

Binary kernel logistic regression: A sparsity-inducing formulation and a convergent decomposition training algorithm

Antonio Consolo ^a, Andrea Manno ^b, Edoardo Amaldi ^a

^a DEIB, Politecnico di Milano, Milano, Italy

^b Centro di Eccellenza DEWS, DISIM, Università degli Studi dell'Aquila, L'Aquila, Italy

ARTICLE INFO

Keywords:

Machine learning
Kernel logistic regression
Sparsity
Decomposition methods
Nonlinear programming

ABSTRACT

Kernel logistic regression (KLR) is a widely used supervised learning method for binary and multi-class classification, which provides estimates of the conditional probabilities of class membership for the data points. Unlike other kernel methods such as Support Vector Machines (SVMs), KLRs are generally not sparse. Previous attempts to deal with sparsity in KLR include a heuristic method referred to as the Import Vector Machine (IVM) and ad hoc regularizations such as the $\ell_{1/2}$ -based one. Achieving a good trade-off between prediction accuracy and sparsity is still a challenging issue with a potential significant impact from the application point of view. In this work, we revisit binary KLR and propose an extension of the training formulation proposed by Keerthi et al., which is able to induce sparsity in the trained model, while maintaining good testing accuracy. To efficiently solve the dual of this formulation, we devise a decomposition algorithm of Sequential Minimal Optimization type which exploits second-order information, and for which we establish global convergence. Numerical experiments conducted on 12 datasets from the literature show that the proposed binary KLR approach achieves a competitive trade-off between accuracy and sparsity with respect to IVM, $\ell_{1/2}$ -based regularization for KLR, and SVM while retaining the advantages of providing informative estimates of the class membership probabilities.

1. Introduction

Logistic Regression (LR) is a popular supervised learning method in Machine Learning (ML) and Statistics, which holds significant relevance for classification and predictive analytics. It belongs to the class of generalized linear models, where the log-odds of an event are modeled as a linear combination of features. Since LR models directly provide estimates of the conditional probabilities that the data points belong to the classes, they are widely used in applications in which this type of information is of interest. Indeed, in a variety of settings (e.g., credit risk scoring) the class membership probabilities turn out to be even more useful than the class membership predictions. Examples of LR applications include: anomaly detection in data for fraud identification (e.g., Mok et al. 2010, Edgar and Manz 2017), disease prediction in medical contexts (e.g., Bagley et al. 2001, Schober and Vetter 2021), and forecasting in business analytics (e.g., Nikolic et al. 2013, De Caigny et al. 2018).

Kernel Logistic Regression (KLR) extends standard LR by using kernel functions to capture complex nonlinear relationships within the data (see e.g. Jaakkola and Haussler 1999). In particular, in KLR the original input data is mapped into a higher-dimensional (implicit)

feature space where metrics are computed using kernel functions. The expressive power of the LR model is substantially increased by generating highly nonlinear decision boundaries with respect to the original input space, while preserving the probabilistic nature of LR.

During the last two decades, KLR has been applied in a variety of fields. Early applications are related, for instance, to speaker recognition and rare event prediction. In Katz et al. (2006) the authors show that KLR and especially sparse KLR perform better in speaker identification experiments than standard Gaussian mixture models and Support Vector Machines (SVMs) on a benchmark dataset. In Maalouf and Trafalis (2011) the authors extend KLR by combining rare events corrections and truncated Newton methods, and demonstrate that, based on the statistical significance test, KLR is more accurate than SVM. Later, in Hong et al. (2015), Chen et al. (2017) KLR is used in geological hazard risk assessment to identify areas susceptible to landslides, showing a better overall performance compared to SVMs and decision trees. Recently, KLR has also been applied in discrete choice modeling, which is central to analyzing individual decision making, with wide-ranging applications in economics, marketing and transportation. In Martín-Baos et al. (2020), KLR is used to specify non-parametric utilities in Random Utility Models and is compared with

* Corresponding author. Currently affiliation of Antonio Consolo: DISCo, Università di Milano-Bicocca, Milano, Italy.
E-mail address: antonio.consolo@unimib.it (A. Consolo).

Multinomial Logit models in terms of goodness of fit and the capability of obtaining the specified utilities. In [Martín-Baos et al. \(2021\)](#) a similar approach is adopted to model random utilities and empirical results show the advantages of KLR with respect to SVMs and Random Forests in terms of accuracy and of the estimation of important indicators in the field of transportation. Finally, in [Martín-Baos et al. \(2025\)](#) the authors introduce a Nyström approximation for KLR on large datasets in order to improve scalability and evaluate performance using large-scale transport mode choice datasets.

As previously mentioned, the distinctive feature of binary KLR is to naturally provide probabilistic estimates of the class membership that are interpretable and are useful to estimate expected costs of classification decisions and confidence levels. It is worth pointing out that this is in contrast with other popular kernel methods such as binary SVM, which frequently achieve similar classification performance (see e.g. [Keerthi et al. 2005](#)). Since SVM predictions are deterministic, several approaches have been proposed to try to derive such valuable probabilistic class membership estimates from the SVM outputs. These include adopting a specific parametric family for the posterior probability ([Wahba et al., 1999](#)), using a series of trigonometric functions ([Vapnik et al., 1998](#)), applying logistic calibration ([Platt et al., 1999b](#)), and employing sequential training of weighted classifiers to obtain an interval-based estimation ([Wang et al., 2008](#)). More recently, several attempts have been made to either propose new alternatives (e.g., [Franc et al. 2011](#), [Benítez-Peña et al. 2024](#)) or address the limitations of previous approaches (e.g., [Böken 2021](#), [Zeng and Zhang 2023](#)). Despite the growing attention devoted to this topic, there seems to be no consensus on the most effective way to extract probabilistic information from SVMs. Therefore, the probabilistic nature of KLR makes it preferable to SVM in a number of settings where reliable probabilistic predictions are required to take aware decisions (see e.g., [Wiens et al. 2019](#), [Kelly et al. 2019](#), [Nicora et al. 2022](#)). Nevertheless, KLR tends to yield denser representations than SVMs, which are natively sparse (only a small subset of data points are needed to specify the decision function) and lead to smaller computational prediction cost.

Sparsity plays a crucial role in machine learning models, as it is closely related to both generalization error and computational efficiency. In statistical learning, the importance of sparsity has been extensively investigated, especially since it leads to sharper generalization bounds (see, e.g., [van de Geer 2008](#), [Bickel et al. 2009](#), [Negahban et al. 2012](#)). In kernel machine models like SVMs, the number of support vectors has been shown to strongly affect generalization, with sparse solutions often providing tighter error bounds, as highlighted in the SVM literature ([Burges, 1998](#); [Vapnik, 2000](#)). As a consequence, several methods have been proposed to obtain sparse solutions in kernel logistic regression. For instance, [Zhu and Hastie \(2005\)](#), [Krishnapuram et al. \(2005\)](#), [Xu et al. \(2013\)](#) show that accurate predictions can be obtained using only a small subset of the data points in the training set. From the computational point of view, sparsity is still of primary importance in practice. Some works, such as [Burges \(1996\)](#), [Wang et al. \(2012\)](#), illustrate the benefits of kernel machine models involving a small number of terms (kernel functions) when subject to limited memory and computational time. More recent works address large-scale settings, aiming to reduce RAM usage and CPU time. For instance, [Koppel et al. \(2019\)](#) studies an online kernel learning framework, [Li and Liao \(2022\)](#) establishes theoretical lower bounds for online kernel selection, and [Martín-Baos et al. \(2025\)](#) considers Nyström approximations for KLR.

In this work, we propose an extension of the nonlinear continuous constrained convex formulation adopted in [Keerthi et al. \(2005\)](#) for training KLR, which induces sparsity in the model without compromising the testing accuracy. The dual of this formulation can be adapted to be efficiently solved by a decomposition method. Inspired by the decomposition algorithm in [Fan et al. \(2005\)](#) for SVM, we devise a Sequential Minimal Optimization (SMO)-type training algorithm exploiting second-order information, for which we establish global

convergence. The proposed formulation and algorithm are tested on 12 datasets from the literature and compared with Import Vector Machine (IVM) greedy heuristic ([Zhu and Hastie, 2005](#)), $\ell_{1/2}$ -KLR regularization technique ([Xu et al., 2013](#)), and SVM. The numerical results shows that our approach is able to achieve a good trade-off between sparsity and testing accuracy, while retaining the advantages of directly providing estimates of the class probabilities.

The remainder of the paper is organized as follows. In [Section 2](#), after recalling the binary KLR model and training formulation, we mention previous work on promoting sparsity in KLR. In [Section 3](#), we describe the proposed nonlinear optimization formulation for training sparse KLR, and its dual. In [Section 4](#), we present the SMO-type decomposition method and state the related convergence guarantees which are proved in [Appendix B](#). The experimental results obtained for 12 datasets from the literature are reported and discussed in [Section 5](#). Finally, [Section 6](#) contains some concluding remarks.

2. Binary KLR and previous works on sparse KLR

In this section, we introduce the KLR model and the associated primal and dual training formulations. Then, we mention previous work to deal with sparsity in KLR.

2.1. Binary KLR model and training formulation

In a binary classification problem with p features, we are given a training set $\{(x_i, y_i)\}_{i \in I}$ where $I = \{1, \dots, N\}$ and each one of the N data points consists of a vector $x_i \in \mathbb{R}^p$ of features and an associated label $y_i \in \{-1, 1\}$. Let $z = \varphi(x)$ denote the higher-dimensional feature space vector corresponding to x , where $\varphi(\cdot)$ represents the feature map which induces the kernel metric $Ker(x, \hat{x}) = \varphi(x) \cdot \varphi(\hat{x})$ and \cdot denotes the inner product in the z space. The binary KLR model can be trained by solving the following unconstrained nonlinear optimization problem:

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 + C \sum_{i \in I} g(-y_i(\omega \cdot z_i - b)), \quad (1)$$

where $z_i = \varphi(x_i)$, C is a regularization parameter, and g is given by:

$$g(\xi) = \log(1 + e^\xi). \quad (2)$$

The function $g(\cdot)$ represents the Negative Log-Likelihood (NLL) associated to the conditional probability:

$$P(y|x) = \frac{1}{1 + e^{-y(\omega \cdot \varphi(x) - b)}}. \quad (3)$$

By introducing the auxiliary variables ξ_i with $i \in I$, problem (1) can be rewritten as

$$\min_{\omega, b, \xi} \frac{1}{2} \|\omega\|^2 + C \sum_{i \in I} g(\xi_i) \quad (4a)$$

$$\text{s.t. } \xi_i = -y_i(\omega \cdot z_i - b) \quad \forall i \in I. \quad (4b)$$

As shown in [Keerthi et al. \(2005\)](#), using the Lagrangian function and the optimality conditions, the Wolfe dual of (4) is as follows

$$\min_{\alpha} \frac{1}{2} \|\omega(\alpha)\|^2 + C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right) \quad (5a)$$

$$\text{s.t. } \sum_{i \in I} \alpha_i y_i = 0, \quad (5b)$$

where the α_i with $i \in I$ are the dual variables associated to constraints (4b), the primal and dual variables are linked by the following relations¹

$$\omega(\alpha) = \sum_{i \in I} \alpha_i y_i z_i, \quad \xi_i(\alpha_i) = g'^{-1}\left(\frac{\alpha_i}{C}\right), \quad (6)$$

¹ Note that $g'(u) = e^u/(1 + e^u)$ has its range in the open interval $(0, 1)$ and is invertible on it.

and the function G , defined as

$$G\left(\frac{\alpha_i}{C}\right) = \frac{\alpha_i}{C} \xi_i - g(\xi_i), \quad (7)$$

is convex. Therefore, training a binary KLR model can be formulated as solving the convex constrained nonlinear optimization problem (5), which is very similar to the SVM dual formulation (see e.g., Cortes and Vapnik 1995). Exploiting this similarity the authors in Keerthi et al. (2005) devise a SMO-type decomposition algorithm based on first-order information to solve formulation (5).

2.2. Previous work on sparse KLR

In the past twenty years, there have been some attempts to deal with sparsity in KLR. In Krishnapuram et al. (2005) the authors combine a bound optimization approach with a coordinate descent update procedure for learning a sparse multi-class classifier within a Bayesian framework. In Zhang et al. (2012) two conservative online learning algorithms are presented to generate sparse KLR. At each iteration, the current classifier is updated in a stochastic manner using Bernoulli random variables to model whether new data points are considered.

In Zhu and Hastie (2005) the authors propose IVMs where KLR sparsity is promoted by solving the problem (1) via a greedy sub-model selection strategy. The approach involves constructing a proxy model for the relation $\omega(\alpha) = \sum_{i \in I} \alpha_i y_i z_i$, by identifying a subset of data points S such that $\sum_{s \in S} \alpha_s y_s z_s$ provides a good approximation of $\sum_{i \in I} \alpha_i y_i z_i$. Since exhaustively searching through all possible subsets of data points is a computationally intractable task, the authors adopt a forward selection strategy, starting with the empty set $S = \emptyset$ and iteratively adding one data point at a time. The included data point is the one producing the best improvement after a one-step Newton–Raphson method. As we shall see in the numerical experiments reported in Section 5, IVMs allow to achieve a significant degree of sparsity. By construction, only a small fraction of data points is considered and then used to index kernel basis functions. However, the greedy heuristic tends to cause premature convergence to suboptimal solutions and to compromise accuracy.

In Roscher et al. (2012a,b) an efficient incremental learning strategy is presented for training sparse IVMs and is applied to the classification of hyperspectral imagery data. New training data points are included to increase the classification accuracy and non-informative data points are deleted to be memory and runtime efficient. In the comparative experiments of IVM and SVM, the number of import vectors is significantly lower than the number of support vectors, and the probabilities provided by IVM are more reliable than the probabilistic information derived from an SVM's output using the method described in Platt et al. (1999a).

Since regularization is a natural way to promote model sparsity, ℓ_1 , ℓ_2 and other ad hoc regularization variants have been considered. In Xu et al. (2013) the $\ell_{1/2}$ regularization is used for kernel logistic regression. In Xu et al. (2012) a thresholding representation theory for $\ell_{1/2}$ regularization is described and the fast iterative convergent half thresholding algorithm is proposed for a different family of loss functions. A series of experiments on compressed sensing problems show that the convergent half thresholding algorithm is an efficient and effective approach for sparse modeling. In Xu et al. (2013) the authors adapt to KLR the $\ell_{1/2}$ -based method and the fast iterative convergent half thresholding algorithm proposed in Xu et al. (2012), showing promising results in terms of accuracy and sparsity compared with IVM, two SVM versions and ℓ_1 regularization.

The above-mentioned approaches allow to achieve different level of sparsity. Nevertheless, since they are based on heuristic techniques or regularization methods, it is still a challenging problem to develop a robust training method that yields sparse KLRs with good testing accuracy.

3. A sparsity-inducing approach for training binary KLR

In this work, aiming at a good trade-off between sparsity and accuracy, we propose an extension of the binary KLR primal formulation (4) in Keerthi et al. (2005) specifically designed to promote sparsity in the dual variables, namely, in the vector of the α_i variables, and we derive a bounded regularized (strictly convex) variant of the dual formulation.

In this section we first describe the sparsity-inducing training formulation and then discuss its main motivation and peculiarities.

3.1. A KLR formulation variant to promote sparsity

Before presenting the formulation for training sparse KLR, we briefly recall some key characteristics of the NLL loss used in the KLR model to justify the introduction of a new variable ρ in the primal formulation, which plays a crucial role in inducing sparsity.

In the literature, several authors (Zhu and Hastie, 2005; Keerthi et al., 2005) have analyzed the similarity between the NLL and the hinge loss, which is often applied in ML models such as SVM. As shown in Fig. 1, the NLL function approximates the hinge loss. However, whereas the hinge loss is capable of inducing sparsity due to its truncation property, the NLL function lacks this capability. As a result, the KLR model typically yields an optimal solution where $\alpha_i > 0$ for all $i \in I$. From the Representer Theorem (Schölkopf et al., 2001), it is known that the classifier parameters ω of the KLR can be expressed as a weighted sum of the data points mapped into the higher-dimensional feature space, where the weights correspond to the variables α_i . The aim is to modify formulation (4) in order to reduce the impact in the objective function of the NLL terms corresponding to the easy-to-classify data points, i.e., those with sensibly negative values of the associated ξ_i variable. For this purpose, we define a nonnegative slack variable ρ that adjusts the ξ_i values corresponding to the easy-to-classify data points, reducing their associated dual variable values α_i .

