Journal homepage http://revistas.unitru.edu.pe/index.php/SSMM

SELECCIONES MATEMÁTICAS
Universidad Nacional de Trujillo
ISSN: 2411-1783 (Online)
2021; Vol. 8(1): 27-36.

A mixed quadratic programming model for a robust support vector machine

Raquel Serna-Diaz, Raimundo Santos Leite and Paulo J. S. Silva

Received, Nov. 04, 2020 *Accepted, Apr. 14, 2021*

How to cite this article:
Serna-Diaz R. et al. *A mixed quadratic programming model for a robust support vector machine.* Selecciones Matemáticas. 2021;8(1):27–36. http://dx.doi.org/10.17268/sel.mat.2021.01.03

Abstract

Support Vector Machines are extensively used to solve classification problems in Pattern Recognition. They deal with small errors in the training data using the concept of soft margin, that allow for imperfect classification. However, if the training data have systematic errors or outliers such strategy is not robust resulting in bad generalization. In this paper we present a model for robust Support Vector Machine classification that can automatically ignore spurius data. We show then that the model can be solved using a high performance Mixed Integer Quadratic Programming solver and present preliminary numerical experiments using real world data that looks promissing.

Keywords . SVM, mixed integer quadratic programming, outliers, classification.

1. Introduction. Support Vector Machines (SVM) are a set of supervised learning methods that is used to build a decision surface that is able to recognize the classes distributed in the space. It achieves this using convex optimization results in order to find a (optimal) decision surface that correctly classifies the training data. In order to avoid overfitting and achieve good generalization an SVM uses two techniques. First, it tries to maximize the distance of the decision surface from the training data. At the same time, it tries to correctly classify the data using a soft margin criterion, which allows part of the training set samples to appear on the wrong side of the separating surface. In this case, all the wrong classifications will have an influence on the determination of the separating surface and consequently on the construction of the decision function. This can limit the quality of generalization of the SVM [1, 2, 3, 4, 5]in the event that the misclassifications come from data with errors, unwanted noise effects, or mislabeling.

In this context, there are several attempts to deal with outliers directly in a given SVM model. For example, in [6] it was considered an SVM model based on an optimization problem whose objective function is convex and which was obtained after relaxing an original model that is not convex. This model will consecutively identify and ignore outliers. Another approach is presented in [7]. It is an SVM model based on the value at risk (VaR) measure and considers an optimization problem that has a nonconvex objective function. The idea is to disregard a percentage of data that should be considered as outliers.

In this work we propose to deal with erroneous data through a modification of the SVM model that is based on Low Order-Value Optimization (LOVO) problem [8]. It will be called the LSVM problem. We will prove that the new model has solutions and will show that it can be solved using a Mixed Integer Quadratic Problem (MIQP) associated to the LSVM problem[9, 10]. Finally, some numerical experiments will be performed, introducing different percentages of artificial outliers in the training data of

*Facultad de Ciencias, Universidad Nacional Agraria la Molina, La molina, Lima, Perú. (rserna@lamolina.edu.pe).
†Instituto de Ciências Exatas e Biológicas , Universidade Federal de Ouro Preto, Campus Universitário Morro do Cruzeiro, CEP: 35400-000, Ouro Preto, MG, Brasil. (raimundo.leite@ufop.edu.br).
‡Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Rua Sérgio Buarque de Holanda, 651 13083-859, Campinas, SP, Brasil. (pjssilva@unicamp.br).

real world data sets available in UCI Machine Learning and Kaggle Repositories. In the experiments, we use the high performance MIQP solver Gurobi [11]. Finally, we close the paper with some conclusions and directions for future research.

2. Optimal separating hyperplane and maximum and soft margin. Let us consider a binary training set T such that

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\} \subseteq (\mathbb{R}^n \times \mathscr{Y}) \tag{2.1}$$

where, each sample (x_i, y_i) is composed by a vector of n features, this is $x_i \in \mathbb{R}^n$, and a label $y_i \in \mathscr{Y} = \{+1, -1\}$ that indicates its classification. The (binary) classification problem consists on finding a decision function or classification rule $f : \mathbb{R}^n \longrightarrow \{+1, -1\}$ that separate the data into two classes and classify correctly new samples belonging to the test set. Samples are assumed to be generated in independently according to an unknown probability distribution $P(x, y)$.

Support vector machines tries to solve the classification problem searching to a hyperplane the succeeds on linearly separating the classes. At the same time it tries to improve generalization by computing a surface that is far away from the training points, creating an empty space between the classes and the linear surface. This area is called the margin of separation. See Figure 2.1. This is achievable solving the following optimization problem:

$$\text{(HSVM)} \qquad \left| \begin{array}{ll} \min\limits_{w,b} & \dfrac{1}{2}\|w\|^2 \\ s.t. & y_i\left[\langle x_i, w\rangle + b\right] \geq 1 \qquad \forall i \in \{1, 2, \ldots, m\}. \end{array} \right.$$

The objective function is associate with the objective of maximizing the margin, which is proportional to $1/\|w\|$, while the constraints enforce the linear separation. This is called the hard margin problem.

