Linear Support Vector Regression with Linear Constraints

Quentin Klopfenstein, Samuel Vaiter

LINEAR SUPPORT VECTOR REGRESSION WITH LINEAR CONSTRAINTS*

QUENTIN KLOPFENSTEIN† AND SAMUEL VAITER‡

Abstract. This paper studies the addition of linear constraints to the Support Vector Regression (SVR) when the kernel is linear. Adding those constraints into the problem allows to add prior knowledge on the estimator obtained, such as finding probability vector or monotone data. We propose a generalization of the Sequential Minimal Optimization (SMO) algorithm for solving the optimization problem with linear constraints and prove its convergence. Then, practical performances of this estimator are shown on simulated and real datasets with different settings: non negative regression, regression onto the simplex for biomedical data and isotonic regression for weather forecast.

Key words. Support Vector Machine, Support Vector Regression, Sequential Minimal Optimization

AMS subject classifications. 90C25, 49J52

1. Introduction.

The Support Vector Machine (SVM) [3] is a class of supervised learning algorithms that have been widely used in the past 20 years for classification tasks and regression. These algorithms rely on two main ideas: the first one is the maximum margin hyperplane which consists in finding the hyperplane that maximises the distance between the vectors that are to be classified and the hyperplane. The second idea is the kernel method that allows the SVM to be used to solve non-linear problems. The technic is to map the vectors in a higher dimensional space which is done by using a positive definite kernel, then a maximum margin hyperplane is computed in this space which gives a linear classifier in the high dimensional space. In general, it leads to a non-linear classifier in the original input space.

From SVM to Support Vector Regression. Different implementations of the algorithms haven been proposed such as C-SVM, ν-SVM [34], Least-Squares SVM [37], Linear Programming SVM [12] among others. Each of these versions have their strenghs and weaknesses depending on which application they are used. They differ in terms of constraints considered for the hyperplane (C-SVM and Least-Squares SVM), in terms of norm considered on the parameters (C-SVM and Linear Programming SVM) and in terms of optimization problem formulation (C-SVM and ν-SVM). Overall, these algorithms are a great tool for classification tasks and they have been used in many different applications like facial recognition [18], image classification [7], cancer type classification [15], text categorization [19] to only cite a few examples. Even though, SVM was first developped for classification, an adaptation for regression estimation was proposed in [11] under the name Support Vector Regression (SVR). In this case, the idea of maximum margin hyperplane is slightly changed into finding a tube around the regressors. The size of the tube is controlled by a hyperparameter chosen by the user: ϵ. This is equivalent to using an ϵ-insensitive loss function, $|y - f(x)|_\epsilon = \max\{0, |y - f(x)| - \epsilon\}$ which only penalizes the error above the chosen ϵ level. As for the classification version of the algorithm, a ν-SVR method exists. In this

*Submitted.

Funding: This work was partly supported by ANR GraVa ANR-18-CE40-0005, Projet ANER RAGA G048CVCRB-2018ZZ and INSERM Plan cancer 18CP134-00.

†Institut Mathématique de Bourgogne, Université de Bourgogne, Dijon, France (quentin.klopfenstein@u-bourgogne.fr).

‡CNRS & Institut Mathématique de Bourgogne, Université de Bourgogne, Dijon, France (samuel.vaiter@u-bourgogne.fr).

version, the hyperparameter ϵ is computed automatically but a new hyperparameter ν has to be chosen by the user which controls asymptotically the proportions of support vectors [34]. SVR has proven to be a great tool in the field of function estimation for many different applications: predicting times series in stock trades [40], travel-time prediction [8] and for estimating the amount of cells present inside a tumor [29].

Incorporating priors. In this last example of application, the authors used SVR to estimate a vector of proportions, however the classical SVR estimator does not take into account the information known about the space in which the estimator lives. Adding this prior information on the estimator may lead to better estimation performance. Incorporating information in the estimation process is a wide field of studies in statistical learning (we refer to Figure 2 in [22] for a quick overview in the context of SVM). A growing interest in prior knowledge incorporated as regularization terms has emerged in the last decades. Lasso [38], Ridge [16], elastic-net [42] regression are examples of regularized problem where a prior information is used to fix an ill-posed problem or an overdetermined problem. The ℓ_1 regularization of the Lasso will force the estimator to be sparse and bring statistical guarantees of the Lasso estimator in high dimensional settings. Another commun way to add prior knowledge on the estimator is to add constraints known a-priori on this estimator. The most commun examples are the ones that constrain the estimator to live in a subspace such as Non Negative Least Squares Regression (NNLS) [23], isotonic regression [2]. These examples belong to a more general type of constraints: linear constraints. Other types of constraints exist like constraints on the derivative of the function that is to be estimated, smoothness of the function for example. Adding those constraints on the Least Squares estimator has been widely studied [2, 24, 4] and similar work has been done for the Lasso estimator in [13]. Concerning the SVR, inequality and equality constraints added as prior knowledge were studied in [22]. In this paper, the authors described a method for adding linear constraints on the Linear Programming SVR [12]. This implementation of the algorithm considers the ℓ_1 norm of the parameters in the optimization problem instead of the classical ℓ_2 norm which leads to a linear programming optimization problem to solve instead of a quadratic programming problem. They also described a method for using information about the derivative of the function that is estimated.

Sequential Minimal Optimization. One of the main challenges of adding these constraints is that it often increases the difficulty of solving the optimization problem related to the estimator. For example, the Least Squares optimization problem has a closed form solution whereas the NNLS uses sophisticated algorithms [4] to approach the solution. SVM and SVR algorithms were extensively studied and used in practise because very efficient algorithms were developped to solve the underlying optimization problems. One of them is called Sequential Minimal Optimization (SMO) [32] and is based on a well known optimization technic called coordinate descent. The idea of the coordinate descent is to break the optimization problem into sub-problems selecting one coordinate at each step and minimizing the function only via this chosen coordinate. The developpement of parallel algorithms have increased the interest in these coordinate descent methods which show to be very efficient for large scale problems. One of the key settings for the coordinate descent is the choice of the coordinate at each step, the choice's strategy will affect the efficiency of the algorithm. There exists three families of strategies for coordinate descent: cyclic [39], random [28] and greedy. The SMO algorithm is a variant of a greedy coordinate descent [41] and is the algorithm implemented in LibSVM [6]. It is very efficient to solve SVM/SVR optimization problems. In the context of linear kernel, other algorithm are used such

as dual coordinate descent [17] or trust region newton methods [25].

Priors and SMO. In one of the application of SVR cited above, information a-priori about the estimator is not used in the estimation process and is only used in a post-processing step. This application comes from the cancer research field, where regression algorithms have been used to estimate the proportions of cell populations that are present inside a tumor (see [27] for a survey). Several estimators have been proposed in the biostatistics litterature, most of them based on constrained least squares [1, 33, 14] but the gold standard is the estimator based on the Support Vector Regression [29]. Our work is motivated by incorporating the fact that the estimator for this application belongs to the simplex: $\mathcal{S} = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x_i \geq 0\}$ in the SVR problem. We believe that for this application, it will lead to better estimation performance. From an optimization point of view, our motivation is to find an efficient algorithm that is able to solve the SVR optimization problem where generic linear constraints is added to the problem as prior knowledge, including simplex prior as described. This work follows the one from [22] except that in our case, we keep the ℓ_2 norm on the parameters in the optimization problem which is the most commun version of the SVR optimization problem and we only focus on inequality and equality constraints as prior knowledge.

Contributions. In this paper, we study a linear SVR with linear constraints optimization problem. We show that the dual of this new problem shares similar properties with the classical ν-SVR optimization problem (Proposition 2.2). We also prove that adding linear constraints to the SVR optimization problem does not change the nature of its dual problem, in the fact that the problem stays a semi-definite positive quadratic function subject to linear constraints. We propose a generalized SMO algorithm that allows the resolution of the new optimization problem. We show that the updates in the SMO algorithm keep a closed form (Definition 3.5) and prove the convergence of the algorithm to a solution of the problem (Theorem 3.7). We illustrate on synthetic and real datasets the usefulness of our new regression estimator under different regression settings: non-negative regression, simplex regression and isotonic regression.

Outline. The article proceeds as follows: we introduce the optimization problem coming from the classical SVR and describe the modifications brought by adding linear constraints in section 2. We then present the SMO algorithm, its generalization for solving constrained SVR and present our result on the convergence of the algorithm in section 3. In section 4, we use synthetic and real datasets on different regression settings to illustrate the practical performance of the new estimator.