The formulation that we propose for training sparse KLR is as follows:

$$\min_{\omega, b, \rho} \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i \in I} g(\xi_i) - \nu \rho \quad (8a)$$

$$s.t. \quad \xi_i - \rho = -y_i(\omega \cdot z_i - b) \quad \forall i \in I \quad (8b)$$

$$\rho \geq 0. \quad (8c)$$

The constraints (8b) expressing the decision for each data point are modified with respect to the original formulation (4) by subtracting a nonnegative slack variable ρ to the left-hand-side of all the equality constraints (4b). This implies a general increment of variables ξ_i for the constraints to be satisfied. Notice that a right shifting of variable ξ_i is equivalent to a left “translation” of the corresponding NLL function. As depicted in Fig. 2(a), the consequence is that the NLL errors of data points with ξ_i values around zero (which are strongly relevant in determining the final classifier), would significantly increase. Conversely, errors of data points with sharply negative ξ_i values lying in the flat left region of the NLL function, whose classification is trivial, would not change significantly.

This effect can also be viewed in terms of derivatives of the NLL function. Indeed, the profile of the NLL second derivatives, shown in Fig. 2(b), has a remarkable positive peak around zero, and it is almost flat and close to zero sufficiently far from it. Therefore, an increase in the value of ξ_i produces a relatively larger first derivatives increase for values of ξ around zero, than for strongly negative/positive values. This fact is expected to strongly influence the training optimization phase, as it leverages first derivatives information.

To amplify the effect of the variable ρ , the latter is maximized by adding the term $-\nu\rho$ in the objective function (8c), with $\nu > 0$. However, ρ cannot grow arbitrarily as, due to constraints (8b), too large values of ρ would imply too large values of the NLL error terms

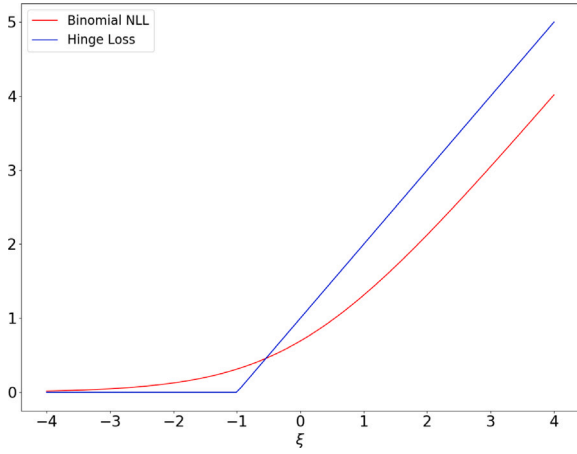


Fig. 1. Plots of the Hinge loss used in Support Vector Machines in blue and of the Binomial Negative Log-Likelihood used in logistic regression in red.

$C \sum_{i \in I} g(\xi_i)$. The desired trade-off between maximization of ρ and minimization of the NLL errors can be calibrated by selecting the values of the C and ν parameters.

By considering the Lagrangian function of (8)

$$L(\omega, b, \xi, \rho) = \frac{1}{2} \|\omega\|^2 + C \sum_{i \in I} g(\xi_i) - \nu \rho + \sum_{i \in I} \alpha_i [-\xi_i + \rho - y_i(\omega \cdot z_i - b)] - \beta \rho \quad (9)$$

with $\beta \geq 0$, the optimality conditions are

$$\nabla_{\omega} L = \omega - \sum_{i \in I} \alpha_i y_i z_i = \mathbf{0} \quad (10)$$

$$\frac{\partial L}{\partial b} = \sum_{i \in I} \alpha_i y_i = 0 \quad (11)$$

$$\frac{\partial L}{\partial \xi_i} = C g'(\xi_i) - \alpha_i = 0 \quad \forall i \in I \quad (12)$$

$$\frac{\partial L}{\partial \rho} = -\nu + \sum_{i \in I} \alpha_i - \beta = 0. \quad (13)$$

Thanks to (10) and (12), we can write ω and ξ_i as functions of the variables α_i :

$$\omega(\alpha) = \sum_{i \in I} \alpha_i y_i z_i, \quad \xi_i(\alpha_i) = g'^{-1}\left(\frac{\alpha_i}{C}\right). \quad (14)$$

As shown in Keerthi et al. (2005), it is possible to define a function

$$G(\delta) = \delta \xi_i - g(\xi_i), \quad \text{with } \delta = \frac{\alpha_i}{C} \quad (15)$$

whose derivative expresses the inverse function g'^{-1} in (14):

$$\frac{dG}{d\delta} = \delta \frac{d\xi_i}{d\delta} + \xi_i - g'(\xi_i) \frac{d\xi_i}{d\delta} = \xi_i = g'^{-1}(\delta).$$

For the case of logistic regression where g is expressed as (2) we have:

$$g'^{-1}(u) = \log\left(\frac{u}{1-u}\right) \quad G(\delta) = \delta \log(\delta) + (1-\delta) \log(1-\delta). \quad (16)$$

Using equations (14) and (15) and recalling that $\beta \geq 0$, the Wolfe dual (which corresponds to the maximization of $L(\omega, b, \xi, \rho)$ subject to (10)–(13)) can be rewritten as:

$$\min_{\alpha} \quad \frac{1}{2} \|\omega(\alpha)\|^2 + C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right) \quad (17a)$$

$$s.t. \quad \sum_{i \in I} \alpha_i y_i = 0 \quad (17b)$$

$$-\nu \geq -\sum_{i \in I} \alpha_i. \quad (17c)$$

Instead of directly solving formulation (17) which includes the complicating constraint (17c), we consider the following “regularized” formulation

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \|\omega(\alpha)\|^2 + C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right) - \lambda \sum_{i \in I} \alpha_i \quad (18) \\ s.t. \quad & \sum_{i \in I} \alpha_i y_i = 0, \end{aligned}$$

where constraint (17c) is replaced by a sort of regularization term $-\lambda \sum_i \alpha_i$ in the objective function, where the weight $\lambda \geq 0$ is a hyperparameter to be tuned. Notice that the hyperparameter ν in constraint (17c) is taken into account by the hyperparameter λ in the objective function.

Some simple observations are in order concerning the impact of the ρ variable introduced in formulation (8) with respect to the other primal variables ξ_i with $i \in I$. The optimal solution $(\hat{\omega}, \hat{b}, \hat{\xi})$ of (8) with $\nu = 0$ (which is equivalent to the optimal solution of the original formulation (4)) is the best one (ideal) in terms of trade-off between $\frac{1}{2} \|\omega\|^2$ and $\sum_{i \in I} g(\xi_i)$ for any given value of C , but it is generally not sparse. Let us consider a sufficiently small positive value of ν which leads to a small positive optimal ρ value, and to an optimal solution of (8) that is a “local” perturbation of $(\hat{\omega}, \hat{b}, \hat{\xi})$. This perturbation implies that constraint (4b) is “replaced” with constraint (8b), and hence that the ideal but not sparse triplet $(\hat{\omega}, \hat{b}, \hat{\xi})$ must be modified to be feasible with respect to (8b). Since for sufficiently small values of μ the optimal solution of (8) is not expected to deviate too much from $(\hat{\omega}, \hat{b}, \hat{\xi})$, a possible way to enforce feasibility, while preserving the decision boundary $(\hat{\omega}, \hat{b})$, is to uniformly increase all ξ_i values by ρ . However, since the increase of each ξ_i has a different impact on the corresponding error terms, a (slight) modification of $(\hat{\omega}, \hat{b})$ in the right-hand side of (8b) would imply heterogeneous updates of the variables ξ_i which could be preferable in terms of the objective function value. In such a case, some ξ_i variables would increase and some others would decrease, which, given the monotonic connection between primal variables ξ and dual variables α_i (see (12)), tend to promote dual sparsity. As empirically shown in Appendix C, for reasonable values of the primal (dual) hyperparameter ν (λ) the norm of the decision boundary hyperplane $(\hat{\omega}, \hat{b})$ increases, while its position and orientation are not significantly affected, which preserves the same classification rates as the ones obtained by the ideal triplet $(\hat{\omega}, \hat{b}, \hat{\xi})$. This is consistent with the fact that positive values of ρ tend to increase the values of the variables ξ_i , and consequently of the relative weight of the error term $C \sum_{i \in I} g(\xi_i)$ in the objective function. This, combined with the presence of the new term $-\nu \rho$ in formulation (8), reduces the relative weight of the regularization term $\frac{1}{2} \|\omega(\alpha)\|^2$.

3.2. Some properties

In this subsection we describe some properties of the optimal dual solution α and the connection between the regularization hyperparameter λ and the primal variable ρ of formulation (8). Additional properties of α as λ tends to infinity can be found in Appendix A.

Since the function $G(\cdot)$ is an entropy function, every dual variable α_i clearly takes values within the compact interval $[0, C]$. For every $i \in I$ and any given λ , let $\alpha(\lambda)$ denote the corresponding optimal solution of (18).

In Appendix A, we prove that for any value of $\lambda \in \mathbb{R}$ all the components $\alpha_i(\lambda)$ of the optimal solution never attain the above-mentioned bounds.

Lemma 1. For any value of $\lambda \in \mathbb{R}$ the optimal solution of formulation (18) $\alpha(\lambda)$ is such that

$$0 < \alpha_i(\lambda) < C \quad \forall i \in I.$$

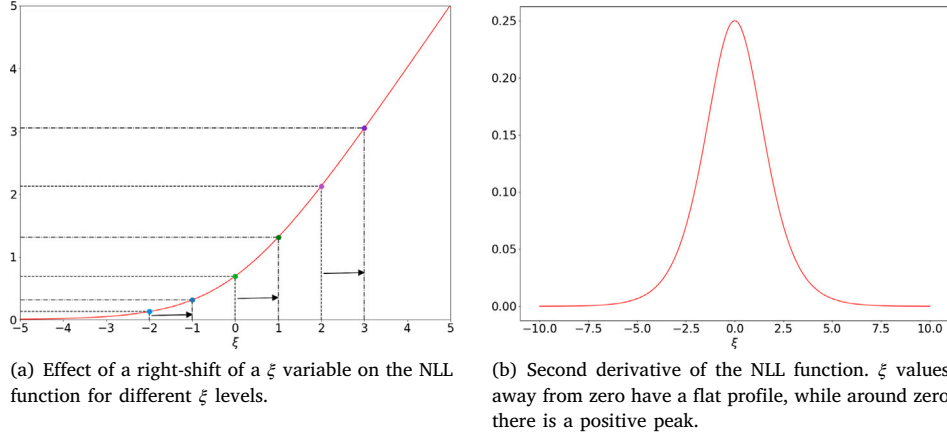


Fig. 2. Illustration of the negative log-likelihood function and its second derivative with respect to the parameter ξ .

Now we analyze the connection between the primal variable ρ and the dual formulation (18). For each $\lambda \in \mathbb{R}$, the corresponding optimality conditions associated with the dual problem (18) lead (via substitution) to the following equations:

$$\lambda - y_i \sum_{j \in I} \alpha_j y_j K_{ij} + y_i \psi = g'^{-1}\left(\frac{\alpha_i}{C}\right) \quad \forall i \in I, \quad (19)$$

where ψ is the Lagrangian multiplier associated to the equality constraint of (18).

Moreover, for every optimal solution of the dual problem (18) the optimality condition of the primal problem (12) and (8b) are also satisfied. Therefore:

$$\rho - y_i(\omega \cdot z_i - b) = g'^{-1}\left(\frac{\alpha_i}{C}\right) \quad \forall i \in I,$$

which, given (10), can be rewritten as:

$$\rho - y_i \sum_{j \in I} \alpha_j y_j K_{ij} + y_i b = g'^{-1}\left(\frac{\alpha_i}{C}\right) \quad \forall i \in I. \quad (20)$$

Any optimal solution $\alpha(\lambda)$ of the dual formulation (18) must satisfy the system of Eqs. (20) in the variables ρ and b , as well as the system of Eqs. (19) in the variables λ and ψ . Since the number N of equations is larger than the number of variables (2), both linear systems admit a unique solution, and these solutions coincide. This directly leads to the following result.

Observation 1. For each $\lambda \in \mathbb{R}$ in the dual formulation (18), the associated primal optimal solution, denoted as $(\omega(\lambda), b(\lambda), \rho(\lambda))$ is such that $\rho(\lambda) = \lambda$. Hence, the optimal value of the primal variable ρ corresponds exactly to the value of the coefficient of the regularization term in the dual formulation (18).

From the above observation, we derive an upper bound on the values of λ which can be used in practice. Since values of $G(\cdot)$ cannot be reliably computed when α_i approaches 0 or C , thanks to Lemma 1, we consider a modified version of formulation (18). Given a tolerance $\gamma > 0$, we define $\bar{0} = \gamma$ and $\bar{C} = C - \gamma$ and add the auxiliary constraints $\bar{0} \leq \alpha_i \leq \bar{C}$ for all $i \in I$ in formulation (18). This is similar to what was done to avoid numerical issues in Keerthi et al. (2005), where the auxiliary bounds are not explicitly added to the dual of the original formulation (4) but are implicitly taken into account in the training algorithm.

Let $I_+ = \{i \in I : y_i = +1\}$, $I_- = \{i \in I : y_i = -1\}$, $n_+ = |I_+|$, $n_- = |I_-|$ and the minority class be the one with the smallest number of data points. Assume without loss of generality that I_- is the minority class. Given any tolerance $\gamma > 0$, it is possible to obtain an upper bound on the value of λ beyond which the optimal solution $\alpha(\lambda)$ satisfies $\alpha_i(\lambda) \geq C - \gamma$ for every data point (x_i, y_i) in the minority class with

$i \in I_-$. Therefore, since $\alpha_i(\lambda) \geq 0$ for $i \in I_-$, the optimal solution is not sparse at least over the components of α corresponding to the minority class.

In particular, in Appendix A we prove the following result.

Proposition 1. Assume that I_- is the minority class. Let $\alpha(0)$ denote the optimal solution of formulation (18) with $\lambda = 0$. Consider any positive tolerance γ such that $\gamma < C - \max_{i \in I_-} \alpha_i(0)$, and define $\bar{C} = C - \gamma$. For any

$$\lambda \geq \max_{i \in I_-} \sum_{j \in I} (\bar{C} - \alpha_j(0)) y_j K_{ji}, \quad (21)$$

the optimal solution $\alpha(\lambda)$ of formulation (18) satisfies $\alpha_i(\lambda) \geq \bar{C}$ for all $i \in I_-$.

Notice that in case of a perfectly balanced dataset (i.e. $n_+ = n_-$), the result of Proposition 1 is valid for every data point of the training set. Indeed, in the case of a balanced dataset, considering $\gamma < C - \max_{i \in I} \alpha_i(0)$, for any $\lambda \geq \max_{i \in I} \sum_{j \in I} (\bar{C} - \alpha_j(0)) y_j K_{ji}$, we have that $\alpha_i(\lambda) \geq \bar{C}$ for all $i \in I$.

From a practical point of view Proposition 1 can be exploited to restrict the search space for the value of the hyperparameter λ , as we aim at a sparse dual variable vector $\alpha(\lambda)$. For datasets with imbalanced classes, the inequality (21) provides an upper bound for λ to avoid that all $\alpha_i(\lambda)$ for all $i \in I_-$ exceed the value \bar{C} , preventing sparsity in the minority class. While, for perfectly balanced datasets ($n_+ = n_-$), the same inequality can be used to avoid that all $\alpha(\lambda)$ components exceed the value \bar{C} , preventing sparsity over all the components of α . Even when the imbalance is mild and the minority class represents slightly less than 50% of data points ($\frac{n_-}{N} < 0.5$), selecting a value of λ above the threshold in (21) implies that the proportion of the nonzero $\alpha(\lambda)$ components is at least $\frac{n_-}{N}$.

3.3. A bounded formulation version

Therefore, for practical purposes we propose to train the following bounded regularized version of formulation (17):

$$\min_{\alpha} f(\alpha) = \frac{1}{2} \|\omega(\alpha)\|^2 + C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right) - \lambda \sum_{i \in I} \alpha_i \quad (22a)$$

$$s.t. \quad \sum_{i \in I} \alpha_i y_i = 0 \quad (22b)$$

$$\bar{0} \leq \alpha_i \leq \bar{C} \quad \forall i \in I. \quad (22c)$$

Notice that the objective function (22a) can be expressed as a function of only the α_i variables as

$$\frac{1}{2} \sum_{i \in I} \sum_{j \in I} y_i y_j \alpha_i \alpha_j K_{ij} + C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right) - \lambda \sum_{i \in I} \alpha_i, \quad (23)$$

where $K_{ij} = \text{Ker}(x_i, x_j)$ is the ij element of the $N \times N$ kernel matrix K .