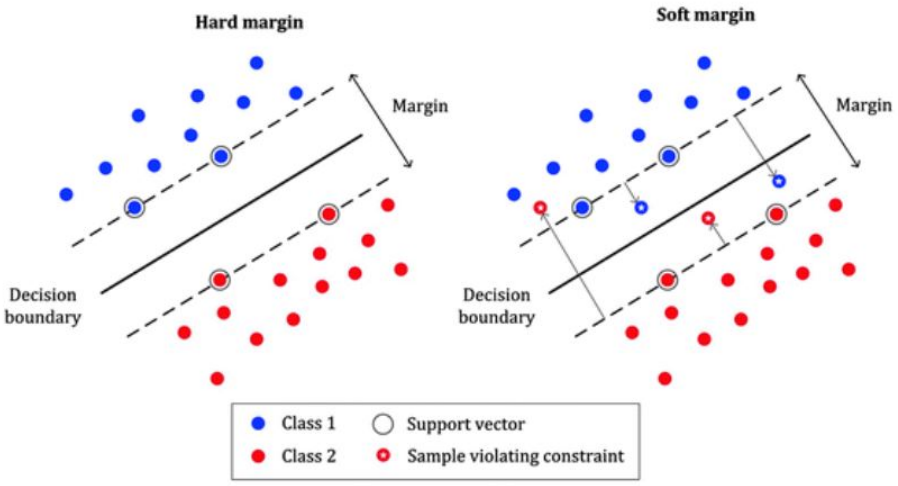

Figure 2.1: In 2D, the decision boundary is simply a line.

However, in many cases, the training points are not linearly separable or can only be fully separated with a small margin. In this situation, the SVM model uses the concept of soft margin: it relaxes the separation constraint in order to improve the margin. To achieve this, we introduce slack variables ξ_i in the separation constraints, that measures by how much the original separation constraint fails to hold. See the right picture in Figure 2.1. More formally, the (soft margin) SVM problem is

$$\text{(SVM)} \qquad \left| \begin{array}{ll} \min\limits_{w,b,\xi} & \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{m} \xi_i \\ s.t. & y_i\left[\langle x_i, w\rangle + b\right] \geq 1 - \xi_i \qquad \forall i \in \{1, 2, \ldots, m\} \\ & \xi_i \geq 0 \qquad \forall i \in \{1, 2, \ldots, m\}, \end{array} \right. ,$$

where $C > 0$ is a penalty parameter which determines the weight between the two terms in the objective function. This parameter weights how flexible the model will be in relation to errors, that is, the smaller the value of C, the more permissive the model will be.

Let (w^*, b^*, ξ^*) is an (SVM) solution, the corresponding decision function turns out to be

$$f(x) = sgn(\langle w^*, x\rangle + b^*).$$

Although it is possible prove directly the existence of a solution for the problem SVM, the existence and calculation of this solution can also be obtained through an associated optimization problem, its dual problem, in a clearer and simpler way due to its special structure 2.2.

Theorem 2.1. *[12, Theorem 2.3.7] The dual associated problem to (SVM) problem is:*

$$\left|\begin{array}{lll} \max\limits_{\lambda} & -\frac{1}{2}\lambda^T Z^T Z\lambda + \sum\limits_{i=1}^{m}\lambda_i & \\ s.t & \sum\limits_{i=1}^{m}\lambda_i y_i = 0, & \\ & 0 \le \lambda_i \le C, & \forall i \in \{1,2,\ldots,m\}, \end{array}\right. \tag{2.2}$$

where Z is the matrix defined as

$$Z := \left[\begin{array}{cccc} y_1 x_1 & y_2 x_2 & \cdots & y_m x_m \end{array}\right].$$

Note that the dual problem is also a quadratic programming problem but with very simple constraints. The constraints are composed only by a box and a simple hyperplane. This is a set where it is easy to project [13]. This fact opens up the path to develop high performance solvers for the dual problem bases on methods like the Spectral Project Gradient method [14] and LIBSVM [15]. In particular, we can prove the following results that shows how to recover an SVM solution from a dual solution.

