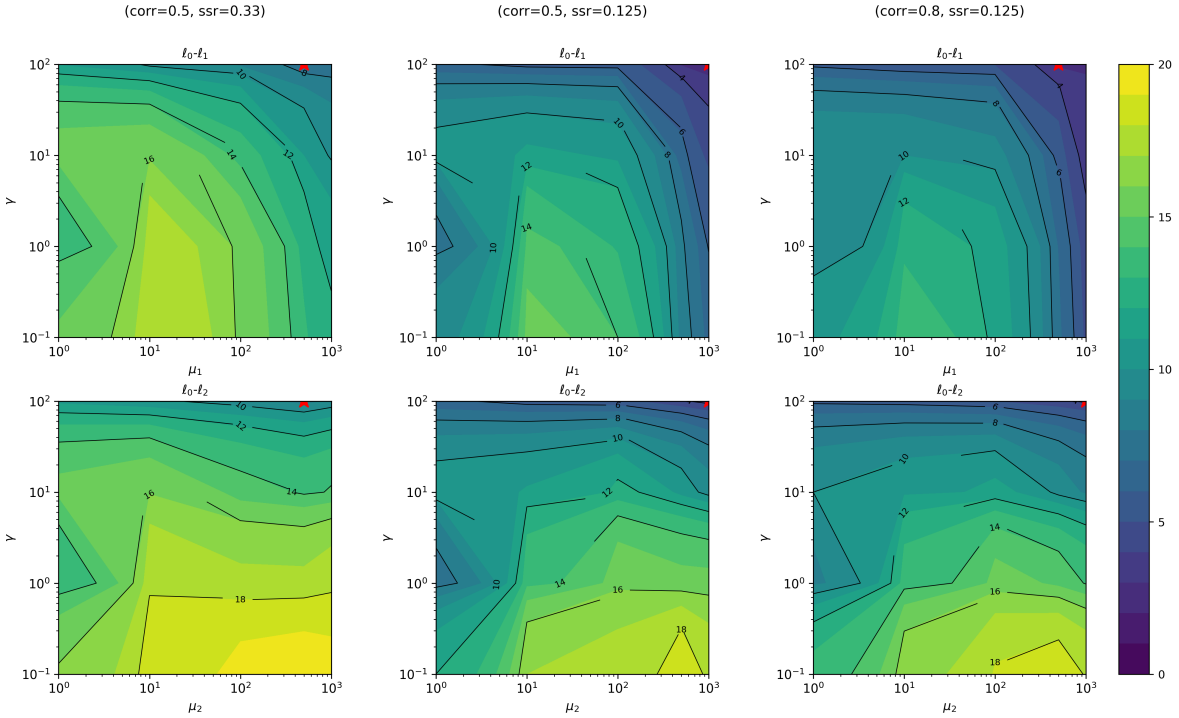


Figure 3: Average number of selected kernels obtained by $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$ under different settings of the regularization hyperparameters $\mu_1$, $\mu_2$ and $\gamma$. Columns correspond to different $(\text{corr}, \text{ssr})$ configurations. Each experiment is repeated 10 times using different random train–test splits, and the reported values represent the mean over the 10 runs.

To further investigate the impact of different kernel-scale control mechanisms, we compare the $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$ formulations by analyzing the relationship between the hyperparameters $\mu_1$, $\mu_2$, and $\gamma$, and the resulting number of selected kernels. As shown in Figure 3, the contour lines of the $\ell_0$-$\ell_2$ formulation are predominantly horizontal, indicating that the number of selected kernels is mainly determined by the $\ell_0$ regularization parameter $\gamma$. This is consistent with the fact that the $\ell_2$ regularization does not introduce additional sparsity. For the $\ell_0$-$\ell_1$ formulation, the contour lines remain nearly horizontal until the $\ell_1$ regularization becomes sufficiently strong, suggesting that sparsity is primarily induced by the $\ell_0$ regularization over a wide range of $\mu_1$. Only when $\mu_1$ becomes very large does the $\ell_1$ regularization also begin to influence the sparsity of the solution. Together, these observations show that, for both the $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$ formulations, sparsity is primarily controlled by the $\ell_0$ regularization parameter $\gamma$. The $\ell_1$ regularization mainly serves as a mechanism for controlling the kernel scale and indirectly induces sparsity for large $\mu_1$.