Notations. We write $||.||$ (resp. $\langle .,. \rangle$) for the euclidean norm (resp. inner product) on vectors. We use the notation $X_{:i}$ (resp. $X_{i:}$) to denote the vector corresponding the the ith column of the matrix X (resp. i^{th} row of the matrix X). Throughout this paper, the design matrix will be $X \in \mathbb{R}^{n \times p}$ and $y \in \mathbb{R}^n$ will be the response vector. X^T will be used for the transposed matrix of X. The vector $\mathbf{e}$ denote the vector with only ones on each of its coordinates and e_j denotes the canonical vector with a one at the jth coordinate. $\nabla_{x_i} f$ is the partial derivative $\frac{\partial f}{\partial x_i}$.

2. Constrained Support Vector Regression.

First we introduce the optimzation problem related to adding linear constraints to the SVR and discuss some interesting properties about this problem.

2.1. Previous work : ν-Support Vector Regression. The ν-SVR estimator [34] is obtained solving the following quadratic optimization problem:

$$\text{(SVR-P)} \qquad \begin{aligned} \min_{\beta,\beta_0,\xi_i,\xi_i^*,\epsilon} \quad & \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\ \text{subject to} \quad & y_i - \beta^T X_{i:} - \beta_0 \leq \epsilon + \xi_i \\ & \beta^T X_{i:} + \beta_0 - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, \epsilon \geq 0. \end{aligned}$$

By solving problem (SVR-P), we seek a linear function $f(x) = \beta^T x + \beta_0$ where $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$, that is at most ϵ deviating from the response vector coefficient y_i. This function does not always exist which is why slack variables $\xi \in \mathbb{R}^n$ and $\xi^* \in \mathbb{R}^n$ are introduced in the optimization problem to allow some observations to break the condition given before. C and ν are two hyperparameters. $C \in \mathbb{R}$ controls the tolerated error and $\nu \in [0, 1]$ controls the number of observations that will lay inside the tube of size 2ϵ given by the two first constraints in (SVR-P). It can be seen as an ϵ-insensitive loss function where a linear penalization is put on the observations that lay outside the tube and the observations that lay inside the tube are not penalized (see [36] for more details).

The different algorithms proposed to solve (SVR-P) often use its dual problem like in [32, 17]. The dual problem is also a quadratic optimization problem with linear constraints but its structure allows an efficient resolution as we will see in more details in section 3. The dual problem of (SVR-P) is the following optimization problem:

$$\text{(SVR-D)} \qquad \begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + y^T(\alpha - \alpha^*) \\ \text{subject to} \quad & 0 \leq \alpha_i, \alpha_i^* \leq \frac{C}{n} \\ & \mathbf{e}^T(\alpha + \alpha^*) \leq C\nu \\ & \mathbf{e}^T(\alpha - \alpha^*) = 0, \end{aligned}$$

where $Q = XX^T \in \mathbb{R}^{2n\times 2n}$.

The equation link between (SVR-P) and (SVR-D) is given by the following formula:

$$\beta = -\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:}.$$

2.2. The constrained optimization problem. We propose a constrained version of problem (SVR-P) that allows the addition of prior knowledge on the linear function f that we seek to estimate. The constrained estimator is obtained solving the optimization problem:

$$
\text{(LSVR-P)} \quad \begin{aligned}
\min_{\beta,\beta_0,\xi_i,\xi_i^*,\epsilon} \quad & \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\
\text{subject to} \quad & \beta^T X_{i:} + \beta_0 - y_i \leq \epsilon + \xi_i \\
& y_i - \beta^T X_{i:} - \beta_0 \leq \epsilon + \xi_i^* \\
& \xi_i, \xi_i^* \geq 0, \epsilon \geq 0 \\
& A\beta \leq b \\
& \Gamma\beta = d,
\end{aligned}
$$

where $A \in \mathbb{R}^{k_1 \times p}$, $\Gamma \in \mathbb{R}^{k_2 \times p}$, $\beta \in \mathbb{R}^p$, ξ, $\xi^* \in \mathbb{R}^n$ and β_0, ϵ, $\in \mathbb{R}$.

The algorithm that we propose in section 3 also uses the structure of the dual problem of (LSVR-P). The next proposition introduces the dual problem and some of its properties.

PROPOSITION 2.1. *If the set* $\{\beta \in \mathbb{R}^n, A\beta \leq b, \Gamma\beta \ = d\}$ *is not empty then,*

1. *Strong duality holds for* (LSVR-P).
2. *The dual problem of* (LSVR-P) *is*

(LSVR-D)

$$
\begin{aligned}
\min_{\alpha,\alpha^*,\gamma,\mu} \quad & \frac{1}{2}\Big[(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + \gamma^T AA^T\gamma + \mu^T\Gamma\Gamma^T\mu \\
& +2\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} - 2\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} - 2\gamma^T A\Gamma^T\mu\Big] \\
& +y^T(\alpha-\alpha^*) + \gamma^T b - \mu^T d \\
\textit{subject to} \quad & 0 \leq \alpha_i^{(*)} \leq \frac{C}{n} \\
& \boldsymbol{e}^T(\alpha+\alpha^*) \leq C\nu \\
& \boldsymbol{e}^T(\alpha-\alpha^*) = 0 \\
& \gamma_j \geq 0.
\end{aligned}
$$

3. *The equation link between primal and dual is*

$$\beta = -\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu.$$

The proof of the first statement of the proposition is given in the discussion below whereas the proofs for the two other statements are given in the Appendix A. We have that α, $\alpha^* \in \mathbb{R}^n$, $\gamma \in \mathbb{R}^{k_1}$ is the vector of Lagrange multipliers associated the the inequality constraint $A\beta \leq b$ which explains the non-negative constraints on its coefficients. $\mu \in \mathbb{R}^{k_2}$ are the Lagrange multipliers associated to the equality constraint $\Gamma\beta = d$ which also explains that there is no constraints in the dual problem on μ. The objective function f which we will write in the stacked form as:

$$f(\theta) = \theta^T\bar{Q}\theta + l^T\theta,$$

where

$$\theta = \begin{bmatrix}\alpha\\ \alpha^*\\ \gamma\\ \mu\end{bmatrix},\ l = \begin{bmatrix}y\\ -y\\ b\\ -d\end{bmatrix} \in \mathbb{R}^{2n+k_1+k_2},\ \bar{Q} = \begin{bmatrix} Q & -Q & XA^T & -X\Gamma^T \\ -Q & Q & -XA^T & X\Gamma^T \\ AX^T & -AX^T & AA^T & -A\Gamma^T \\ -\Gamma X^T & \Gamma X^T & -\Gamma A^T & \Gamma\Gamma^T \end{bmatrix}$$

is a square matrix of size $2n + k_1 + k_2$.

An important observation is that this objective function is always convex. The matrix $\bar{Q}$ is the product of the matrix $\begin{bmatrix} X \\ -X \\ A \\ -\Gamma \end{bmatrix}$ and its transpose matrix. It means that $\bar{Q}$ is a Gramian matrix and it is positive semi-definite which implies that f is convex. The problem (LSVR-D) is then a quadratic programming optimization problem which meets Slater's condition if there exists a θ that belongs to the feasible domain which we will denote by $\mathcal{F}$. If there is such a θ we have strong duality holding between problem (LSVR-P) and (LSVR-D). The only condition we need to have on A and Γ is that they define a non-empty polyhedron in order to be able to solve the optimization problem.

Our second observation on problem (LSVR-D) is that the inequality constraints $\mathbf{e}^T(\alpha + \alpha^*) \leq C\nu$ is replaced by an equality constraints in the same way that it was suggested in [5] for the classical problem (SVR-D).

Proposition 2.2. *If $\epsilon > 0$, all optimal solutions of (LSVR-D) satisfy*

1. $\alpha_i \alpha_i^* = 0$, $\forall i$
2. $\boldsymbol{e}^T(\alpha + \alpha^*) = C\nu$

The proof is given in Appendix B. This observation will be important for the algorithm that we propose in section 3.

3. Generalized Sequential Minimal Optimization. In this section we propose a generalization of the SMO algorithm [32] to solve problem (LSVR-D) and present our main result on the convergence of the proposed algorithm to the solution of (LSVR-D). The SMO algorithm is a variant of greedy coordinate descent [41] taking into consideration non-separable constraints, which in our case are the two equality constraints. We start by describing the previous algorithm that solve (SVR-D).