Proposition 2.1. *[12, Theorem 2.3.9] Consider a training set T as in (2.1) and let be a solution λ^* for the dual problem (2.2), then a (SVM) primal solution (w^*, b^*, ξ^*) can be obtained as follows*

$$w^* = \sum_{i=1}^{m}\lambda_i^* y_i x_i. \tag{2.3}$$

And, if exist a λ^{opt} component λ_j^{opt} such that $\lambda_j^{opt} \in]0, C[$ with $j \in \{1,2,\ldots,m\}$, then

$$b^{opt} = y_j - \langle w^{opt}, x_j\rangle = y_j - \sum_{i=1}^{m}\lambda_i^{opt} y_i \langle x_i, x_j\rangle \tag{2.4}$$

or

$$b^{opt} = \frac{\sum\limits_{j\in J_C} y_j - \langle w^{opt}, x_j\rangle}{n(J_C)} \quad \text{where } J_C = \{j \in \{1,2,\ldots,m\} : 0 < \lambda_j^{opt} < C\}.$$

Besides that

$$\xi_i^{opt} = \left\{\begin{array}{lll} 0 & if \quad i \in \{1,2,\ldots,m\}: & 0 \le \lambda_i^{opt} < C \\ 1 - y_i\left[\langle x_i, w^{opt}\rangle + b^{opt}\right] & if \quad i \in \{1,2,\ldots,m\}: & \lambda_i^{opt} = C. \end{array}\right.$$

3. Ignoring outliers of corrupted data. In this section we are interested in a problem where the training data have some samples with measurement errors or some kind of systematic corruption that should be identified. Note that in this case using simply the soft margin approach is not desirable, as the errors will probably affect the decision boundary in a systematic way. This can limit the generalization quality of the SVM. Our objective is to change the model in such a way that we can ignore the largest errors in order to alleviate this effect.

Therefore, with the intention of eliminating the influence of wrong samples, we propose an SVM model that uses selective sampling in order to control the unwanted effects of the corrupted data. To achieve this we consider only that we have an estimate p of the number of samples that have systematic errors. We start by introducing a notation that will be used throughout the text.

As in (SVM), let $\xi \in \mathbb{R}_+^m$ represent classification errors. In order to ignore the largest errors, we reorder these values as $\xi_{[1]} \ge \xi_{[2]} \ge \ldots \ge \xi_{[m]}$ (decreasing order), where the brackets indicate the new index (used in [16, 17]) necessary for the ordering to hold. In this case, the p largest values will be ignored, so the errors terms to ignore are simply

$$\xi_{[1]}, \xi_{[1]}, \cdots, \xi_{[p]} \quad , \qquad \forall \xi \in \mathbb{R}^m \tag{3.1}$$

Now we can introduce our modified model. Let be a training sample data

$$T = \{(x_{1d}, y_1), (x_2, y_2), \cdots, (x_m, y_m)\} \subseteq (\mathbb{R}^n \times \mathscr{Y}), \qquad \text{where} \qquad \mathscr{Y} = \{+1, -1\} \tag{3.2}$$

and $p \in \mathbb{N}$ such that $p < m$, that is an estimate of the maximum number of erroneous samples to consider.

Ignoring the p largest values of the errors means that the term $\sum_{i=1}^{p} \xi_{[i]}$ will be ignored in the objective function of the soft margin SVM, so only the $m-p$ smallest errors should be minimized in the objective function of the soft margin SVM. Consequently, the primal soft margin problem reformulated for this case is

$$\text{(LSVM)} \qquad \left| \begin{array}{lll} \min\limits_{w,b,\xi} & \frac{1}{2}\|w\|^2 + C \sum\limits_{i=p+1}^{m} \xi_{[i]} & \\ s.t. & y_i\left[\langle x_i, w\rangle + b\right] \geq 1 - \xi_i & \forall i \in \{1,2,\ldots,m\} \\ & \xi_i \geq 0 & \forall i \in \{1,2,\ldots,m\} \end{array} \right.$$

where $\xi_{[1]} \geq \xi_{[2]} \geq \ldots \geq \xi_{[m]}$ and $C > 0$.

Even though the model looks similar to a regular SVM, it has a very different nature. In fact this optimization model is highly nonconvex because the objective function in this case is not convex and also nondifferentiable. This creates clear complications, as the usual methods used to train SVMs can not be applied anymore. In particular, continuous optimization methods can no longer guarantee convergence to global optima, but only to stationary points that are probably only local minima.

Moreover, the usual (Lagrangian) dual problem is probably much less useful, as there is duality gap due to the nonconvexity. This means that we can not resort to methods that solve the dual problem with its simple constraint set. We need to deal directly with the primal formulation.

In the next section we show that we can recast this model as a MIQP problem, opening the path to solve it using high performance MIQP solvers showing that the problem is actually tractable.

4. A MIQP model for LSVM. The objective of this section is to develop a Mixed Integer Quadratic Programming model that is equivalent to LSVM. This will open up the possibility of using a high performance computational solver, like Gurobi [11], to solve it. This can be achieved adding extra variables to bound the slacks ξ together with on-off variables. We do this in the next result.

