ECONSTOR
Make Your Publications Visible.

A Service of

ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Burgard, Jan Pablo; Pinheiro, Maria Eduarda; Schmidt, Martin

Article — Published Version

Mixed-integer quadratic optimization and iterative clustering techniques for semi-supervised support vector machines

TOP

Provided in Cooperation with:
Springer Nature

Suggested Citation: Burgard, Jan Pablo; Pinheiro, Maria Eduarda; Schmidt, Martin (2024) : Mixed-integer quadratic optimization and iterative clustering techniques for semi-supervised support vector machines, TOP, ISSN 1863-8279, Springer, Berlin, Heidelberg, Vol. 32, Iss. 3, pp. 391-428, https://doi.org/10.1007/s11750-024-00668-w

This Version is available at:
https://hdl.handle.net/10419/315697

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

http://creativecommons.org/licenses/by/4.0/

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

TOP (2024) 32:391–428
https://doi.org/10.1007/s11750-024-00668-w

ORIGINAL PAPER

Mixed-integer quadratic optimization and iterative clustering techniques for semi-supervised support vector machines

Jan Pablo Burgard[1] · Maria Eduarda Pinheiro[2] · Martin Schmidt[2]

Received: 22 March 2023 / Accepted: 7 February 2024 / Published online: 16 May 2024
© The Author(s) 2024

Abstract

Among the most famous algorithms for solving classification problems are support vector machines (SVMs), which find a separating hyperplane for a set of labeled data points. In some applications, however, labels are only available for a subset of points. Furthermore, this subset can be non-representative, e.g., due to self-selection in a survey. Semi-supervised SVMs tackle the setting of labeled and unlabeled data and can often improve the reliability of the results. Moreover, additional information about the size of the classes can be available from undisclosed sources. We propose a mixed-integer quadratic optimization (MIQP) model that covers the setting of labeled and unlabeled data points as well as the overall number of points in each class. Since the MIQP's solution time rapidly grows as the number of variables increases, we introduce an iterative clustering approach to reduce the model's size. Moreover, we present an update rule for the required big-M values, prove the correctness of the iterative clustering method as well as derive tailored dimension-reduction and warm-starting techniques. Our numerical results show that our approach leads to a similar accuracy and precision than the MIQP formulation but at much lower computational cost. Thus, we can solve larger problems. With respect to the original SVM formulation, we observe that our approach has even better accuracy and precision for biased samples.

Keywords Semi-supervised learning · Support vector machines · Clustering · Mixed-integer quadratic optimization

✉ Martin Schmidt
martin.schmidt@uni-trier.de

Jan Pablo Burgard
burgardj@uni-trier.de

Maria Eduarda Pinheiro
pinheiro@uni-trier.de

1 Department of Economic and Social Statistics, Trier University, Universitätsring 15, 54296 Trier, Germany

2 Department of Mathematics, Trier University, Universitätsring 15, 54296 Trier, Germany

Mathematics Subject Classification 90C11 · 90C90 · 90-08 · 68T99

1 Introduction

Support vector machines (SVMs) are a standard approach for supervised binary classification (Boser et al. 1992; Cortes and Vapnik 1995). The core idea is to find a separating hyperplane that optimally splits the feature space in a positive and a negative side according to the positive and negative labels of the data.

Obtaining labels for all units of interest can be costly. This is especially the case if one has to do a classic survey to obtain the labels. In this case, it would be favorable to train the SVM on only partly labeled data. This yields a semi-supervised learning setting. Bennett and Demiriz (1998) formulate and solve the semi-supervised SVM (S^3VM) as a mixed-integer linear problem (MILP). Many strategies for solving S^3VM have been proposed in the following decades such as the transductive approach (TSVM) by Joachims (2002) and Yu et al. (2012) or manifold regularization (LapSVM) by Belkin et al. (2006) and Melacci and Belkin (2009). Some researchers also consider a balancing constraint as done in meanS3VM by Kontonatsios et al. (2017) and in c^3SVM by Chapelle et al. (2006). Moreover, the balancing constraint proposed by Chapelle and Zien (2005) enforces that the proportion of unlabeled and labeled data on both sides is similar to the proportion given by the labeled data.

In many cases, however, the aggregated information about the number of positive and negative cases in a population is known from an external source. For example, in population surveys, there are population figures from official statistics agencies. This setting is studied, e.g., by Burgard et al. (2021), who develop a cardinality-constrained multinomial logit model and apply it in the context of micro-simulations. As another example, in some businesses, the total amount of positive labels could be known but not which customer has a positive or a negative label. An intuitive example is a supermarket for which the amount of cash payments is known. However, this information is not ex-post attributable to the individual customers. We propose to add this aggregated additional information to the optimization model by imposing a cardinality constraint on the predicted labels for the unlabeled data. As will be shown in our numerical experiments, this improves the accuracy of the classification of the unlabeled data. Furthermore, the inclusion of such a cardinality constraint is very useful in the case in which the labeled data is not a representative sample from the population. When obtaining the labels from process data or from online surveys, the inclusion process of the labeled data is generally not known. This is subsumed under the non-probability sample. In this case, inverse inclusion probability weighting, as typically done in survey sampling, is not applicable. By not controlling the inclusion process, strong over- or under-coverage of relevant information in the data set is possible and should be taken into account in the analysis. Not accounting for possible biases in the data generally leads to biased results.

We propose a big-M-based MIQP to solve the semi-supervised SVM problem with a cardinality constraint for the unlabeled data. Here, we restrict ourselves to the linear kernel. Other kernels such as Gaussian and polynomial ones can, in principle, be used as well. However, this would lead to additional nonlinear constraints in a our mixed-integer model and would thus significantly increase the computational challenge of solving the problem. Although we strongly suspect that the problem is NP-hard, we have no proof for it since we focus here on solution techniques and not on a formal complexity analysis of the problem. The cardinality constraint helps to account for biased samples since the number of positive predictions on the population is bounded by the constraint. The computation time for this MIQP grows rapidly with the number of variables—especially for an increasing number of integer variables. We develop an algorithm that uses a clustering-based model reduction to reduce the computation time. Similar reduction approaches can be found for the classic SVM using, e.g., fuzzy clustering (Almasi and Rouhani 2016; Cervantes et al. 2006), clustering-based convex hulls (Birzhandi and Youn 2019), and k-means clustering (de Almeida et al. 2000; Yao et al. 2013). We prove the correctness of our iterative clustering method and further show that it computes feasible points for the original problem. Hence, it also delivers proper upper bounds. Within our iterative approach, we additionally derive a scheme for updating the required big-M values and present tailored dimension-reduction as well as warm-starting techniques.

The paper is organized as follows. In Sect. 2, we describe our optimization problem and the big-M-based MIQP formulation. Afterward, the clustering-based model reduction technique is presented in Sect. 3. There, we also present our algorithm that combines the model reduction and the MIQP formulation. In Sect. 4, we discuss some algorithmic improvements such as the handling of data points that are far away from the hyperplane and the choice of M in the big-M formulation. In Sect. 5, we present how to use the solution of our algorithm to obtain the solution of the initial MIQP formulation by fixing some points on the correct side of the hyperplane. Finally, in Sect. 6, numerical results are reported and discussed and we conclude in Sect. 7.

2 An MIQP formulation for a cardinality-constrained semi-supervised SVM

Let $X \in \mathbb{R}^{d \times N}$ be the data matrix with $X_l = [x^1, \dots, x^n]$ being the labeled data and $X_u = [x^{n+1}, \dots, x^N]$ being the unlabeled data. Hence, we have $x^i \in \mathbb{R}^d$ for all $i \in [1, N] := \{1, \dots, N\}$. We set $m := N - n$ and $y \in \{-1, 1\}^n$ is the vector of class labels for the labeled data. When the data is linearly separable, the SVM provides a hyperplane (ω, b) that separates the positively and negatively labeled data. In the case that the data is not linearly separable, the standard approach is to use the ℓ_2-SVM by Cortes and Vapnik (1995) given by

$$\min_{\omega,b,\xi} \quad \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i \tag{P1a}$$

$$\text{s.t.} \quad y_i(\omega^\top x^i - b) \geq 1 - \xi_i, \quad i \in [1, n], \tag{P1b}$$

$$\xi_i \geq 0, \quad i \in [1, n]. \tag{P1c}$$

Here and in what follows, $\|\cdot\|$ denotes the Euclidean norm. However, other norms such as the 1- or the max-norm could be used as well. For being able to include unlabeled data in the optimization process, Bennett and Demiriz (1998) propose the semi-supervised SVM (S^3VM). In many applications, the aggregated information on the labels is available, e.g., from census data. In the following, we know the total number τ of positive labels for the unlabeled data from an external source. We adapt the idea of the S^3VM such that we can use τ as an additional information in the optimization model. Our goal is to find optimal parameters $\omega^* \in \mathbb{R}^d$, $b^* \in \mathbb{R}$, $\xi^* \in \mathbb{R}^n$, and $\eta^* \in \mathbb{R}^2$ that solve the optimization problem

$$\min_{\omega,b,\xi,\eta} \quad \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2) \tag{P2a}$$

$$\text{s.t.} \quad y_i(\omega^\top x^i - b) \geq 1 - \xi_i, \quad i \in [1, n], \tag{P2b}$$

$$\tau - \eta_1 \leq \sum_{i=n+1}^{N} h_{\omega,b}(x^i) \leq \tau + \eta_2, \tag{P2c}$$

$$\xi_i \geq 0, \quad i \in [1, n], \tag{P2d}$$

$$\eta_1, \eta_2 \geq 0, \tag{P2e}$$

with

$$h_{\omega,b}(x) = \begin{cases} 1, & \text{if } \omega^\top x + b \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

Note that the objective function in (P2a) is a compromise between maximizing the distance between the two classes as well as minimizing the classification error for the label and the unlabeled data. The penalty parameters $C_1 > 0$ and $C_2 > 0$ aim to control the importance of the slack variables ξ and η, respectively. Constraint (P2b) enforces on which side of the hyperplane the labeled data x^i should lie. Constraint (P2c) ensures that we have τ unlabeled data on the positive side. If $\eta_1^* > 0$ holds for a solution $(\omega^*, b^*, \xi^*, \eta^*)$, then less than τ unlabeled points are classified as positive. On the other hand, if $\eta_2^* > 0$ holds, more than τ unlabeled points are classified as positive. If $\eta_1^* = \eta_2^* = 0$ holds, exactly τ unlabeled points are classified in the positive class. Note that, having assigned a very high value to C_1 or C_2, the objective function value is dominated by these slack variables.

The function $h_{\omega,b}(\cdot)$ in Constraint (P2c) is not continuous, which means that Problem (P2) cannot be easily solved by standard solvers. A typical way to overcome this problem is to add binary variables to turn on or off the enforcement of a constraint. By introducing binary variables $z_i \in \{0, 1\}$, $i \in [n + 1, N]$, we can reformulate the optimization Problem (P2) using the following big-M formulation:

$$\min_{\omega,b,\xi,\eta,z} \quad \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2) \tag{P3a}$$

$$\text{s.t.} \quad y_i(\omega^\top x^i + b) \geq 1 - \xi_i, \quad i \in [1, n], \tag{P3b}$$

$$\omega^\top x^i + b \leq z_i M, \quad i \in [n+1, N], \tag{P3c}$$

$$\omega^\top x^i + b \geq -(1 - z_i)M, \quad i \in [n+1, N], \tag{P3d}$$

$$\tau - \eta_1 \leq \sum_{i=n+1}^{N} z_i \leq \tau + \eta_2, \tag{P3e}$$

$$\xi_i \geq 0, \quad i \in [1, n], \tag{P3f}$$

$$\eta_1, \eta_2 \geq 0, \tag{P3g}$$

$$z_i \in \{0, 1\}, \quad i \in [n+1, N], \tag{P3h}$$

where M needs to be chosen sufficiently large. As z_i is binary, Constraints (P3c) and (P3d) lead to

$$\omega^\top x^i + b > 0 \implies z_i = 1, \quad i \in [n+1, N],$$
$$\omega^\top x^i + b < 0 \implies z_i = 0, \quad i \in [n+1, N].$$

If x^i lies on the hyperplane, i.e., $\omega^\top x^i + b = 0$, Constraints (P3c) and (P3d) hold for $z_i = 1$ and $z_i = 0$. In this case, it can be counted either on the positive or on the negative side. For this reason, Problem (P3) is not formally equivalent to Problem (P2). Reformulation (P3) is a mixed-integer quadratic problem (MIQP) in which all constraints are linear but the objective function is quadratic. We refer to this problem as CS^{3}VM.

Since we now stated our first model, let us shed some light on the results depending on whether the standard SVM or CS^{3}VM is used. Figure 1 shows a 2-dimensional example data set and the corresponding hyperplanes for SVM and CS^{3}VM. In this case, $\tau = 11$, i.e., 11 unlabeled points belong the positive class. Note that SVM only classifies 6 unlabeled points as positive, while CS^{3}VM classifies 11 as such. The point that lies on the CS^{3}VM hyperplane is classified as positive because the binary variable regarding this point is 1. This example shows that using τ as additional information can improve the classification of unlabeled points.