3.1. Previous work : Sequential Minimal Optimization. In this subsection, we define $f(\alpha, \alpha^*) = \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + y^T(\alpha - \alpha^*)$ and we note $\nabla f \in \mathbb{R}^{2n}$ its gradient. From [20], we rewrite the Karush-Kuhn-Tucker (KKT) conditions in the following way:

$$\min_{i \in I_{\text{up}}} \nabla_{\alpha_i} f \geq \max_{j \in I_{\text{low}}} \nabla_{\alpha_j} f \tag{3.1}$$

where

$$I_{\text{up}}(\alpha) = \{i \in \{1, \ldots, l\} : \alpha_i < \frac{C}{l}\}$$

$$I_{\text{low}}(\alpha) = \{i \in \{1, \ldots, l\} : \alpha_i > 0\}.$$

The same condition is written for the α^* variables replacing α_i by α_i^* above. These conditions leads to an important definition for the rest of this paper.

Definition 3.1. *We will say that (i, j) is a violating pair of variables if one of these two conditions is satisfied:*

$$i \in I_{up}(\alpha), j \in I_{low}(\alpha) \text{ and } \nabla_{\alpha_i} f < \nabla_{\alpha_j} f$$
$$i \in I_{low}(\alpha), j \in I_{up}(\alpha) \text{ and } \nabla_{\alpha_i} f > \nabla_{\alpha_j} f.$$

Because the algorithm SMO does not provide in general an exact solution in a finite number of steps there is a need to relax the optimality conditions which gives a new definition.

DEFINITION 3.2. *We will say that (i, j) is a τ-violating pair of variables if one of these two conditions is satisfied:*

$$i \in I_{up}(\alpha), j \in I_{low}(\alpha) \text{ and } \nabla_{\alpha_i} f < \nabla_{\alpha_j} f - \tau$$
$$i \in I_{low}(\alpha), j \in I_{up}(\alpha) \text{ and } \nabla_{\alpha_i} f > \nabla_{\alpha_j} f + \tau.$$

The SMO algorithm will then choose at each iteration a pair of violating variables in the α block or in the α^* block. Once the choice is done, a subproblem of size two is solved, considering that only the two selected variables are to be minimized in problem (SVR-D). The outline of the algorithm is presented in Algorithm 3.1.

The choice of the violating pair of variables presented in [21] was to always work with the most violating pairs of variables, which means the variables that leads to the largest gap compared to the optimality conditions given in (3.1). This choice is what makes a link with greedy coordinate descent, however greedy here is related to the largest gap with the optimality score and is not related to the largest decrease in the objective function.

The resolution of the subproblem of size two has a closed form. The idea is to use the two equality constraints to go from a problem of size two to a problem of size one. Then, the goal is to minimize a quadratic function of one variable under box constraints which is done easily. We will give more details of the resolution of these subproblems in subsection 3.3 for our proposed algorithm.

The proof of convergence of SMO algorithm was given in [20] without convergence rate. The proof relies on showing that the sequence defined by the algorithm $f(\alpha^k, (\alpha^*)^k)$ is a decreasing sequence and that there cannot be the same violating pair of variables infinitely many times. The linear convergence rate was proved later by Schmidt and She [35] as well as the identification of the support vectors in finite time.

3.2. Optimality conditions for the constrained SVR. In this subsection we define f as the objective function of problem (LSVR-D) and $\nabla f \in \mathbb{R}^{2n+k_1+k_2}$ its gradient. The Lagrangian of optimization problem (LSVR-D) is defined by :

$$L = f - \sum_{i=1}^{n}(\lambda_i \alpha_i + \lambda_i^* \alpha_i^*) + \sum_{i=1}^{n} \beta_i(\alpha_i - \frac{C}{n}) + \beta_i^*(\alpha_i^* - \frac{C}{n})$$
$$- \sigma(\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) - C\nu) - \delta \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) - \sum_{j=1}^{k_1} \eta_j \gamma_j.$$

We then give KKT conditions for each block of variables:

Algorithm 3.1 SMO algorithm

Require: $\tau > 0$
  Initializing $\alpha^0 \in \mathbb{R}^n$, $(\alpha^*)^0 \in \mathbb{R}^n$ in $\mathcal{F}$ and set $k = 0$
  while $\Delta > \tau$ **do**
    $i \leftarrow \underset{i \in I_{\text{up}}}{\text{argmin}} \nabla_{\alpha_i} f \quad j \leftarrow \underset{i \in I_{\text{low}}}{\text{argmax}} \nabla_{\alpha_j} f$
    $i^* \leftarrow \underset{i \in I^*_{\text{up}}}{\text{argmin}} \nabla_{\alpha^*_i} f \quad j^* \leftarrow \underset{i \in I^*_{\text{low}}}{\text{argmax}} \nabla_{\alpha^*_j} f$
    $\Delta_1 \leftarrow \nabla_{\alpha_j} f - \nabla_{\alpha_i} f$
    $\Delta_2 \leftarrow \nabla_{\alpha^*_j} f - \nabla_{\alpha^*_i} f$
    $\Delta \leftarrow \max(\Delta_1, \Delta_2)$ ▷ Select the maximal violating pair

    if $\Delta = \Delta_1$ **then**
      $\alpha^{k+1} \leftarrow$ Solution of subproblem for variables α_i and α_j

    else
      $(\alpha^*)^{k+1} \leftarrow$Solution of subproblem for variables α_{i^*} and α_{j^*}

    $k \leftarrow k + 1$
  return $\alpha^k, (\alpha^*)^k$

The α block

$$\begin{aligned} \nabla_{\alpha_i} L = \nabla_{\alpha_i} f - \lambda_i + \beta_i - \sigma - \delta &= 0 \\ \lambda_i \alpha_i &= 0 \\ \beta_i(\alpha_i - \frac{C}{n}) &= 0 \\ \lambda_i &\geq 0 \\ \beta_i &\geq 0 \end{aligned}$$

We will consider different possiblities of value for α_i.

Case 1- $\alpha_i = 0$ then $\beta_i = 0$ and $\lambda_i \geq 0$

$$\nabla_{\alpha_i} f - \sigma - \delta \geq 0$$

Case 2- $\alpha_i = \frac{C}{n}$ then $\lambda_i = 0$ and $\beta_i \geq 0$

$$\nabla_{\alpha_i} f - \sigma - \delta \leq 0$$

Case 3- $0 < \alpha_i < \frac{C}{n}$ then $\beta_i = 0$, $\theta_i = 0$

$$\nabla_{\alpha_i} f - \sigma - \delta = 0$$

We then consider the set of indices :

$$I_{\text{up}}(\alpha) = \{i \in \{1, \ldots, n\} : \alpha_i < \frac{C}{n}\}$$

$$I_{\text{low}}(\alpha) = \{i \in \{1, \dots, n\} : \alpha_i > 0\}$$

The optimality conditions are satisfied if and only if

$$\min_{i \in I_{\text{up}}} \nabla_{\alpha_i} f \geq \max_{j \in I_{\text{low}}} \nabla_{\alpha_j} f.$$

The α^* block In this block, the conditions are very similar to the ones given for the block α, the only difference here is that we will have two new sets of indices:

$$I^*_{\text{up}}(\alpha^*) = \{i \in \{1, \dots, n\} : \alpha^*_i < \frac{C}{n}\}$$

and

$$I^*_{\text{low}}(\alpha) = \{i \in \{1, \dots, n\} : \alpha^*_i > 0\}$$

which gives the following optimality condition:

$$\min_{i \in I^*_{\text{up}}} \nabla_{\alpha^*_i} f \geq \max_{j \in I^*_{\text{low}}} \nabla_{\alpha^*_j} f.$$

The γ block

$$\nabla_{\gamma_j} L = \nabla_{\gamma_j} f - \eta_j = 0 \eta_j \gamma_j = 0 \eta_j \geq 0$$

We will consider different possiblities of value for γ_j.