Theorem 4.1. *Consider the MIQP model*

$$\text{(MLSVM)} \qquad \begin{array}{lll} \min\limits_{w,b,\xi,\zeta,u} & \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{m} \zeta_i & \\ s.t. & y_i(\langle w, x^i\rangle + b) \geq 1 - \xi_i, \ i = 1,\ldots,m, & \\ & \sum\limits_{i=1}^{m} u_i \leq p & \\ & \xi \leq \zeta + Mu & \\ & \xi, \zeta \geq 0, \ u \in \{0,1\}^m, & \end{array}$$

where $M \in \mathbb{R}$. If M is large enough, the components w, b, ξ of any solution to (MLSVM) *also comprise a solution to* (LSVM) *and there is at least one solution of* (LSVM) *that can be extended to a solution of* (MLSVM)*. In particular, both problems have the same optimal value.*

Before proving this theorem, let us prove an auxiliary lemma that will allow us to find what is a good value to the M constant.

Lemma 4.1. *The problem* (LSVM) *has at least one solution.*

Proof: Remember, first, that (SVM) always have a solution [12, Theorem 2.3.2]. Now, let I denote an arbitrary subset of $\{1,\ldots,m\}$ with $m-p$ elements. Let us call SVM_I the associated SVM problem where only the samples with indexes in I are considered. Therefore, SVM_I have a solution for all possible I. Let I^*, denote the index set associated to a SVM_I that has the smallest optimal value among all possible I and let us denote by $w^*, b^*, \xi^*_{I^*}$ its solution. Remember that $\xi^*_{I^*}$ has coordinates only in the index set I^*,this is the reason for its special notation.

Now consider a feasible point (w, b, ξ) in (LSVM) and, as before, let $[1], [2], \ldots [m]$ denotes an order for the indexes of ξ such that $\xi_{[1]} \geq \xi_{[2]} \geq \ldots \geq \xi_{[m]}$. Now, define $I = \{[p+1], [p+2], \ldots [m]\}$. Clearly (w, b, ξ_I) is a feasible point of SVM_I and, hence, its objective value, which coincides with the objective value of (LSVM) for (w, b, ξ), is larger than the optimal value of SVM_{I^*}. This show that the optimal value of (LSVM) is a larger than the optimal value of SVM_{I^*}.

On the other hand, a solution of $(w^*, b^*, \xi^*_{I^*})$ of SVM_{I^*} can be naturally extended to a feasible point of (LSVM). Simply define the coordinates of ξ^* outside I^* as values that are large enough to ensure the validity of the relaxed soft margin constraints and there are also larger than the coordinates already in $\xi^*_{I^*}$. We conclude that this extended solution is also a feasible to (LSVM) with objective equal to the optimal value of SVM_{I^*}. Therefore, the optimal values of SVM_{I^*} and (LSVM) are equal.

We can now proceed to prove the Theorem 4.1.

Proof: Let (w^*, b^*, ξ^*) be a solution to (LSVM). We will prove the result for an (MLSVM) defined with $M = \|\xi^*\|_\infty$. We start, considering a problem which is (LSVM) with an extra constraint:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=p+1}^{m} \xi_{[i]} \\ \text{s.t.} \quad & y_i(\langle w, x^i\rangle + b) \geq 1 - \xi_i,\ i = 1,\dots,m, \\ & \xi \leq M \\ & \xi \geq 0. \end{aligned} \tag{4.1}$$

Observe that (w^*, b^*, ξ^*) is feasible to (4.1). Moreover, since by construction this last problem is a restriction of (LSVM) that has one of its solutions as a feasible point, it has the same optimal value as (LSVM). Hence, it is a feasible problem whose solutions are also solutions to (LSVM) and there is at least one solution to (LSVM), namely (w^*, b^*, ξ^*), which is also a solution to (4.1). That is (4.1) has exactly the properties we want to prove for (MLSVM). Let us study the relationship between these three problems.

First, let us start with a feasible point (w, b, ξ) of (4.1), with associated order $\xi_{[1]} \geq \xi_{[2]} \geq \ldots \geq \xi_{[m]}$. Define

$$\begin{aligned} u_{[1]} := u_{[2]} := \ldots u_{[p]} := 1, \\ u_{[p+1]} := u_{[p+2]} := \ldots = u_{[m]} := 0, \end{aligned}$$

and

$$\begin{aligned} \zeta_{[1]} := \max(0, \xi_{[1]} - M) = 0, \zeta_{[2]} := \max(0, \xi_{[2]} - M) = 0, \ldots, \zeta_{[p]} := \max(0, \xi_{[p]} - M) = 0, \\ \zeta_{[p+1]} := \xi_{[p+1]}, \zeta_{[p+2]} := \xi_{[p+2]}, \ldots, \zeta_{[m]} := \xi_{[m]}. \end{aligned}$$

These equations present a feasible point of (MLSVM) with exactly the same objective value as the original point for (4.1). This implies that the optimal value of (4.1), which is equal to the optimal value of (LSVM), is greater or equal to the optimal value of (MLSVM).