Overall, on this noisy classification test set, $\ell_0$-$\ell_1$ achieves the best performance in recovering the true informative kernels among all four methods. Since the $\ell_1$ norm is the standard mechanism for controlling the kernel scale in the MKL literature, these results suggest that combining $\ell_0$ regularization with $\ell_1$-based

kernel scale control is particularly effective. The comparison between the $\ell_0$-$\ell_1$ and $\ell_0$-$\ell_2$ formulations suggests that, in the MKL setting, the $\ell_1$ norm is an effective kernel scale controller, improving support recovery while maintaining the sparsity induced by the $\ell_0$ regularization. In the following experiments, we focus on the comparison between $\ell_0$-$\ell_1$ and $\ell_1$.

### 6.3 Classification Accuracy under Fixed Kernel Scale

In Section 6.2, we observed that with the Lagrangian penalty formulation, methods incorporating $\ell_0$ regularization are able to achieve solutions that are significantly sparser than $\ell_1$-based methods, while maintaining comparable classification accuracy.

A key difference between the penalized formulation and the explicit constraint formulation lies in how the kernel scale is controlled. Under the penalized form, the kernel scale is not explicitly constrained and is only implicitly regulated through the Lagrangian penalty, making it difficult to align across models with different parameterizations. For convex optimization problems, the Lagrangian and constrained formulations are equivalent. However, this equivalence is no longer guaranteed when the binary variables $z$ are introduced in Formulations equation 19 and equation 18, making the problem non-convex. The constrained formulation is adopted solely to provide a common kernel scale, allowing a direct comparison of different kernel-scale control mechanisms.

We now compare the $\ell_0$-$\ell_1$ and $\ell_1$ methods under a common kernel scale constraint of the form $\sum_m \theta_m \leq 1$. Both formulations can be viewed as extensions of Formulation equation 18. The classical $\ell_1$-MKL formulation is recovered by removing the $\ell_0$ regularization and the associated binary variables. In the previous section, we observed that $\ell_0$-$\ell_1$ achieves better recovery performance in terms of precision. Here, we investigate whether this advantage carries over in the presence of a kernel scale control constraint. The experiments were conducted on the same datasets as those used in Section 6.2. The sparsity penalty $\gamma$ is chosen from $[10^{-2}, 10^2]$. For different values of $\gamma$, we report, among the solutions of $\ell_0$-$\ell_1$ that are sparser than those obtained by $\ell_1$, the one achieving the highest test accuracy.

| ssr | corr | #Kernels | | Accuracy | | True mass | |
|---|---|---|---|---|---|---|---|
| | | $\ell_0$-$\ell_1$ | $\ell_1$ | $\ell_0$-$\ell_1$ | $\ell_1$ | $\ell_0$-$\ell_1$ | $\ell_1$ |
| 0.125 | 0.2 | **9.7** ± 1.49 | 9.9 ± 1.37 | **0.890** ± 0.04 | 0.882 ± 0.04 | 0.767 ± 0.05 | **0.771** ± 0.06 |
| 0.125 | 0.5 | **3.9** ± 0.99 | 9.0 ± 1.63 | **0.900** ± 0.06 | 0.878 ± 0.06 | **0.842** ± 0.09 | 0.771 ± 0.06 |
| 0.125 | 0.8 | **7.4** ± 1.78 | 7.9 ± 1.60 | **0.895** ± 0.06 | 0.885 ± 0.06 | **0.794** ± 0.06 | 0.770 ± 0.09 |
| 0.2 | 0.2 | **9.0** ± 1.49 | 9.1 ± 1.60 | **0.893** ± 0.04 | 0.887 ± 0.04 | **0.839** ± 0.03 | 0.828 ± 0.03 |
| 0.2 | 0.5 | **7.9** ± 1.37 | 8.1 ± 1.52 | **0.900** ± 0.03 | 0.897 ± 0.03 | **0.846** ± 0.03 | 0.837 ± 0.04 |
| 0.2 | 0.8 | **4.2** ± 0.79 | 8.0 ± 1.15 | **0.912** ± 0.03 | 0.887 ± 0.05 | **0.902** ± 0.06 | 0.827 ± 0.06 |
| 0.33 | 0.2 | **5.0** ± 1.05 | 9.4 ± 1.84 | **0.882** ± 0.05 | 0.870 ± 0.06 | **0.914** ± 0.14 | 0.841 ± 0.06 |
| 0.33 | 0.5 | **5.1** ± 0.74 | 8.2 ± 0.92 | **0.880** ± 0.04 | 0.870 ± 0.06 | **0.925** ± 0.06 | 0.841 ± 0.08 |
| 0.33 | 0.8 | **4.6** ± 0.97 | 7.7 ± 0.82 | **0.887** ± 0.06 | 0.882 ± 0.07 | **0.950** ± 0.08 | 0.864 ± 0.06 |