In the big-M formulation, the choice of M is crucial. If M is too small, the problem can become infeasible or optimal solutions could be cut off. If M is chosen too large, the respective continuous relaxations usually lead to bad lower bounds and solvers may encounter numerical troubles. The choice of M is discussed in the following lemma and theorem. In Lemma 1 we show how M is related to the objective function and the given data. This is then used in Theorem 2 to derive a provably correct big-M.

Lemma 1 *Given a feasible point for Problem* (P3) *with an objective function value* f, *an optimal solution* $(\omega^*, b^*, \xi^*, \eta^*, z^*)$ *of* (P3) *satisfies*

$$\|\omega^*\| \leq \sqrt{2f} \quad \text{and} \quad |b^*| \leq \|\omega^*\| \max_{i \in [1,N]} \|x^i\| + 1$$

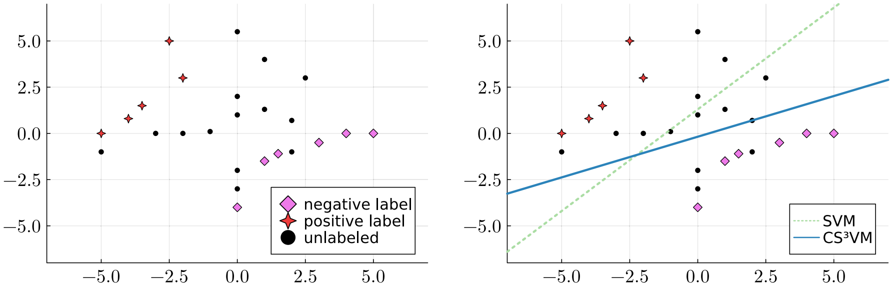

Fig. 1 A 2-dimensional example (left) and the hyperplanes resulting from the SVM and the CS^{3}VM (right)

and, consequently, every optimal solution satisfies (P3c) *and* (P3d) *for*

$$M = 2\sqrt{2f} \max_{i \in [1,N]} \|x^i\| + 1.$$

Proof Due to optimality, we get

$$\frac{\|\omega^*\|^2}{2} \leq \frac{\|\omega^*\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i^* + C_2(\eta_1^* + \eta_2^*) \leq f \implies \|\omega^*\| \leq \sqrt{2f}.$$

The second inequality is shown by contradiction. To this end, we w.l.o.g. assume that $\tilde{b} = \|\omega^*\| \max_{i \in [1,N]} \|x^i\| + 1 + \delta$ is part of an optimal solution for some $\delta > 0$. Using the inequality of Cauchy–Schwarz then yields

$$\begin{aligned}(\omega^*)^\top x^i + \tilde{b} &= (\omega^*)^\top x^i + \|\omega^*\| \max_{j \in [1,N]} \|x^j\| + 1 + \delta \\ &\geq -\|\omega^*\| \|x^i\| + \|\omega^*\| \max_{j \in [1,N]} \|x^j\| + 1 + \delta \\ &> 1\end{aligned}$$

for all $i \in [1, N]$. Hence, for all $i \in [1, n]$ with $y_i = 1$, we get $\tilde{\xi}_i = 0$ from Constraint (P3b) and the objective function. Moreover, for $i \in [1, n]$ with $y_i = -1$, the same reasoning implies

$$-(\omega^*)^\top x^i - \tilde{b} = 1 - \tilde{\xi}_i \implies \tilde{\xi}_i = 2 + (\omega^*)^\top x^i + \|\omega^*\| \max_{j \in [1,N]} \|x^j\| + \delta.$$

Besides that, for the unlabeled data $i \in [n+1, N]$, since $(\omega^*)^\top x^i + \tilde{b} > 1$, we get $\tilde{z}_i = 1$, which leads to

$$\sum_{i=n+1}^{N} \tilde{z}_i = m \implies \tilde{\eta}_1 = 0, \ \tilde{\eta}_2 = m - \tau.$$

This means that the objective function value for the point $(\omega^*, \tilde{b}, \tilde{\xi}, \tilde{\eta}, \tilde{z})$ is given by

$$\tilde{f} := \frac{\|\omega^*\|^2}{2} + C_1 \sum_{i:y_i=-1} \left(2 + (\omega^*)^\top x^i + \|\omega^*\| \max_{j\in[1,N]} \|x^j\| + \delta\right) + C_2(m-\tau).$$

However, if we set $\bar{b} := \|\omega^*\| \max_{i\in[1,N]} \|x^i\| + 1$, we get

$$(\omega^*)^\top x^i + \bar{b} \geq 1, \quad i \in [1, N],$$

i.e., $z_i = 1$ for all $i \in [n+1, N]$, $\bar{\eta}_1 = 0$, $\bar{\eta}_2 = m - \tau$, and $\bar{\xi}_i = 0$ for i with $y_i = 1$. Moreover, for $i \in [1, n]$ with $y_i = -1$, from Constraint (P3b) we obtain

$$-(\omega^*)^\top x^i - \bar{b} = 1 - \bar{\xi}_i \implies \bar{\xi}_i = 2 + (\omega^*)^\top x^i + \|\omega^*\| \max_{i\in[1,N]} \|x^i\|.$$

All this implies that the objective function value $\bar{f}$ for the point $(\omega^*, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ satisfies

$$\bar{f} := \frac{\|\omega^*\|^2}{2} + C_1 \sum_{i:\, y_i=-1} (2 + (\omega^*)^\top x^i + \|\omega^*\| \max_{j\in[1,N]} \|x^j\|) + C_2(m-\tau) < \tilde{f},$$

which contradicts the assumption that $\tilde{f}$ is optimal. Hence,

$$|b^*| \leq \|\omega^*\| \max_{i\in[1,N]} \|x^i\| + 1$$

holds, which proves the second inequality. Note further that

$$(\omega^*)^\top x^i + b^* \leq \|\omega^*\| \|x^i\| + |b^*| \leq 2\sqrt{2f} \max_{j\in[1,N]} \|x^j\| + 1 = M$$

and

$$(\omega^*)^\top x^i + b^* \geq -\|\omega^*\| \|x^i\| - |b^*| \geq -2\sqrt{2f} \max_{j\in[1,N]} \|x^j\| - 1 = -M$$

holds for all $i \in [n+1, N]$. □

We now use the result from the last technical lemma to obtain a provably correct big-M.

Theorem 2 *A valid big-M for Problem* (P3) *is given by*

$$M = 2\sqrt{2(2C_1\bar{n} + C_2(m-\tau))} \max_{i\in[1,N]} \|x^i\| + 1 \tag{1}$$

with $\bar{n} := |\{i \in [1, n] \colon y_i = -1\}|$.

Proof Consider the feasible point of (P3) given by $\omega = 0 \in \mathbb{R}^d$ and $b = 1$. Since $\omega^\top x^i + b = 1$ holds for all $i \in [1, N]$, Constraint (P3b) implies

$$\xi_i = \begin{cases} 2, & \text{if } y_i = -1, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, using Constraints (P3c)–(P3e) leads to

$$z_i = 1, \ i \in [n+1, N], \quad \eta_1 = 0, \quad \eta_2 = m - \tau,$$

which implies that the objective function for the point $(\omega, b, \xi, \eta, z)$ is given by

$$f = 0 + 2C_1\bar{n} + C_2(m - \tau).$$

Finally, from Lemma 1, we get

$$M = 2\sqrt{2(2C_1\bar{n} + C_2(m - \tau))} \max_{i \in [1,N]} \|x^i\| + 1.$$

□

3 A re-clustering method for solving CS^{3}VM

In Model (P3) of the last section, each binary variable is related to an unlabeled point. The larger the number of unlabeled data, the larger the number of binary variables and, hence, the larger the computational burden to solve Problem (P3). To reduce this computational burden, we propose to cluster the unlabeled data. This way, only one binary variable per cluster is needed. For every cluster, we use its centroid as its representative point. To obtain clusterings, we use minimum sum-of-squares clustering (MSSC). The MSSC problem is NP-hard; see, e.g., Aloise et al. (2009), Mahajan et al. (2012), and Dasgupta (2007). However, we do not need a globally optimal solution for the MSSC problem as will be shown below. Given a number k of clusters and a matrix $S = [s^1, \dots, s^p] \in \mathbb{R}^{d \times p}$ of given points, the goal of the MSSC is to find mean vectors $c^j \in \mathbb{R}^d$, $j \in [1, k]$, that solve the problem

$$c^* = \arg\min_c \ell(S, c), \quad c = (c^j)_{j=1,\dots,k},$$

where the loss function ℓ is the sum of the squared Euclidean distances, i.e.,

$$\ell(S, c) = \sum_{j=1}^{k} \sum_{s^i \in \mathcal{C}_j} \|s^i - c^j\|^2$$

with $\mathcal{C}_j \subset \mathbb{R}^d$ being the set of data points that are assigned to cluster j.

We solve this problem heuristically using the k-means algorithm (MacQueen 1967; Lloyd 1982) for $S = X_u$, i.e., we cluster the unlabeled data. Then, instead of using all unlabeled data as in the last section, we only use the clusters' centroids $c^1, \dots, c^k$ and the numbers $e_1, \dots, e_k$ of data points in each cluster to obtain the problem

$$\begin{aligned}
\min_{\omega,b,\xi,\eta,z} \quad & \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2) && \text{(P4a)}\\
\text{s.t.} \quad & y_i(\omega^\top x^i + b) \geq 1 - \xi_i, \quad i \in [1, n], && \text{(P4b)}\\
& \omega^\top c^j + b \leq z_j M, \quad j \in [1, k], && \text{(P4c)}\\
& \omega^\top c^j + b \geq -(1 - z_j)M, \quad j \in [1, k], && \text{(P4d)}\\
& \tau - \eta_1 \leq \sum_{j=1}^{k} e_j z_j \leq \tau + \eta_2, && \text{(P4e)}\\
& \xi_i \geq 0, \quad i \in [1, n], && \text{(P4f)}\\
& \eta_1, \eta_2 \geq 0, && \text{(P4g)}\\
& z_j \in \{0, 1\}, \quad j \in [1, k]. && \text{(P4h)}
\end{aligned}$$

A valid big-M is still given by (1) as shown in the next proposition.

Proposition 1 *If $e_j \geq 1$ for all $j \in [1, k]$, a valid big-M for Problem (P4) is given by (1).*

Proof The proof follows the same lines as the proofs of Lemma 1 and Theorem 2 with the additional observation that for all $j \in [1, k]$, it holds

$$\|c^j\| = \frac{1}{e_j} \left\| \sum_{i:x^i \in \mathcal{C}_j} x^i \right\| \leq \frac{e_j \max_{i \in [n+1,N]} \|x^i\|}{e_j} = \max_{i \in [n+1,N]} \|x^i\|. \qquad \square$$

It can happen that the hyperplane given by (ω^*, b^*) that results from the solution of Problem (P4) cuts through some cluster. This means that not all data points of the cluster actually lie on the same side of the hyperplane. If this happens, the solution of Problem (P4) does not satisfy the cardinality constraint (P3e) of Problem (P3). To fix this, we propose an iterative method that is formally listed in Algorithm 1. Note that the use the k-means algorithm is helpful here as it automatically provides the convex hulls of the clusters. Hence, it is easy to check if the hyperplane cuts through some cluster or not.

If Algorithm 1 terminates it holds that all points in a cluster are on the same side of the final hyperplane. This implies the cardinality constraint (P3e) is satisfied. Note that the k-means algorithm is only called once to initialize the clustering. For all other iterations, we manually split clusters if they are cut by the hyperplane of the respective iteration and compute the new centroids directly.

The next theorem establishes that Algorithm 1 always terminates after finitely many iterations.

Algorithm 1: Re-Clustering Method (RCM)

Input: $X \in \mathbb{R}^{d\times N}$, $y \in \{-1, 1\}^n$, $k^1 \in \mathbb{N}$, $C_1 > 0$, $C_2 > 0$, and $\tau \in \mathbb{N}$.

1 Set $t \leftarrow 1$, compute M^t as in (1), compute a clustering of X_u in k^1 many clusters using the k-means algorithm, and obtain the centroids $c^1, \dots, c^{k^1}$ as well as the numbers $e_1, \dots, e_{k^1}$ of data points in each cluster.
2 Solve Problem (P4) to compute the hyperplane (ω^t, b^t) as well as ξ^t, η^t, z^t.
3 **if** *the hyperplane* (ω^t, b^t) *cuts a cluster* **then**
4   Set $k^{t+1} \leftarrow k^t$.
5   **for** *each cluster that is cut by the hyperplane* (ω^t, b^t) **do**
6   Split the cluster into two new clusters so that neither of the two new clusters is cut by the hyperplane (ω^t, b^t).
7   Update the centroids of the newly created clusters.
8   Set $k^{t+1} \leftarrow k^{t+1} + 1$.
9   **end**
10   Update $t \leftarrow t + 1$ and go to Step 2.
11 **else**
12   Return the hyperplane (ω^t, b^t) as well as ξ^t, η^t, z^t.
13 **end**

Theorem 3 *Suppose that $e_j \geq 1$ for all $j \in [1, k^1]$ after Step 1 of Algorithm 1. Then, Algorithm 1 terminates after at most $m - k^1$ iterations, where m is the number of the unlabeled data points and k^1 is the number of initial clusters.*

Proof Observe that since we cluster m unlabeled points, the maximum number of clusters we can obtain is m. Besides that, if in an iteration t, Algorithm 1 does not terminate, at least one cluster is split Step 6. Because we start with k^1 clusters and since in each iteration, we increase the number of clusters at least by one, the maximum number of iterations is $m - k^1$. □

Note that the point obtained by Algorithm 1 is not necessarily a minimizer of Problem (P3). However, the objective function value of the point obtained by Algorithm 1 is an upper bound for the objective function value of Problem (P3).