On the other hand, let (w, b, ξ, ζ, u) be feasible for (MLSVM). Redefine u, if necessary, so that $u_i := 1$ whenever ξ is among the p largest values of ξ, and 0 otherwise. Analogously, redefine $\zeta_i := \max(0, \xi_i - Mu_i)$. This new point is also feasible for (MLSVM). Its objective is

$$\frac{1}{2}\|w\|^2 + C\sum_{u_i=0} \xi_i + C\sum_{u_i=1} \max(0, \xi_i - M).$$

The first two terms are the objective of (w, b, ξ) in (LSVM) and the last one is non-negative. Therefore, we conclude the reverse inequality between the optimal value of (LSVM) and (MLSVM). Therefore, the optimal values of (LSVM), (4.1), and (MLSVM) are all equal and the solution (w^*, b^*, ξ^*) can be extended to a solution of (MLSVM) using the construction employed to derive the first inequality.

Finally, let $(w^\star, b^\star, \xi^\star, \zeta^\star, u^\star)$. As above, by redefining $u^\star$ and $\zeta^\star$ if necessary, we can assume that its objective has the form

$$\frac{1}{2}\|w^\star\|^2 + C\sum_{u_i^\star=0} \xi_i^\star + C\sum_{u_i^\star=1} \max(0, \xi_i^\star - M),$$

where the first term is the objective value of $(w^\star, b^\star, \xi^\star)$ in (LSVM). Since the optimal values are equal, this implies that $\max(0, \xi_i^\star - M) = 0$ for the p largest values of $\xi^\star$, that is $\xi^\star \leq M$ and hence $(w^\star, b^\star, \xi^\star)$ is a solution of (4.1) and (LSVM). The proof is complete.

This result opens up the path to use Mixed Integer Quadratic Programming solvers to tackle (MLSVM) and consequently the robust SVM variation (LSVM). This will allow us to avoid spurious local minima that may attract methods that can only ensure local convergence as described in [18].

5. Numerical experiments. In order to test the efficiency of (MLSVM) in alleviating the error introduced by corrupted training data, we have performed some preliminary numerical experiments using the Gurobi solver [11]. Our experiments focused on problems with corrupted labels instead of corrupted

features, as they are less likely to be amenable to treatment by usual statistical methods to identify outliers and spurious measurements.

Even though it is usual in the literature to use artificially generated data do test such modifications [7], we have to chosen to use not to do so in order to test the results under a more realistic setting. We have then selected three test sets that are easily available and are amenable to linear classification.

The first set is the Breast Cancer Wisconsin (Diagnostic) data, that can be obtained at the UCI Machine Learning Repository. This test set is composed of 569 samples with 30 features extracted from images of a breast mass from patients which are labeled as malignant and benign. This is a classical data set that is linearly separable [19, 20].

The next data set is called Ionosphere. It is also available at the UCI repository. It is composed of 350 samples with 34 features representing radar data from electrons in the ionosphere. The patterns are labeled either good (when there is evidence of some kind of structure in the ionosphere) or bad (otherwise) [21].

Finally, the last data set has more samples. It has 1107 samples selected from the Credit Card Fraud Detection data set to balance between the two classes that represent regular and fraudulent credit card transactions. The data is available at Kaggle. The features were extracted using the PCA transformation of the original data to preserve confidentiality, only the first 29 most important PCA features are present [22].

Now, we have to introduce systematic errors in the data. Instead of adding large random noise to some samples like in [7], we preferred to do something more subtle to test the capacity of the (MLSVM) model. We only performed changes in the labels. The idea is to introduce a region of "confusion", where part of the samples that were originally from a class are mislabeled creating regions where nearby samples of both class coexist. The objective is to model regions where the classification is difficult, or situations where the person that is labeling the training data has incentives to mislabel in one direction. This happens, for example, in medical applications where it is dangerous to consider a sick patient as healthy, for example.

To achieve this we start setting the number of samples that will be changed, let us call it p as before. We then randomly selected one sample from each class. For each of these base samples, we sort all the other samples of the same label from the nearest to the farthest flipping the label of $p/2$ every other two samples, starting from the closest one to the farthest from the base sample. This will create two regions of "confusion" in the data with $p/2$ nearby samples with flipped labels close to other samples with correct information. As was said before, what we are trying to access is whether the (MLSVM) model will be less sensitive to the wrong data than the regular (SVM) model.