According to the Karush–Kuhn–Tucker (KKT) conditions, α is a stationary point of (22a)–(22c) if and only if there exist two nonnegative vectors σ , η and a scalar b such that

$$\nabla f(\alpha) + b\mathbf{y} = \eta - \sigma$$

$$\eta_i(\alpha_i - \bar{0}) = 0, \sigma_i(\bar{C} - \alpha_i) = 0, \eta_i \geq 0, \sigma_i \geq 0 \quad \forall i \in I.$$

The KKT conditions can be rewritten as

$$\nabla f(\alpha)_i + by_i \geq 0 \quad \text{if } \alpha_i < \bar{C} \quad (24)$$

$$\nabla f(\alpha)_i + by_i \leq 0 \quad \text{if } \alpha_i > \bar{0}. \quad (25)$$

Defining as in Chen et al. (2006)

$$I_{\text{up}}(\alpha) = \{i \mid \alpha_i < \bar{C}, y_i = 1 \text{ or } \alpha_i > \bar{0}, y_i = -1\} \quad (26)$$

$$I_{\text{low}}(\alpha) = \{i \mid \alpha_i < \bar{C}, y_i = -1 \text{ or } \alpha_i > \bar{0}, y_i = 1\},$$

the conditions (24) and (25) are equivalent to

$$-y_i \nabla f(\alpha)_i \leq b \quad \forall i \in I_{\text{up}}(\alpha)$$

$$-y_i \nabla f(\alpha)_i \geq b \quad \forall i \in I_{\text{low}}(\alpha).$$

Thus a feasible solution α is a stationary point of (22a)–(22c) if

$$\max_{i \in I_{\text{up}}(\alpha)} -y_i \nabla f(\alpha)_i \leq \min_{i \in I_{\text{low}}(\alpha)} -y_i \nabla f(\alpha)_i. \quad (27)$$

3.4. Motivation for the sparsity inducing formulation

In this section, we provide illustrative examples to analyze the way formulation (8) induces sparsity in the KLR model. In particular, we show how the decision boundary, the sparsity level, and the slack variable ρ vary for different values of the hyperparameter ν , and we point out the differences with respect to heuristics based on thresholding the dual variables α_i . For the ease of representation, we consider a linear kernel and a synthetic not linearly separable dataset, referred to as Synth, for which reasonable linear decision boundaries (hyperplanes) exist. Synth includes $p = 2$ features and consists of 99 data points (50 belonging to the first class and 49 to the second one).

3.4.1. The role of the ρ variable in promoting sparsity

By comparing the original formulation (4) and the proposed sparse KLR formulation (8), for any identical value of hyperparameter C and sufficiently large values of hyperparameter ν , we show that the presence of variable ρ in (8) leads to a different decision boundary (ω, b) with respect to the one obtained with the formulation (4) (formulation (8) with $\nu = 0$) which is determined by a relatively smaller subset of the training data points. As we will see via some illustrative examples, this behavior tends to promote sparsity in the associated dual variables α_i .

First, we observe that for any given value of C there exists a $\bar{\nu} > 0$ such that any given $\nu > \bar{\nu}$ implies that the optimal value of ρ is strictly positive. Since problems (1) and (8) are strictly convex, we denote by $(\hat{\omega}, \hat{b})$ and (ω^*, b^*, ρ^*) the unique optimal solutions of (1) and (8), respectively. For the sake of computational simplicity, let us consider for problem (8) the following equivalent bound constrained formulation obtained by substituting in the objective function the ξ_i variables according to the equality constraints (8b):

$$\min_{\omega, b, \rho} \bar{f}(\omega, b, \rho) = \frac{1}{2} \|\omega\|^2 + C \sum_{i \in I} g(\rho - y_i(\omega \cdot z_i - b)) - \nu \rho$$

$$\text{s.t. } \rho \geq 0.$$

To guarantee that $\rho^* > 0$, the partial derivative of \bar{f} with respect to ρ at $(\omega^*, b^*, 0)$ should be negative, i.e. $\frac{\partial \bar{f}(\omega^*, b^*, 0)}{\partial \rho} < 0$, otherwise $(\omega^*, b^*, 0)$

is optimal due to the constraint $\rho \geq 0$. Notice that in such a case (ω^*, b^*) coincides with $(\hat{\omega}, \hat{b})$. Therefore, to obtain $\rho^* > 0$ we need to impose that

$$\frac{\partial \bar{f}(\omega^*, b^*, 0)}{\partial \rho} = \frac{\partial \bar{f}(\hat{\omega}, \hat{b}, 0)}{\partial \rho} = -\nu + C \sum_{i \in I} \frac{1}{1 + e^{y_i(\hat{\omega} \cdot z_i - \hat{b})}} < 0,$$

implying that

$$\nu > \bar{\nu} = C \sum_{i \in I} \frac{1}{1 + e^{y_i(\hat{\omega} \cdot z_i - \hat{b})}}. \quad (29)$$

Given any fixed values of C and $\nu > \bar{\nu}$, for any $(\omega, b) \in \mathbb{R}^{p+1}$ the positivity of the corresponding optimal ρ together with constraint (8c) lead to a general increase in the variables ξ_i with respect to formulation (4). This results in an increase in the weight of the second term $\sum_{i \in I} g(\xi_i)$ of the objective function (8a). While all the variables ξ_i increase by the same amount ρ , the increase in each $g(\xi_i)$ term of the objective function substantially differs depending on the data point. Clearly, given the negative log-likelihood g depicted in Fig. 2(a), the data points with larger ξ_i (i.e., those closer to the decision boundary, whether correctly or incorrectly classified) will have a greater impact on the increase in the error term. We remark that, due to the nature of g , this is not equivalent to increasing C since this would increment all $g(\xi_i)$ terms by the same multiplicative factor for all the data points.

The general increase in $\sum_{i \in I} g(\xi_i)$ leads to a variation of the components of the decision boundary vector ω , as $\|\omega\|$ tends to be larger due to a reduced impact of the first term $\frac{1}{2} \|\omega\|^2$ in the objective function. Since the increase in $\sum_{i \in I} g(\xi_i)$ is more affected by the increment in the terms $g(\xi_i)$ related to the data points which are closer to the decision boundary (those with larger α_i values), the change in the components of ω is more influenced by such data points.

Fig. 3 provides a visualization of the impact of the variable ρ in the formulation (8) for the Synth dataset with $C = 60$ through four scatterplots. As the value of ν increases, ranging from $\nu = 0$ (which amounts to solve the original formulation (4)) to $\nu = 625$, the number of selected data points decreases from the 92.93% to the 37.37%, where the selected data points are those associated to α_i values larger than a threshold set to $1e^{-5}$. It is worth pointing out that not only the number of selected data points substantially decreases but also the focus is progressively restricted to those that are closer to the decision boundary.

Fig. 4 indicates how the number of data points selected by formulation (8) and the norm of the vector ω vary when the value of ν increases from 500 to 1000 in 8 equally spaced steps. The x -axis corresponds to the ν values, while the left y -axis corresponds to the number of selected data points (blue curve), and the right y -axis to the norm of the decision boundary coefficient vector $\|\omega\|$ (red curve). Clearly, as the value of ν grows the presence of the ρ term in the KLR primal formulation (8) induces progressively sparser solutions, decreasing the number of selected data points down to about 20%. Larger values of ν also leads to an increase in the norm of ω due to the reduced impact of the $\|\omega\|^2$ term in the objective function.

3.4.2. Differences with respect to thresholding heuristics

As highlighted above, the increase in ν leads to higher values of α_i related to the data points closer to the decision boundary and lower values for α_i associated with well-classified points further away from the boundary. Since the introduction of the $\nu \rho$ term in our sparse KLR formulation (8) implies that the larger the value of hyperparameter ν the larger $\sum_{i \in I} \alpha_i$ (see the KKT conditions (17c)), then the sum of the dual variables α_i corresponding to the data points which lie closer to the decision boundary increases.

It is worth pointing that this is not equivalent to applying in the original KLR dual formulation (5) a thresholding method to sparsify the trained model. A natural way to define a thresholding heuristic after the training phase is to set to zero all dual variables α_i below a certain threshold. Notice that, by considering the connection between dual and primal variables, namely $\xi_i = -y_i(\omega \cdot z_i - b)$ and $\xi_i(\alpha_i) = g^{-1}(\frac{\alpha_i}{C})$

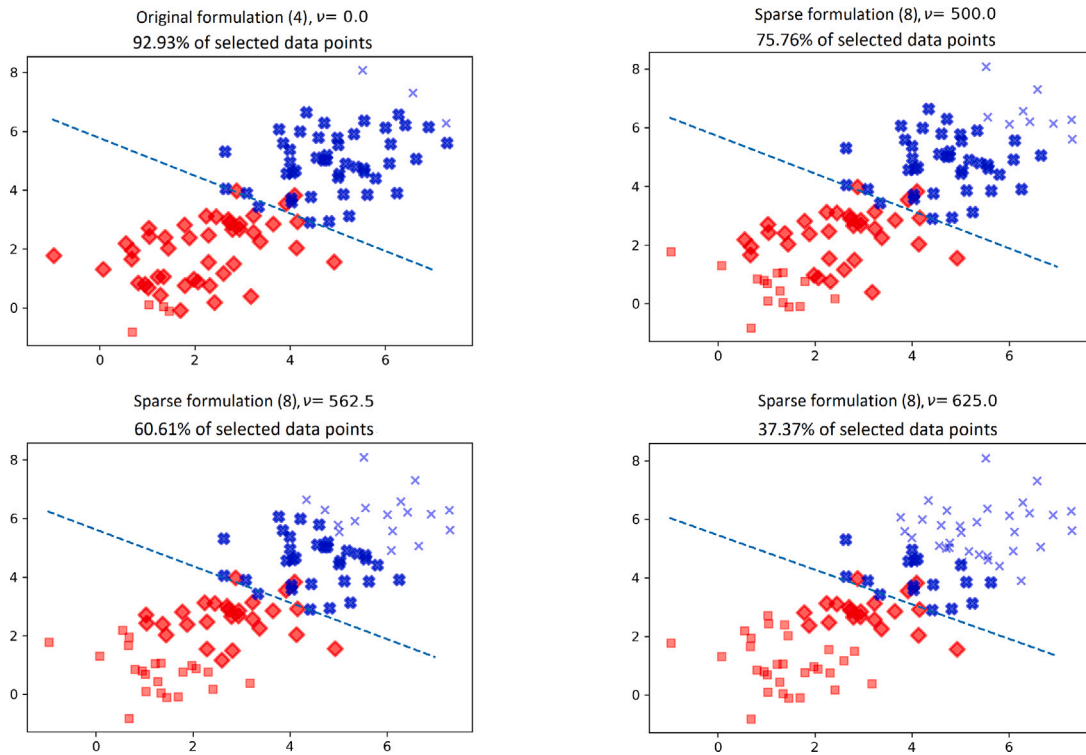


Fig. 3. Scatterplots of the Synth dataset with 2 classes (red and blue), $p = 2$ and $N = 99$, for four values of the hyperparameter $\nu \in \{0, 500, 562.5, 625\}$. Setting $\nu = 0$ of the top-left plot amounts to solving the original formulation (4). Data points selected by sparsity-inducing formulation (with nonzero α_i) are represented with bold red rhombuses and bold blue crosses, discarded/excluded data points are represented with thin blue crosses and red small squares.

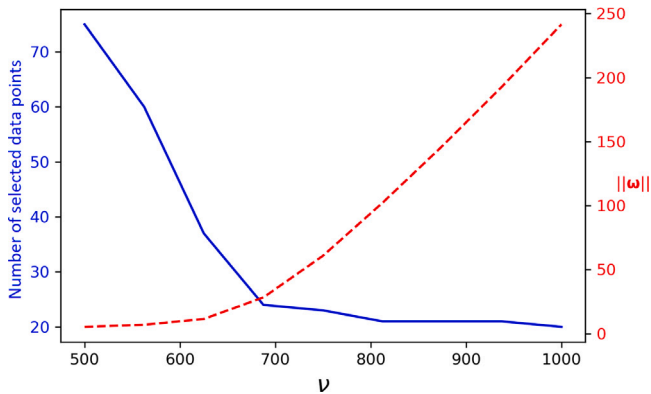


Fig. 4. Plots of the number of data points selected by the sparsity-inducing formulation (in blue) and of the norm of the vector ω (in red) as functions of the hyperparameter ν .

(see (4b) and (6)), this is equivalent to set to zero all α_i associated to data points whose distance from the decision boundary is larger than a certain value.

We implemented two versions of the thresholding heuristic to illustrate the difference with respect to our approach. In the first version, we solve the (non-sparse) primal formulation (2) (with $\nu = 0$) and construct the decision boundary by considering only the α_i exceeding a threshold that guarantees the same level of sparsity obtained by solving our (sparse) formulation (8) for a given $\nu > \bar{\nu}$. In the second version, we remove from the dataset those data points whose corresponding α_i have been set to zero according to the first version of the heuristic, and we train again the KLR model solving formulation (2) with respect to the reduced dataset. Note that the second version of the heuristic does not exploit at all information related to easy-to-classify data points,

since it only considers the data points selected by the first version of the heuristic. For our illustrative example, the second version of the heuristic yields a naive classifier that assigns the same class label to all data points. Hence, we briefly discuss the second version below, without reporting the results.

For the Synth dataset we solve formulation (4) with $C = 60$, by setting $\nu = 0$ for the non-sparse solution and $\nu = 1100$ for the sparse one. For $\nu = 1100$, the optimal solution selects 20 data points, while, for $\nu = 0$, 92 data points are considered. To guarantee the same level of sparsity we select from the optimal solution obtained with $\nu = 0$ the 20 data points with higher α_i s.

Fig. 5 shows a scatter plot of the Synth dataset, where class membership is represented by color (red and blue). The linear classifiers produced by Sparse KLR (depicted as a blue dashed line) and the thresholding heuristic (shown as an orange solid line) are shown. Data points represented with bold rhombuses and crosses are those selected for classification by both our Sparse KLR and the thresholding heuristic. It is interesting to note that the two methods select different data points. In particular, the cyan data point is only selected by the Sparse KLR, while the black data point is only selected by the thresholding heuristic. Moreover, not only the two hyperplane classifiers differ substantially but also the one obtained with the sparse formulation turns out to be more accurate (95% accuracy vs. 92%). Indeed, the Sparse KLR classifier tends to be very similar to the one obtained by training the non sparse KLR formulation (depicted in Fig. 5 as a green-colored dashed line) with the same $C = 60$, which is the best possible in terms of training accuracy.

Another interesting aspect influencing the difference between the two classifiers is the relative weight of each α_i with respect to the sum of all other selected data points, namely $\alpha_i / \sum_{i \in I} \alpha_i$. Fig. 6 shows histograms with the relative weights of the 20 data points (whose indices are reported in the x-axis) selected by the sparse formulation (in blue) and the thresholding heuristic (in yellow). In our sparse formulation, the relative weights of the selected data points are more

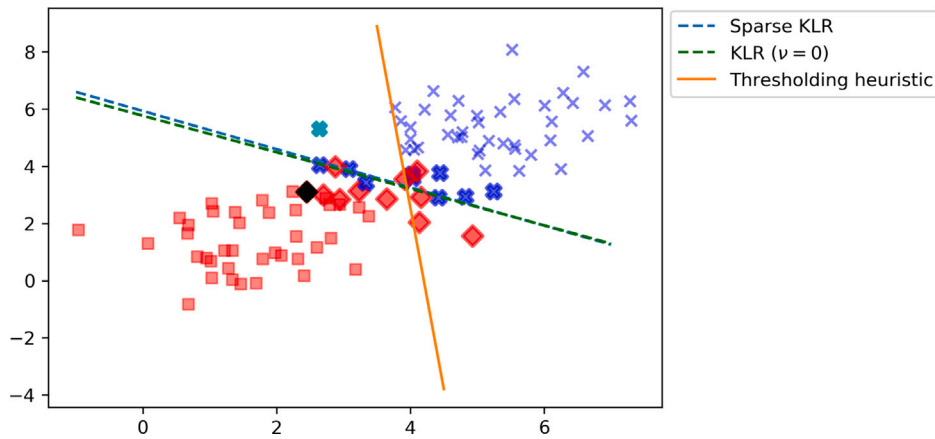


Fig. 5. Scatter plot of the Synth dataset. The three lines represent the decision boundary related to Sparse KLR (blue dashed line), Thresholding heuristic (orange solid line), and the KLR formulation with $\nu = 0$ (green dashed line). The rhombuses and crosses represent the selected data points for the red and blue classes, respectively. The cyan data point is selected solely by the sparse formulation while the black one is selected by imposing the threshold.