Theorem 4 *Let $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ be the point returned by Algorithm 1. Then, $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ is feasible for Problem (P3) with*

$$M = 2\sqrt{2\bar{f}} \max_{i \in [1,N]} \|x^i\| + 1$$

and, consequently,

$$\bar{f} := \frac{\|\bar{\omega}\|^2}{2} + C_1 \sum_{i=1}^{n} \bar{\xi}_i + C_2(\bar{\eta}_1 + \bar{\eta}_2)$$

is an upper bound of Problem (P3).

Proof For all clusters $\mathcal{C}_j$, $j \in \{1, \dots, k^t\}$, where t is the final iteration of Algorithm 1, we set $\tilde{z}_i = \bar{z}_j$ for all i with $x^i \in \mathcal{C}_j$. We now show that $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \tilde{z})$ is a feasible point for Problem (P3). Indeed, Constraints (P3b), (P3f), (P3g), and (P3h) are clearly fulfilled. Furthermore, since

$$\sum_{i \in \mathcal{C}_j} \tilde{z}_i = e_j \bar{z}_j$$

for all $j \in [1, k^t]$, using (P4e) we get

$$\sum_{i=n+1}^{N} \tilde{z}_i = \sum_{j=1}^{k^t} e_j \bar{z}_j \implies \tau - \bar{\eta}_1 \leq \sum_{i=n+1}^{N} \tilde{z}_i \leq \tau + \bar{\eta}_2$$

and Constraint (P3e) is satisfied. Besides that,

$$\frac{\|\bar{\omega}\|^2}{2} \leq \bar{f} \implies \|\bar{\omega}\| \leq \sqrt{2\bar{f}} \tag{2}$$

holds and as in Lemma 1, we get

$$|\bar{b}| \leq \|\bar{\omega}\| \max_{i \in [1,N]} \|x^i\| + 1. \tag{3}$$

Moreover, by construction, for all $i \in \{n+1, \dots N\}$ with $\tilde{z}_i = 1$, x^i belongs to a cluster $\mathcal{C}_j$ such that $\bar{\omega}^\top c^j + \bar{b} \geq 0$. Using the fact that all points in $\mathcal{C}_j$ are on the same side of the hyperplane, this side must be the positive one. This fact together with (2) and (3) implies

$$\begin{aligned} -(1 - \tilde{z}_i)M = 0 \leq \bar{\omega}^\top x^i + \bar{b} &\leq \|\bar{\omega}\| \max_{i \in [1,N]} \|x^i\| + |\bar{b}| \\ &\leq 2\sqrt{2\bar{f}} \max_{i \in [1,N]} \|x^i\| + 1 = M = \tilde{z}_i M. \end{aligned}$$

Similarly, for all $i \in \{n+1, \dots N\}$ with $\tilde{z}_i = 0$, we get

$$-M = -(1 - \tilde{z}_i)M \leq \bar{\omega}^\top x^i + \bar{b} \leq 0 = \tilde{z}_i M$$

and (P3c) as well as (P3d) are fulfilled. Because $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \tilde{z})$ is a feasible point for Problem (P3), $\bar{f}$ is an upper bound to the Problem (P3). □

Note, finally, that since the point obtained from Algorithm 1 is feasible for Problem (P3), we can use it for warm starting.

4 Further algorithmic enhancements

In order to reduce computational costs, we propose two additional enhancements. The first one (see Sect. 4.1) makes use of the fact that the SVM is mostly influenced by data points that are close to the separating hyperplane. The second one (see Sect. 4.2) introduces a rule for updating M in each iteration of Algorithm 1.

4.1 Handling points far from the hyperplane

In Algorithm 1, the number of clusters increases in each iteration. Hence, the time to solve Problem (P4) increases from iteration to iteration in general. Like in the original SVM, the points closest to the hyperplane influence the resulting hyperplane more than the other points. Obviously, eliminating points that do not strongly influence the hyperplane decreases the size of the problem. Some approaches to eliminate these points have also been proposed for the original SVM. For a survey, see, e.g., Birzhandi et al. (2002). However, most of these approaches are heuristics and do not necessarily yield a feasible point of the problem.

The idea for our setting is the following. Clusters that are far away from the hyperplane could be omitted as this will not change the solution. The farther a cluster is from the hyperplane in an iteration, the less likely it is that the cluster will be split or change sides completely in a future iteration. Hence, the clusters farthest from the current hyperplane mainly add information about their side and capacity. However, in a later iteration, the cluster may become relevant again. Thus, we need to find a way to discard detailed information on certain clusters but also a way to reactivate the discarded clusters if necessary.

We propose the following procedure to reduce the amount of clusters that have to be considered in the current iteration of the algorithm. If the number of clusters exceeds a fixed value k^+, we first fix the cluster with the centroid farthest from the hyperplane as a kind of residual cluster on a side if this side has points far from the hyperplane. Second, we discard all clusters in which all points are farther from the hyperplane than some Δ^t and assign them to the residual cluster on their side of the hyperplane. This way the cardinality constraint remains valid. Moreover, all formerly discarded clusters are checked for re-consideration. If a discarded cluster has a point with a distance to the hyperplane less than Δ^t or if any point in the cluster changed the side, the cluster is reactivated.

Let $\bar{S} = (s_{\alpha(1)}, \dots, s_{\alpha(d)})^\top$ be the vector of increasingly sorted values of $S = \{s_1, \dots, s_d\}$ and let $a \in (0, 1)$. The a-quantile of S, as proposed by Hyndman and Fan (1996), is given by

$$P_S(a) := s_{\alpha(q)} + \frac{s_{\alpha(q)} - s_{\alpha(r)}}{q - r} \left((d-1)a - q + 1\right)$$

with

$$q := \max_{i \in [1,d]} \left\{ i \colon \frac{i-1}{d-1} \leq a \right\}, \quad r := \min_{i \in [1,d]} \left\{ i \colon \frac{i-1}{d-1} \geq a \right\}.$$

Given a parameter $\hat{\Delta}^t \in (0, 1)$, we choose Δ^t in each iteration t according to

$$\Delta^t = P_{D^t}(\hat{\Delta}^t) \quad \text{with} \quad D^t_j = \left| (\omega^t)^\top c_j + b^t \right| \quad \text{for all } j \in \left[1, k^t\right]. \tag{4}$$

Note that if in an iteration t, a point in some discarded cluster changed the side, the vector z as part of the current solution does not fit to this change. This happens when, e.g., $(\omega^{t-1})^\top x^i + b^{t-1} > 0$ and $(\omega^t)^\top x^i + b^t < 0$ but $z_j^t > 0$ with $\mathcal{C}_j$ being the cluster with centroid farthest from the hyperplane on the positive side. To avoid that this happens too often, $\hat{\Delta}^{t+1}$ is increased by a fixed value $\tilde{\Delta} \in (0, 1)$ when there is some point in some discarded cluster that has changed sides.

Motivated by the above discussions, we add new steps in the Algorithm 1 that can be seen in Algorithm 2. In Step 5, if the number of clusters exceeds k^+, clusters far from the hyperplane are discarded. In Steps 9 and 10, clusters discarded with a point that changed sides or that is closer to the hyperplane than Δ^t are reactivated. In Step 12, $\hat{\Delta}^t$ is updated.

Algorithm 2: Improved Re-Clustering Method (IRCM)

Input: $X \in \mathbb{R}^{d\times N}$, $y \in \{-1, 1\}^n$, $k^1 \in \mathbb{N}$, $C_1 > 0$, $C_2 > 0$, $\tau \in \mathbb{N}$, $\hat{\Delta}^1 \in (0, 1)$, $\tilde{\Delta} \in (0, 1)$, $\mathcal{G}^1 = \emptyset$, $k^+ \in \mathbb{N}$.

1 Set $t = 1$, compute M^t as in (1), cluster X_u in k^1 clusters using k-means, leading to centroids $c^1, \dots, c^{k^1}$ and the numbers $e_1, \dots, e_{k^1}$ of data points in each cluster.
2 Solve Problem (P4) to compute the hyperplane (ω^t, b^t) as well as ξ^t, η^t, z^t.
3 Compute Δ^t as in (4).
4 **if** $k^t > k^+$ **then**
5 | update $\mathcal{G}^{t+1} \leftarrow \mathcal{G}^t \cup \{\mathcal{C}_j : |(\omega^t)^\top x^\ell + b^t| > \Delta^t \; \forall x^\ell \in \mathcal{C}_j\}$.
6 **else**
7 | set $\mathcal{G}^{t+1} \leftarrow \mathcal{G}^t$.
8 **end**
9 Set $\mathcal{J}^t := \{\mathcal{C}_j \in \mathcal{G}^t : \exists x^\ell \in \mathcal{C}_j : \operatorname{sign}((\omega^t)^\top x^\ell + b^t) \neq \operatorname{sign}((\omega^{t+1})^\top x^\ell + b^{t+1})\}$.
10 Update $\mathcal{G}^{t+1} \leftarrow \mathcal{G}^{t+1} \setminus (\{\mathcal{C}_j \in \mathcal{G}^t : \exists x^\ell \in \mathcal{G}_j^t \text{ with } |(\omega^t)^\top x^\ell + b^t| \leq \Delta^t\} \cup \mathcal{J}^t)$.
11 **if** $\mathcal{J}^t \neq \emptyset$ **then**
12 | update $\hat{\Delta}^{t+1} \leftarrow \min\{\hat{\Delta}^t + \tilde{\Delta}, 1\}$.
13 **else**
14 | set $\hat{\Delta}^{t+1} \leftarrow \hat{\Delta}^t$
15 **end**
16 Compute M^{t+1} as in (8).
17 **if** $\mathcal{J}^t \neq \emptyset$ *or the hyperplane* (ω^t, b^t) *cuts a cluster* **then**
18 | Set $k^{t+1} \leftarrow k^t$.
19 | **for** *each cluster that is cut by the hyperplane* (ω^t, b^t) **do**
20 | | Split the cluster into two new clusters so that neither of the two new clusters is cut by the hyperplane (ω^t, b^t).
21 | | Update the centroids of the newly created clusters.
22 | | Set $k^{t+1} \leftarrow k^{t+1} + 1$.
23 | **end**
24 | Update $t \leftarrow t + 1$ and back to Step 2.
25 **else**
26 | Return the hyperplane (ω^t, b^t) as well as ξ^t, η^t, z^t.
27 **end**

4.2 Updating the Big-M

As discussed in Sect. 2, M needs to be sufficiently large. However, the bigger the M, the more likely we face numerical issues. As shown in Sect. 2, the smaller the objective function provided by a feasible point, the smaller the value of M can be chosen. Based on that, we update M in each iteration with the aim of decreasing it. We do this by adding Step 16 in Algorithm 2 and the next theorem justifies this.