We have then implemented the MIQP problem representing (MLSVM) using the Python programming language extending the regular support vector machine implementation of the Scikit Learn library [23]. In order to solve the MIQP we used Gurobi version 9.0.2 [11]. The code was run on a computer using 8 cores of an AMD Ryzen Threadripper 1950X CPU, with 64 GB of RAM, and running Ubuntu Linux 20.04. We have also used the Scikit Learn library to perform basic pattern recognition operations like randomly selecting samples, performing classical SVM training, and choosing the C hyper parameter using cross validation with a regular SVM. Remember that the problem is a MIQP with many integer variables. Fortunately, Gurobi was able to find good quality solutions fast. We left each problem running until an optimal solution was found with a timeout of 10 minutes. This was necessary to make the test run in a reasonable amount of time.

Data set	Error	p level	Ideal	SVM	MLSVM
Breast cancer	0%	2%	97.72%	97.72%	**97.81%**
	4%	2%	97.63%	96.75%	**97.11%**
	4%	6%	97.63%	96.75%	97.37%
	7%	5%	97.54%	95.70%	**96.75%**
	7%	9%	97.54%	95.70%	97.63%
	10%	8%	97.46%	94.47%	**96.40%**
	10%	12%	97.46%	94.47%	97.63%
Ionosphere	0%	2%	**86.71%**	**86.71%**	86.57%
	4%	2%	87.00%	85.57%	86.29%
	4%	6%	87.00%	**85.57%**	84.14%

	7%	5%	87.14%	**83.71%**	**83.71%**
	7%	9%	87.14%	83.71%	84.00%
	10%	8%	86.71%	82.29%	82.71%
	10%	12%	86.71%	82.29%	**82.57%**
Credit card	0%	2%	**94.73%**	**94.73%**	94.19%
	4%	2%	94.73%	93.87%	**94.14%**
	4%	6%	94.73%	93.87%	94.46%
	7%	5%	94.73%	93.78%	94.05%
	7%	9%	94.73%	93.78%	**93.87%**
	10%	8%	94.95%	92.30%	93.29%
	10%	12%	94.95%	92.30%	**93.02%**

Table 5.1: Comparison between generalization estimates for (regular) SVM and the robust version MLSVM

The results are described in Table 5.1. The column "Error" has the amount of error that was introduced in the data using the procedure described above. For example, if it says 4%, it means that 4% of the training data had the labels flipped before the training. The column "p level" describes how many samples MLSVM may ignore. Note that we never choose the exact error, because this value is never really known in applications. We only test with a lower and upper estimate to see whether the robust method behaves better in one of these cases.

All the other columns present an estimate for the generalization of the trained machines obtained by the following procedure. We start randomly separating 20% of the data *with the original, correct labels,* to estimate the generalization. The other 80% has "p level" of its labels corrupted as explained above and is then used to train the machines. What we present are the mean values after ten runs of these procedures.

For each test we present the generalization results for three variations of the support vector machine classifier:

1. "Ideal": in the case we first delete all the corrupted data and use the remaining information to train a regular SVM. This simulates what is the best generalization than can be achieved if a method is able to find *exactly* the corrupted data and ignore it.
2. "SVM": a regular SVM is trained using training data that is partially corrupted. This is what would be achieved in a standard method was used: a method that does not try to take into account that the data has errors.
3. "MLSVM": the robust version of the SVM described in Section 4.

The variation with the best generalization for each error and "p-level", that are represented by lines in the table, is either in boldface or in red. The red color indicates that the variation was the best for the given error level considering the two "p levels".

As it can be seen from the results, in the presence of corrupted data, MLSVM is almost always better than a regular SVM when there is some error in the data, i.e. if the error is greater than 0. This can be clearly seen as all red labeled data appears in the MLSVM column showing the effectiveness of the model proposed in this work.

However, at least in one case the improvement is by a small margin: Ionosphere. In this case, which is the one with the lowest generalization level in all tests, the generalization degrades quickly when the error is introduced and the MLSVM model only has a limited capability to alleviate this phenomenon.

By analyzing this results we can see that the extra slack given to MLSVM is sometimes used badly overfitting to the training data instead of improving generalization. In this situation a natural remedy in Machine Learning is to try to use some kind of regularization that imposes a penalty to the overfitting. To achieve this, we changed objective of MLSVM to only ignore a sample if it brings a sensible improvement to the original objective. In other words, a sample will be ignored only if this improves the MLSVM optimization criterion significantly. Formally, let ξ^* be the slack of a solution of the regular SVM in the

training data. Using the ordering notation from (LSVM), we change the original objective of (MLSVM) from

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\zeta_i$$

to

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\zeta_i + RC\left(\frac{\sum_{i=1}^{m-p}\xi^*_{[i]}}{p}\right)\sum_{i=1}^{m}u_i, \tag{5.1}$$

where $R \geq 0$ is a parameter, typically in $(0,1)$. The idea of the extra term is that in a solution it should be interesting to erase the samples associated to the largest p errors only if they represent a gain proportional to the mean error margin of an SVM that does not try to ignore the errors. This change should preclude ignoring samples that marginally improve the objective that we know is just a rough approximation of the actual generalization capacity of the machine.