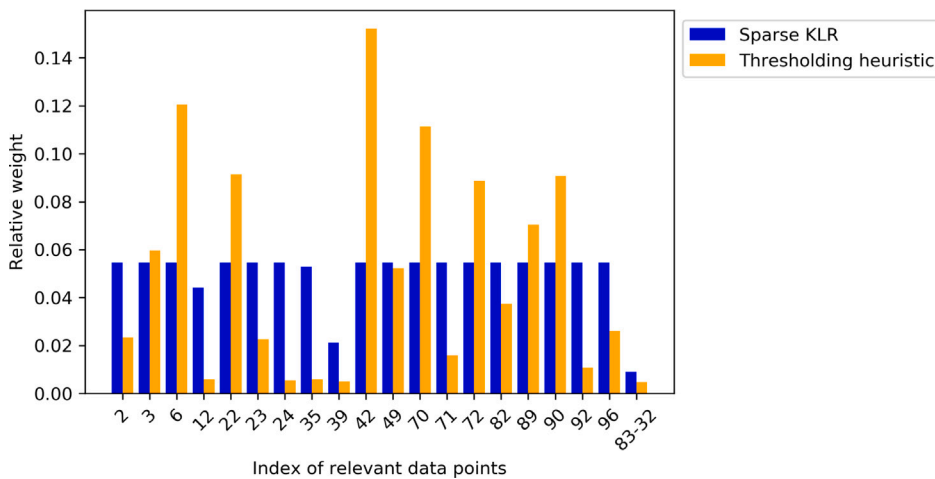


Fig. 6. Histograms representing the relative weights ($\alpha_i / \sum_{i \in I} \alpha_i$) of selected data points in the solutions provided by the Sparse KLR formulation (blue) and the thresholding heuristic (yellow).

evenly distributed than those obtained by the thresholding heuristic. In summary, the sparse KLR formulation and the thresholding heuristic yield two distinct decision boundaries because they select different subsets of data points and lead to different distributions of the relative weight associated to the corresponding dual variables.

A second possible version of the thresholding heuristic consists in training the non sparse KLR formulation on the restricted dataset containing only the data points identified by the first thresholding heuristic. When comparing our sparse KLR classifier with that of the second thresholding heuristic, we notice that the latter achieves a substantially lower accuracy. This is because, when solving formulation (4) considering only the 20 selected data points, we focus only on the data points that are either misclassified or very close to the boundary, without taking into account the easily classifiable data points. We do not report this classifier in the figure since it classifies all the data points in the same class.

4. A decomposition method for training sparse binary KLR

Decomposition methods are widely used in training ML models whenever, due to dimensionality, standard optimization methods involving first or second order information are out of reach (see e.g., Platt 1998, Lucidi et al. 2009, Manno et al. 2018, 2016). They can be designed to leverage the underlying structure of the training

problems, with positive impacts on the computational efficiency and on the ability to escape from poor quality solutions (see e.g., Grippo et al. 2015, Amaldi et al. 2023, Consolo et al. 2026).

Decomposition algorithms iteratively split the original problem into a sequence of smaller subproblems in which, at each iteration, only a subset of variables, the so-called working set, is involved in the optimization process, while the other ones are kept fixed to their current values. The choice of the variables to be included, commonly denoted as working set selection, is crucial for the computational efficiency and for the theoretical convergence properties.

In the next subsection, we present a SMO-type decomposition method for formulation (22), and we state its asymptotical global convergence property which is proved in Appendix B.

4.1. A SMO-type decomposition algorithm based on second-order information

The structure of the dual KLR formulation (22) is amenable to decomposition. In particular, as similarly noticed in Keerthi et al. (2005) for formulation (5), (22) allows for the adaptation to KLR of the efficient SMO-type decomposition algorithms developed for SVMs.

In SMO-type algorithms, at each iteration k the working set involves only two variables, whose indices are referred to as $\{i^*, j^*\} \subseteq$

$\{1, \dots, N\}$.² Notice that, given a current feasible solution α and the working set indices $\{i^*, j^*\}$, due to equality constraint (22b), the only feasible directions for variables α_{i^*} and α_{j^*} are, respectively, of the form $\frac{t}{y_{i^*}}$ and $-\frac{t}{y_{j^*}}$, where t represents the steplength. Therefore, the two-dimensional SMO subproblem can be recast into a one-dimensional problem with respect to the steplength variable t as follows

$$\min_t \quad \phi_{i^*, j^*}(t) = \frac{1}{2} \|\omega(\tilde{\alpha}(t))\|^2 + C \sum_{i \in I} G\left(\frac{\tilde{\alpha}_i(t)}{C}\right) - \lambda \sum_{i \in I} \tilde{\alpha}_i(t) \quad (30a)$$

$$\text{s.t.} \quad \bar{0} \leq \tilde{\alpha}_{i^*}(t) \leq \bar{C} \quad (30b)$$

$$\bar{0} \leq \tilde{\alpha}_{j^*}(t) \leq \bar{C}, \quad (30c)$$

where $\tilde{\alpha}_{i^*}(t) = \alpha_{i^*} + \frac{t}{y_{i^*}}$, $\tilde{\alpha}_{j^*}(t) = \alpha_{j^*} - \frac{t}{y_{j^*}}$ and $\tilde{\alpha}_s(t) = \alpha_s$ for all s with $s \neq i^*$ and $s \neq j^*$.

Before describing the proposed working set selection procedure, we introduce the concept of ‘‘violating pair’’, consisting of a pair of indices $\{i, j\}$ whose associated variables α_i and α_j violate together the optimality conditions (27). A violating pair, which is a good candidate to be included in the working set, can be determined by selecting one index $i \in I_{up}$ and one index $j \in I_{low}$ such that

$$-y_i \nabla f(\alpha)_i > -y_j \nabla f(\alpha)_j. \quad (31)$$

Among all possible violating pairs, consider $\{i^{MVP}, j^{MVP}\}$ associated to the maximal violation of the optimality conditions (27), the so called ‘‘Maximal Violating Pair’’ (MVP). The MVP can be easily determined (at the cost of ordering elements $-y_i \nabla f(\alpha)_i$) as

$$i^{MVP} = \arg \max_{i \in I_{up}(\alpha)} -y_i \nabla f(\alpha)_i \quad \text{and} \quad j^{MVP} = \arg \min_{j \in I_{low}(\alpha)} -y_j \nabla f(\alpha)_j. \quad (32)$$

In the SMO-type decomposition method for KLR proposed in Keerthi et al. (2005), the working set selection procedure consists in iteratively selecting the MVP according to first-order information.

Inspired by the computationally more efficient working set selection procedure proposed in Fan et al. (2005) for SVMs, we devise for the KLR dual formulation (22) a working set selection procedure which takes advantage of second-order information. After setting the first index i^* as i^{MVP} , one selects, among all possible candidate indices $j \in I_{low}(\alpha)$ satisfying (31), the index j^* for which the quadratic approximation of the objective function (30a) of the sub-problem associated to $\{i^*, j^*\}$ is minimized. In particular, for a given pair of indices $\{i^*, j^*\}$ we approximate the objective function (30a) of the corresponding sub-problem, with its second-order Taylor expansion in a neighborhood of $t = 0$, as

$$\phi_{i^*, j^*}(0) + \phi'_{i^*, j^*}(0)t + \frac{1}{2} \phi''_{i^*, j^*}(0)t^2. \quad (33)$$

Then, the index j^* is selected as the one that minimizes (33) the most. The minimizer of (33) can be obtained analytically. Indeed, for a given index j , the t value minimizing (33), denoted by t^* , is $t^* = -\frac{v_{i^*, j}}{2q_{i^*, j}}$ and

the corresponding objective function value is $-\frac{v_{i^*, j}^2}{2q_{i^*, j}}$ (up to a constant term), where

$$v_{i^*, j} = \sum_{s \in I} \alpha_s y_s \text{Ker}(\mathbf{x}_j, \mathbf{x}_s) + y_j G'\left(\frac{\alpha_j}{C}\right) - \lambda y_j - \sum_{s \in I} \alpha_s y_s \text{Ker}(\mathbf{x}_{i^*}, \mathbf{x}_s) - y_{i^*} G'\left(\frac{\alpha_{i^*}}{C}\right) + \lambda y_{i^*} \quad (34)$$

and

$$q_{i^*, j} = \text{Ker}(\mathbf{x}_{i^*}, \mathbf{x}_{i^*}) + \text{Ker}(\mathbf{x}_j, \mathbf{x}_j) - 2\text{Ker}(\mathbf{x}_{i^*}, \mathbf{x}_j) + \frac{C}{\alpha_{i^*}(C - \alpha_{i^*})} + \frac{C}{\alpha_j(C - \alpha_j)}. \quad (35)$$

² For simplicity of notation, the dependence of $\{i^*, j^*\}$ from iteration k is omitted.

It is important to note that the solution i^* minimizing problem (33) is used exclusively for selecting the second index j^* in the working set selection procedure. Once the pair (i^*, j^*) is determined, the update of the dual variables $(\alpha_{i^*}, \alpha_{j^*})$ is performed by solving subproblem (30).

To summarize, we propose the following second-order working set selection procedure:

Second-order Working Set Selection (WSS)

- (1) Select $i^* = \arg \max_i \{-y_i \nabla f(\alpha)_i \mid i \in I_{up}(\alpha)\} = \arg \max_i \{-\sum_{j \in I} \alpha_j y_j \text{Ker}(\mathbf{x}_i, \mathbf{x}_j) - y_i G'\left(\frac{\alpha_i}{C}\right) - \lambda y_i \mid i \in I_{up}\}$
- (2) Select $j^* = \arg \min_j \{-\frac{v_{i^*, j}^2}{q_{i^*, j}} \mid j \in I_{low}(\alpha), -y_j \nabla f(\alpha)_j < -y_{i^*} \nabla f(\alpha)_{i^*}\}$, where $v_{i^*, j}, q_{i^*, j}$ are as in (34) and (35)
- (3) Return the pair $\{i^*, j^*\}$

and the SMO-type decomposition training algorithm described in SMO Algorithm.

SMO Algorithm Sequential Minimal Optimization decomposition algorithm

- 1: Choose α^0 feasible starting solution and set $k = 0$
 - 2: if α^k satisfies (27) then
 - 3: Set $\alpha^* = \alpha^k$
 - 4: else
 - 5: Select $\{i, j\}$ according to WSS procedure
 - 6: Determine α^{new} as the optimal solution of (30)
 - 7: Let $\alpha^{k+1} = \alpha^{new}$, set $k = k + 1$ and go back to 2
 - 8: end if
 - 9: return α^*
-

For any infinite sequence generated by SMO Algorithm we can establish global asymptotic convergence. The convergence analysis, which is reported in Appendix B, is an adaptation of the one for the second-order SMO-type method presented in Chen et al. (2006) for SVM. Since due to the presence of the term $C \sum_{i \in I} G\left(\frac{\alpha_i}{C}\right)$ the objective function (22a) is convex but not quadratic (like for SVM), we consider a quadratic approximation in the WSS.

The convergence result is as follows.

Proposition 2. Assume that the kernel matrix K is positive semidefinite. Let $\{\alpha^k\}$ be the infinite sequence generated by the SMO-type method Algorithm SMO Algorithm. Then $\{\alpha^k\}$ globally converges to the unique optimal solution of problem (22).

It is worth pointing out that the assumption in Proposition 2 of positive semidefiniteness of kernel matrix K (i.e., satisfying the Mercer’s Theorem Mercer, 1909) is very mild from a practical point of view. Indeed, several kernel functions usually adopted in practical applications (e.g., linear, polynomial, Gaussian, and Laplacian kernels) induce positive semidefinite (or definite) kernel matrices.

5. Computational experiments

In this section, we first compare the Sparse KLR training formulation (22) solved to optimality via SMO Algorithm (referred to as S-KLR), with IVM, $\ell_{1/2}$ -KLR, and SVM on a collection of 12 benchmark datasets in terms of testing accuracy and sparsity. Then, a second set of experiments is carried out to assess independently the impact of the proposed formulation variant in terms of sparsity, and of the second order WSS procedure in terms of CPU time reduction.

5.1. Experimental settings and datasets

S-KLR is implemented in Python 3.9.7, using the Cython extension to convert the SMO Algorithm part of the code in C in order to improve efficiency. For SVM training, the scikit-learn 1.1.2

Table 1
Description of 12 datasets.

Dataset	Abbreviation	N	p
banknote	banknote	1372	4
coil2000	coil2000	9822	85
diabetes	diabetes	768	8
ionosphere	ionosphere	351	34
magic	magic	19020	10
monk2	monk2	432	6
ringnorm	ring	7400	20
sonar	sonar	208	60
spambase	spambase	4597	57
twonorm	twonorm	7400	20
waveform	waveform	5000	21
wisconsin-breast-cancer	wisconsin	569	30

package is used. To train IVM, the Import Vector Machine Classifier software (Roscher et al., 2012a,b) (version 4.4) in Matlab 2022b is adopted. As to the $\ell_{1/2}$ -KLR, the half thresholding has been implemented in Python 3.9.7 using the jax 0.4.23 package for the efficient calculation of the gradient. The experiments are carried out on a PC with Intel(R) Core(TM) i7-11370H CPU @ 3.30 GHz with 16 GB of RAM.

To initialize S-KLR we follow the approach outlined in Keerthi et al. (2005). For SMO Algorithm, it is essential to guarantee that the starting solution α^0 is feasible with respect to constraints (22b) and (22c). By denoting with m_1 and m_2 the number of data points in the first and second class, respectively, if $\max\{m_1, m_2\} < \tau^{-1}$, then α^0 components α_i^0 can be set to $\frac{1}{m_1}$ if the i th data point is in the first class and $\frac{1}{m_2}$ otherwise. In the unlikely case in which $\max\{m_1, m_2\} > \tau^{-1}$, then if $m_1 > m_2$ we can set $\alpha_i = \bar{0}$ for i in the first class, and $\alpha_i = \frac{m_1 \bar{C}}{m_2}$ otherwise, provided that $\frac{m_1 + m_2}{m_2} \tau \leq C$.

Two alternative stopping criteria are considered for SMO Algorithm: a maximum number of iterations (set to 10000), and the satisfaction of the optimality conditions (27) up to a certain tolerance $\tau = 1e^{-5}$. As for the parameter γ appearing in the definition of the bounds $\bar{0}$ and \bar{C} , we set $\gamma = 1e^{-5}$.

Notice that, due to the auxiliary bounds of constraints (22c), no optimal dual variables will assume exactly the value zero, as every component α_i of any feasible solution must satisfy $\alpha_i \geq \bar{0} = \gamma > 0$. However, in the reported experiments the components α_i^* of the optimal solution α^* whose values reach the lower bound ($\alpha_i^* = \gamma$) are considered equal to zero and are not involved in the model for prediction.

Subproblem (30) is solved with the “Newton-CG” algorithm (see e.g., Nocedal and Wright 1999) implemented in the Scipy 1.9.0 Python package. Numerical experiments are conducted on 12 datasets from the UCI Machine Learning Repository (Asuncion and Newman, 2007) and the LIBSVM repository (Chang and Lin, 2011).³ The number of data points N in each dataset ranges from 208 to 19020, while the number of features p ranges from 4 to 85 (see Table 1 for details). For all datasets, all features are scaled in the interval $[0, 1]$, and the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ with $\sigma = 1$ is adopted.

The testing accuracy and sparsity of the ML models obtained using all the compared methods are evaluated through k -fold cross-validation, with $k = 5$. Since the values of hyperparameters C and λ can significantly affect the model performance, we conducted a grid search to determine good estimates. For every fold and model, the values of C and λ were selected using a validation set consisting of 5% of the training data points. After the selection of both hyperparameters, the

³ The three-class dataset waveform, considered in both Keerthi et al. (2005) and Zhu and Hastie (2005), has been transformed into a binary classification problem where the data points in class 1 must be separated from those in classes 0 and 2. It is worth noting that the separation of the data points in class 0 or 2 does not significantly affect the overall results.

model was retrained on the entire training set with the chosen values. For all methods, both performance measures are computed by varying the regularization parameter C in the interval $\{10^r : -4 \leq r \leq 4, r \in \mathbb{Z}\}$. For S-KLR, a further hyperparameter λ related to the sparsity inducing term must be set. For a given regularization value C , λ varies over 10 equally spaced values in the interval $[0, C]$ (including 0 and C).

Although the above-mentioned grid search for S-KLR over both C and λ is computationally heavier than in the presence of a single hyperparameter (like for SVMs), in Appendix C we provide experimental evidence that the λ fine-tuning can be avoided. Indeed, for a fixed value of C , λ can be set to a relatively small value depending on C , without compromising S-KLR performance.

Concerning $\ell_{1/2}$ -KLR, in Xu et al. (2013) the authors adapt an approach proposed originally in Xu et al. (2012) for which the value of the hyperparameter C is automatically selected based on a predefined level of sparsity. They assume that a priori information on a “proper” sparsity level is available and do not carry out a grid search on the C value. In this work, we consider the general setting where no prior information is available and we apply in the $\ell_{1/2}$ -KLR experiments the same standard grid search used for S-KLR, IVM and SVM.

5.2. Numerical results

We have carried out two types of experiments. First, we compare the performance of the proposed S-KLR formulation and decomposition algorithm with that of IVM, $\ell_{1/2}$ -KLR, and SVM in terms of testing accuracy and sparsity. Second, we investigate how the proposed formulation variant (22) improves model sparsity with respect to the formulation in Keerthi et al. (2005), and we assess the impact of the second order WSS procedure in terms of CPU time with respect to a benchmarking version of S-KLR implementing the same first order WSS procedure in Keerthi et al. (2005).