Theorem 5 *Consider X, y, C_1, C_2, τ, as well as $c^1, \dots, c^{k^t}$ and $e_1, \dots e_{k^t}$ in an iteration t of Algorithm 1. Then, the optimal solution $(\bar{\omega}^t, \bar{b}^t, \bar{\xi}^t, \bar{\eta}^t, \bar{z}^t)$ of Problem (P4) provides an upper bound*

$$\tilde{f}_t := \frac{\|\bar{\omega}^t\|^2}{2} + C_1 \sum_{i=1}^{n} \bar{\xi}_i + C_2(\tilde{\eta}_1 + \tilde{\eta}_2), \tag{5}$$

with

$$\tilde{z}_j = \begin{cases} 1, & \text{if } \left(\bar{\omega}^t\right)^\top \tilde{c}_j + \bar{b}^t \geq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

and

$$\tilde{\eta}_1 = \max\left\{0, \tau - \sum_{j=1}^{s} e_j \tilde{z}_j\right\}, \quad \tilde{\eta}_2 = \max\left\{0, \sum_{j=1}^{s} e_j \tilde{z}_j - \tau\right\}, \tag{7}$$

for Problem (P4) with $c^1, \dots, c^{k^{t+1}}$ and $e_1, \dots e_{k^{t+1}}$ as updated in iteration t with

$$M = 2\sqrt{2\tilde{f}_t} \max_{i \in [1,N]} \|x^i\| + 1. \tag{8}$$

Proof Consider $\tilde{z}$ as given in (6) and $\tilde{\eta}_1, \tilde{\eta}_2$ as given in (7). We now show that $(\bar{\omega}^t, \bar{b}^t, \bar{\xi}^t, \tilde{z}, \tilde{\eta})$ is a feasible point for Problem (P4). Indeed, Constraints (P4b) and (P4e)–(P4h) are clearly satisfied. Moreover, $(\bar{\omega}^t, \bar{b}^t, \bar{\xi}^t, \bar{\eta}^t, \bar{z}^t)$ provides the objective function value given by (5) and

$$\|\bar{\omega}^t\| \leq \sqrt{2\tilde{f}_t}, \quad |\bar{b}^t| \leq \|\bar{\omega}^t\| \max_{i \in [1,N]} \|x^i\| + 1,$$

see the proof of Lemma 1. This together with $\|c^j\| \leq \max_{i \in [n+1,N]} \|x^i\|$ implies

$$\left(\bar{\omega}^t\right)^\top c^j + \bar{b}^t \leq \|\bar{\omega}^t\| \max_{i \in [n+1,N]} \|x^i\| + |\bar{b}^t| \leq 2\sqrt{2\tilde{f}_t} \max_{i \in [1,N]} \|x^i\| + 1 = M$$

and

$$\left(\bar{\omega}^t\right)^\top c^j + \bar{b}^t \geq -M.$$

Hence, Constraints (P4c) and (P4d) are satisfied. Since $(\bar{\omega}^t, \bar{b}^t, \bar{\xi}^t, \tilde{z}, \tilde{\eta})$ is a feasible point for Problem (P4), $\tilde{f}_t$ is an upper bound for Problem (P4). □

Using Theorem 5, we can update M in each iteration of Algorithm 2 as in (8). The following theorem establishes that as Algorithm 1, Algorithm 2 always terminates after finitely many iterations.

Theorem 6 *The Algorithm 2 terminates after at most*

$$2m - k^1 + \frac{(1-\hat{\Delta}^1)}{\tilde{\Delta}}$$

iterations, where m is the number of unlabeled data points, k^1 is the number of initial clusters, and $\hat{\Delta}^1$, $\tilde{\Delta}$ are inputs of Algorithm 2.

Proof In Algorithm 2, the number of iterations can only be greater as in Algorithm 1 if there is some iteration t for which $\mathcal{J}^t \neq \emptyset$ holds but the hyperplane does not cut any cluster. At each iteration in which this happens, $\hat{\Delta}^t$ is increased and, in the worst case, i.e.,

$$\hat{t} := m - k^1 + \frac{(1-\hat{\Delta}^1)}{\tilde{\Delta}},$$

we get $\hat{\Delta}^{\hat{t}} = 1$. This implies that for all further iterations t,

$$\Delta^t = \max_{j\in[1,k^t]} |(\omega^t)^\top c^j + b^t|$$

holds. Thus, no cluster is added to the set $\mathcal{G}^t$. Since $|\mathcal{G}^{\hat{t}}| \leq m$ and $\mathcal{J}^t \subset \mathcal{G}^{\hat{t}}$, Algorithm 2 can only have m more iterations with $\mathcal{J}^t \neq \emptyset$. This means that the maximum number of iterations is $2m - k^1 + (1-\hat{\Delta}^1)/\tilde{\Delta}$. □

Although Theorem 6 shows that, in the worst case, Algorithm 2 can take more iterations than Algorithm 1 to terminate, Algorithm 2 solves problems with less binary variable in every iteration, which means that the time per iteration will be lower compared to Algorithm 1.

Note that the objective function value obtained by Algorithm 2 is an upper bound for the objective function value of Problem (P3).

Theorem 7 *Let $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ be the point returned by Algorithm 2. Then, $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ is feasible for Problem (P3) with*

$$M = 2\sqrt{2\bar{f}} \max_{i\in[1,N]} \|x^i\| + 1$$

and, consequently,

$$\bar{f} := \frac{\|\bar{\omega}\|^2}{2} + C_1 \sum_{i=1}^{n} \bar{\xi}_i + C_2(\bar{\eta}_1 + \bar{\eta}_2)$$

is an upper bound of Problem (P3).

Proof Since Algorithm 2 terminates when no cluster changes the side and no cluster is cut by the hyperplane, the proof is the same as for Theorem 5. □

As before, we can use the point obtained from Algorithm 2 to warm start Problem (P3).

5 Using IRCM for warm-starting

As stated in Theorem 7, the solution found by Algorithm 2 is feasible for Problem (P3). Hence, we can use it for warm-starting the solution process of Problem (P3). The next lemma establishes that unlabeled points can be fixed to be in one side of the hyperplane.

Lemma 8 *Let $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ be a feasible point of Problem (P3) with objective function value $\bar{f}$. Furthermore, let $(\omega^*, b^*, \xi^*, \eta^*, z^*)$ be an optimal solution of Problem (P3) with objective function value f^*. Set*

$$P_u := \left\{i \in [n+1, N] \colon (\omega^*)^\top x^i + b^* > 0\right\},$$
$$N_u := \left\{i \in [n+1, N] \colon (\omega^*)^\top x^i + b^* < 0\right\},$$

and let $S_p \subseteq P_u$, $S_n \subseteq N_u$ be arbitrarily chosen subsets and let $x^s \notin S_n$ be an unlabeled point with $\bar{\omega}^\top x^s + \bar{b} < 0$. Then, the objective function value $\tilde{f}$ given by any feasible point of the problem

$$\min_{\omega,b,\xi,\eta,z} \quad \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2) \tag{P5a}$$
$$\text{s.t} \quad y_i(\omega^\top x^i + b) \geq 1 - \xi_i, \quad i \in [1, n], \tag{P5b}$$
$$\omega^\top x^i + b \leq z_i M, \quad i \in [n+1, N] \setminus (\{s\} \cup S_p \cup S_n), \tag{P5c}$$
$$\omega^\top x^i + b \geq -(1 - z_i)M, \quad i \in [n+1, N] \setminus (\{s\} \cup S_p \cup S_n), \tag{P5d}$$
$$\omega^\top x^i + b \geq 0, \quad i \in S_p, \tag{P5e}$$
$$\omega^\top x^i + b \leq 0, \quad i \in S_n, \tag{P5f}$$
$$0 \leq \omega^\top x^s + b \leq z_s M, \tag{P5g}$$
$$\tau - \eta_1 \leq |S_p| + \sum_{i \in [n+1,N] \setminus (S_p \cup S_n)} z_i \leq \tau + \eta_2, \tag{P5h}$$
$$\xi_i \geq 0, \quad i \in [1, n], \tag{P5i}$$
$$\eta_1, \eta_2 \geq 0, \tag{P5j}$$
$$z_i \in \{0, 1\}, \quad i \in [n+1, N] \setminus (S_p \cup S_n), \tag{P5k}$$

with M as defined in (8), satisfies the following properties:

(a) *$\tilde{f}$ is an upper bound for f^*,*

(b) *if $\tilde{f}$ is the optimal objective function value of Problem (P5) and $\bar{f} < \tilde{f}$ is satisfied, it holds $(\omega^*)^\top x^s + b^* < 0$, i.e., $x^s \in N_u$.*

Proof (a) The points that satisfy Constraints (P5b)–(P5k) are feasible for Problem (P3) and provide an objective function value $\tilde{f}$. Since f^* is the optimal objective function value of Problem (P3), $f^* \leq \tilde{f}$ holds.

(b) Consider by contradiction that $(\omega^*)^\top x^s + b^* \geq 0$ holds. This means that $(\omega^*, b^*, \xi^*, \eta^*, z^*)$ satisfies (P5b)–(P5k). Moreover, since $\tilde{f}$ is the objective function for Problem (P5), we get $f^* = \tilde{f}$. However, $f^* \leq \bar{f}$ holds. Thus,

$$f^* \leq \bar{f} < \tilde{f} = f^*$$

yields a contradiction. □

Note that the last lemma can be adapted for the case $\bar{\omega}^\top x^s + \bar{b} > 0$. In this case, the constraints (P5g) need to be replaced with

$$-(1 - z_s)M \leq \omega^\top x^s + b \leq 0 \tag{9}$$

and (b) needs to be replaced with $(\omega^*)^\top x^s + b^* > 0$, i.e., $x^s \in P_u$. Note that the more points we have fixed on one side, the solution of Problem (P3) tends to be faster as there are fewer binary variables.

Moreover, the solution of Problem (P3) can be found by solving the problem

$$\min_{\omega,b,\xi,\eta,z} \quad \frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2) \tag{P6a}$$

$$\text{s.t.} \quad y_i(\omega^\top x^i - b) \geq 1 - \xi_i, \quad i \in [1, n], \tag{P6b}$$

$$\omega^\top x^i + b \leq z_i M, \quad i \in [n+1, N] \setminus (S_p \cup S_n), \tag{P6c}$$

$$\omega^\top x^i + b \geq -(1 - z_i)M, \quad i \in [n+1, N] \setminus (S_p \cup S_n), \tag{P6d}$$

$$\omega^\top x^i + b \geq 0, \quad i \in S_p, \tag{P6e}$$

$$\omega^\top x^i + b \leq 0, \quad i \in S_n, \tag{P6f}$$

$$\tau - \eta_1 \leq |S_p| + \sum_{i \in [n+1,N] \setminus (S_p \cup S_n)} z_i \leq \tau + \eta_2, \tag{P6g}$$

$$\xi_i \geq 0, \quad i \in [1, n], \tag{P6h}$$

$$\eta_1, \eta_2 \geq 0 \tag{P6i}$$

$$z_i \in \{0, 1\}, \quad i \in [n+1, N] \setminus (S_p \cup S_n), \tag{P6j}$$

where S_p and S_n are subsets of P_u and N_u, respectively.

Based on these results, we propose the following. We compute the point $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ using Algorithm 2, leading to an objective function value $\bar{f}$ for Problem (P3). Afterward, we sort the indices $i \in \{n+1, N\}$, indicated by the permutation $\alpha\colon \{n+1, N\} \to \{n+1, N\}$, so that $|\bar{\omega}^\top x^{\alpha(i)} + \bar{b}| \geq |\bar{\omega}^\top x^{\alpha(i)+1} + \bar{b}|$ holds.

Consider now a given and fixed parameter $B_{\max}$, a factor $\gamma \in (1, m/B_{\max}]$, and let β be $\gamma B_{\max}$ rounded to the next integer. While the number of fixed points is smaller than $B_{\max}$, we do the following. For $i \in \{1, \dots, \beta\}$, if $\bar{\omega}^\top x^{\alpha(i)} + \bar{b} < 0$ holds, we try to solve Problem (P5) using the limit time of $T_{\max}$ and the upper bound $\bar{f}$. If there is a feasible point of this problem, we set $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ to this point and update the objective function value $\bar{f}$ accordingly. If no feasible point could be computed and if the limit time was not reached, we fix x^s to be in the negative side.

Similarly, we do the same if $\bar{\omega}^\top x^{d_\ell} + \bar{b} > 0$ holds with (P5g) replaced by (9). The method is formally described in Algorithm 3. Finally note that, although Problem (P5) is an MIQP, it is a feasibility problem, which is often easier to solve than an optimization problem in practice. Besides that, if the point obtained from Algorithm 2 is close to the optimum of Problem (P3), many unlabeled points will be fixed and Problem (P3) will be faster to solve.

Algorithm 3: Improved & Warm-Started Re-Clustering Method (WIRCM)

Input: $X \in \mathbb{R}^{d\times N}$, $y \in \{-1, 1\}^n$, $k^1 \in \mathbb{N}$, $C_1 > 0$, $C_2 > 0$, $\tau \in \mathbb{N}$, $\hat{\Delta}^1 \in (0, 1)$, $\tilde{\Delta} \in (0, 1)$, $\mathcal{G}^1 = \emptyset$, $k^+ \in \mathbb{N}$, $T_{\max} > 0$, $B_{\max} \in \mathbb{N}$, and $\gamma \in (1, m/B_{\max}]$.

Compute the hyperplane $(\bar{\omega}, \bar{b})$ and $\bar{\xi}, \bar{\eta}, \bar{z}$ using Algorithm 2, leading to the objective function value $\bar{f}$. Let M be the last M^t of Algorithm 2.
Sort the indices $i \in \{n + 1, N\}$ such that $|\bar{\omega}^\top x^{\alpha(i)} + \bar{b}| \geq |\bar{\omega}^\top x^{\alpha(i)+1} + \bar{b}|$ holds and set β to be $\gamma B_{\max}$ rounded to the next integer.
for $i \in \{1, \dots, \beta\}$ **do**
  if $|S_p| + |S_n| \leq B_{\max}$ **then**
    if $\bar{\omega}^\top x^s + \bar{b} < 0$ **then**
      Solve Problem (P5) with upper bound $\bar{f}$ and a time limit $T_{\max}$.
      if *there is a feasible point* **then**
        update $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$, and $\bar{f}$
      else if $T_{\max}$ *was not reached* **then**
        $S_n \leftarrow S_n \cup \{s\}$
      end
    else if $\bar{\omega}^\top x^s + \bar{b} > 0$ **then**
      Solve the problem (P5) with (P5g) replaced by (9), using $\bar{f}$ as an upper bound and a time limit of $T_{\max}$.
      if *there is a feasible point* **then**
        update $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$, and $\bar{f}$
      else if $T_{\max}$ *was not reached* **then**
        $S_p \leftarrow S_p \cup \{s\}$
      end
    end
  end
end
Compute the solution $(\omega^*, b^*, \xi^*, \eta^*, z^*)$ of Problem (P6) with $(\bar{\omega}, \bar{b}, \bar{\xi}, \bar{\eta}, \bar{z})$ value and $\bar{f}$ as an upper bound.