Table 1: Performance of $\ell_0$-$\ell_1$ and $\ell_1$ on synthetic datasets controlled by ssr and corr, in terms of the number of selected kernels, accuracy, and true mass. All experiments are repeated 10 times using different random seeds for train–test splits, and the mean and standard deviation are reported.

In Table 1, we observe that when the total kernel scale is constrained to be less than 1, the $\ell_0$-$\ell_1$ method consistently outperforms $\ell_1$ in terms of sparsity, test accuracy, and true mass. This is consistent with our findings in Section 6.2: under a fixed scale, incorporating an $\ell_0$ constraint increases the weight assigned to informative kernels, which in turn leads to improved classification accuracy on the test set. Importantly, this improvement in accuracy is observed consistently across all datasets except one that, not coincidentally, has the most degraded signal and smallest number of active features.

Overall, these results suggest that, while $\ell_0$ directly models sparsity, such explicit sparsity modeling in the MKL framework leads to more effective kernel selection under a fixed scale. In other terms, the $\ell_0$ regularization cannot be replaced by a constraint on the kernel scale. It has been observed on this synthetic

benchmark that $\ell_0$ enables the model to achieve higher accuracy using fewer kernels, which contrasts with observations made in the MKL literature (Kloft et al., 2011; Cortes et al., 2012) that using more kernels typically leads to better performance.

### 6.4 Classification Accuracy on Real-world Datasets

We now investigate the behavior of the different methods on real-world datasets, specifically the four binary classification datasets from the UCI repository with moderate dimensionality (at most 60 features) namely, *Ionosphere*(34), *Sonar*(60), *Spambase*(57), and *WDBC*(30). For datasets with a large number of samples, we perform downsampling to 500 instances while preserving the class distribution, and use the resulting data to train the SVM models. For each experiment, we repeat the runs 10 times using different random seeds. We compare $\ell_1$-MKL and $\ell_0$-$\ell_1$ MKL (Formulation equation 18) in terms of classification accuracy and kernel selection sparsity, reporting the mean and standard deviation.

The SVM regularization parameter is fixed at $C = 100$. The $\ell_0$ regularization parameter $\gamma$ is selected by 3-fold cross-validation on the training set from the candidate set $0.01, 0.05, 0.1$. The optimality gap of the MILP main loop is set to 0.01, with a maximum of 1000 iterations.

| Dataset | Accuracy | | #Kernels | |
|---|---|---|---|---|
| | $\ell_0$-$\ell_1$ | $\ell_1$ | $\ell_0$-$\ell_1$ | $\ell_1$ |
| Ionosphere(34) | **0.893** ± 0.014 | 0.883 ± 0.024 | **20.8** ± 1.3 | 21.8 ± 2.0 |
| Sonar(60) | **0.757** ± 0.074 | 0.745 ± 0.072 | **36.2** ± 2.8 | 37.4 ± 3.2 |
| Spambase(57) | 0.906 ± 0.030 | 0.906 ± 0.032 | **37.2** ± 4.3 | 38.2 ± 4.2 |
| WDBC(30) | **0.960** ± 0.015 | 0.959 ± 0.017 | 13.4 ± 1.0 | 13.4 ± 0.8 |

Table 2: Performance comparison between $\ell_0$-$\ell_1$ and $\ell_1$ MKL on UCI datasets. Results are reported as mean ± standard deviation over 10 runs.