Case 1- $\gamma_j = 0$ then

$$\nabla_{\gamma_j} f \geq 0$$

Case 2- $\gamma_j > 0$

$$\nabla_{\gamma_j} f = 0$$

Definition 3.3. *We will say that j is a τ-violating variable for the block γ if*

$$\nabla_{\gamma_j} f + \tau < 0.$$

The μ block

$$\nabla_{\mu_j} L = \nabla_{\mu_j} f = 0$$

Definition 3.4. *We will say that j is a τ-violating variable for the block μ if*

$$|\nabla_{\mu_j} f| > \tau.$$

From these conditions on each block, we build an optimization strategy that follows the idea of the SMO described in subsection 3.1. For each block of variables, we

Algorithm 3.2 Generalized SMO algorithm

Require: $\tau > 0$
Initializing $\alpha^0 \in \mathbb{R}^n$, $(\alpha^*)^0 \in \mathbb{R}^n$, $\gamma^0 \in \mathbb{R}^{k_1}$ and $\mu^0 \in \mathbb{R}^{k_2}$ in $\mathcal{F}$ and set $k = 0$
while $\Delta > \tau$ **do**

$i \leftarrow \underset{i \in I_{\text{up}}}{\text{argmin}} \nabla_{\alpha_i} f$ $\quad$ $j \leftarrow \underset{i \in I_{\text{low}}}{\text{argmax}} \nabla_{\alpha_j} f$

$i^* \leftarrow \underset{i \in I^*_{\text{up}}}{\text{argmin}} \nabla_{\alpha^*_i} f$ $\quad$ $j^* \leftarrow \underset{i \in I^*_{\text{low}}}{\text{argmax}} \nabla_{\alpha^*_j} f$

$\Delta_1 \leftarrow \nabla_{\alpha_j} f - \nabla_{\alpha_i} f$ $\quad$ $\Delta_2 \leftarrow \nabla_{\alpha^*_j} f - \nabla_{\alpha^*_i} f$

$\Delta_3 \leftarrow - \underset{j \in \{1,\dots,k_1\}}{\min} \nabla_{\gamma_j} f$ $\quad$ $\Delta_4 \leftarrow \underset{j \in \{1,\dots,k_2\}}{\max} |\nabla_{\mu_j} f|$

$\Delta \leftarrow \max(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$ $\quad$ ▷ Select the maximal violating variables

if $\Delta = \Delta_1$ **then**
$\alpha^{k+1} \leftarrow$Solution of subproblem for variables α_i and α_j

else if $\Delta = \Delta_2$ **then**
$(\alpha^*)^{k+1} \leftarrow$Solution of subproblem for variables α_{i^*} and α_{j^*}

else if $\Delta = \Delta_3$ **then**
$u = \underset{i \in \{1,\dots,k_1\}}{\text{argmin}} \nabla_{\gamma_i} f$
$\gamma^{k+1} \leftarrow$Solution of subproblem for variable γ_u

else
$u = \underset{i \in \{1,\dots,k_2\}}{\text{argmax}} \nabla_{\mu_i} f$
$\mu^{k+1} \leftarrow$ Solution of subproblem for variable μ_u

$k \leftarrow k + 1$
return $\alpha^k, (\alpha^*)^k, \gamma^k, \mu^k$

compute what we call a *violating optimality score* based on the optimality conditions given above. Once the scores are computed for each block, we select the block which has the largest score and solve an optimization subproblem in the block selected. If the block α or the block α^* is selected, we will update a pair of variables by solving a minization problem of size two. However if the block γ or the block μ is selected, we will update only one variable at a time. This is justified by the fact that the variables α and α^* have non-separable equality constraints linking them together. The rest of this section will be dedicated to the presentation of our algorithm and to giving some interesting properties such as a closed form for updates on each of the blocks and a convergence theorem.

3.3. Updates rules and convergence. The first definition describes the closed form updates for the different blocks of variables.

Definition 3.5. *The update between iterate k and iterate $k+1$ of the generalized SMO algorithm has the following form:*

1. *if the block α is selected and (i, j) is the most violating pair of variable then*

the update will be as follows:

$$\alpha_i^{k+1} = \alpha_i^k + t^*$$
$$\alpha_j^{k+1} = \alpha_j^k - t^*,$$

where $t^* = \min(\max(I_1, -\frac{(\nabla_{\alpha_i} f - \nabla_{\alpha_j} f)}{(Q_{ii} - 2Q_{ij} + Q_{jj})}), I_2)$ *with* $I_1 = \max(-\alpha_i^k, \alpha_j^k - \frac{C}{n})$ *and* $I_2 = \min(\alpha_j^k, \frac{C}{n} - \alpha_i^k)$.

2. *if the block* α^* *is selected and* (i^*, j^*) *is the most violating pair of variable then the update will be as follows:*

$$(\alpha_i^*)^{k+1} = (\alpha_i^*)^k + t^*$$
$$(\alpha_j^*)^{k+1} = (\alpha_j^*)^k - t^*,$$

where $t^* = \min(\max(I_1, -\frac{(\nabla_{\alpha_i^*} f - \nabla_{\alpha_j^*} f)}{(Q_{ii} - 2Q_{ij} + Q_{jj})}), I_2)$ *with* $I_1 = \max(-(\alpha_i^*)^k, (\alpha_j^*)^k - \frac{C}{n})$ *and* $I_2 = \min((\alpha_j^*)^k, \frac{C}{n} - (\alpha_i^*))^k$.

3. *if the block* γ *is selected and* i *is the index of the most violating variable in this block then the update will be as follows:*

$$\gamma_i^{k+1} = \max(-\frac{\nabla_{\gamma_i} f}{(AA^T)_{ii}} + \gamma_i^k, 0).$$

4. *if the block* μ *is selected and* i *is the index of the most violating variable in this block then the update will be as follows:*

$$\mu_i^{k+1} = -\frac{\nabla_{\mu_i} f}{(\Gamma\Gamma^T)_{ii}} + \mu_i^k.$$

This choice of updates comes from solving the optimization problem (LSVR-D) considering that only one or two variables are updated at each step. One of the key elements of the algorithm is to make sure that at each step the iterate belongs to $\mathcal{F}$. Let's suppose that the block α is selected as the block in which the update will happen and let (i, j) be the most violating pair of variables. The update is the resolution of a subproblem of size 2, considering that only α_i and α_j are the variables, the rest remains constant. The two equality constraints in (LSVR-D), $\sum_{i=1}^n \alpha_i - \alpha_i^* = 0$ and $\sum_{i=1}^n \alpha_i + \alpha_i^* = C\nu$, lead to the two following equalities: $\alpha_i^{k+1} + \alpha_j^{k+1} = \alpha_i^k + \alpha_j^k$. The later yields to using a parameter t for the update of the variables leading to:

$$\alpha_i^{k+1} = \alpha_i^k + t,$$
$$\alpha_j^{k+1} = \alpha_j^k - t.$$

Updating the variable in the block α this way will force the iterates of Algorithm 3.1 to meet the two equalities constraints at each step. We find t by solving (LSVR-D) considering that we minimize only over t. Let $u \in \mathbb{R}^{2n+p+k_1+k_2}$ be the vector that contains only zeros except at the ith coordinate where it is equal to t and at j^{th} coordinate where it is equal to $-t$. Therefore, we find t by minimizing the following optimization problem:

$$\begin{aligned} \min_{t \in \mathbb{R}} \quad & \psi(t) = \frac{1}{2}\left[(\theta^k + u)^T \bar{Q}(\theta^k + u)\right] + l^T(\theta^k + u) \\ \text{subject to} \quad & 0 \leq \alpha_i^{k+1}, \alpha_j^{k+1} \leq \frac{C}{n}. \end{aligned}$$

First we minimize the objective function without the constraints and since it is a quadratic function of one variable we just clip the solution of unconstrained problem to have the solution of the constrained problem. We will use the term "clipped update" or "clipping" when the update is projected unto the constraints space and is not the result of the unconstrainted optimization problem. As we only consider size one problem for the updates, it will mean that the update will be a bound of an interval. We will use the notation K as a term containing the terms that do not depend on t. We write that

$$\begin{aligned}\psi(t) &= \frac{1}{2}u^T\bar{Q}u + u^T\bar{Q}\theta^k + l^T u + K \\ &= \frac{1}{2}t^2(\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij}) + u^T\nabla f(\theta^k) + K \\ &= \frac{1}{2}t^2(\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij}) + t(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)) + K.\end{aligned}$$

It follows that the unconstrained minimum of $\psi(t)$ is $t_q = \frac{-(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k))}{(\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij})}$. Taking the constraints into account we have that:

$$\begin{aligned}0 \leq \alpha_i^k + t \leq \frac{C}{n}, \\ 0 \leq \alpha_j^k - t \leq \frac{C}{n},\end{aligned}$$

it yields to $t^* = \min(\max(I_1, t_q), I_2)$ with $I_1 = \max(-\alpha_i, \alpha_j - \frac{C}{n})$ and $I_2 = \min(\alpha_j, \frac{C}{n} - \alpha_i)$. The definition of the updates for the block α^* relies on the same discussion.