As to the first type of experiments, Table 2 reports the average testing accuracy and sparsity level across the $k = 5$ folds. These results were obtained by selecting the best values of the hyperparameters (C for all methods and also λ for S-KLR) through validation over the aforementioned grid of values. For S-KLR, $\ell_{1/2}$ -KLR, and SVM, the table also reports the average results obtained by selecting the hyperparameters, via validation, that yield the sparsest model among the three most accurate ones. These model versions are referred to as “S-KLR sparsest of 3 most accurate”, “ $\ell_{1/2}$ -KLR sparsest of 3 most accurate”, and “SVM sparsest of 3 most accurate”. It is worth noting that in some cases, such as for S-KLR on twonorm dataset or for SVM on the spambase dataset, the highest level of sparsity does not necessarily coincide with the best testing accuracy.

The results in Table 2 indicate that both S-KLR and SVM achieve equivalent and substantially higher accuracy levels than IVM and $\ell_{1/2}$ -KLR. In particular, S-KLR and SVM yield an average testing accuracy of approximately 92.5%, while IVM of 84.9% and $\ell_{1/2}$ -KLR of 86.7%. S-KLR attains a good average degree of sparsity (42.18% of data points are used), but SVM yields sparser models (only 28.8% of data points are support vectors). It is worth noting that, when considering the experiments for the sparsest of the 3 most accurate models, the average testing accuracy of S-KLR still amounts to 92.5%, whereas that of SVM decreases to 90.5%, and the sparsity gap between S-KLR and SVM narrows to 6.3% (33.1% and 26.8%, respectively). Moreover, when comparing the sparsest model among the 3 most accurate S-KLR model with SVM, the average sparsity difference further decreases to less than 4.5%, while maintaining the same level of accuracy.

IVM, which is based on a greedy heuristic, tends to achieve remarkable sparsity levels (5.6%) but at the cost of significantly compromising the testing accuracy (on average 84.9%). For the banknote, diabetes, ionosphere, magic, monk2, and ring datasets, IVM yields testing accuracies which are lower by at least 6.7% and by up to 21% than those provided by S-KLR and SVM. Note that the standard deviation of the

Table 2

Comparative results in terms of average testing accuracy and average percentage of data points involved in the model sparse representation (% data points) obtained with S-KLR, $\ell_{1/2}$ -KLR, IVM and SVM. For S-KLR, $\ell_{1/2}$ -KLR and SVM the additional columns “sparsest of 3 most accurate” represent the best sparsity results over the three most accurate models in terms of accuracy. The standard deviation of the average accuracy is reported in parentheses. In the last row, all the average testing accuracy and the average ratio are highlighted in bold.

Dataset	S-KLR sparsest of 3 most accurate		S-KLR		$\ell_{1/2}$ -KLR sparsest of 3 most accurate		$\ell_{1/2}$ -KLR		IVM		SVM sparsest of 3 most accurate		SVM	
	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points	accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points
banknote	1	0.06	1	0.162	0.835	0.576	0.935	0.985	0.882	0.008	1	0.009	1	0.009
coil2000	0.94	0.155	0.94	0.265	0.94	0.0002	0.94	0.0002	0.94	0.0001	0.94	0.119	0.94	0.119
diabetes	0.77	0.529	0.767	0.724	0.678	0.171	0.755	0.747	0.653	0.002	0.745	0.49	0.767	0.531
ionosphere	0.946	0.488	0.946	0.488	0.726	0.011	0.946	0.304	0.879	0.221	0.926	0.444	0.934	0.459
magic	0.857	0.69	0.857	0.754	0.648	0	0.741	0.989	0.719	0.0001	0.875	0.293	0.874	0.299
monk2	0.958	0.292	0.958	0.297	0.528	0	0.937	0.413	0.833	0.027	0.974	0.222	0.976	0.286
ring	0.978	0.085	0.978	0.113	0.505	0.135	0.744	0.857	0.763	0.002	0.974	0.07	0.978	0.09
sonar	0.866	0.924	0.856	0.941	0.534	0	0.756	0.464	0.822	0.347	0.643	0.924	0.851	0.923
spambase	0.935	0.246	0.935	0.518	0.675	0.302	0.804	0.913	0.902	0.008	0.931	0.173	0.935	0.206
twonorm	0.976	0.075	0.978	0.141	0.487	0	0.976	0.752	0.977	0.002	0.975	0.075	0.977	0.101
waveform	0.91	0.282	0.91	0.516	0.899	0.568	0.905	0.922	0.884	0.005	0.909	0.287	0.909	0.287
wisconsin	0.974	0.145	0.975	0.143	0.805	0.039	0.96	0.707	0.936	0.052	0.965	0.115	0.975	0.147
Average	0.926 (0.066)	0.331	0.925 (0.067)	0.422	0.688 (0.157)	0.15	0.867 (0.097)	0.671	0.849 (0.096)	0.056	0.905 (0.106)	0.268	0.926 (0.067)	0.288

Table 3

Average testing accuracy and average percentage of data points involved in the model sparse representation (% data points) achieved by KLR-1ord, KLR-2ord, and S-KLR. KLR-1ord coincides with the method proposed in Keerthi et al. (2005). The standard deviation of the average accuracy is reported in parentheses.

Dataset	KLR 1-ord (Keerthi et al., 2005)		KLR 2-ord		S-KLR	
	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points	Accuracy	Ratio of selected data points
banknote	0.977	1	1	0.694	1	0.162
coil2000	0.94	0	0.94	0.155	0.94	0.265
diabetes	0.763	1	0.776	1	0.767	0.724
ionosphere	0.943	1	0.94	1	0.946	0.488
magic	0.839	1	0.807	0.69	0.857	0.754
monk2	0.96	1	0.972	0.96	0.958	0.297
ring	0.949	1	0.978	0.817	0.978	0.113
sonar	0.823	1	0.823	1	0.856	0.941
spambase	0.925	1	0.934	0.95	0.935	0.518
twonorm	0.978	1	0.979	0.987	0.978	0.141
waveform	0.909	1	0.911	0.998	0.91	0.516
wisconsin	0.952	1	0.951	0.978	0.975	0.143
Average	0.913 (0.068)	0.917	0.918 (0.074)	0.852	0.925 (0.068)	0.422

average testing accuracy across all datasets is substantially higher for IVM (0.096) than for S-KLR and SVM (about 0.066).

$\ell_{1/2}$ -KLR achieves considerably lower testing accuracy compared with S-KLR and SVM (on average 5.8% lower), and considerably lower sparsity level (67.11% of the data points are used) with respect to S-KLR, SVM and to IVM. As far as the 3 most accurate model versions are concerned, $\ell_{1/2}$ -KLR yields testing accuracies that are comparable to those provided by S-KLR and SVM for the coil2000 and waveform datasets, whereas it achieves testing accuracies which are at least 6.7% lower for all the other datasets.

To summarize, S-KLR and SVM achieve a better trade-off between testing accuracy and sparsity with respect to IVM and $\ell_{1/2}$ -KLR. Concerning the comparison between S-KLR and SVM, both approaches provide equivalent and substantially higher accuracy levels than IVM and $\ell_{1/2}$ -KLR. It is worth emphasizing that S-KLR achieves a remarkable degree of sparsity compared with standard KLR and that the moderate sparsity difference in favor of the (intrinsically sparse) SVM is compensated by the more informative KLR probabilistic information.

In the second type of experiments, we first assess the improvement in model sparsity and then we evaluate the reduction in CPU training time.

As far as sparsity is concerned, we compare the proposed S-KLR approach with the original formulation (5) solved using a SMO-type algorithm based on the first-order working set selection procedure in Keerthi et al. (2005), denoted as KLR-1ord. We also consider KLR-2ord which extends KLR-1ord by adopting the same second-order WSS procedure devised for S-KLR.⁴ Notice that KLR-2ord is a special case of S-KLR obtained by setting $\lambda = 0$, that is, S-KLR generalizes KLR-2ord with the sparsity inducing variant of the KLR formulation. It is worth pointing out that the comparison between S-KLR and KLR-2ord aims at verifying that the high level of sparsity achieved by S-KLR is essentially due to the ρ term in the sparse KLR formulation (22), and not to the second-order WSS procedure. As mentioned in Section 4.1, the latter is devised for efficiency reasons.

The results of the sparsity experiments are shown in Table 3. In particular, for each dataset, we report the average testing accuracy

⁴ For a better comparison with KLR-2ord and S-KLR, the KLR-1ord implementation slightly differs from the method in Keerthi et al. (2005). To improve numerical stability, in Keerthi et al. (2005) whenever a variable reaches an auxiliary bound it is fixed at that bound and only reconsidered in a second refinement phase. Although in KLR-1ord the auxiliary bounds are only used to constrain subproblem (30), the difference with respect to the method in Keerthi et al. (2005) is negligible due to the strict convexity of the original problem (both versions converge to the same unique solution).

of KLR-1ord and KLR-2ord, obtained by selecting the optimal hyperparameter C on a validation set, as explained above. In the case of S-KLR, the reported accuracy corresponds to the average performance achieved by jointly tuning C and λ . Comparing S-KLR with KLR-2ord, we observe that the sparsity term in S-KLR yield a substantial reduction in the number of data points that are actually used to determine the decision boundaries (from 85.2% to 42.2%). The second-order WSS procedure in KLR-2ord seems to lead to a slight increase in the degree of sparsity with respect to KLR-1ord. In terms of accuracy, both S-KLR and KLR-2ord yield a moderate improvement with respect to KLR-1ord.

Concerning CPU training time, Table 4 reports the training times in seconds required by the S-KLR using the second order WSS procedure (denoted as S-KLR-2ord) and a benchmarking version of S-KLR implementing the same first order WSS in Keerthi et al. (2005) (denoted as S-KLR-1ord). For each dataset, the average CPU times over all the values of the grid search on the C and λ hyperparameters are indicated. In particular, the first two columns correspond to S-KLR-1ord and S-KLR-2ord, while the last column shows the ratio between the latter and the former CPU times.

The computational results show that S-KLR-2ord yields an average CPU time saving of about 18%. Except for spambase, where S-KLR-1ord is slightly faster (about 1%), S-KLR-2ord always leads to a CPU time reduction, often above the 20% of time savings, with peaks of 36.6% (ionosphere) and 38.2% (wisconsin). It is worth emphasizing that such favorable results of S-KLR-2ord with respect to S-KLR-1ord are not obvious. Indeed, since the objective function of the sparse KLR formulation (22) is not quadratic, the second-order information used to select the second WSS index may not be more reliable than the first-order one used in S-KLR-1ord. This clearly differs from the SVM case in which, due to the quadratic objective function, the information used in the second-order WSS is exact.

6. Concluding remarks

We revisited the problem of training sparse binary KLR so as to improve the balance between model sparsity and testing accuracy with respect to the available approaches. In particular, we presented a variant of the strictly convex KLR formulation capable of inducing sparsity and a SMO-type training algorithm to exactly solve its dual, which exploits second-order information and admits asymptotic convergence guarantees.

S-KLR was tested on 12 benchmark datasets and compared with the sparse KLR heuristic IVM, the regularization approach $\ell_{1/2}$ -KLR, and the popular sparse kernel model SVM. The numerical results indicate that S-KLR and SVM achieve a better trade-off between testing accuracy

Table 4

Average training time required by S-KLR-1ord and S-KLR-2ord for all the 12 datasets.

Dataset	CPU time (s)		Ratio 2ord/1ord
	S-KLR-1ord	S-KLR-2ord	
banknote	9.649	7.559	0.783
coil2000	22.654	16.633	0.734
diabetes	8.36	7.068	0.845
ionosphere	10.955	6.989	0.638
magic	53.779	52.841	0.982
monk2	6.388	4.548	0.712
ring	21.12	20.011	0.947
sonar	6.85	5.204	0.759
spambase	16.282	16.525	1.015
twonorm	25.21	24.73	0.981
waveform	16.658	14.077	0.845
wisconsin	8.839	5.464	0.618
Average	17.23 (13.2)	15.13 (13.6)	0.821

and sparsity than IVM and $\ell_{1/2}$ -KLR. S-KLR and SVM obtain both similar high accuracy levels and good sparsity levels. The sensibly lower sparsity of S-KLR with respect to SVM is compensated by the informative class probability estimates provided by the former model. As far as training times are concerned, the proposed second order WSS procedure in our SMO algorithm turns out to have a significant impact. From a theoretical point of view, we investigated the relationship between primal variable ρ enforcing sparsity in the primal formulation and the dual hyperparameter λ . Moreover, we derived upper bounds on the λ hyperparameter beyond which a subset of all the dual variables α_i of all of them reach C up to a small tolerance required for numerical stability. Empirical evidence suggests that S-KLR does not require a time consuming grid search procedure involving both hyperparameters C and λ . Indeed, for the considered datasets, the value of the sparsity hyperparameter λ can be set according to a simple rule without compromising the performance.

Interesting research directions include: the adoption of algorithmic strategies like kernel caching, low-rank kernel approximations, and parallel training, to improve the SMO-type algorithm in order to scale to larger datasets; the development of a KLR variant which induces feature sparsity (only a subset of features are actually used); the extension of the S-KLR method to multiclass classification. Finally, we could also try to gain further theoretical insight into the level of sparsity obtained by the S-KLR model via the proposed formulation.

CRedit authorship contribution statement

Antonio Consolo: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Andrea Manno:** Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Edoardo Amaldi:** Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization.

Appendix A. Considerations on dual variables and the regularization parameter

For the sake of notation we define $I_+ = \{i \in I : y_i = +1\}$, $I_- = \{i \in I : y_i = -1\}$, $n_+ = |I_+|$, $n_- = |I_-|$, and $Q_{ij} = y_i y_j K_{ij}$.

We recall that we assume matrix Q to be positive semidefinite. For convenience we report the convex formulation (18):

$$\begin{aligned} \min \quad & f_\lambda(\alpha) = \frac{1}{2} \alpha^\top Q \alpha + C \sum_{i=1}^n G\left(\frac{\alpha_i}{C}\right) - \lambda \sum_{i=1}^n \alpha_i \\ \text{s.t.} \quad & y^\top \alpha = 0. \end{aligned}$$

We define

$$h(\alpha) = \frac{1}{2} \alpha^\top Q \alpha + C \sum_{i=1}^n G\left(\frac{\alpha_i}{C}\right) \text{ and } f_\lambda(\alpha) = h(\alpha) - \lambda \mathbf{1}^\top \alpha.$$

It is easy to show that the function $h(\alpha)$ is bounded on $[0, C]^n$.

Lemma 2. *There exists $M < \infty$ such that $|h(\alpha)| \leq M$ for every $\alpha \in [0, C]^n$.*

Proof. Since Q is symmetric and positive semidefinite, on $[0, C]^n$ we have that $\alpha^\top Q \alpha \leq \text{eig}_{\max}(Q) \|\alpha\|^2 \leq \text{eig}_{\max}(Q) n C^2$ where $\text{eig}_{\max}(Q)$ is the maximum eigenvalue of Q . Moreover $G(\delta) \in [-\log 2, 0]$ for $\delta \in [0, 1]$. Thus $|h(\alpha)| \leq M$ with $M = \frac{1}{2} \text{eig}_{\max}(Q) n C^2 + C n \log 2$. \square

By considering formulation (18), the presence of $G(\cdot)$ in the objective function implies that the optimal solution is contained in the hypercube $[0, C]^n$, let us define this bounded version of the feasible region $\mathcal{F} = \{\alpha \in [0, C]^n : y^\top \alpha = 0\}$ and $m^* = \max_{\alpha \in \mathcal{F}} \sum_{i \in I} \alpha_i$.