The new results are then presented in Table 5.2. This table has an extra column named "Reg. MLSVM" for the regularized version with the objective described in (5.1). We also present the previous information to facilitate the comparison.

It can be seen that the regularization was not effective for the first problem, Breast Cancer, where the original MLSVM already is able to compete, and in some cases even outperform, the ideal method that erases only the corrupted samples. Anyhow, in this case, the regularized version also have a very good performance.

Data set	Error	p level	Ideal	SVM	MLSVM	Reg. MLSVM
Breast cancer	0%	2%	97.72%	97.72%	**97.81%**	97.72%
	4%	2%	97.63%	96.75%	**97.11%**	96.75%
	4%	6%	97.63%	96.75%	97.37%	97.19%
	7%	5%	97.54%	95.70%	**96.75%**	95.53%
	7%	9%	97.54%	95.70%	97.63%	97.19%
	10%	8%	97.46%	94.47%	**96.40%**	**96.40%**
	10%	12%	97.46%	94.47%	97.63%	97.11%
Ionosphere	0%	2%	86.71%	86.71%	86.57%	**86.86%**
	4%	2%	87.00%	85.57%	**86.29%**	85.29%
	4%	6%	87.00%	85.57%	84.14%	86.71%
	7%	5%	87.14%	83.71%	83.71%	**85.29%**
	7%	9%	87.14%	83.71%	84.00%	86.14%
	10%	8%	86.71%	82.29%	82.71%	83.57%
	10%	12%	86.71%	82.29%	82.57%	**82.71%**
Credit card	0%	2%	94.73%	**94.73%**	94.19%	94.28%
	4%	2%	94.73%	93.87%	**94.14%**	93.87%
	4%	6%	94.73%	93.87%	94.46%	94.50%
	7%	5%	94.73%	93.78%	94.05%	93.83%
	7%	9%	94.73%	93.78%	93.87%	**94.01%**
	10%	8%	94.95%	92.30%	**93.29%**	93.06%
	10%	12%	94.95%	92.30%	93.02%	93.51%

Table 5.2: Comparison between generalization estimates with the regularized variant of the MLSVM

On the other hand, on the problems Ionosphere and Credit Card the regularization proves to be effective. The improvement is specially high in the Ionosphere problem, which is the most difficult. In this case, it can avoid the degradation of the generalization similarly to what would be achieved with the ideal method up to the corruption level of 7%. Note that the non regularized version have generalization estimates that are 2% lower than what is achieved in the regularized MLSVM.

6. Conclusions. In this work we have introduced a MIQP model that is equivalent to solve a variation of the Support Vector Machine Model that ignores corrupted samples, generating a robust classifier. The new model was originally deduced using ideas from Low-Order Value Optimization that naturally lead to nonconvex and nondifferentiable problems. To avoid the difficulties associated with the solution of such optimization models we developed the MIQP variant that is equivalent to the robust model and can be solved using state-of-the-art solvers like Gurobi.

We then performed initial tests using real world data in the difficult case of flipped labels. The test showed that the solutions obtained from the MIQP model is able to alleviate the loss of generalization when the training uses corrupted data. The initial results are encouraging and suggest that more research should be done in this are.

A major problem with this approach is that the MLSVM model is not convex. This introduces duality gap in the natural dual formulation destroying the nice interrelationship between the primal and dual. This has two main consequences. First, methods of solution bases on the simple constraint structure of the dual are not useful anymore, leaving us with methods to solve the primal problem directly. Second, it is not possible to use kernels to perform nonlinear separation [24]. This is major drawback that appears in previous robust versions of SVM [6, 7].

However, LOVO problems can usually be recast as a difference of convex problem as a LOVO function can be seen as the full sum minus the largest value of up to p entries [8, 18]. We are now trying to explore this fact and the rich literature of difference of convex optimization that presents its rich dual results [25, 26, 27] to try to derive a dual version of (MLSVM) that is amenable to the kernel trick allowing nonlinear separation.

7. Acknowledgments. This research was partially supported by the Brazilian funding agencies CNPq (grant 04301/2019-1) and FAPESP (grants 2013/07375-0, 2018/24293-0).