Let's now make an observation that will explain the definition of the updates for the blocks γ and μ. Let i be the index of the variable that will be updated. Solving the problem:

$$\theta_i^{k+1} = \underset{\theta_i}{\operatorname{argmin}} \frac{1}{2}\theta^T\bar{Q}\theta + l^T\theta,$$

leads to the following solution $\theta_i^{k+1} = \frac{-\nabla_i f(\theta^k)}{\bar{Q}_{ii}} + \theta_i^k$.

Let's recall that the update for the block γ has to keep the coefficient of γ positive to stay in $\mathcal{F}$ hence we have to perform the following clipped update with $i \in \{2n + p + 1, \ldots, 2n + p + k_1\}$:

$$\theta_i^{k+1} = \max(\frac{-\nabla_{\gamma_i} f(\theta^k)}{\bar{Q}_{ii}} + \theta_i^k, 0).$$

Then noticing that $\bar{Q}_{ii} = AA_{ii}^T$ for this block, we obtain the update for the block γ.

There are no constraints on the variables in the blok μ, so the update comes from the fact that $\bar{Q}_{ii} = \Gamma\Gamma_{ii}^T$ for $i \in \{2n+p+k_1+1, \ldots, 2n+p+k_1+k_2\}$ which corresponds to the indices of the block μ.

From these updates we have to make sure that $Q_{ii} + Q_{jj} - 2Q_{ij} \neq 0$, let us recall that $Q_{ij} = \langle X_{i:}, X_{j:}\rangle$ which means that $Q_{ii} + Q_{jj} - 2Q_{ij} = ||X_{i:} - X_{j:}||^2$. This quantity is zero only when $X_{i:} = X_{j:}$ coordinate wise. It would mean that the same row appears two times in the design matrix which does not bring any new information for the regression and can be avoided easily. $(AA^T)_{ii} = \langle A_{i:}, A_{i:}\rangle$ is zero if and only

if $A_{i:} = 0$ which means that a row of the matrix A is zero, so there is no constraint on any variable of the optimization problem which will never happen. It is the same discussion for $(\Gamma\Gamma^T)_{ii}$.

The next proposition makes sure that once a variable (resp. pair of variables) is updated, it cannot be a violating variable (resp. pair of variables) at the next step. This proposition makes sure, for the two blocks α and α^*, that the update t^* cannot be 0.

PROPOSITION 3.6. *If (i,j) (resp.i) was the pair of most violating variable (resp. the most violating variable) in the block α or α^* (resp. block γ or μ) at iteration k then at iteration $k+1$, (i,j) (resp. i) cannot be violating the optimality conditions.*

The proof of this proposition is left in the Appendix C.

Finally, we show that the algorithm converges to a solution of (LSVR-D) and since strong duality holds it allows us to have a solution of (LSVR-P).

THEOREM 3.7. *For any given $\tau > 0$ the sequence of iterates $\{\theta^k\}$, defined by the generalized SMO algorithm, converges to an optimal solution of the optimization problem* (LSVR-D)

The proof of this theorem relies on the same idea as the one proposed in [26] for the classical SMO algorithm and is given in Appendix D. We show that it can be extended to our algorithm with some new observations. The general idea of the proof is to see that the distance between the primal vector generated by the SMO-algorithm and the optimal solution of the primal is controled by the following expression $\frac{1}{2}||\beta^k - \beta^{\text{opt}}|| \leq f(\theta^k) - f(\theta^{\text{opt}})$, where β^k is the kth primal iterate obtained via the relationship primal-dual and θ^k and where β^{opt} is a solution of (LSVR-P). From this observation, we show that we can find a subsequence of the SMO-algorithm θ^{k_j} that converges to some $\bar{\theta}$, solution of the dual problem. Using the continuity of the objective function of the dual problem, we have that $f(\theta^{k_j}) \to f(\bar{\theta})$. Finally, we show that the sequence $\{f(\theta^k)\}$ is decreasing and bounded which implies its convergence and from the convergence monotone theorem we know that to $f(\theta^k)$ converges to $f(\bar{\theta})$ since one of its subsequence converges. This proves that $||\beta^k - \beta^{\text{opt}}|| \to 0$ and finishes the proof. The convergence rate for the SMO algorithm is difficult to obtain considering the greedy choice of the blocks and the greedy choice inside the blocks. A proof for the classical SMO exists but with uniformly at random choice of the block [35]. Convergence rate for greedy algorithms in optimization can be found in [30] for example but the assumption that the constraints must be separable is a major issue for our case. The study of this convergence rate is out of scope of this paper.

4. Numerical experiments. The code for the different regression settings is available on a GitHub repository[1], each setting is wrapped up in a package and is fully compatible with scikit learn [31] `BaseEstimator` class.

In order to compare the estimators, we worked with the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) which are given by the following expressions:

$$\text{MAE} = \frac{1}{p}\sum_{i=1}^{p} |\beta_i^* - \hat{\beta}_i|,$$

$$\text{RMSE} = \sqrt{\frac{1}{p}||\beta^* - \hat{\beta}||^2},$$

[1] https://github.com/Klopfe/LSVR

where β^* are the ground truth coefficients and $\hat{\beta}$ are the estimated coefficients. We also used the Signal-To-Noise Ratio (SNR) to control the level noise simulated in the data. We used the following definition:

$$\text{SNR} = 10 \log 10 (\frac{\mathbb{E}(X\beta(X\beta)^T)}{\text{Var}(\epsilon)}).$$

4.1. Non Negative regression. First, the constraints are set to force the coefficient of β to be positive and we compare our constrained-SVR estimator with the NNLS [23] estimator which is the result of the following optimization problem:

$$\text{(NNLS)} \qquad \begin{aligned} \min_{\beta} \quad & \frac{1}{2}||y - X\beta|| \\ \text{subject to} \quad & \beta_i \geq 0. \end{aligned}$$

In this special case of non-negative regression, $A = -I_p$, $b = 0$, $C = 0$, $d = 0$, the constrained-SVR optimization problem which we will call Non-Negative SVR (NNSVR) then becomes:

$$\text{(NNSVR)} \qquad \begin{aligned} \min_{\beta, \beta_0, \xi_i, \xi_i^*, \epsilon} \quad & \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\ \text{subject to} \quad & \beta^T X_{i:} + \beta_0 - y_i \leq \epsilon + \xi_i \\ & y_i - \beta^T X_{i:} - \beta_0 \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, \epsilon \geq 0 \\ & \beta_i \geq 0. \end{aligned}$$

Synthetic data. We generated the design matrix X from a gaussian distribution $\mathcal{N}(0, 1)$ with 500 samples and 50 features. The true coefficients to be found β^* were gererated taking the exponential of a gaussian distribution $\mathcal{N}(0, 2)$ in order to have positive coefficients. Y was simply computed as the product between X and β^*. We wanted to test the robustness of our estimator compared to NNLS and variant of SVR estimators. To do so, we simulated noise in the data using different types of distributions, we tested gaussian noise and laplacian noise under different levels of noise. For this experiment, the noise distributions were generated to have a SNR equals to 10 and 20, for each type of noise we performed 50 repetitions. The noise was only added in the matrix Y the design matrix X was left noiseless. We compared different estimators NNLS, NNSVR, the Projected-SVR (P-SVR) which is simply the projection of the classical SVR estimator unto the positive orthant and also the classical SVR estimator without constraints. The results of this experiment are in Table 4.1. We see that for a low gaussian noise level (SNR = 20) the NNLS has a lower RMSE and lower MAE. However, we see that the differences between the four compared methods are small. When the level of noise increases (SNR = 10), the NNSVR estimator is the one with the lowest RMSE and MAE. The NNLS estimator performs poorly in the presence of high level of noise in comparison to the SVR based estimator. When a laplacian noise is added to the data, the NNSVR is the estimator that has the lowest RMSE and MAE for low level of noise SNR = 20 and high level of noise SNR = 10.

4.2. Regression unto the simplex. In this subsection, we study the performance of our proposed estimator on simplex constraints Simplex Support Vector Regression (SSVR). In this case, $A = -I_p$, $b = 0$, $\Gamma = \mathbf{e}$ and $d = 1$. The optimization problem that we seek to solve is:

Table 4.1: Results for the Support Vector Regression (SVR), Projected Support Regression (P-SVR), Non-Negative Support Vector Regression (NNSVR) and Non-Negative Least Squares (NNLS) for simulated data with $n = 500$ and $p = 50$. The mean (standard deviation) of the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) over 50 repetitions are reported. Different noise distribution (gaussian and laplacian) and different Signal to Noise Ratio (SNR) values were tested.