From Table 2, we observe that $\ell_0$-$\ell_1$ MKL achieves better classification accuracy than $\ell_1$-MKL on the UCI datasets of moderate size. Across the four UCI benchmark datasets reported in Table 2, $\ell_0$–$\ell_1$ MKL achieves comparable or better test accuracy than $\ell_1$-MKL.

A similar trend is observed in kernel selection: $\ell_0$-$\ell_1$ MKL selects fewer kernels than $\ell_1$-MKL. Thus, the findings of Sections 6.2 and 6.3 on a toy problem extend to real-world datasets: incorporating $\ell_0$ regularization can stably reduces the number of selected kernels while slightly improving classification accuracy. With these real datasets, the difference between $\ell_0$-$\ell_1$ and $\ell_1$-MKL is not as visible as for the synthetic test cases because there are not truly useless features and the class boundaries are not perfectly linear, which introduces other performance variation sources.

### 6.5 Computational Time

The synthetic datasets introduced earlier are of varying sizes which allows to evaluate the practical, computational, scalability of the proposed $\ell_0$-MKL methods. Based on the previous experiments, we focus on the best-performing $\ell_0$-$\ell_1$ model with an explicit constraint formulation, i.e., incorporating a convex combination constraint on the kernel weights. We consider a sample size $n = 500$ and kernel dictionary sizes $M \in \{20, 40, 60, 80, 100\}$. For each configuration, we solve the optimization problem to a prescribed optimality tolerance (relative gap of 0.01) or until reaching a maximum of 1000 iterations. A time limit of an hour (3600 seconds) is imposed for each run. The relative optimality gap is reported as a percentage. We record the evolution of the optimality gap over time to assess the convergence behavior. Figure 4 illustrates the optimality gap (in percentage) as a function of elapsed time for different problem sizes.

From Figure 4, we observe that all configurations converge to the prescribed optimality tolerance within the 1-hour time limit. This indicates that, although the proposed method is theoretically of exponential complexity, it remains practically solvable for moderate problem sizes. $\ell_0$-MKL with about 100 features

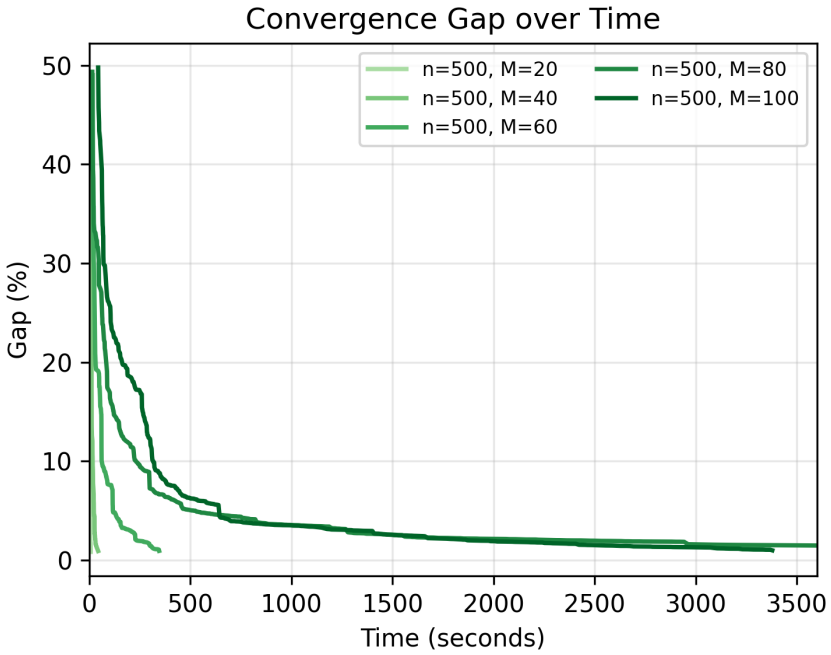


Figure 4: Optimality gap (%) versus time for different sample sizes $n$ and kernel dictionary sizes $M$. Each curve corresponds to a specific $(n, M)$ configuration.

and 500 datapoints can be solved to optimality within an hour of computations on standard computing installations.