Lemma 3. *Let $n_{\min} = \min\{n_+, n_-\}$. Then*

$$m^* = 2C n_{\min}.$$

Proof. Let define $S_+ = \sum_{i \in I_+} \alpha_i$ and $S_- = \sum_{i \in I_-} \alpha_i$, then $S_+ \leq C n_+$ and $S_- \leq C n_-$. From the equality constraints $\alpha^\top y = 0$ we have that $S_+ = S_- \leq C n_-$ ($S_- = S_+ \leq C n_+$), thus $S_+ = S_- \leq C \min\{n_+, n_-\} = C n_{\min}$. Therefore $m^* \leq 2C n_{\min}$. To obtain equality, it is sufficient to set $\alpha_i = C$ for all data points belonging to the minority class and for n_{\min} data points from the majority class, while setting $\alpha_i = 0$ for all remaining data points. \square

Proposition 3. *For any given $\lambda \in \mathbb{R}$, let $\alpha(\lambda)$ denote the minimizer of (18). Then*

$$\lim_{\lambda \rightarrow \infty} \sum_{i \in I} \alpha_i(\lambda) = m^* = 2C n_{\min}.$$

Proof. For any $\epsilon > 0$, if $\alpha \in \mathcal{F}$ satisfies $\sum_{i \in I} \alpha_i \leq m^* - \epsilon$, from Lemma 2 we have

$$f_\lambda(\alpha) = h(\alpha) - \lambda \sum_{i \in I} \alpha_i \geq -M - \lambda(m^* - \epsilon).$$

From Lemma 3 we know that there exists $\bar{\alpha} \in \mathcal{F}$ with $\sum_{i \in I} \bar{\alpha}_i = m^*$ and thus $f_\lambda(\bar{\alpha}) \leq M - \lambda m^*$. For $\lambda > 2M/\epsilon$ we have $f_\lambda(\alpha) > f_\lambda(\bar{\alpha})$. Therefore, for sufficiently large values of λ , any minimizer $\alpha(\lambda)$ must satisfy $\sum_{i \in I} \alpha_i(\lambda) \geq m^* - \epsilon$. Since this reasoning can be applied for any arbitrary small ϵ , we have:

$$\lim_{\lambda \rightarrow \infty} \sum_{i \in I} \alpha_i(\lambda) = m^* = 2C n_{\min}.$$

Since the sequence of $\alpha(\lambda)$ for growing λ is defined on a compact set \mathcal{F} , it admits a limit point $\alpha(\lambda)$ which, from the strict convexity of the objective function in (18), is the unique optimal solution. \square

In the case of balanced classes ($n_+ = n_-$) we have $n_{\min} = \frac{n}{2}$ and $m^* = Cn$. From Proposition 3 follows:

Corollary 1. *For any dataset with balanced classes ($n_+ = n_-$) the optimal solution $\alpha(\lambda)$ of formulation (18) is such that*

$$\lim_{\lambda \rightarrow \infty} \alpha(\lambda) = C \mathbf{1}.$$

Lemma 1. *For any value of $\lambda \in \mathbb{R}$ the optimal solution of formulation (18) $\alpha(\lambda)$ is such that*

$$0 < \alpha_i(\lambda) < C \quad \forall i \in I.$$

Proof. Let us proceed by contradiction. Suppose that $\alpha_k(\lambda) = 0$ for some index k . The two sets

$$R = \{i : y_i = y_k, \alpha_i(\lambda) > 0\} \text{ and } S = \{i : y_i = -y_k, \alpha_i(\lambda) < C\}$$

cannot both be empty. Indeed, if they were, we would have that

$$\mathbf{y}^T \boldsymbol{\alpha}(\lambda) = \sum_{i: y_i=y_k} \alpha_i(\lambda) y_k + \sum_{i: y_i=-y_k} \alpha_i(\lambda) (-y_k) = 0 - C y_k |\{i : y_i = -y_k\}| \neq 0,$$

thus $\boldsymbol{\alpha}(\lambda) \notin \mathcal{F}$. Therefore, we have either (i) at least one $m \in \mathcal{R}$, or (ii) at least one $i \in \mathcal{S}$.

Let us consider case (i). Then, we can define a direction \mathbf{d} with $d_k = 1$, $d_m = -1$, and $d_j = 0 \forall j \neq m, k$ such that $\mathbf{y}^T \mathbf{d} = 0$ and $\boldsymbol{\alpha}(\lambda) + \tau \mathbf{d} \in \mathcal{F}$ for any $\tau \in (0, \bar{\tau}]$ where $\bar{\tau} = C - \alpha_k(\lambda)$, hence \mathbf{d} is a feasible direction at $\boldsymbol{\alpha}(\lambda)$. It is easy to see that $\nabla f_k(\boldsymbol{\alpha}(\lambda)) d_k < 0$. Indeed, while the derivatives of $\frac{1}{2} \|\boldsymbol{\omega}(\boldsymbol{\alpha})\|^2$ and $-\lambda \sum_i \alpha_i$ are bounded over \mathcal{F} , function G is equal to $-\infty$ at zero (recall $\alpha_k = 0$). This contradicts the optimality condition $\nabla f(\boldsymbol{\alpha}(\lambda))^T \mathbf{d} \geq 0$.

Concerning case (ii), we can follow the same argument of case (i) (replacing \mathcal{R} with \mathcal{S} and considering $d_m = 1$) to obtain the same contradiction. Therefore, for any $\lambda \in \mathbb{R}$ we can guarantee that $\boldsymbol{\alpha}(\lambda) > \mathbf{0}$.

We omit the proof of $\boldsymbol{\alpha}(\lambda) < C$ that can be obtained with the same arguments of the previous cases (i) and (ii) related to $\boldsymbol{\alpha}(\lambda) > \mathbf{0}$ by simply inverting the orientation of the feasible direction \mathbf{d} . \square

We now recall the statement of [Proposition 1](#) (in [Section 3.2](#)) and report the proof.

Proposition 1. *Assume without loss of generality that I_- is the minority class. Let $\boldsymbol{\alpha}(0)$ denote the optimal solution of formulation (18) with $\lambda = 0$. Consider any positive tolerance γ such that $\gamma < C - \max_{i \in I_-} \alpha_i(0)$, and define $\bar{C} = C - \gamma$. For any*

$$\lambda \geq \max_{i \in I_-} \sum_{j \in I} (\bar{C} - \alpha_j(0)) y_j K_{ji}, \quad (\text{A.1})$$

the optimal solution $\boldsymbol{\alpha}(\lambda)$ of formulation (18) satisfies $\alpha_i(\lambda) \geq \bar{C} \forall i \in I_-$.

Proof. From [Proposition 3](#), we know that as λ increases, $\alpha_i(\lambda) \forall i \in I_-$ tends to C . Therefore, there exists a sufficiently large $\bar{\lambda}$ for which for any $\lambda \geq \bar{\lambda}$ we have $\alpha_i(\lambda) > \alpha_i(0) \forall i \in I_-$. For any $\lambda \geq \bar{\lambda}$, recalling the monotone relationship between dual α_i and primal ξ_i (see (12)), we have that $\xi_i(\lambda) > \xi_i(0)$, hence from (8b)

$$\rho(\lambda) - y_i(\boldsymbol{\omega}(\lambda) \cdot \mathbf{z}_i - b(\lambda)) \geq -y_i(\boldsymbol{\omega}(0) \cdot \mathbf{z}_i - b(0)) \quad \forall i \in I_-. \quad (\text{A.2})$$

Omitting the variable b for convenience⁵ and using Eq. (10) into (A.2), we obtain

$$\rho(\lambda) > y_i \sum_{j \in I} (\alpha_j(\lambda) - \alpha_j(0)) y_j K_{ij} \quad \forall i \in I_-.$$

Thus, recalling that $\rho(\lambda) = \lambda$, to obtain $\alpha_i(\lambda) \geq \bar{C} \forall i \in I_-$ it is sufficient to set

$$\lambda \geq \max_{i \in I_-} \sum_{j \in I} (\bar{C} - \alpha_j(0)) y_j K_{ij}. \quad \square$$

Appendix B. Asymptotic convergence analysis

In this section we provide the convergence analysis leading to the result of [Proposition 2](#) stated in [Section 4](#).

We start with a technical Lemma concerning the convergence of “shifted” sub-sequences of convergent sub-sequences of any sequence $\{\boldsymbol{\alpha}^k\}$, $k \in \mathbb{N}$, made up of feasible solutions for problem (22).

Lemma 4. *If a sub-sequence $\{\boldsymbol{\alpha}^k\}$, $k \in \mathcal{K}$ converges to $\bar{\boldsymbol{\alpha}}$, then for any given positive integer s , the sub-sequence $\{\boldsymbol{\alpha}^{k+s}\}$, $k \in \mathcal{K}$, converges to $\bar{\boldsymbol{\alpha}}$ as well.*

⁵ The intercept can be absorbed into the kernel by augmenting each input with a constant feature (or equivalently adjusting the kernel), so the explicit intercept term may be omitted without loss of generality.

Proof. First, we verify that the inequality

$$f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^{k+1}) \geq \frac{2}{C} \|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\|^2 \quad (\text{B.1})$$

holds for every iteration index $k \geq 0$. This inequality guarantees the sufficient decrease of the objective function with respect to the magnitude of the step.

Assuming that t^* denotes the optimal solution of the one-dimensional subproblem (30), we can express the second-order Taylor expansion of $\phi(t)$ around t^* as:

$$\phi(t) = \phi(t^*) + \frac{1}{2} \phi''(\bar{t})(t - t^*)^2, \quad (\text{B.2})$$

with \bar{t} such that $t \leq \bar{t} \leq t^*$. By considering the second-order derivative of $\phi(t)$:

$$\begin{aligned} \phi''(t) &= \text{Ker}(\mathbf{x}_{i^*}, \mathbf{x}_{j^*}) + \text{Ker}(\mathbf{x}_{j^*}, \mathbf{x}_{i^*}) - 2\text{Ker}(\mathbf{x}_{i^*}, \mathbf{x}_{j^*}) \\ &+ \frac{1}{C} \left[G''\left(\frac{\tilde{\alpha}_{i^*}(t)}{C}\right) + G''\left(\frac{\tilde{\alpha}_{j^*}(t)}{C}\right) \right] \end{aligned}$$

and the fact that $G''(\cdot) \geq 4$ and the sum of the first three terms (involving $\text{Ker}(\cdot, \cdot)$) is greater or equal to zero, we have $\phi''(t) \geq \frac{8}{C}$. From (B.2), for $t = 0$ we have:

$$f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^{k+1}) = \phi(0) - \phi(t^*) \geq \frac{4}{C} (t^*)^2 = \frac{2}{C} \|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\|^2,$$

which amounts to the sufficient decrease inequality.

To prove the lemma, we use the previous inequality (B.1) and the fact that the sequence $\{f(\boldsymbol{\alpha}^k)\}$ is not only decreasing but bounded, since the objective function (22a) is continuous over a compact set.

Considering any sub-sequence $\{\boldsymbol{\alpha}^k\}$, $k \in \mathcal{K}$, which converges to $\bar{\boldsymbol{\alpha}}$, for the sub-sequence $\{\boldsymbol{\alpha}^{k+1}\}$, $k \in \mathcal{K}$, we have

$$\begin{aligned} \lim_{k \in \mathcal{K}, k \rightarrow \infty} \|\boldsymbol{\alpha}^{k+1} - \bar{\boldsymbol{\alpha}}\| &\leq \lim_{k \in \mathcal{K}, k \rightarrow \infty} (\|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\| + \|\boldsymbol{\alpha}^k - \bar{\boldsymbol{\alpha}}\|) \\ &\leq \lim_{k \in \mathcal{K}, k \rightarrow \infty} \left(\sqrt{\frac{2}{C} (f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^{k+1}))} + \|\boldsymbol{\alpha}^k - \bar{\boldsymbol{\alpha}}\| \right) = 0. \end{aligned}$$

Thus, the shifted sub-sequence $\{\boldsymbol{\alpha}^{k+1}\}$, $k \in \mathcal{K}$, converges to $\bar{\boldsymbol{\alpha}}$. Applying the same argument to the sub-sequence $\{\boldsymbol{\alpha}^{k+1}\}$, $k \in \mathcal{K}$, we can also prove that $\lim_{k \in \mathcal{K}, k \rightarrow \infty} \boldsymbol{\alpha}^{k+2} = \bar{\boldsymbol{\alpha}}$. By induction, we have that $\lim_{k \in \mathcal{K}, k \rightarrow \infty} \boldsymbol{\alpha}^{k+s} = \bar{\boldsymbol{\alpha}}$ for any given s . \square

We now verify that the WSS is a special case of WSS with constant factor violating pair (WSS-CF) discussed in [Chen et al. \(2006\)](#).⁶ In the WSS-CF procedure reported below, we consider only pairs $i \in I_{\text{up}}(\boldsymbol{\alpha})$ and $j \in I_{\text{low}}(\boldsymbol{\alpha})$ such that the corresponding violation of the optimality condition (27) is at least as large as the image of a contraction function h^7 applied to the violation of the maximal violating pair, namely,

$$-y_i{}_{MVP} \nabla f(\boldsymbol{\alpha})_{iMVP} - (-y_j{}_{MVP} \nabla f(\boldsymbol{\alpha})_{jMVP}).$$

WSS-CF:

- (1) Consider a fixed $0 < \theta \leq 1$ for all iterations.
- (2) Select any pair $i^* \in I_{\text{up}}(\boldsymbol{\alpha})$ and $j^* \in I_{\text{low}}(\boldsymbol{\alpha})$ such that

$$-y_{i^*} \nabla f(\boldsymbol{\alpha})_{i^*} - (-y_{j^*} \nabla f(\boldsymbol{\alpha})_{j^*}) \geq \theta (-y_i{}_{MVP} \nabla f(\boldsymbol{\alpha})_{iMVP} - (-y_j{}_{MVP} \nabla f(\boldsymbol{\alpha})_{jMVP})) > 0 \quad (\text{B.3})$$

- (3) Return the pair $\{i^*, j^*\}$

It is easy to verify that, if the $N \times N$ kernel matrix K is positive semidefinite, WSS is a special case of WSS-CF. Indeed, consider the

⁶ Notice that in [Chen et al. \(2006\)](#) WSS-CF is referred to as WSS2.

⁷ As mentioned in [Chen et al. \(2006\)](#), $h: \mathbb{R} \rightarrow \mathbb{R}$ can be any function that is strictly increasing for $x \geq 0$, satisfying the conditions $h(x) \leq x$ for all $x \geq 0$, and $h(0) = 0$.

pair $\{i^*, j^*\}$ selected by WSS where $i^* = i^{MVP}$. At each iteration k , by definition of WSS, the current feasible solution α^k satisfies

$$\begin{aligned} & -\frac{(-y_{i^*} \nabla f(\alpha^k)_{i^*} - (-y_{j^*} \nabla f(\alpha^k)_{j^*}))^2}{q_{i^*j^*}} \\ & \leq -\frac{(-y_{i^{MVP}} \nabla f(\alpha^k)_{i^{MVP}} - (-y_{j^{MVP}} \nabla f(\alpha^k)_{j^{MVP}}))^2}{q_{i^{MVP}j^{MVP}}} \end{aligned}$$

with $q_{i^*j^*}$ and $q_{i^{MVP}j^{MVP}}$ positive. Since the feasible set of (22) is compact, for any pair $\{i, j\}$ the function $q_{ij} = Ker(\mathbf{x}_i, \mathbf{x}_i) + Ker(\mathbf{x}_j, \mathbf{x}_j) - 2Ker(\mathbf{x}_i, \mathbf{x}_j) + \frac{C}{\alpha_i(C-\alpha_i)} + \frac{C}{\alpha_j(C-\alpha_j)}$ is continuous in α and admits a maximum and a minimum over the feasible region of (22). Taking the minimum and the maximum of q_{ij} over all the pairs of indices $\{i, j\}$ and all the feasible α , we have the following inequality:

$$\begin{aligned} & (-y_{i^*} \nabla f(\alpha^k)_{i^*} - (-y_{j^*} \nabla f(\alpha^k)_{j^*})) \\ & \geq \sqrt{\frac{\min_{ij} \min_{\alpha} q_{ij}}{\max_{ij} \max_{\alpha} q_{ij}}} (-y_{i^{MVP}} \nabla f(\alpha^k)_{i^{MVP}} - (-y_{j^{MVP}} \nabla f(\alpha^k)_{j^{MVP}})). \end{aligned}$$

Thus, WSS is a special case of WSS-CF with $\theta = \sqrt{\frac{\min_{ij} \min_{\alpha} q_{ij}}{\max_{ij} \max_{\alpha} q_{ij}}}$.

Now we are ready to prove Proposition 2 after recalling it.

Proposition 2. Assume that the kernel matrix K is positive semidefinite. Let $\{\alpha^k\}$ be the infinite sequence generated by the SMO-type method Algorithm SMO Algorithm. Then $\{\alpha^k\}$ globally converges to the unique optimal solution of problem (22).