Distribution	**Estimator**	**RMSE**	**MAE**
Gaussian noise	SVR	2.238 (0.081)	29.288 (2.452)
	P-SVR	2.178 (0.087)	27.248 (2.545)
SNR = 20	NNSVR	2.174 (0.089)	27.224 (2.480)
($\sigma = 773.1$)	NNLS	**2.120 (0.114)**	**25.226 (2.699)**
Gaussian noise	SVR	2.732 (0.099)	44.764 (4.230)
	P-SVR	2.584 (0.154)	39.687 (5.963)
SNR = 10	NNSVR	**2.536 (0.105)**	**37.740 (3.866)**
($\sigma = 2444.9$)	NNLS	3.478 (0.208)	60.553 (7.923)
Laplacian noise	SVR	2.086 (0.109)	25.538 (3.181)
	P-SVR	2.039 (0.109)	23.978 (3.059)
SNR = 20	NNSVR	**2.035 (0.115)**	**23.827 (3.146)**
(b = 546.7)	NNLS	2.115 (0.103)	25.028 (2.571)
Laplacian noise	SVR	2.665 (0.148)	42.245 (5.777)
	P-SVR	2.526 (0.198)	37.745 (7.271)
SNR = 10	NNSVR	**2.480 (0.157)**	**35.786 (5.761)**
(b = 1728.8)	NNLS	3.463 (0.230)	63.940 (8.375)

$$
\text{(SSVR)} \quad
\begin{aligned}
\min_{\beta,\beta_0,\xi_i,\xi_i^*,\epsilon} \quad & \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\
\text{subject to} \quad & \beta^T X_{i:} + \beta_0 - y_i \leq \epsilon + \xi_i \\
& y_i - \beta^T X_{i:} - \beta_0 \leq \epsilon + \xi_i^* \\
& \xi_i, \xi_i^* \geq 0, \epsilon \geq 0 \\
& \beta_i \geq 0 \\
& \sum_i \beta_i = 1.
\end{aligned}
$$

Synthetic data. We first tested on simulated data generated by the function `make_regression` of scikit-learn. Once the design matrix X and the response vector y were generated using this function, we had access to the ground truth that we will write β^*. This function was not designed to generate data with a β^* that belongs to the simplex so we first projected β^* unto the simplex and then recomputed y multiplying the design matrix by the new projected $\beta^*_{\mathcal{S}}$. We added a centered gaussian noise in the data with the standard deviation of the gaussian was chosen such as the signal-to-noise ratio (SNR) was equal to a defined number, we used the following formula for a given SNR:

$$\sigma = \sqrt{\frac{\text{Var}(y)}{10^{SNR/10}}},$$

where σ is the standard deviation used to simulate the noise in the data. The choice of the two hyperparameters C and ν was done using 5-folds cross validation on a grid of possible pairs. The values of C were taken evenly spaced in the $log10$ base between $[-3, 3]$, we considered 10 different values. The values of ν were taken evenly spaced in the linear space between $[0.05, 1.0]$ and we also considered 10 possible values. We tested different size for the matrix $X \in \mathbb{R}^{n\times p}$ to check the potential effects of the dimensions on the quality of the estimation and we did 50 repetitions for each point of the curves. The measure that was used to compare the different estimators is the RMSE between the true β and the estimated $\hat{\beta}$.

We compared the RMSE of our estimator to the Simplex Ordinary Least Squares (SOLS) which is the result of the following optimization problem:

$$\text{(SOLS)} \qquad \begin{aligned} \min_{\beta} \quad & \frac{1}{2}||y - X\beta|| \\ \text{subject to} \quad & \beta_i \geq 0, \\ & \sum_{i=1}^{p} \beta_i = 1, \end{aligned}$$

and to the estimator proposed in the biostatics litterature that is called Cibersort. This estimator is simply the result of using the classical SVR and project the obtained estimator unto the simplex. The RMSE curves as a function of the SNR are presented in subsection 4.2. We observe that the SSVR is generally the estimator with the lowest RMSE, this observation becomes clearer as the level of noise increases in the data. We notice that when there is a low level of noise and when n is not too large in comparison to p, the three compared estimator perform equally. However, there is a setting when n is large in comparison to p (in this experiment for $n = 250$ or 500 and $p = 5$) where the SSVR estimator has a higher RMSE than the Cibersort and SOLS estimator untill a certain level of noise (SNR < 15). Overall, this simulation shows that there is a significant improvement in the estimation performance of the SSVR mainly when there is noise in the data.

Real dataset. In the cancer research field, regression algorithms have been used to estimate the proportions of cell populations that are present inside a tumor. Indeed, a tumor is composed of different types of cells such as cancer cells, immune cells, healthy cells among others. Having access to the information of the proportions of these cells could be a key to understanding the interactions between the cells and the cancer treatment called immunotherapy [9]. The modelization done is that the RNA extracted from the tumor is seen as a mixed signal composed of different pure signals coming from the different types of cells. This signal can be unmixed knowing the different pure RNA signal of the different types of cells. In other words, y will be the RNA signal coming from a tumor and X will be the design matrix composed of the RNA signal from the isolated cells. The number of rows represent the number of genes that we have access to and the number of columns of X is the number of cell populations that we would like to quantify. The hypothesis is that there is a linear relationship between X and y. As said above, we want to estimate proportions which means that the estimator has to belong to the probability simplex $\mathscr{S} = \{x : x_i \geq 0 \ , \ \sum_i x_i = 1\}$.

Several estimators have been proposed in the biostatistics litterature most of them based on constrained least squares [33, 14, 1] but the gold standard is the estimator based on the SVR.

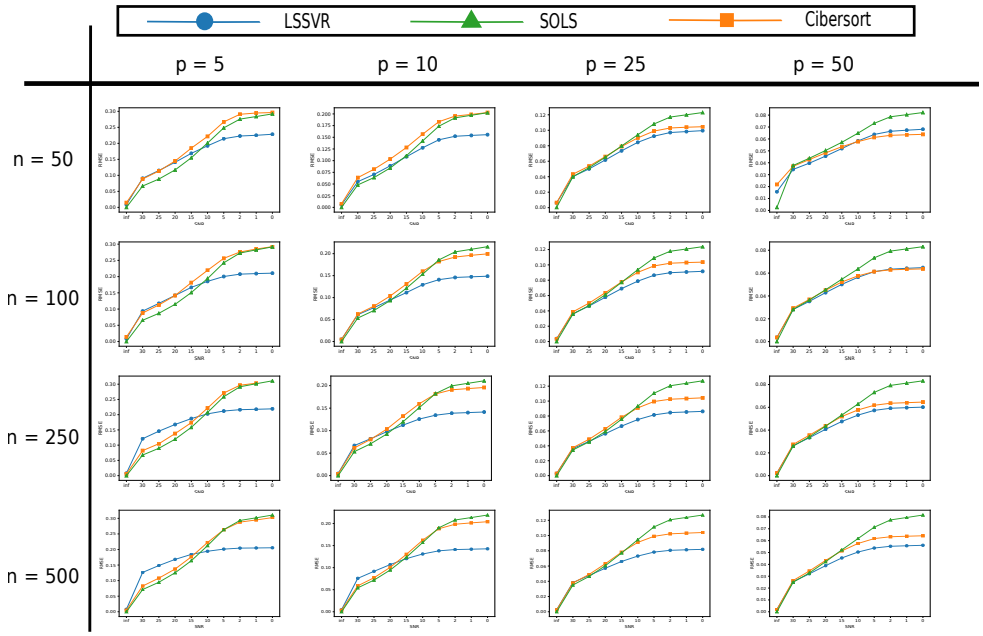


Fig. 4.1: The Root Mean Squared Error (RMSE) as a function of the Signal to Noise Ration (SNR) is presented. Different dimensions for the design matrix X and the response vector y were considered. n represents the number of rows of X and p the number of columns. For each plot, the blue line represents the RMSE for the Linear Simplex SVR (LSSVR) estimator, the green one the Simplex Ordinary Least Squares (SOLS) estimator and the orange on the Cibersort estimator. Each point of the curve is the mean RMSE of 50 repetitions. The noise in the data has a gaussian distribution.