ORCID and License
Raquel Serna-Diaz https://orcid.org/0000-0003-1213-6063
Raimundo Santos Leite https://orcid.org/0000-0001-7730-9127
Paulo J. S. Silva https://orcid.org/0000-0003-1340-965X

References

[1] Zhan Y Shen D. Increasing Efficiency of SVM by Adaptively Penalizing Outliers. In: Rangarajan A., Vemuri B., Yuille A.L. (eds) Energy Minimization Methods in Computer Vision and Pattern Recognition. EMMCVPR 2005. Lecture Notes in Computer Science, vol 3757. Springer, Berlin, Heidelberg; 2005. p. 539–551.
[2] Seetha H, Narasimha Murty M, Saravanan R. On Improving the Generalization of SVM Classifier. vol. 157. Springer, Berlin, Heidelberg: Computer Networks and Intelligent Computing; 2011.
[3] Hoak J. The Effects of Outliers on Support Vector Machines; Portland; 2010.
[4] Thongkam J, XuYanchun G, Huang ZF. Support VectorMachine for Outlier Detection in Breast Cancer Survivability Prediction. vol. 4977 (99-109). Springer, Berlin, Heidelberg: Advanced Web and Network Technologies, and Applications. Lecture Notes in Computer Science; 2008.
[5] Debruyne M. An Outlier Map for Support Vector Machine Classification. Universiteit Antwerpen: The Annals of Applied Statistics; 2009.
[6] Xu L, Crammer K, Schuurmans D. Robust Support Vector Machine Training via Convex Outlier Ablation. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence. EUA; 2006. p. 536–542.
[7] Tsyurmasto P, Zabarankin M, Uryasev S. Value-at-Risk Support Vector Machine: Stability to Outliers. J. of Combinatorial Optimization; 2014; 28:218–232.
[8] Andreani R, Martínez JM, Martínez L, Yano FS. Low Order-Value. J. of Optimization and Applications. Global Optimization; 2009; 43(01):1-22.
[9] Hess EJ, Brooks JP. The Support Vector Machine and Mixed Integer Linear Programming: Ramp Loss SVM with L1-Norm Regularization. 14th INFORMS Computing Society Conference; 2015.
[10] Anguita D, Ghio A, Pischiutta S, Ridella S. A support vector machine with integer parameters. Neurocom puting; 2008; 72:480-489.
[11] Gurobi Optimization L. Gurobi Optimizer Reference Manual[Internet]; 2020. Available from: http://www.gurobi.com.
[12] Deng N, Tian Y, Zhang C. Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Boca Raton: CRC Press, Taylor & Francis Group; 2013.

[13] Cominetti R, Mascarenhas WF, Silva PJS. A Newton's Method for the Continuous Quadratic Knapsack Problem. Mathematical Programming Computation; 2014; vol. 6:151–169.
[14] Birgin EG, Martínez JM, Raydan M. Nonmonotone Spectral Projected GradientMethods for Convex Sets. SIAM J. on Optimization; 2000; vol. 10:1196–1211.
[15] Chang CC, Lin CJ. LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology; 2011; vol. 2(7):1–27.
[16] Boyd SP, Vandenberghe L. Convex Optimization. Cambridge, UK ; New York: Cambridge University Press; 2004.
[17] Gaudioso M, Gorgone E, Hiriart-Urruty J. Feature selection in SVM via Polyhedral k-norm. Optimization Letters. 2020, (19-36).
[18] Leite RS. Support Vector Machines and the Low Order-Value Optimization. Campinas: Universidad de Campinas; 2019.
[19] Street WN, Wolberg WH, Mangasarian OL. Nuclear Feature Extraction for Breast Tumor Diagnosis. In: Biomedical Image Processing and Biomedical Visualization. vol.39 1905-07-29. International Society for Optics and Photonics; 1993; p. 861-870.
[20] Mangasarian OL, Street WN, Wolberg WH. Breast Cancer Diagnosis and Prognosis Via Linear Programming. Operations Research; 1995; vol. 43(08):570-577.
[21] Sigillito VG,Wing SP, Hutton LP, Baker KB. Classification of Radar Returns from the Ionosphere Using Neural Networks, 19. Johns Hopkins APL Technical Digest; 1989.
[22] Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Transactions on Neural Networks and Learning Systems; 2018; vol. 29:3784–3797.
[23] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-Learn: Machine Learning in Python. J. of Machine Learning Research; 2011; vol. 12. Available from: http://jmlr.org/papers/v12/ pedregosa11a.html.
[24] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods. Cambridge ; New York: Cambridge University Press; 2000.
[25] Horst R, Thoai NV. DC Programming: Overview. J. of Optimization Theory and Applications; 1999; vol. 103:1–43.
[26] Martinez-Legaz JE, VolleM. Duality in D.C. Programming: The Case of Several D.C. Constraints. J. of Mathematical Analysis and Applications; 1999; vol. 237:657–671.
[27] Tao PD, An LTH. Convex Analysis Approach to D. C. Programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica. 1997; vol. 22:289–355.