6 Numerical results

In this section, we present and discuss our computational results that illustrate the benefits of knowing the total amount of each class of unlabeled data and of using our approaches to speed up the solution process. We evaluate this on different test sets from the literature. The test sets are described in Sect. 6.1, while the computational setup is depicted in Sect. 6.2. The evaluation criteria are described in Sect. 6.3 and the numerical results are discussed in Sect. 6.4.

6.1 Test sets

For the computational analysis of the proposed approaches, we consider the subset of instances presented by Olson et al. (2017) that are suitable for classification problems and that have at most three classes. We restrict ourselves to instances of at most three classes to obtain an overall test set of manageable size. Repeated instances are removed and instances with missing information are reduced to the observations without missing information. If three classes are given in an instance, we transform them into two classes such that the class with label 1 represents the positive class, and the other two classes represent the negative class. This results in a final test set of 97 instances; see Table 1 in "Appendix A".

To avoid numerical instabilities, we re-scale all data sets as follows. For each coordinate $j \in [1, d]$, we compute

$$l_j = \min_{i \in [1,N]} \{x_j^i\}, \quad u_j = \max_{i \in [1,N]} \{x_j^i\}, \quad m_j = 0.5\left(l_j + u_j\right)$$

and shift each coordinate j of all data points x^i via $\bar{x}_j^i = x_j^i - m_j$. If we do this for all data points, they get centered around the origin. Moreover, if a coordinate j of the re-scaled points is still large, i.e., if $\tilde{l}_j = l_j - m_j < -10^2$ or $\tilde{u}_j = u_j - m_j > 10^2$ holds, it is re-scaled via

$$\tilde{x}_j^i = (\overline{v} - \underline{v}) \frac{\bar{x}_j^i - \tilde{l}_j}{\tilde{u}_j - \tilde{l}_j} + \overline{v},$$

with $\overline{v} = 10^2$ and $\underline{v} = -10^2$. The corresponding 29 instances that we re-scaled are marked with an asterisk in Table 1. Note that we use a linear transformation to scale the datasets. Hence, after computing the hyperplane for the scaled data, the respective hyperplane for the original data can also be computed ex post by applying another suitably chosen linear transformation as well.

In our computational study, we want to highlight the importance of cardinality constraints, especially for the case of non-representative biased samples. Biased samples occur frequently in non-probability surveys, which are surveys for which the inclusion process is not monitored and, hence, the inclusion probabilities are unknown as well. Correction methods like inverse inclusion probability weighting are therefore not applicable. For an insight into inverse inclusion probability weighting, see Skinner and D'arrigo (2011) and references therein.

To mimic this situation, we create 5 biased samples with 10 % of the data being labeled for each instance. Different from a simple random sample in which each point has an equal probability of being chosen as labeled data, in the biased sample, the labeled data is chosen with probability 85 % for being on the positive side of the hyperplane. Then, for each instance, with a time limit of 3600 s, we apply the approaches listed in Sect. 6.2. In Appendix C, we also provide the results under simple random sampling, which produces unbiased samples. We see that the results form the proposed methods are similar to the plain SVM in that setting. Hence, besides the additional computational burden, there is no downside to use the proposed method in case of an unknown sampling process.

6.2 Computational setup

Our algorithm has been implemented in Julia 1.8.5 and we use Gurobi 9.5.2 and JuMP (Dunning et al. 2017) to solve Problem (P1), (P3), and (P4). All computations were executed on the high-performance cluster "Elwetritsch", which is part of the "Alliance of High-Performance Computing Rheinland-Pfalz" (AHRP). We used a single Intel XEON SP 6126 core with 2.6 GHz and 64 GB RAM.

For each one of the 485 instances described in Sect. 6.1, the following approaches are compared:

(a) SVM as given in Problem (P1), where only labeled data are considered;
(b) CS^{3}VM as given in Problem (P3) with M as given in (1);
(c) IRCM as described in Algorithm 2;
(d) WIRCM as described in Algorithm 3.

Based on our preliminary experiments, we set the penalty parameters $C_1 = C_2 = 1$. For WIRCM, we impose a time limit for solving Problem (P5) of $T_{\max} = 40$ s. Moreover, we choose $\gamma = 1.2$ and the maximum number $B_{\max}$ of unlabeled points that can be fixed as

$$B_{\max} = \begin{cases} 0.2m, & \text{if } m \in [1, 100], \\ 0.25m, & \text{if } m \in (100, 500], \\ 0.35m, & \text{if } m \in (500, 1000], \\ 0.45m, & \text{otherwise.} \end{cases}$$

Finally, for IRCM and WIRCM, we set $\hat{\Delta}^1 = 0.8$, $\tilde{\Delta} = 0.1$, $k^+ = 50$, and the initial number of clusters is set to

$$k^1 = \begin{cases} 10, & \text{if } m \in [1, 500], \\ 20, & \text{if } m \in (500, 1000], \\ 50, & \text{otherwise.} \end{cases}$$

A more detailed discussion of the choice of hyperparameters is given in Appendix D.

6.3 Evaluation criteria

The first evaluation criterion is the run time of SVM, CS^{3}VM, IRCM, and WIRCM. The results will help to contextualize other evaluation criteria such as accuracy and precision. To compare run times, we use empirical cumulative distribution functions (ECDFs). Specifically, for S being a set of solvers (or approaches as above) and for P being a set of problems, we denote by $t_{p,s} \geq 0$ the run time of approach $s \in S$ applied to problem $p \in P$ in seconds. If $t_{p,s} > 3600$, we consider problem p as not being solved by approach s. With these notations, the performance profile of approach s is the graph of the function $\gamma_s \colon [0, \infty) \to [0, 1]$ given by

$$\gamma_s(\sigma) = \frac{1}{|P|} \left| \left\{ p \in P \colon t_{p,s} \leq \sigma \right\} \right|. \tag{10}$$

The second evaluation criterion is based on Theorem 5, where we show that the objective function value of the point obtained by IRCM is an upper bound for CS^{3}VM, and consequently for Problem (P4) that is solved with WIRCM. Note that SVM also provides a feasible point for CS^{3}VM and, consequently, provides an upper bound as well. Consider (ω, b, ξ) the solution of SVM, we compute the binary variables z_i, $i \in [n+1, N]$ as follows:

$$z_i = \begin{cases} 1, & \text{if } \omega^\top x^i + b > 0, \\ 0, & \text{if } \omega^\top x^i + b < 0. \end{cases}$$

If $\omega^\top x^i + b = 0$ for some x^i, we set

$$z_i = \begin{cases} 1, & \text{if } \sum_{j \in [n+1, N] \colon \omega^\top x^j + b \neq 0} z_i \leq \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we set

$$\eta_1 = \max \left\{ 0, \tau - \sum_{i=n+1}^{N} z_i \right\}, \quad \eta_2 = \max \left\{ 0, \sum_{i=n+1}^{N} z_i - \tau \right\},$$

and the objective function value can be computed as

$$\frac{\|\omega\|^2}{2} + C_1 \sum_{i=1}^{n} \xi_i + C_2(\eta_1 + \eta_2).$$

Based on that, we compare how close the objective function values obtained from SVM, CS^{3}VM, IRCM, and WIRCM are to the optimal solution. To this end, we use ECDFs, for which we replace $t_{p,s}$ by $f_{p,s}$ in Eq. (10) with

$$f_{p,s} := \frac{b_{p,s} - f_p^*}{f_p^*}, \tag{11}$$

where f_p^* is the optimal objective function value of problem p and $b_{p,s}$ is the objective function value obtained by approach s.

Besides that, for each instance and for each approach described in Sect. 6.2, after computing the hyperplane (ω, b), we classify all points x^i as being on the positive side if $\omega^\top x^i + b > 0$ and as being on the negative side if $\omega^\top x^i + b < 0$ holds. For CS^{3}VM and WIRCM, if the hyperplane (ω, b) satisfies $\omega^\top x^i + b = 0$ for some unlabeled point x^i, we classify this point as positive or negative depending on the respective binary variable z_i. On the other hand, for IRCM, if $\omega^\top x^i + b = 0$ for some unlabeled point x^i, we classify this point as positive or negative depending on z_j with j so that $x^i \in \mathcal{C}_j$. For the labeled points in these three approaches and for all points in the SVM, if $\omega^\top x^i + b = 0$ holds, we classify the point on the correct side. Note that for the cases in which the IRCMs take more than 3600 s to solve the instance, we use the last hyperplane found by the algorithm. If we hit the time limit in Gurobi when solving CS^{3}VM (either standalone or in the final phase of the WIRCM), we take the best solution found so far.

Knowing the true label of all points, we then distinguish all points in four categories: true positive (TP) or true negative (TN) if the point is classified correctly in the positive or negative class, respectively, as well as false positive (FP) if the point is misclassified in the positive class and as false negative (FN) if the point is misclassified in the negative class. Based on that we compute two classification metrics, for which a higher value indicates a better classification. The first one is accuracy (AC). It measures the proportion of correctly classified points and is given by

$$\mathrm{AC} := \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \in [0, 1]. \tag{12}$$

The second metric is precision (PR). It measures the proportion of correctly classified points among all positively classified points and is computed by

$$\mathrm{PR} := \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \in [0, 1]. \tag{13}$$

The main comparison in terms of accuracy and precision is w.r.t. the "true hyperplane", i.e., the solution of Problem (P1) on the complete data with all N points and all labels available. The main question is how close the accuracy and precision is to the one of the true hyperplane. Hence, we compute the ratios of the accuracy and precision according to

$$\widehat{\mathrm{AC}} := \frac{\mathrm{AC}}{\mathrm{AC}_{\mathrm{true}}}, \quad \widehat{\mathrm{PR}} := \frac{\mathrm{PR}}{\mathrm{PR}_{\mathrm{true}}}, \tag{14}$$

where $\mathrm{AC}_{\mathrm{true}}$ and $\mathrm{PR}_{\mathrm{true}}$ are computed as in Eqs. (12) and (13) for the true hyperplane.

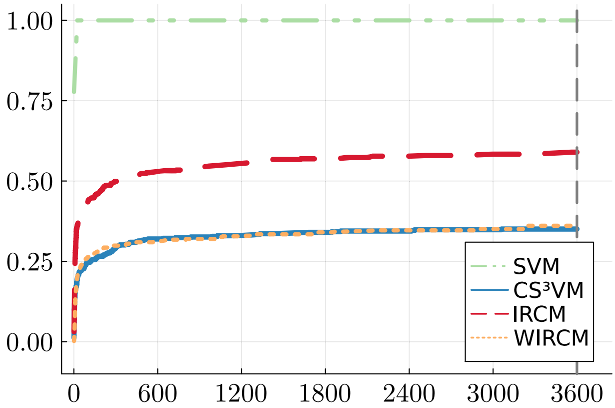

Fig. 2 ECDFs for run time (in seconds)

We also compare the measures with the SVM method, which only considers the information of the labeled data. For this purpose, we compute

$$\overline{\mathrm{AC}} := \frac{\mathrm{AC} - \mathrm{AC}_{\mathrm{SVM}}}{\mathrm{AC}_{\mathrm{SVM}}}, \quad \overline{\mathrm{PR}} := \frac{\mathrm{PR} - \mathrm{PR}_{\mathrm{SVM}}}{\mathrm{PR}_{\mathrm{SVM}}}, \tag{15}$$

where $\mathrm{AC}_{\mathrm{SVM}}$ and $\mathrm{PR}_{\mathrm{SVM}}$ are computed as in (12) and (13) for the SVM hyperplane. To keep the numerical results section concise, we report on recall and the false positive rate in Appendix B.

6.4 Numerical results

6.4.1 Run time

Figure 2 shows the ECDFs for the measured run times. Clearly, SVM is the fastest algorithm. This is expected as the SVM does not include any binary variables related to the unlabeled points, which is in contrast to other approaches. It can be seen that the IRCM outperforms both CS^{3}VM and WIRCM. This shows that the idea to cluster unlabeled data points significantly decreases the run time. However, we need to be careful with the interpretation of these run times since termination of SVM and IRCM does not imply that a globally optimal point is found, whereas this is guaranteed CS^{3}VM and the WIRCM. The quality of the points found by SVM and IRCM will be discussed in the next section. The figure also clearly indicates that Problem (P2) is rather challenging: Even IRCM, which terminates for the most instances within the time limit (indicated by the gray and dashed vertical line) only does so for 57 % of the instances. Note that the WIRCM has the worst efficiency. This obviously needs to be the case since due to Step 1 of Algorithm 3, its run time always includes the run time of the IRCM. To shed some light on the scalability of the approaches, we also present a brief analysis of the run times in dependence of the number of samples in Appendix E.