Proof. First, since K is positive semidefinite it is easy to verify that for any given pair $\{i, j\}$ selected in the WSS procedure, the corresponding one-dimensional subproblem (30) is strictly convex. Indeed,

$$\begin{aligned} \phi''(t) &= Ker(\mathbf{x}_i, \mathbf{x}_i) + Ker(\mathbf{x}_j, \mathbf{x}_j) - 2Ker(\mathbf{x}_i, \mathbf{x}_j) \\ &+ \frac{1}{C} \left[G''\left(\frac{\tilde{\alpha}_i(t)}{C}\right) + G''\left(\frac{\tilde{\alpha}_j(t)}{C}\right) \right] > 0. \end{aligned}$$

Since the feasible set of (22) is compact, $\{\alpha^k\}$ admits a convergent sub-sequence. Suppose that $\bar{\alpha}$ is the limit point of a convergent sub-sequence of $\{\alpha^k\}$. If $\bar{\alpha}$ is not a stationary point of (22) (the optimality conditions (27) are not satisfied) there exists a pair $\{\bar{i}, \bar{j}\}$ with

$$\bar{i} = \arg \max_{s \in I_{up}(\bar{\alpha})} -y_s \nabla f(\bar{\alpha})_s \text{ and } \bar{j} = \arg \min_{s \in I_{low}(\bar{\alpha})} -y_s \nabla f(\bar{\alpha})_s, \quad (B.4)$$

whose maximal violation is denoted as

$$\Delta = -y_{\bar{i}} \nabla f(\bar{\alpha})_{\bar{i}} + y_{\bar{j}} \nabla f(\bar{\alpha})_{\bar{j}} > 0. \quad (B.5)$$

We further define the smallest (non zero) absolute value of the difference between any possible pair of terms $-y_s \nabla f(\bar{\alpha})_s$ as:

$$\Delta' = \min_{i,s \in I} \left\{ \left| -y_i \nabla f(\bar{\alpha})_i + y_s \nabla f(\bar{\alpha})_s \right| \mid -y_i \nabla f(\bar{\alpha})_i \neq -y_s \nabla f(\bar{\alpha})_s \right\} > 0. \quad (B.6)$$

Given Lemma 4, the continuity of $\nabla f(\alpha)$, and the fact that $\theta \frac{\Delta'}{2} > 0$ due to (B.6), we can verify the claims (B.7)–(B.12). For any given r , there exists $\bar{k} \in \mathcal{K}$ such that for all $k \in \mathcal{K}$, with $k \geq \bar{k}$, the following hold:

for the pair of indices $\{\bar{i}, \bar{j}\}$ as in (B.4) and for $u = 0, \dots, r$, we have

$$-y_{\bar{i}} \nabla f(\alpha^{k+u})_{\bar{i}} + y_{\bar{j}} \nabla f(\alpha^{k+u})_{\bar{j}} > \Delta', \quad (B.7)$$

$$\text{if } i \in I_{up}(\bar{\alpha}), \text{ then } i \in I_{up}(\alpha^k), \dots, i \in I_{up}(\alpha^{k+r}), \quad (B.8)$$

$$\text{if } i \in I_{low}(\bar{\alpha}), \text{ then } i \in I_{low}(\alpha^k), \dots, i \in I_{low}(\alpha^{k+r}), \quad (B.9)$$

$$\begin{aligned} & \text{if for a given pair } \{i, j\} \text{ we have } -y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j > 0, \text{ then for} \\ & u = 0, \dots, r, \text{ we have } -y_i \nabla f(\alpha^{k+u})_i + y_j \nabla f(\alpha^{k+u})_j > \Delta', \quad (B.10) \end{aligned}$$

$$\text{if for a given pair } \{i, j\} \text{ we have } -y_i \nabla f(\bar{\alpha})_i = -y_j \nabla f(\bar{\alpha})_j, \text{ then for}$$

$$u = 0, \dots, r, \text{ we have } \left| -y_i \nabla f(\alpha^{k+u})_i + y_j \nabla f(\alpha^{k+u})_j \right| < \theta \Delta', \quad (B.11)$$

if $\{i, j\}$ is the working set at the $(k+u)$ -th with $0 \leq u \leq r-1$ and

$$-y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j > 0 \text{ then } i \notin I_{up}(\alpha^{k+u+1}) \text{ or } j \notin I_{low}(\alpha^{k+u+1}). \quad (B.12)$$

Let us verify the claim (B.7) which states that, for any iteration $k \geq \bar{k}$ the violation of the pair $\{\bar{i}, \bar{j}\}$ for α^{k+u} is larger than Δ' , for $u = 0, \dots, r$. Lemma 4 implies that the sequences $\{\alpha^k\}, \{\alpha^{k+1}\}, \dots, \{\alpha^{k+r}\}, k \in \mathcal{K}$, all converge to $\bar{\alpha}$. From the continuity of $\nabla f(\alpha)$ and relation (B.6), for any given u with $0 \leq u \leq r$ and the corresponding sequence $\{\alpha^{k+u}\}, k \in \mathcal{K}$, there exists k_u such that (B.7) holds for all $k \geq k_u, k \in \mathcal{K}$. Since r is finite, by setting \bar{k} to be the largest of these k_u with $u = 0, \dots, r$, (B.7) is valid for all $u = 0, \dots, r$. The derivations of (B.8)–(B.11) follow from similar arguments so they are omitted.

We now verify the claim (B.12) which states that, for any iteration $k \geq \bar{k}$, if at iteration $k+u$ with $0 \leq u \leq r-1$ the working set pair $\{i, j\}$ has a positive violation associated to $\bar{\alpha}$, then at iteration $k+u+1$ either i does not belong to $I_{up}(\alpha^{k+u+1})$ or j does not belong to $I_{low}(\alpha^{k+u+1})$. First, we define $\alpha(t)$ as $\alpha_s(t) = \alpha_s^{k+u}$ with $s \neq i, j$ and $\alpha_i(t) = \alpha_i^{k+u}(t^*)$ and $\alpha_j(t) = \alpha_j^{k+u}(t^*)$, where $\alpha_i^{k+u}(t^*)$ and $\alpha_j^{k+u}(t^*)$ are the optimal values of, respectively, α_i and α_j generated by solving the one-dimensional problem (30) at $k+u$ iteration. Similar to the optimality condition (27) for problem (22), since $\alpha(t)$ is stationary for the subproblem then

$$\max_{s \in I_{up}(\alpha(t)): s \in \{i, j\}} -y_s \nabla f(\alpha(t))_s \leq \min_{s \in I_{low}(\alpha(t)): s \in \{i, j\}} -y_s \nabla f(\alpha(t))_s. \quad (B.13)$$

By noticing that $\alpha(t) = \alpha^{k+u+1}$, if

$$i \in I_{up}(\alpha^{k+u+1}) \text{ and } j \in I_{low}(\alpha^{k+u+1}),$$

then from (B.13) we have

$$-y_i \nabla f(\alpha^{k+u+1})_i + y_j \nabla f(\alpha^{k+u+1})_j \leq 0. \quad (B.14)$$

However, assuming that $-y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j > 0$, claim (B.10) for α^{k+u+1} implies that $-y_i \nabla f(\alpha^{k+u+1})_i + y_j \nabla f(\alpha^{k+u+1})_j > \Delta'$, so there is a contradiction.

For ease of proof presentation, the indices of $\bar{\alpha}$ are reordered as follows:

$$-y_1 \nabla f(\bar{\alpha})_1 \leq \dots \leq -y_N \nabla f(\bar{\alpha})_N. \quad (B.15)$$

We also introduce

$$S_1(k) = \sum \{i \mid i \in I_{up}(\alpha^k)\} \text{ and } S_2(k) = \sum \{N - i \mid i \in I_{low}(\alpha^k)\} \quad (B.16)$$

Clearly

$$N \leq S_1(k) + S_2(k) \leq N(N-1). \quad (B.17)$$

If $\{i, j\}$ is selected at the $(k+u)$ -th iteration (with $u = 0, \dots, r$), then we must have that

$$-y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j > 0. \quad (B.18)$$

Indeed, using an adaptation of claim (B.10) where we assume that $-y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j < 0$, we obtain $-y_i \nabla f(\alpha^{k+u})_i + y_j \nabla f(\alpha^{k+u})_j < 0$ which violates WSS condition (B.3). Moreover, if $-y_i \nabla f(\bar{\alpha})_i + y_j \nabla f(\bar{\alpha})_j = 0$ then

$$\begin{aligned} & -y_i \nabla f(\alpha^{k+u})_i + y_j \nabla f(\alpha^{k+u})_j < \theta(\Delta') < \theta(-y_{\bar{i}} \nabla f(\alpha^{k+u})_{\bar{i}} + y_{\bar{j}} \nabla f(\alpha^{k+u})_{\bar{j}}) \\ & \leq \theta \left(\max_{s \in I_{up}(\alpha^{k+u})} -y_s \nabla f(\alpha^{k+u})_s - \min_{s \in I_{low}(\alpha^{k+u})} -y_s \nabla f(\alpha^{k+u})_s \right). \quad (B.19) \end{aligned}$$

Starting from the left, the first strict inequality is due to (B.11), and the second strict inequality is due to (B.7). The third inequality is implied by claims (B.8) and (B.9) where $\bar{i} \in I_{up}(\bar{\alpha})$, and respectively $\bar{j} \in I_{low}(\bar{\alpha})$. As a consequence, (B.18) is valid since (B.19) contradicts (B.3).⁸

To obtain the contradiction of (B.4) and (B.5) we exploit a counting procedure. Considering two consecutive iterations k th and $k+1$ -th, the (B.18) established above and (B.12) imply

$$i \notin I_{up}(\alpha^{k+1}) \text{ or } j \notin I_{low}(\alpha^{k+1}).$$

⁸ Notice that in the last inequality in (B.19) we do not use the notation i^{MVP} and j^{MVP} because we want to explicitly refer to the $k+u$ -th iterations.

Table C.5

Results in terms of testing accuracy and sparsity obtained by λ -heuristic ((C, λ) set to $(C, \frac{C}{10})$) and by S-KLR. The standard deviation of the average accuracy is reported in parentheses.

Dataset	λ -choice sparsest of 3 most accurate		λ -choice		S-KLR sparsest of 3 most accurate		S-KLR	
	Acc	% data points	Acc	% data points	Acc	% data points	Acc	% data points
	banknote	1	0.059	1	0.059	1	0.05	1
coil2000	0.94	0.059	0.94	0.059	0.94	0.059	0.94	0.059
diabetes	0.768	0.523	0.7695	0.565	0.771	0.567	0.771	0.567
ionosphere	0.943	0.488	0.949	0.976	0.952	0.496	0.952	0.496
magic	0.861	0.357	0.861	0.357	0.861	0.357	0.865	0.99
monk2	0.958	0.292	0.965	1	0.97	0.319	0.972	0.357
ring	0.978	0.081	0.978	0.081	0.978	0.08	0.978	0.08
sonar	0.889	0.925	0.889	0.925	0.894	0.92	0.894	0.92
spambase	0.932	0.241	0.932	0.241	0.932	0.241	0.939	0.918
twonorm	0.974	0.071	0.974	0.185	0.977	0.08	0.978	0.082
waveform	0.902	0.32	0.912	0.99	0.911	0.291	0.911	0.291
wisconsin	0.975	0.144	0.975	0.144	0.977	0.187	0.977	0.187
AVERAGE	0.927 (0.064)	0.297	0.929 (0.064)	0.4651	0.9303 (0.064)	0.304	0.9315 (0.064)	0.4164

Let us suppose that $i \notin I_{\text{up}}(\alpha^{k+1})$. Given (B.12) and (B.18), from (B.8) we would have $i \notin I_{\text{up}}(\bar{\alpha})$ and hence $i \in I_{\text{low}}(\bar{\alpha})$. Due to (B.9) and the selection rule (B.3), we have $i \in I_{\text{low}}(\alpha^k) \cap I_{\text{up}}(\alpha^k)$. Thus

$$i \in I_{\text{low}}(\alpha^k) \cap I_{\text{up}}(\alpha^k) \text{ and } i \notin I_{\text{up}}(\alpha^{k+1}).$$

Since $j \in I_{\text{low}}(\alpha^k)$, we would have

$$S_1(k+1) \leq S_1(k) - i + j \leq S_1(k) - 1 \text{ and } S_2(k+1) \leq S_2(k), \quad (\text{B.20})$$

where $-i + j \leq -1$ comes from (B.15).

Now suppose that $j \notin I_{\text{low}}(\alpha^{k+1})$. With similar arguments we would obtain that

$$j \in I_{\text{low}}(\alpha^k) \cap I_{\text{up}}(\alpha^k) \text{ and } j \notin I_{\text{low}}(\alpha^{k+1}).$$

Since $i \in I_{\text{up}}(\alpha^k)$, we would have

$$S_1(k+1) \leq S_1(k) \text{ and } S_2(k+1) \leq S_2(k) - (N - j) + (N - i) \leq S_2(k) - 1. \quad (\text{B.21})$$

The same reasoning (based on (B.18) and (B.12)) can be applied for any two consecutive iterations, from iteration k to iteration $k+r$. By setting $r = N(N-1)$, (B.20) and (B.21) imply that $S_1(k) + S_2(k)$ is reduced to zero, which contradicts (B.17). Therefore, (B.4) and (B.5) cannot hold are wrong and any limit point of the sequence satisfies the optimality conditions (27).

Since problem (22) is strictly convex ($K \geq 0$ and $C \sum_{i \in I} G(\frac{\alpha_i}{C}) - \lambda \sum_{i \in I} \alpha_i$ is strictly convex), and its feasible set is compact, (22) admits a unique global optimal solution α^* .

From the previous arguments, all limit points of $\{\alpha^k\}$ satisfy the optimality conditions, thus $\{\alpha^k\}$ globally converges to α^* . \square

Appendix C. On hyperparameters tuning

The values of hyperparameters C and λ can significantly impact the performance of the S-KLR method. Conducting a grid search to determine good estimates of these hyperparameters can be time-consuming and computationally expensive. This appendix investigates the trend of testing accuracy and sparsity as functions of the two hyperparameters, considering the same training-testing split of the datasets reported in Section 5.2. Since we are interested in the overall impact of hyperparameter changes rather than in model comparison, the validation set used during tuning is not considered.

Fig. C.7 highlights the impact on the performance of these hyperparameters, by showing the accuracy and sparsity trends obtained when varying C and λ according to the grid search described in Section 5.2 for all considered datasets. In particular, the y-axis represents, respectively, the accuracy on the left-hand-side plots, and the sparsity level

on the right-hand-side plots. The colors of the profiles represent the C values (see the colorbar for values details), and the x-axis represent 10 equally values in the interval $[0, C]$ for λ .

We observed two main types of behaviors:

- (i) For a fixed value of C , the variation of λ only slightly affect the accuracy (see plots on the left).
- (ii) For small values of C ,⁹ the value of λ does not significantly affect the model sparsity. However, for larger values of C , a small value of λ is sufficient to sensibly increase the model sparsity. This behavior is consistent, with minor variations, across all 12 datasets.

The observations (i) and (ii) suggest that the λ value can be heuristically set to a relatively small value, without the need of a refined and expensive grid search.

In the sequel, we set $\lambda = \frac{C}{10}$ and we refer to it as λ -choice. table reports the average results (over the $k = 5$ folds) of the sparsest model among the 3 most accurate ones.

In Table C.5, we show the average results obtained with λ -choice and with S-KLR employing the standard grid search on both λ and C . As in Table 3, for both strategies we report the results corresponding to the best accuracy as well as the ones corresponding to the sparsest model among the 3 most accurate ones. Concerning the best accuracy criterion, λ -choice yields comparable accuracy scores and a slight loss of average sparsity (46.51% against 41.64%). As to the sparsest among the 3 most accurate criterion, λ -choice and S-KLR achieve comparable average performance for both accuracy and sparsity.

To summarize, we observe that the simple λ -choice yields S-KLR models whose sparsity is not significantly worse than the one provided by S-KLR with a refined grid search for both C and λ values. Whereas the accuracy of both approaches is comparable. This shows that also in the case of KLR it is just sufficient to search for the value of the single hyperparameter (C) as it is the case of SVM, with no need for time consuming two dimensional grid search. Considering, as an example, the grid search applied to the experiments of Section 5.2, λ -choice would have implied a computational time reduction by approximately a factor of 8 (the number of considered grid λ values minus one).

⁹ In some sparsity plots the profiles related to small C values (e.g., $C = 0.0001$ or $C = 0.001$) are overlapped by the ones associated to medium C values (e.g., $C = 1$ or $C = 10$).