## 7 Conclusions

In this paper, we study the properties of $\ell_0$-regularized kernel selection in the context of multiple kernel learning. To accurately model and solve the resulting problem, we adopt a mixed-integer programming framework for $\ell_0$-regularized kernel selection. We consider a rank-one linear kernel dictionary, where each feature corresponds to a base kernel. On this predefined dictionary, we establish an explicit connection between $\ell_0$ regularization on the kernel weights and the resulting kernel construction, providing a structural interpretation of sparse kernel selection. To decouple the effects of sparsity induction and kernel scale control in standard $\ell_1$-MKL, we further introduce $\ell_0$-$\ell_1$, $\ell_0$-$\ell_2$, and $\ell_0$-$\ell_\infty$ formulations for comparison. On synthetic data, where labels are generated from a small subset of informative kernels, methods incorporating $\ell_0$ regularization achieve better recovery of the true kernels, reflected in higher precision and true mass, while maintaining comparable classification accuracy. When kernel weights are constrained by an explicit budget (as in standard $\ell_1$-MKL), we further observe that $\ell_0$-based methods can improve test accuracy. This finding completes a current understanding in the MKL literature that using more kernels leads to better performance. Our results demonstrate how, starting from large kernel dictionaries, sparser solutions that generalize better can be achieved in MKL. Finally, we validate these findings on UCI real-world datasets, where consistent improvements are observed.

Future work includes exploring alternative kernel dictionary constructions where the features are created by a large, predefined, nonlinear set of mappings.

## References

Francis R Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(6), 2008.

Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression. *The Annals of Statistics*, 48 (1):300–323, 2020.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, pp. 813–852, 2016.

Dimitris Bertsimas, Caio de Próspero Iglesias, and Nicholas A. G. Johnson. Sparse multiple kernel learning: Alternating best response and semidefinite relaxations. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=Y5icwFwkyh`.

Serhat S Bucak, Rong Jin, and Anil K Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369, 2013.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.

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

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL `https://www.gurobi.com`.

Franck Iutzeler and Jérôme Malick. Nonsmoothness in machine learning: specific structure, proximal identification, and applications. *Set-Valued and Variational Analysis*, 28(4):661–678, 2020.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12:953–997, 2011.

Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

André CA Nascimento, Ricardo BC Prudêncio, and Ivan G Costa. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC bioinformatics*, 17(1):46, 2016.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24 (2):227–234, 1995.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

Alain Rakotomamonjy, Rémi Flamary, Gilles Gasso, and Stéphane Canu. $\ell_p$-$\ell_q$ penalty for sparse linear and sparse multiple kernel multitask learning. *IEEE Transactions on Neural Networks*, 22(8):1307–1320, 2011.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

Niranjan Subrahmanya and Yung C Shin. Sparse multiple kernel learning for signal processing applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):788–798, 2009.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Tinghua Wang and Xiaoqiang Tu. Multiple kernel learning using nonlinear lasso. *IEEJ Transactions on Electrical and Electronic Engineering*, 14(5):760–767, 2019.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 1175–1182, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

## A Appendix

We propose an OA algorithm (Algorithm 1) for solving the $\ell_0$-SMKL problem. For clarity and uniformity, Algorithm 1 is presented using the constraint formulation. In practice, the corresponding penalty forms with Lagrange multipliers can also be used for implementation. The outer problem is a mixed-integer program, while the inner problem is a standard SVM. Optimality is achieved via a cutting-plane scheme.

We first introduce the SVM dual feasible set

$$\mathcal{Y} := \left\{ \alpha \in \mathbb{R}^n \;\middle|\; 0 \le \alpha_i \le C,\ \sum_{i=1}^{n} \alpha_i y_i = 0 \right\}.$$

Define the SVM dual objective under kernel weights $\theta$ as

$$h(\theta, \alpha) := \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{m=1}^{M} \theta_m \alpha^\top Y K_m Y \alpha.$$