We compared the three same estimators on a real biological dataset where the real quantities of cells to obtain were known. The dataset can be found on the GEO website under the accession code GSE11103[2]. For this example $n = 584$ and $p = 4$ and we have access to 12 different samples that are our repetitions. Following the same idea than previous benchmark performed in this field of application, we increased the level of noise in the data and compared the RMSE of the different estimators. gaussian and laplacian distributions of noise were added to the data. The choice of the two hyperparameters C and ν was done using 5-folds cross validation on a grid of possible pairs. The values of C were taken evenly spaced in the $\log_{10}$ base between $[-5, -3]$, we considered 10 different values. The interval of C is different than the simulated data because of the difference in the range value of the dataset. The values of ν were taken evenly spaced in the linear space between $[0.05, 1.0]$ and we also considered 10 possible values.

We see that when there is no noise in the data (SNR $= \infty$) both Cibersort and SSVR estimator perform equally. The SOLS estimator already has a higher RMSE than the two others estimator probably due to the noise already present in the data.

[2]The dataset can be downloaded from the Gene Expression Omnibus website under the accession code GSE11103.

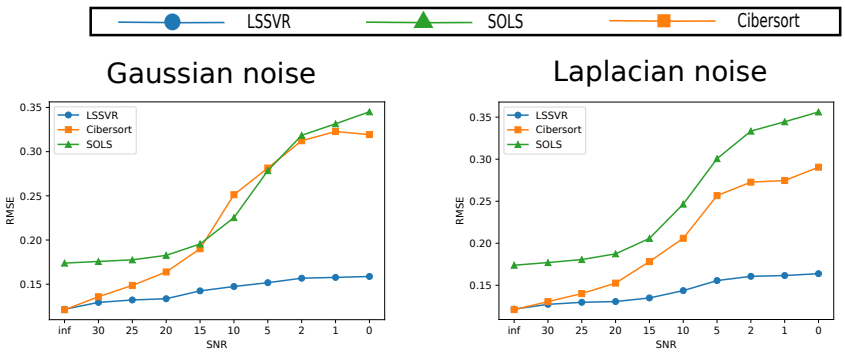


Fig. 4.2: The Root Mean Squared Error (RMSE) as a function of the Signal to Noise Ration (SNR) is presented on a real dataset where noise was manually added. Two different noise distribution were tested: gaussian and laplacian. Each point of the curve is the mean RMSE of 12 different response vectors and we repeated the process four times for each level of noise. This would be equivalent to having 48 different repetitions.

As the level of noise increases, the SSVR estimator remains the estimator with the lowest RMSE in both gaussian and laplacian noise settings.

4.3. Isotonic regression. In this subsection, we will consider constraints that impose an order on the variables. This type of regression is usually called isotonic regression. Such constraints appear when prior knowledge are known on a certain order on the variables. This partial order on the variables can also be seen as an acyclic directed graph. More formally, we note $G = (V, E)$ a directed acyclic graph where V is the set of vertices and E is the set of nodes. On this graph, we define a partial order on the vertices. We will say for $u, v \in V$ that $u \leq v$ if and only if there is a path joining u and v in G. This type of constraints seems natural in different applications such as biology, medicine, weather forecast.

The most simple example of this type of constraints might be the monotonic regression where we force the variables to be in a increasing or decreasing order. It means that with our former notations that we would impose that $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_p$ on the estimator. This type of constraints can be coded in a finite difference matrix (or more generally any incidence matrix of a graph)

$$A = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & 1 & -1 \end{bmatrix}$$

and $\Gamma = 0$, $b = 0$, $d = 0$ forming linear constraints as in the scope of this paper. The Isotonic Support Vector Regression (ISVR) optimization problem is written as follows:

$$
\text{(ISVR)} \qquad \begin{aligned}
\min_{\beta,\beta_0,\xi_i,\xi_i^*,\epsilon} \quad & \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\
\text{subject to} \quad & \beta^T X_{i:} + \beta_0 - y_i \leq \epsilon + \xi_i \\
& y_i - \beta^T X_{i:} - \beta_0 \leq \epsilon + \xi_i^* \\
& \xi_i, \xi_i^* \geq 0, \epsilon \geq 0 \\
& \beta_1 \leq \beta_2 \leq \ldots \leq \beta_n.
\end{aligned}
$$

We compare our proposed ISVR estimator with the classical least squares isotonic regression (IR) [2] which is the solution of the following problem:

$$
\text{(IR)} \qquad \begin{aligned}
\min_{\beta} \quad & \frac{1}{2}||\beta - y||^2 \\
\text{subject to} \quad & \beta_1 \leq \beta_2 \leq \ldots \leq \beta_n.
\end{aligned}
$$

Synthetic dataset. We first generated data from a gaussian distribution ($\mu = 0$, $\sigma = 1$) that we sorted and then added noise in the data following the same process as described in subsection 4.2 with different SNR values (10 and 20). We tested gaussian noise and laplacian noise. We compared the estimation quality of both methods using MAE and RMSE. In this experiment, the design matrix X is the identity matrix. We performed grid search selection via cross validation for the hyperparameters C and ν. C had 5 different possible values taken on the logscale from 0 to 3, and ν had 5 different values taken between 0.05 and 1 on the linear scale. The dimension of the generated gaussian vector was 50 and we did 50 repetitions. We present in Table 4.2 the results of the experiment, the value inside a cell is the mean RMSE or MAE over the 50 repetitions and the value between brackets is the standard deviation over the repetitions. Under a low level of gaussian noise or laplacian noise, both methods are close in term of RMSE and MAE with a little advantage for the classical isotonic regression estimator. When the level of noise is important (SNR = 10), our proposed ISVR has the lowest RMSE and MAE for the two noise distribution tested.

Real dataset. Isotonic types of constraints can be found in different applications such as biology, ranking and weather forecast for example. Focusing on global warming type of data, reserchers have studied the anomaly of the average temperature over a year in comparison to the years 1961-1990. These temperature anomalies have a monotenous trend and keep increasing since 1850 untill 2015. Isotonic regression estimator was used on this dataset[3] in [13] and we compared our proposed ISVR estimator for anomaly prediction. The hyperparameter for the ISVR were set manually for this simulation. subsection 4.3 shows the result for the two estimators. The classical isotonic regression estimator perform better than our proposed estimator globally which is confirmed by the RMSE and MAE values of $\text{RMSE}_{IR} = 0.0067$ against $\text{RMSE}_{ISVR} = 0.083$ and $\text{MAE}_{IR} = 0.083$ against $\text{MAE}_{ISVR} = 0.116$. Howevever, we notice that in the portions where there is a significant change like between 1910-1940 and 1980-2005, the IR estimation looks like a step function whereas the ISVR estimation follows an increasing trend without these piecewise constant portions. Note that the bias induced by the use of constraints can be overcome with reffiting methods such as [10].

[3]This dataset can be downloaded from the Carbon Dioxide Information Analysis Center at the Oak Ridge National Laboratory.

Table 4.2: Results for the Isotonic Support Vector Regression (ISVR), and the Isotonic regression (IR) for simulated data with $p = 50$. The mean (standard deviation) of the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) over 50 repetitions are reported. Different noise distribution (gaussian and laplacian) and different Signal to Noise Ratio (SNR) values were tested.

Distribution	**Estimator**	**RMSE**	**MAE**
Gaussian noise	ISVR	0.212 (0.02)	0.254 (0.06)
SNR = 20	IR	**0.203 (0.02)**	**0.229 (0.04)**
Gaussian noise	ISVR	**0.284 (0.04)**	**0.446 (0.12)**
SNR = 10	IR	0.311 (0.04)	0.534 (0.12)
Laplacian noise	ISVR	**0.202 (0.03)**	0.223 (0.05)
SNR = 20	IR	0.203 (0.02)	**0.221 (0.04)**
Laplacian noise	ISVR	**0.276 (0.05)**	**0.414 (0.11)**
SNR = 10	IR	0.312 (0.05)	0.513 (0.13)