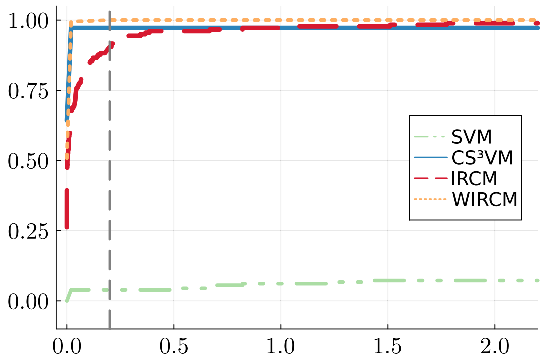

Fig. 3 ECDFs for the quality of the obtained upper bounds

6.4.2 Quality of the obtained upper bounds

As discussed in the last section, for some instances none of the three approaches that actually consider the unlabeled data terminate within the given time limit. This means we do not obtain the optimal objective function value for these instances, which we, moreover, can only provably obtain by CS^{3}VM and the WIRCM. In fact, we have the optimal solution for 179 instances. These are the baseline instances for Fig. 3, which shows the ECDFs for the upper bound quality, as defined in (11). Note that the objective function value obtained by SVM is very far from the optimal value, while the IRCM finds an objective function value rather close to the optimal value (with $f_{ps} \leq 0.2$, see the gray dashed vertical line) in 90 % of these instances. Besides that, the WIRCM outperforms CS^{3}VM in this comparison, which means using the IRCM as a warm start improves the result.

The consequences of the results so far are the following. If one is interested in getting a rather good feasible point as quickly as possible, one should use the IRCM. If one is able to spend some more run time, one should use the WIRCM. Hence, both novel methods derived in this paper have their advantage over just solving the CS^{3}VM with a standard MIQP solver.

6.4.3 Accuracy

For some instances, none of the three approaches that actually tackle the unlabeled data terminate within the given time limit. Hence, our first comparison only considers instances for which CS^{3}VM terminates within the time limit.

As can be seen in Fig. 4, the relative accuracy $\widehat{\mathrm{AC}}$ (w.r.t. the true hyperplane) of CS^{3}VM, is closer to 1 than the relative accuracy of SVM—especially for the unlabeled data. This means that using the unlabeled points as well as the cardinality constraint allows to re-produce the classification of the true hyperplane with higher accuracy than the standard SVM does. Besides that, the relative accuracy of the SVM is more spread than the one of the other approaches, indicating that there is comparable more variation in the results as compared the results of CS^{3}VM. The box in the boxplot depicts the range of the medium 50 % of the values; 25 % of the values are below and 25 % are above the box.

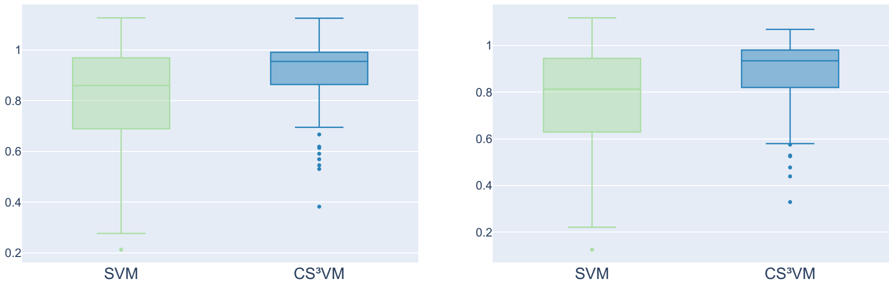

Fig. 4 Relative accuracy $\widehat{\text{AC}}$ w.r.t. the true hyperplane; see (14). Only those instances are considered for which CS^{3}VM terminated. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

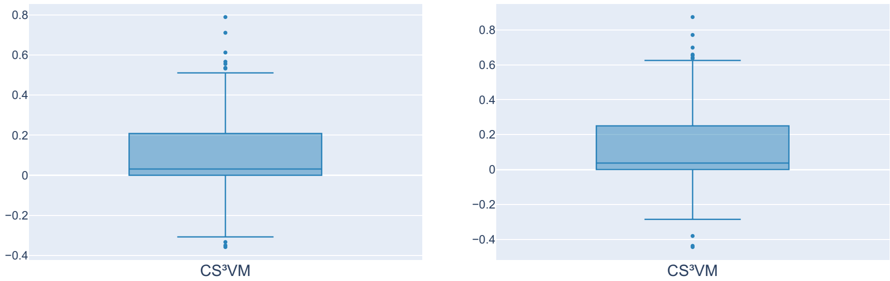

Fig. 5 Accuracy values $\overline{\text{AC}}$ w.r.t. the SVM; see (15) only consider the instances that CS^{3}VM terminated. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

Figure 5 shows that, in almost 75 % of the cases, CS^{3}VM, has $\overline{\text{AC}}$ values larger than zero, where zero means the same accuracy as the SVM itself. In the others 25 % of the cases, the $\overline{\text{AC}}$ of CS^{3}VM is slightly smaller than SVM.

The second comparison considers only those three approaches that actually consider the unlabeled data, i.e., CS^{3}VM, IRCM, and WIRCM for all instances. As can be seen in Fig. 6, even though IRCM does not have an optimality guarantee, it has a better relative accuracy $\widehat{\text{AC}}$ than the hyperplane obtained from CS^{3}VM within the time limit. Consequently, as the hyperplane obtained from IRCM is used as a warm-start in WIRCM, it also has better accuracy.

Figure 7 shows that, in almost 75 % of the cases, CS^{3}VM, the IRCM, and the WIRCM have $\overline{\text{AC}}$ values larger than zero. That is, in general, our methods have greater accuracy than the SVM. Though, some cases indicate worse $\overline{\text{AC}}$ values for our methods than for the SVM. This happens because for some instances, the methods (mainly for CS^{3}VM; see also Fig. 2) do not terminate within the time limit. Hence, we expect that the number of negative values will decrease if we would increase the time limit.

6.4.4 Precision

We again separate the comparisons as in Sect. 6.4.3. Figure 8 shows that the SVM's relative precision $\widehat{\text{PR}}$ is lower than the relative precision of CS^{3}VM. This means that

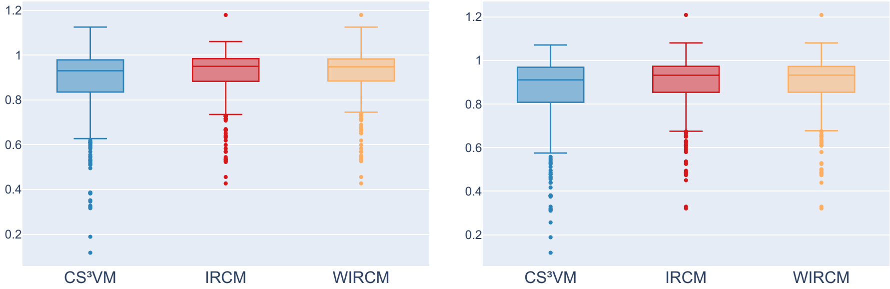

Fig. 6 Relative accuracy $\widehat{AC}$ w.r.t. the true hyperplane; see (14). Left: Comparison for all data points. Right: Comparison only for unlabeled data points

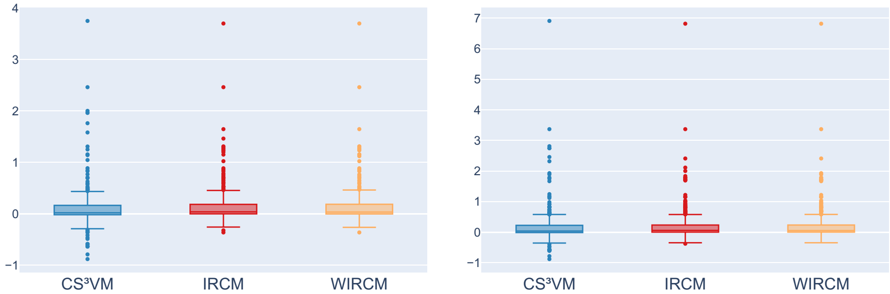

Fig. 7 Accuracy values $\overline{AC}$ w.r.t. the SVM; see (15) consider all instances. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

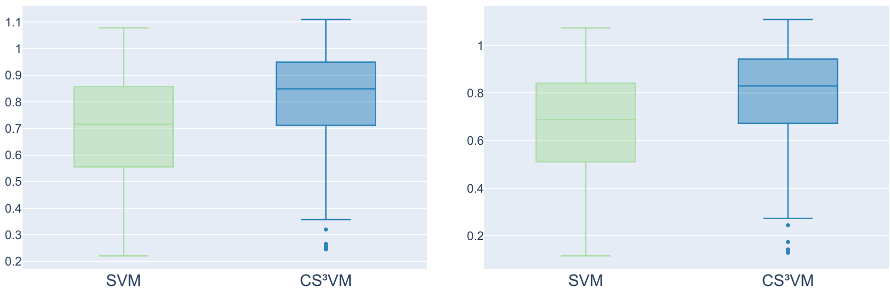

Fig. 8 Relative precision $\widehat{PR}$ w.r.t. the true hyperplane as; see (14). Only those instances are considered for which CS^{3}VM terminated. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

CS^{3}VM re-produces the classification of the true hyperplane with higher precision than the original SVM. Hence, SVM has more false-positive results. This happens because the biased sample is more likely to have positively labeled data and due to having no information about the unlabeled data, the SVM ends up classifying points on the positive side. As can be seen in Fig. 9, CS^{3}VM has slightly higher $\overline{PR}$ values than 0, which is the baseline here that refers to the SVM itself. This means, CS^{3}VM is slightly more precise than the SVM.

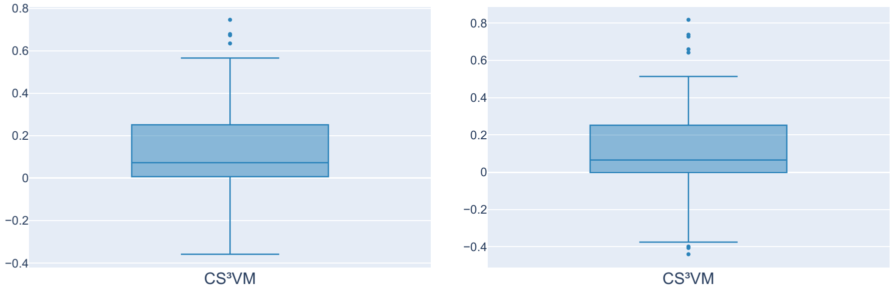

Fig. 9 Precision values $\overline{\text{PR}}$ w.r.t. the SVM; see (15). Only those instances are considered for which CS^{3}VM terminated. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

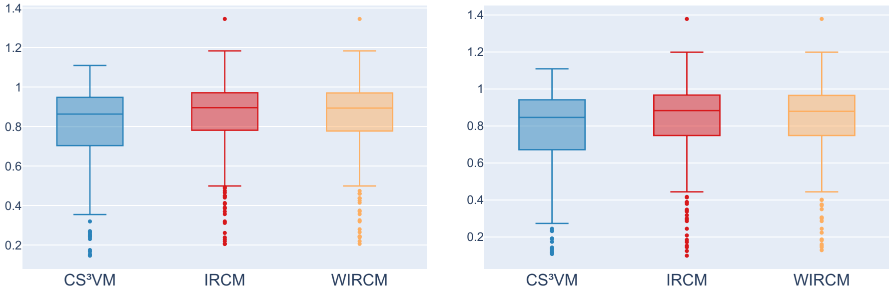

Fig. 10 Relative precision $\widehat{\text{PR}}$ w.r.t. the true hyperplane as; see (14). Left: Comparison for all data points. Right: Comparison only for unlabeled data points

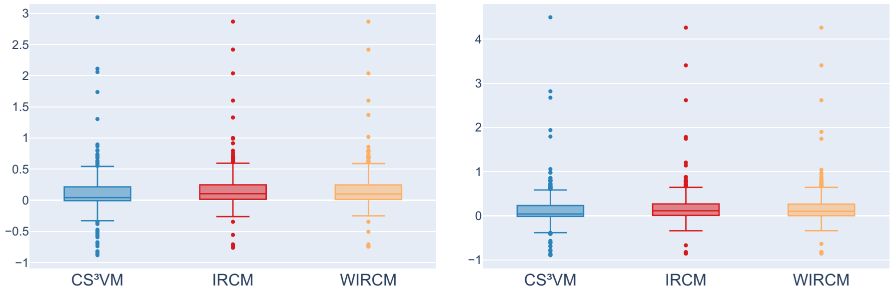

Fig. 11 Precision values $\overline{\text{PR}}$ w.r.t. the SVM; see (15). Left: Comparison for all data points. Right: Comparison only for unlabeled data points

Figure 10 shows that the $\widehat{\text{PR}}$ values of the IRCM and the WIRCM are less spread than the ones of CS^{3}VM. The reason most likely is that the CS^{3}VM approach terminates on fewer instances than the IRCM and the WIRCM. As can be seen in Fig. 11, the IRCM and the WIRCM also have slightly higher $\overline{\text{PR}}$ values than 0. This means that our methods are slightly more precise than the SVM. The negative outliers most likely are due to the same reason as those for the respective accuracy values.