**Subproblem (SVM dual).** Given $\theta^k$, the subproblem is

$$\alpha^k \in \arg\max_{\alpha \in \mathcal{Y}} h(\theta^k, \alpha). \tag{21}$$

**Master problem.** Let $\mathcal{A}$ denote the set of accumulated cuts, and let $\mathcal{T} \subset \mathbb{R}^M \times \{0,1\}^M$ denote the feasible set of $(\theta, z)$. The master problem is formulated as

$$\begin{aligned} \min_{(\theta,z)\in\mathcal{T},\,\eta} \quad & \eta + \gamma \sum_{m=1}^{M} z_m \\ \text{s.t.} \quad & \eta \ge h(\theta, \alpha^*), \quad \forall \alpha^* \in \mathcal{A}. \end{aligned} \tag{22}$$

Here, $\eta$ is an auxiliary variable representing the value of the SVM dual objective. The feasible set $\mathcal{T}$ is defined as

$$\mathcal{T} := \{(\theta, z) \mid 0 \le \theta_m \le \mathcal{M} z_m,\ z_m \in \{0,1\},\ \theta \in \Theta,\ m \in [M]\},$$

where $\Theta$ specifies additional constraints on $\theta$ depending on the formulation (e.g., $\ell_1$ or $\ell_2$ constraints).

**Cutting plane.** Given $\alpha^k$, we generate the following valid cut:

$$\eta \;\ge\; h(\theta, \alpha^k) = \sum_{i=1}^{n} \alpha_i^k - \frac{1}{2} \sum_{m=1}^{M} \theta_m \alpha^{k\top} Y K_m Y \alpha^k.$$

The algorithm alternates between solving the master problem and the SVM subproblem, iteratively refining the approximation of the value function $\phi(\theta) := \max_{\alpha\in\mathcal{Y}} h(\theta, \alpha)$.

We now establish the convergence of Algorithm 1.

**Proposition 2** (Epigraph Reformulation)**.** *The problem*

$$\min_{(\theta,z)\in\mathcal{T}} \gamma \sum_{m=1}^{M} z_m + \phi(\theta), \quad \phi(\theta) = \max_{\alpha\in\mathcal{Y}} h(\theta,\alpha),$$

*is equivalent to*

$$\min_{(\theta,z)\in\mathcal{T},\eta} \gamma \sum_{m=1}^{M} z_m + \eta \quad s.t. \quad \eta \geq h(\theta,\alpha),\ \forall \alpha \in \mathcal{Y},$$

*Proof.* For any fixed $(\theta, z)$, $\phi(\theta)$ is the pointwise maximum of $h(\theta,\alpha)$ over $\mathcal{Y}$. Thus, $\eta \geq h(\theta,\alpha)$ for all $\alpha \in \mathcal{Y}$ if and only if $\eta \geq \phi(\theta)$. Minimizing $\eta$ therefore yields $\eta = \phi(\theta)$ at optimality, establishing the equivalence. □

**Proposition 3.** *Algorithm 1 returns an $\varepsilon$-optimal solution upon termination.*

*Proof.* At each iteration, the master problem is a relaxation of the original problem, since each cut

$$\eta \geq h(\theta,\alpha)$$

is valid for the epigraph of $\phi(\theta)$. Therefore, the master objective value

$$LB^k = \eta^k + \gamma \sum_m z_m^k$$

provides a valid lower bound, and the sequence $\{LB^k\}$ is nondecreasing. For the current solution $\theta^k$, the subproblem computes $\alpha^k$ such that

$$h(\theta^k, \alpha^k) = \phi(\theta^k),$$

yielding an upper bound

$$UB^k = \phi(\theta^k) + \gamma \sum_m z_m^k.$$

While the sequence $\{UB^k\}$ is not necessarily monotonic, the best upper bound

$$\overline{UB}^k := \min_{t\leq k} UB^t$$

is nonincreasing. Moreover, since the binary vector $z$ takes values in a finite set, only finitely many integer assignments can be generated. Each iteration adds a cut that is exact at the current solution and permanently tightens the master problem. Therefore, the algorithm terminates after finitely many iterations. Upon termination,

$$\overline{UB}^k - LB^k \leq \varepsilon.$$

Since

$$LB^k \leq v^\star \leq \overline{UB}^k,$$

the returned solution is $\varepsilon$-optimal. □