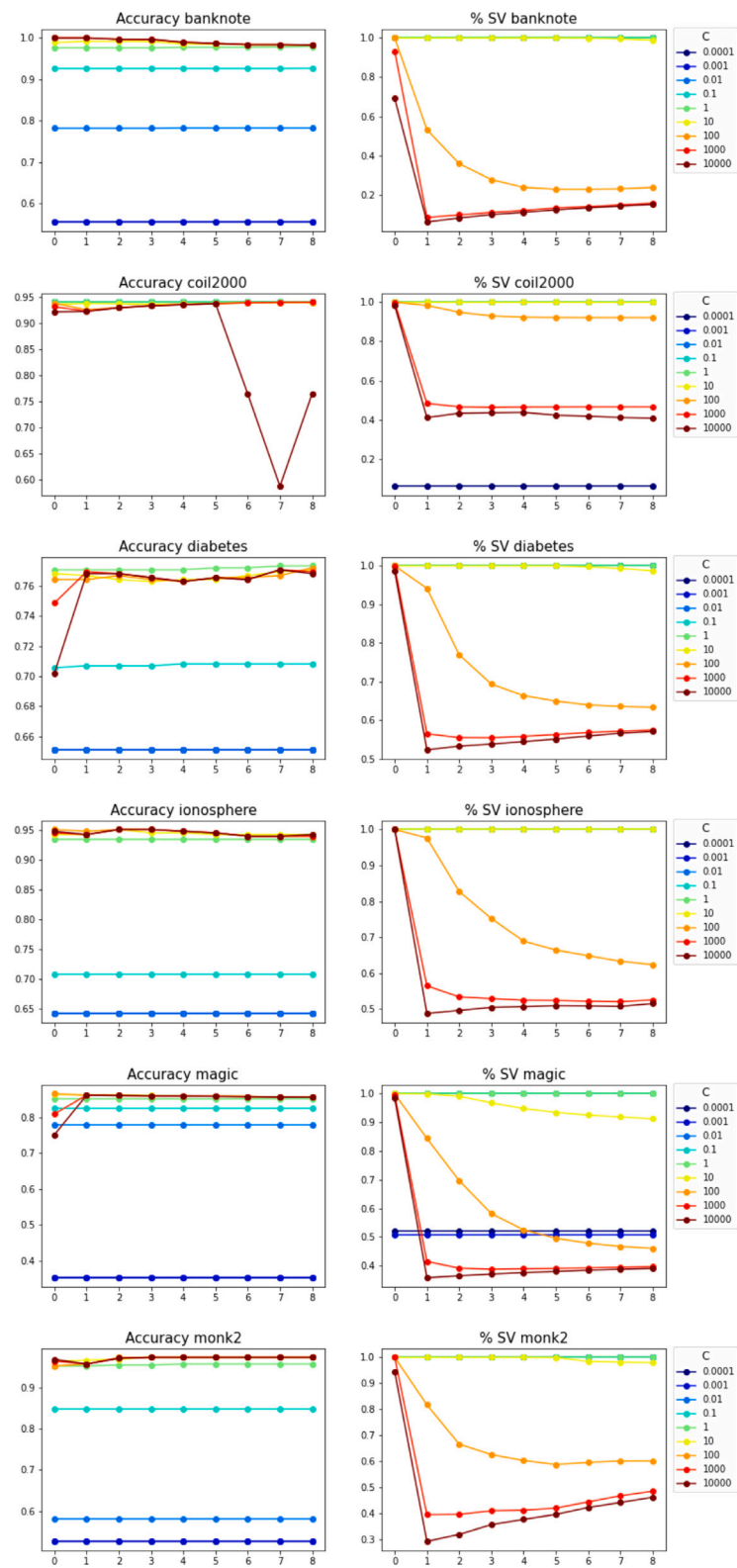


Fig. C.7. Testing accuracy and sparsity in terms of the regularization term C and in the x -axis the sparsity term λ for S-KLR (colored dotted line). The sparsity term λ varies from 0 to C in 9 steps (i.e., 10 different values of λ).

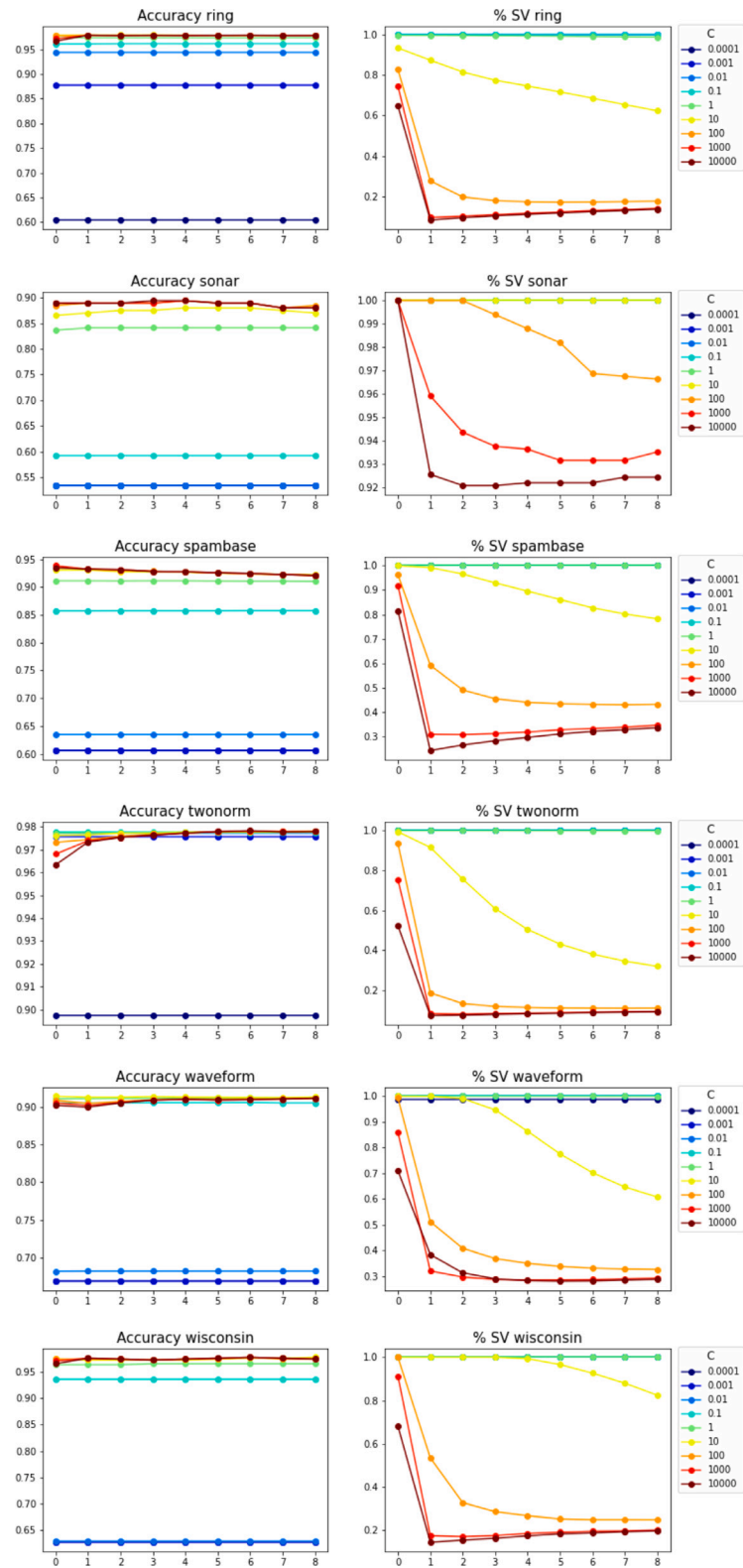


Fig. C.7. (continued).

Data availability

All data used in this work are publicly available.

References

- Amaldi, E., Consolo, A., Manno, A., 2023. On multivariate randomized classification trees: l_0 -based sparsity, VC dimension and decomposition methods. *Comput. Oper. Res.* 151, 106058.
- Asuncion, A., Newman, D., 2007. UCI Machine Learning Repository.
- Bagley, S.C., White, H., Golomb, B.A., 2001. Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain. *J. Clinical Epidemiology* 54 (10), 979–985.
- Benítez-Peña, S., Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., 2024. Cost-sensitive probabilistic predictions for support vector machines. *European J. Oper. Res.* 314 (1), 268–279.
- Bickel, P.J., Ritov, Y., Tsybakov, A.B., 2009. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.* 37 (4), 1705–1732.
- Böken, B., 2021. On the appropriateness of Platt scaling in classifier calibration. *Inf. Syst.* 95, 101641.
- Burges, C.J.C., 1996. Simplified Support Vector Decision Rules. In: *Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*. Morgan Kaufmann, pp. 71–77.
- Burges, C.J., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* 2 (2), 121–167.
- Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol. (TIST)* 2 (3), 1–27.
- Chen, P.-H., Fan, R.-E., Lin, C.-J., 2006. A Study on SMO-type Decomposition Methods for Support Vector Machines. *IEEE Trans. Neural Netw.* 17 (4), 893–908.
- Chen, W., Xie, X., Peng, J., Wang, J., Duan, Z., Hong, H., 2017. GIS-based landslide susceptibility modelling: a comparative assessment of kernel logistic regression, Naïve-Bayes tree, and alternating decision tree models. *Geomatics, Nat. Hazards Risk* 8 (2), 950–973.
- Consolo, A., Amaldi, E., Manno, A., 2026. Soft regression trees: a model variant and a decomposition training algorithm. *European J. Oper. Res.* 328 (2), 607–619.
- Cortes, C., Vapnik, V., 1995. Support-Vector Networks. *Mach. Learn.* 20, 273–297.
- De Caigny, A., Coussement, K., De Bock, K.W., 2018. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European J. Oper. Res.* 269 (2), 760–772.
- van de Geer, S.A., 2008. High-Dimensional Generalized Linear Models and the Lasso. *Ann. Statist.* 36 (2), 614–645.
- Edgar, T.W., Manz, D.O., 2017. Chapter 4 - exploratory study. In: Edgar, T.W., Manz, D.O. (Eds.), *Research Methods for Cyber Security*. Syngress, pp. 95–130.
- Fan, R.-E., Chen, P.-H., Lin, C.-J., Joachims, T., 2005. Working Set Selection Using Second Order Information for Training Support Vector Machines. *J. Mach. Learn. Res.* 6 (12).
- Franc, V., Zien, A., Schölkopf, B., 2011. Support Vector Machines as Probabilistic Models. In: *Proceedings of the 28th International Conference on Machine Learning*, vol. 11, pp. 665–672.
- Grippo, L., Manno, A., Sciandrone, M., 2015. Decomposition techniques for multilayer perceptron training. *IEEE Trans. Neural Networks Learn. Syst.* 27 (11), 2146–2159.
- Hong, H., Pradhan, B., Xu, C., Bui, D.T., 2015. Spatial prediction of landslide hazard at the Yihuang area (China) using two-class kernel logistic regression, alternating decision tree and support vector machines. *Catena* 133, 266–281.
- Jaakkola, T.S., Haussler, D., 1999. Probabilistic Kernel Regression Models. In: *Seventh International Workshop on Artificial Intelligence and Statistics*. PMLR.
- Katz, M., Schaffner, M., Andelic, E., Kruger, S.E., Wendemuth, A., 2006. Sparse Kernel Logistic Regression using Incremental Feature Selection for Text-Independent Speaker Identification. In: *2006 IEEE Odyssey-the Speaker and Language Recognition Workshop*. IEEE, pp. 1–6.
- Keerthi, S.S., Duan, K., Shevade, S.K., Poo, A.N., 2005. A Fast Dual Algorithm for Kernel Logistic Regression. *Mach. Learn.* 61 (1), 151–165.
- Kelly, C.J., Karthikesalingam, A., Suleyman, M., Corrado, G., King, D., 2019. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* 17, 1–9.
- Koppel, A., Warnell, G., Stump, E., Ribeiro, A., 2019. Parsimonious Online Learning with Kernels via Sparse Projections in Function Space. *J. Mach. Learn. Res.* 20 (3), 1–44.
- Krishnapuram, B., Carin, L., Figueiredo, M.A., Hartemink, A.J., 2005. Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds. *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (6), 957–968.
- Li, J., Liao, S., 2022. Worst-case regret analysis of computationally budgeted online kernel selection. *Mach. Learn.* 111 (3), 937–976.
- Lucidi, S., Palagi, L., Risi, A., Sciandrone, M., 2009. A convergent hybrid decomposition algorithm model for SVM training. *IEEE Trans. Neural Netw.* 20 (6), 1055–1060.
- Maalouf, M., Trafalis, T.B., 2011. Robust weighted kernel logistic regression in imbalanced and rare events data. *Comput. Statist. Data Anal.* 55 (1), 168–183.
- Manno, A., Palagi, L., Sagratella, S., 2018. Parallel decomposition methods for linearly constrained problems subject to simple bound with application to the SVMs training. *Comput. Optim. Appl.* 71 (1), 115–145.
- Manno, A., Sagratella, S., Livi, L., 2016. A convergent and fully distributable SVMs training algorithm. In: *2016 International Joint Conference on Neural Networks*. IJCNN, IEEE, pp. 3076–3080.
- Martín-Baos, J.Á., García-Ródenas, R., López-García, M.L., Rodríguez-Benitez, L., 2020. Discrete choice modeling using Kernel Logistic Regression. *Transp. Res. Procedia* 47, 457–464.
- Martín-Baos, J.Á., García-Ródenas, R., Rodríguez-Benitez, L., 2021. Revisiting kernel logistic regression under the random utility models perspective. *An interpretable machine-learning approach*. *Pattern Lett.* 13 (3), 151–162.
- Martín-Baos, J.Á., García-Ródenas, R., Rodríguez-Benitez, L., Bierlaire, M., 2025. Scalable kernel logistic regression with Nyström approximation: Theoretical analysis and application to discrete choice modelling. *Neurocomputing* 617, 128975.
- Mercer, J., 1909. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Phil. Trans. R. Soc. A* 209, 415–446.
- Mok, M.S., Sohn, S.Y., Ju, Y.H., 2010. Random effects logistic regression model for anomaly detection. *Expert Syst. Appl.* 37 (10), 7162–7166.
- Negahban, S., Ravikumar, P., Wainwright, M.J., Yu, B., 2012. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statist. Sci.* 27 (4), 538–557.
- Nicora, G., Rios, M., Abu-Hanna, A., Bellazzi, R., 2022. Evaluating pointwise reliability of machine learning prediction. *J. Biomed. Informatics* 127, 103996.
- Nikolic, N., Zarkic-Joksimovic, N., Stojanovski, D., Joksimovic, I., 2013. The application of brute force logistic regression to corporate credit scoring models: Evidence from Serbian financial statements. *Expert Syst. Appl.* 40 (15), 5932–5944.
- Nocedal, J., Wright, S.J., 1999. *Numerical Optimization*. Springer.
- Platt, J., 1998. Sequential Minimal Optimization: A fast algorithm for training Support Vector Machines. *Tech. Rep. (MSR-TR-98-14)*, Microsoft.
- Platt, J., Cristianini, N., Shawe-Taylor, J., 1999a. Large Margin DAGs for Multiclass Classification. *Adv. Neural Inf. Process. Syst.* 12.
- Platt, J., et al., 1999b. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.* 10 (3), 61–74.
- Roscher, R., Förstner, W., Waske, B., 2012a. I2VM: Incremental import vector machines. *Image Vis. Comput.* 30 (4), 263–278.
- Roscher, R., Waske, B., Förstner, W., 2012b. Incremental Import Vector Machines for Classifying Hyperspectral Data. *IEEE Trans. Geosci. Remote Sens.* 50 (9), 3463–3473.
- Schober, P., Vetter, T.R., 2021. Logistic Regression in Medical Research. *Anesth. Analg.* 132 (2), 365.
- Schölkopf, B., Herbrich, R., Smola, A.J., 2001. A Generalized Representer Theorem. In: *International Conference on Computational Learning Theory*. Springer, pp. 416–426.
- Vapnik, V.N., 2000. *The Nature of Statistical Learning Theory*, second ed. Springer, New York.
- Vapnik, V.N., Vapnik, V., et al., 1998. *Statistical learning theory*.
- Wahba, G., et al., 1999. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. *Adv. Kernel Methods-Support Vector Learn.* 6, 69–87.
- Wang, Z., Crammer, K., Vucetic, S., 2012. Breaking the Curse of Kernelization: Budgeted Stochastic Gradient descent for Large-Scale SVM Training. *J. Mach. Learn. Res.* 13 (1), 3103–3131.
- Wang, J., Shen, X., Liu, Y., 2008. Probability estimation for large-margin classifiers. *Biometrika* 95 (1), 149–167.
- Wiens, J., Saria, S., Sendak, M., Ghassemi, M., Liu, V.X., Doshi-Velez, F., Jung, K., Heller, K., Kale, D., Saeed, M., et al., 2019. Do no harm: a roadmap for responsible machine learning for health care. *Nature Med.* 25 (9), 1337–1340.
- Xu, Z., Chang, X., Xu, F., Zhang, H., 2012. $L_{1/2}$ regularization: A Thresholding Representation Theory and a Fast Solver. *IEEE Trans. Neural Networks Learn. Syst.* 23 (7), 1013–1027.
- Xu, C., Peng, Z., Jing, W., 2013. Sparse kernel logistic regression based on $L_{1/2}$ regularization. *Sci. China Inf. Sci.* 56, 1–16.
- Zeng, L., Zhang, H.H., 2023. Sparse Learning and Class Probability estimation with Weighted Support Vector Machines. *arXiv preprint arXiv:2312.10618*.
- Zhang, L., Jin, R., Chen, C., Bu, J., He, X., 2012. Efficient Online Learning for Large-Scale Sparse Kernel Logistic Regression. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, (1), pp. 1219–1225.
- Zhu, J., Hastie, T., 2005. Kernel Logistic Regression and the Import Vector Machine. *J. Comput. Graph. Statist.* 14 (1), 185–205.