7 Conclusion

For many classification problems, it can be costly to obtain labels for the entire population of interest. However, aggregate information on how many points are in each class can be available from external sources. For this situation, we proposed a semi-supervised SVM that can be modeled via a big-M-based MIQP formulation. We also presented a rule for updating the big-M in an iterative re-clustering method and derived further computational techniques such as tailored dimension reduction and warm-starting to reduce the computational cost.

In case of simple random samples, our proposed semi-supervised methods perform as good as the classic SVM approach. However, in many applications, the available data is coming from non-probability samples. Hence, there is the risk of obtaining biased samples. Our numerical study shows that our approaches have better accuracy and precision than the original SVM formulation in this setting.

The problem of considering a cardinality constraint is computationally challenging. Our proposed clustering approach significantly helps to decrease the run time and to find an objective function value that is very close to the optimal value. Besides that, the clustering approach maintains the same accuracy and precision as the MIQP formulation. Moreover, using the clustering approach as a warm-start and fixing some unlabeled points on one side of the hyperplane helps to improve the quality of the objective function value again. Hence, the newly proposed methods lead to a significant improvement compared to just solving the classic MIQP formulation using a standard solver.

Despite these contributions, there is still room for improvement and future work. First, we only considered the linear SVM kernel. For future work, the development of methods for other kernels, such as a Gaussian kernel, can be a valuable topic. Moreover, the use of other norms than the 2-norm could be analyzed as well and the formal hardness of the considered problem should be settled. Finally, the adaptation of our approaches for multiclass SVMs using a one-vs.-rest strategy may be another reasonable future work.

Appendix A: Detailed information on the instances

See Table 1.

Table 1 Overview over the entire test set with the number of points (N) and the dimension (d)

ID	Instance	N	d
1	prnn_synth	250	2
2*	analcatdata_asbestos	73	3
3*	lupus	87	3
4	analcatdata_boxing1	120	3
5	analcatdata_boxing2	132	3
6	haberman	289	3
7	analcatdata_happiness	60	3

Table 1 continued

ID	Instance	N	d
8*	analcatdata_aids	50	4
9	analcatdata_lawsuit	263	4
10	iris	147	4
11	hayes_roth	93	4
12	balance_scale	625	4
13	parity5	32	5
14*	bupa	341	5
15	irish	470	5
16	phoneme	5349	5
17	tae	110	5
18	new_thyroid	215	5
19*	analcatdata_bankruptcy	50	6
20*	analcatdata_creditscore	100	6
21	mux6	64	6
22	monk3	357	6
23	monk1	432	6
24	monk2	432	6
25	appendicitis	106	7
26	prnn_crabs	200	7
27*	penguins	333	7
28	postoperative_patient_data	78	8
29*	biomed	209	8
30*	pima	768	8
31*	cars	392	8
32	analcatdata_japansolvent	52	9
33	glass2	162	9
34	breast_cancer	272	9
35	saheart	462	9
36	threeOf9	512	9
37	profb	672	9
38	breast_w	463	9
39	tic_tac_toe	958	9
40	xd6	512	9
41	cmc	1425	9
42	analcatdata_cyyoung9302	92	10
43	analcatdata_cyyoung8092	97	10
44	breast	691	10
45	flare	315	10
46	parity5+5	1024	10
47	magic	18,905	10

Table 1 continued

ID	Instance	N	d
48	analcatdata_fraud	42	11
49	heart_statlog	270	13
50	heart_h	293	13
51	hungarian	293	13
52*	cleve	302	13
53*	heart_c	302	13
54	wine_recognition	178	13
55*	australian	690	14
56*	adult	48,790	14
57*	schizo	340	14
58*	buggyCrx	690	15
59	labor	57	16
60	house_votes_84	342	16
61	hepatitis	155	19
62*	credit_g	1000	20
63	gametes_e_0.1H	1599	20
64	gametes_e_0.4H	1600	20
65	gametes_e_0.2H	1600	20
66	gametes_h_50	1592	20
67	gametes_h_75	1599	20
68*	churn	5000	20
69*	ring	7400	20
70	twonorm	7400	20
71	waveform_21	5000	21
72	ann_thyroid	7129	21
73	spect	228	22
74	horse_colic	357	22
75	agaricus_lepiota	8124	22
76*	hypothyroid	3086	25
77*	dis	3711	29
78*	allhypo	3709	29
79*	allbp	3711	29
80*	breast_cancer_wisconsin	569	30
81	backache	180	32
82	ionosphere	351	34
83	chess	3196	36
84	waveform_40	5000	40
85	connect_4	67,557	42
86	spectf	267	44

Table 1 continued

ID	Instance	N	d
87*	tokyo1	959	44
88	molecular_biology_promoters	106	57
89*	spambase	4210	57
90	sonar	208	60
91	splice	2903	60
92	coil2000	8380	85
93*	Hill_Valley_without_noise	1212	100
94*	clean1	476	168
95*	clean2	6598	168
96	dna	3002	180
97	gametes_e_1000atts	1600	1000

Appendix B: Further numerical results

Besides the measures of accuracy and precision, we compare two further measures in this section. First, recall (RE) measures the percentage of points with positive label that are actually classified as positive. It is formally given by

$$\mathrm{RE} := \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{16}$$

Note that for applications such as cancer diagnosis, it is relevant to evaluate recall because it is more important to flag cancer rather than to do not. Also in cases of rare positive labels, recall is often the favored metric. Note that values close to 1 indicate a better classification here.

Second, we also compare the false positive rate (FPR), which measures the probability of points with negative labels being classified as positive:

$$\mathrm{FPR} := \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{17}$$

This quantity is important in some applications such as quality control, where a false positive can cause more issues than a false negative. Note that for FPR, the lower the value, the better the classification.

The main comparison in terms of recall and false positive rate is w.r.t. the "true hyperplane", i.e., the solution of Problem (P1) on the complete data with all N points and all labels available. The main question is how close the recall and false positive rate is to the one of the true hyperplane. Hence, we compute the ratios of the recall and false positive rate according to

$$\widehat{\mathrm{RE}} := \frac{\mathrm{RE}}{\mathrm{RE}_{\mathrm{true}}}, \quad \widehat{\mathrm{FPR}} := \frac{\mathrm{FPR}}{\mathrm{FPR}_{\mathrm{true}}}, \tag{18}$$

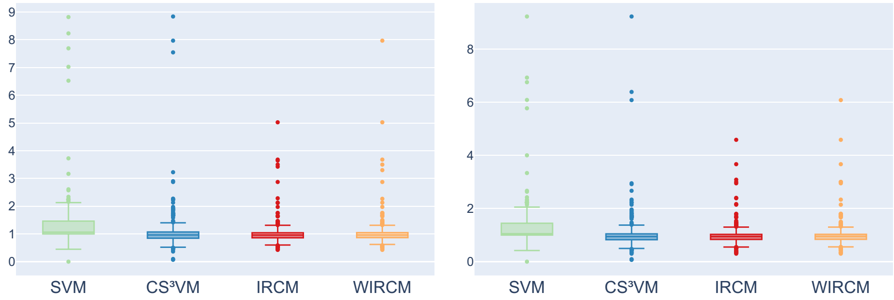

Fig. 12 Relative recall $\widehat{\mathrm{RE}}$ w.r.t. the true hyperplane; see (18). Left: Comparison for all data points. Right: Comparison only for unlabeled data points

Fig. 13 Relative false positive rate $\widehat{\mathrm{FPR}}$ w.r.t. the true hyperplane; see (18). Left: Comparison for all data points. Right: Comparison only for unlabeled data points

where $\mathrm{RE}_{\mathrm{true}}$ and $\mathrm{FPR}_{\mathrm{true}}$ are computed as in (16) and (17) for the true hyperplane.

As can be seen in Fig. 12, the SVM's relative recall is a little bit larger than the one of the other methods. As in Sect. 6.4.4, this happens because the biased sample is more likely to have positive labeled data and having no information about the unlabeled data, the SVM ends up classifying points on the positive side.

Figure 13 shows that CS^{3}VM, the IRCM, and the WIRCM have lower $\widehat{\mathrm{FPR}}$ values than the original SVM. This means that the newly proposed methods have a lower false positive rate than the original SVM. The fact that CS^{3}VM terminates for less instances than the IRCM explains why the IRCM has a lower relative false positive rate than CS^{3}VM. Finally, since the WIRCM uses the IRCM for warm-starting, the WIRCM also has better relative false positive rates than CS^{3}VM.

Appendix C: Numerical results for simple random samples

In Sect. 6, we focused our computational study on non-representative, biased samples. The common baseline scenario to check the performance of estimators is to apply them on simple random samples. Hence, for completeness, we also present the results under simple random sampling. That is, each unit in the data set has the same probability $\pi_i = n/N$ to be included into the sample of size n. The instances are the same as

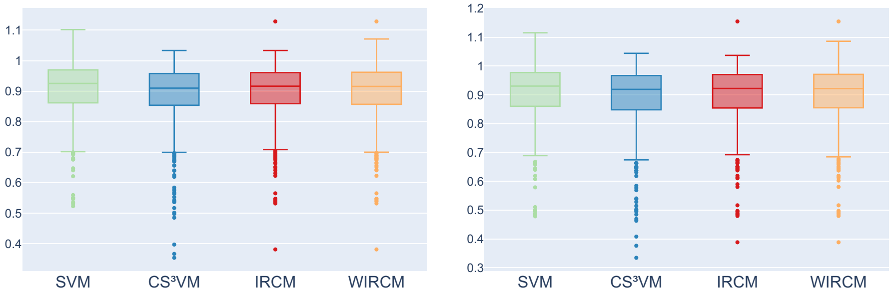

Fig. 14 Relative accuracy $\widehat{\mathrm{AC}}$ w.r.t. the true hyperplane; see (14), for the simple random samples. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

Fig. 15 Relative precision $\widehat{\mathrm{PR}}$ w.r.t. the true hyperplane; see (14), for the simple random samples. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

described in Sect. 6.1. The computational setup follows the description in Sect. 6.2. As before, the used evaluation criteria are $\widehat{\mathrm{AC}}$, $\widehat{\mathrm{PR}}$ as in (14) and $\widehat{\mathrm{RE}}$, $\widehat{\mathrm{FPR}}$ as in (18).

Figures 14 and 15 show similar accuracy and precision performance for all approaches. This is as expected, as the sample is not biased and hence the cardinality constraint does not contribute relevant additional information to the problem. Therefore, the SVM does not tend to classify the points as positive as it is the case for the biased samples. The outliers, mainly present for $\mathrm{CS}^3\mathrm{VM}$, are due those instances that are not solved within the time limit. As can be seen in Figs. 16 and 17, recall and false positive rate are also similar for all approaches.

Hence, for the simple random samples our approaches have almost the same results as the SVM. Note that for the biased samples, they outperformed the SVM. Hence, in cases for which the type of sample is not known, it is "safe" to use the newly proposed approaches for classification.

Appendix D: Choosing the hyperparameters

Each parameter of Algorithm 2 and 3 as well as in Problem (P1) and (P3) can be chosen from a range. In Table 2 we present plausible ranges for these parameters.

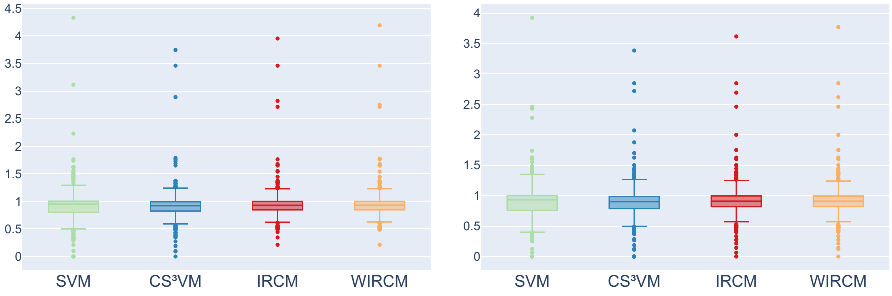

Fig. 16 Relative recall $\widehat{\mathrm{RE}}$ w.r.t. the true hyperplane; see (18), for the simple random samples. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

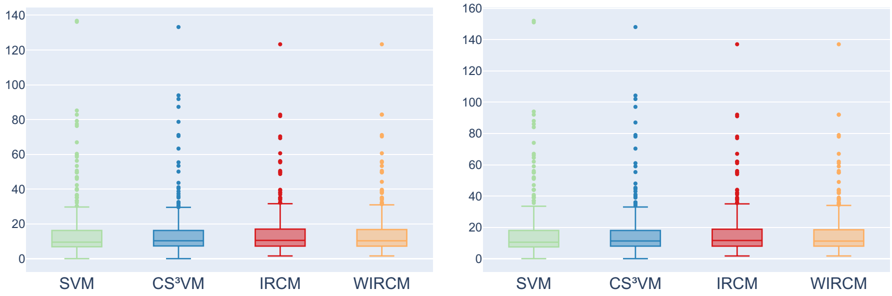

Fig. 17 Relative false positive rate $\widehat{\mathrm{FPR}}$ w.r.t. the true hyperplane; see (18), for the simple random samples. Left: Comparison for all data points. Right: Comparison only for unlabeled data points

Table 2 Plausible ranges for the hyperparameters

Parameter	Plausible range	Current choice
C_1	$\mathbb{R}_{\geq 0}$	1
C_2	$[0.5C_1, 2C_1]$	1
k^1	$[2, m]$	10, 20, 50
k^+	$[k^1, m]$	50
$\hat{\Delta}^1$	$[0.5, 0.9]$	0.8
$\tilde{\Delta}$	$[0.1, 1 - \hat{\Delta}^1]$	0.1
$B_{\max}$	$[1, m]$	$0.2m, 0.25m, 0.35m, 0.45m$
γ	$[1.1, m/B_{\max}]$	1.2
$T_{\max}$	$[10, 100]s$	$40s$

Clearly, $C_1 \in \mathbb{R}_{\geq 0}$ holds. However, the closer the value is to 1, the more equally important are maximizing the margin and minimizing the classification error for the labeled data. The range of C_2 is based on C_1 in order to indicate how much more important the unlabeled data is compared to the labeled data. Again, we choose $C_2 = 1$ so that both data have the same importance. Besides that, if C_2 is much bigger than C_1, our preliminary tests showed that this leads to focus on minimizing the classification error for the unlabeled data, which implies focusing on the binary variable and, hence, leads to larger run times.

For choosing the other parameters, we consider the first 3 datasets presented in Table 1 and varied the parameter choices in a preliminary numerical study. Based on the results, we now discuss how to choose the remaining parameters. The parameter k^1 can be between 2 and m since we cluster m unlabeled points. Note that, the smaller k^1, the less time per iteration is needed since we have fewer binary variables. However, more iterations may be needed to find the solution. On the other hand, the bigger k^1, the more time per iteration is required. We choose to start with a small value of k^1 because in preliminary numerical tests, when the algorithm terminated, the number of clusters never exceeded $m/3$. Moreover, in our preliminary tests, if the algorithm exceeds $k^t = 50$ for some iteration t, it takes a lot of time to solve Problem (P4). To decrease this time, we reduced the number of clusters, eliminating the ones being far from the hyperplane. This is the reason why we choose $k^+ = 50$.

The parameter $\hat{\Delta}^1$ indicates that clusters with a distance to the hyperplane greater than the $\hat{\Delta}^1$-quantile of all distances will be deactivated. It is between 0.5 and 0.9 because a smaller value than 0.5 means removing points that are too close to the hyperplane. This implies that in next iterations many clusters can be reactivated. On the other hand, if it is larger than 0.9, it means that almost no clusters can be deactivated. We choose 0.8 because in our preliminary numerical tests we noticed that with a smaller value, many clusters were activated again, which increased the required time per iteration. The range of $\tilde{\Delta}$ is justified by the fact that for all t, the maximum value of $\hat{\Delta}^t$ is 1. We chose 0.1 because the higher the value we choose, the smaller the possibility to eliminate clusters becomes. If chosen smaller, $\hat{\Delta}^t$ and $\hat{\Delta}^{t+1}$ would be very similar and some clusters would be deactivated and reactivated several times.

Because we have m unlabeled points, we can fix at most m unlabeled points, which justifies the range of $B_{\max}$ and the maximum value of γ. Since some points are not fixed on some side—they may be on the wrong side or it could take more than $T_{\max}$ to solve Problem (P5)—we try to fix at least more than 10 % of $B_{\max}$ many unlabeled points. This is why the minimum value of γ is 1.1. The maximum value of $T_{\max}$ is 100 s because, if chosen smaller, we observe that there is often not enough time to solve Problem (P5). On the other hand, if it is larger, we observe that the time needed to solve the Algorithm 3 increases.

Appendix E: Run times in dependence of the number of data points

In this section, we complement Sect. 6 by presenting the run times in dependence of the number of points in the data set in order to shed some light on the scalability of our approaches. To this end, we split the entire data set in three subsets.

The first subset only considers those 46 data sets with $N \leq 500$. As can be seen in Fig. 18 (top), IRCM solves more than 75 % of the instances while CS^{3}VM and WIRCM solve more than 50 %. The second subset contains 11 data sets with $N \in (500, 1500]$. Figure 18 (middle) shows that for these test sets, IRCM solves about 40 % of the instances while CS^{3}VM and WIRCM solve more than 10 %. The last subset contains those 21 data sets with $N > 1500$. Figure 18 (bottom) shows that CS^{3}VM and WIRCM

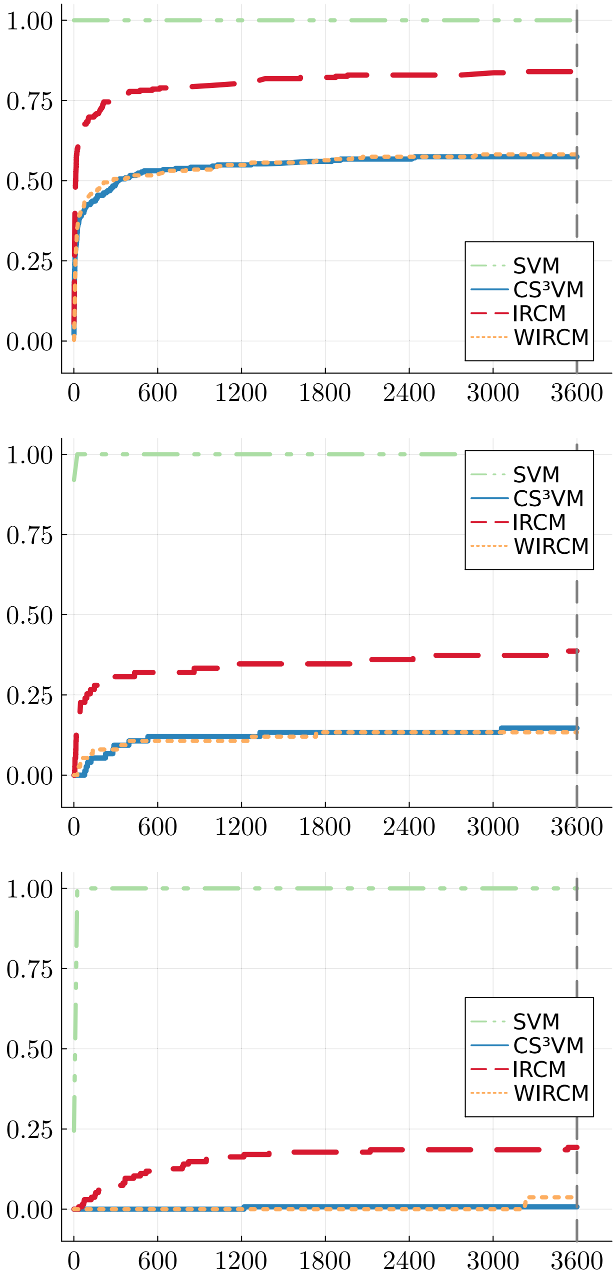

Fig. 18 ECDFs for run time (in seconds). Top: Instances with $N \geq 500$. Middle: Instances with $N \in (500, 1500]$. Bottom: Instances with $N > 1500$

do not solve any of these instances and IRCM solves about 20 %. As expected, the larger the number of points and, thus, the larger the number of binary variables, the more challenging it is to solve the instances. Besides that, SVM solves all instances, which is expected since it does not include any binary variables.

Acknowledgements The authors thank the DFG for their support within RTG 2126 "Algorithmic Optimization".

Funding Open Access funding enabled and organized by Projekt DEAL

Data availability We use the data from the literature that we properly cite.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Almasi ON, Rouhani M (2016) Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets. Turk J Electr Eng Comput Sci 24:219–233. https://doi.org/10.3906/elk-1304-139

Aloise D, Deshpande A, Hansen P, Popat P (2009) NP-hardness of Euclidean sum-of-squares clustering. Mach Learn 75(2):245–248. https://doi.org/10.1007/s10994-009-5103-0

Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434

Bennett KP, Demiriz A (1998) Semi-supervised support vector machines. In: Proceedings of the 11th international conference on neural information processing systems. NIPS'98. MIT Press, Cambridge, pp 368–374. https://proceedings.neurips.cc/paper/1998/file/b710915795b9e9c02cf10d6d2bdb688c-Paper.pdf

Birzhandi P, Youn HY (2019) CBCH (clustering-based convex hull) for reducing training time of support vector machine. J Supercomput 75(8):5261–5279. https://doi.org/10.1007/s11227-019-02795-9

Birzhandi P, Kim KT, Youn HY (2002) Reduction of training data for support vector machine: a survey. Soft Comput 26(8):3729–3742. https://doi.org/10.1007/s00500-022-06787-5

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory. COLT '92. ACM Press, Pittsburgh, pp 144–152. https://doi.org/10.1145/130385.130401

Burgard JP, Krause J, Schmaus S (2021) Estimation of regional transition probabilities for spatial dynamic microsimulations from survey data lacking in regional detail. Comput Stat Data Anal 154:107048. https://doi.org/10.1016/j.csda.2020.107048

Cervantes J, Li X, Yu W (2006) Support vector machine classification based on fuzzy clustering for large data sets, vol 4293. Springer, Berlin, pp 572–582. https://doi.org/10.1007/11925231_54

Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. In: Cowell RG, Ghahramani Z (eds) Proceedings of the tenth international workshop on artificial intelligence and statistics, vol R5. Proceedings of machine learning research. PMLR, pp 57–64. http://proceedings.mlr.press/r5/chapelle05b/chapelle05b.pdf

Chapelle O, Chi M, Zien A (2006) A continuation method for semi-supervised SVMs. In: Proceedings of the 23rd international conference on machine learning. ICML '06. Association for Computing Machinery, New York, pp 185–192https://doi.org/10.1145/1143844.1143868

Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20:273–297. https://doi.org/10.1007/BF00994018

Dasgupta S (2007) The hardness of k-means clustering. https://cseweb.ucsd.edu//~dasgupta/papers/kmeans.pdf

de Almeida MB, de Pádua Braga A, Braga JP (2000) SVM-KM: speeding SVMs learning with a priori cluster selection and k-means. Proceedings of the sixth Brazilian symposium on neural networks 1:162–167. https://doi.org/10.1109/SBRN.2000.889732

Dunning I, Huchette J, Lubin M (2017) JuMP: a modeling language for mathematical optimization. SIAM Rev 59(2):295–320. https://doi.org/10.1137/15M1020575

Hyndman RJ, Fan Y (1996) Sample quantiles in statistical packages. Am Stat 50(4):361–365. https://doi.org/10.2307/2684934

Joachims T (2002) Training transductive support vector machines. In: Learning to classify text using support vector machines. Springer, New York, pp 163–174. https://doi.org/10.1007/978-1-4615-0907-3_9

Kontonatsios G, Brockmeier AJ, Przybyła P, McNaught J, Mu T, Goulermas JY, Ananiadou S (2017) A semi-supervised approach using label propagation to support citation screening. J Biomed Inf 72:67–76. https://doi.org/10.1016/j.jbi.2017.06.018

Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137. https://doi.org/10.1109/TIT.1982.1056489

MacQueen J (1967) Classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp 281–297

Mahajan M, Nimbhorkar P, Varadarajan K (2012) The planar k-means problem is NP-hard. Theor Comput Sci 442:13–21. https://doi.org/10.1016/j.tcs.2010.05.034

Melacci S, Belkin M (2009) Laplacian support vector machines trained in the primal. J Mach Learn Res. arXiv:0909.5422

Olson RS, La Cava W, Orzechowski P, Urbanowicz RJ, Moore JH (2017) PMLB: a large benchmark suite for machine learning evaluation and comparison. BioData Min 10(36):1–13. https://doi.org/10.1186/s13040-017-0154-4

Skinner CJ, D'arrigo, (2011) Inverse probability weighting for clustered nonresponse. Biometrika 98(4):953–966. https://doi.org/10.1093/biomet/asr058

Yao Y, Liu Y, Yu Y, Xu H, Lv W, Li Z, Chen X (2013) K-SVM: an effective SVM algorithm based on K-means clustering. J Comput. https://doi.org/10.4304/jcp.8.10.2632-2639

Yu X, Yang J, Zhan J-P (2012) A transductive support vector machine algorithm based on spectral clustering. AASRI Proc 1:384–388. https://doi.org/10.1016/j.aasri.2012.06.059

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.