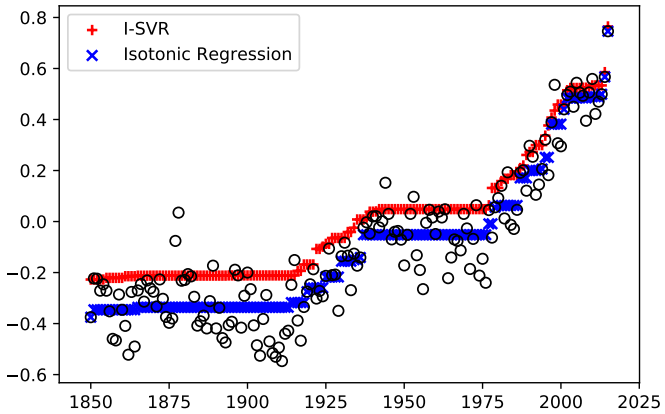
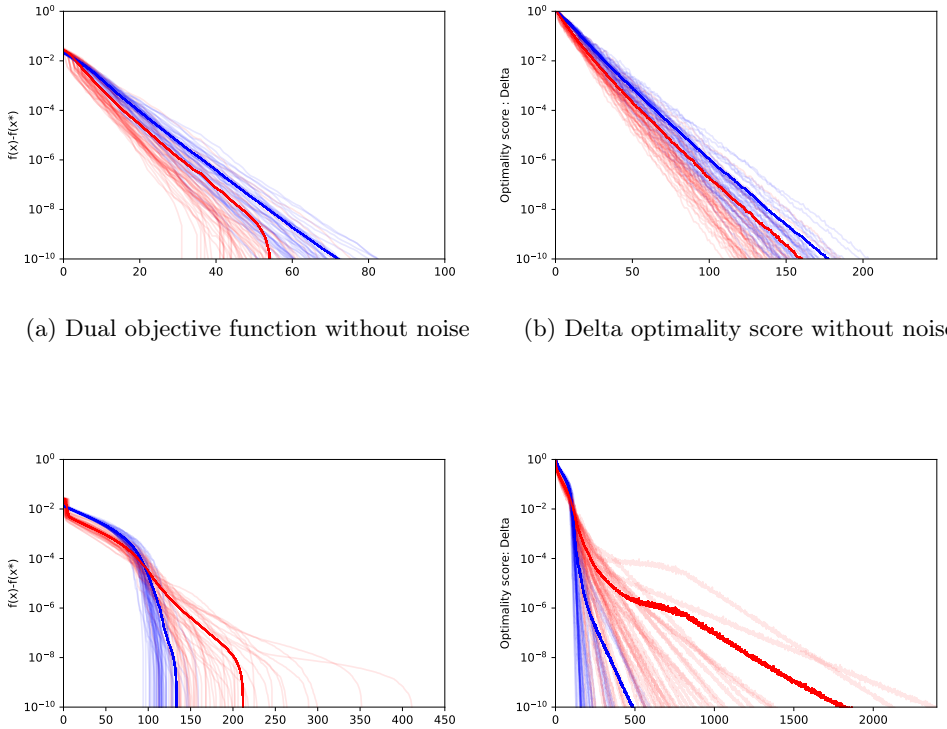


Fig. 4.3: Global warming dataset. Annual temperature anomalies relative to 1961-1990 average, with estimated trend using Isotonic Support Vector Regression (ISVR) and the classical Isotonic Regression (IR) estimator.

4.4. Performance of the GSMO versus SMO. We compared the efficiency of the SMO algorithm to solve the classical SVR optimization problem and the SSVR optimization problem. To do so, we used the same data simulation process described earlier in this subsection and set the number of rows of the matrix X, $n = 200$ and the number of columns $p = 25$. Two different settings were considered here, one without any noise in the data and another one with gaussian noise added such that the SNR would be equal to 30. The transparent trajectories represent the decrease of the objective function or the optimality score Δ for the classical SMO in blue and for the generalized SMO in red for the 50 repetitions considered. The average trajectory is represented in dense color. Figure 4.4a and Figure 4.4b are the results for the noiseless setting and Figure 4.4c and Figure 4.4d for the setting with noise. When



(a) Dual objective function without noise

(b) Delta optimality score without noise

(c) Dual objective function with noise

(d) Optimality score: Delta with noise

Fig. 4.4: Plots of 50 trajectories of the dual objective function value (Figure 4.4a, Figure 4.4c) and the optimality score (Figure 4.4b, Figure 4.4d) in function of the number of iterations for the classical SMO algorithm in blue and the proposed generalized SMO in red. Two settings were used, one without noise and another one with additive gaussian noise.

there is not noise in the data, the generalized SMO decreases faster than the classical SMO. It is important to remind that the true vector here belongs to the simplex so without any noise it is not surprising that our proposed algorithm goes faster than the classical SMO. However, when noise is adding to the data, it takes more iterations for the generalized SMO to find the solution of the optimization problem.

5. Conclusion. In this paper, we studied the optimization problem related to SVR with linear constraints. We showed that for this optimization problem, strong duality holds and that the dual problem is convex. We presented a generalized SMO algorithm that solve the dual problem and we proved its convergence to a solution. This algorithm uses a coordinate descent strategy where a closed form of the updates were defined. The proposed algorithm is easy to implement and shows good performance in practise. We demonstrated the good performance of our proposed estimator on different regression settings. In presence of high level of noise, our estimator has shown to be robust and has better estimation performance in comparison to Least

Squares based estimators or projected SVR estimators.

This work leaves several open questions for future works. The question of the convergence rate of the algorithm is very natural and will have to be address in the future. Another natural question rises about the possiblity to extend our method on non-linear function estimation with linear constraints. From our point of view, it is a very challenging question because the dual optimization problem of the linearly constrained SVR loses its only dependance on the inner product between the columns of X, crossed terms appear in the objective function which makes it difficult to use the kernel trick as it would naturally be used for classical SVR.

REFERENCES

[1] A. R. Abbas, K. Wolslegel, D. Seshasayee, Z. Modrusan, and H. F. Clark, *Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus*, PLOS ONE, 4 (2009), pp. 1–16, https://doi.org/10.1371/journal.pone.0006098.

[2] R. E. Barlow and H. D. Brunk, *The isotonic regression problem and its dual*, Journal of the American Statistical Association, 67 (1972), pp. 140–147, https://doi.org/10.1080/01621459.1972.10481216.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*, in Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, New York, NY, USA, 1992, ACM, pp. 144–152, https://doi.org/10.1145/130385.130401.

[4] R. Bro and S. De Jong, *A fast non-negativity-constrained least squares algorithm*, Journal of Chemometrics, 11 (1997), pp. 393–401, https://doi.org/10.1002/(SICI)1099-128X(199709/10)11:5<393::AID-CEM483>3.0.CO;2-L.

[5] C. Chang and C. Lin, *Training v-support vector regression: Theory and algorithms*, Neural Comput., 14 (2002), pp. 1959–1977, https://doi.org/10.1162/089976602760128081.

[6] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27.

[7] O. Chapelle, P. Haffner, and V. N. Vapnik, *Support vector machines for histogram-based image classification*, IEEE Transactions on Neural Networks, 10 (1999), pp. 1055–1064, https://doi.org/10.1109/72.788646.

[8] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee, *Travel-time prediction with support vector regression*, IEEE Transactions on Intelligent Transportation Systems, 5 (2004), pp. 276–281, https://doi.org/10.1109/TITS.2004.837813.

[9] J. Couzin-Frankel, *Cancer immunotherapy*, Science, 342 (2013), pp. 1432–1433, https://doi.org/10.1126/science.342.6165.1432.

[10] C.-A. Deledalle, N. Papadakis, J. Salmon, and S. Vaiter, *Clear: Covariant least-square refitting with applications to image restoration*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 243–284, https://doi.org/10.1137/16M1080318.

[11] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, *Support vector regression machines*, in Advances in Neural Information Processing Systems 9, M. C. Mozer, M. I. Jordan, and T. Petsche, eds., MIT Press, 1997, pp. 155–161.

[12] T.-T. Friel and R. Harrison, *Linear programming support vector machines for pattern classification and regression estimation: and the sr algorithm: Improving speed and tightness of vc bounds in sv algorithms*, research report, February 1998.

[13] B. R. Gaines, J. Kim, and H. Zhou, *Algorithms for fitting the constrained lasso*, Journal of Computational and Graphical Statistics, 27 (2018), pp. 861–871, https://doi.org/10.1080/10618600.2018.1473777.

[14] T. Gong, N. Hartmann, I. S. Kohane, V. Brinkmann, F. Staedtler, M. Letzkus, S. Bongiovanni, and J. D. Szustakowski, *Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples*, PLOS ONE, 6 (2011), pp. 1–11, https://doi.org/10.1371/journal.pone.0027156.

[15] D. Haussler, D. W. Bednarski, M. Schummer, N. Cristianini, N. Duffy, and T. S. Furey, *Support vector machine classification and validation of cancer tissue samples using microarray expression data* , Bioinformatics, 16 (2000), pp. 906–914, https://doi.org/10.1093/bioinformatics/16.10.906.

[16] A. E. Hoerl and R. W. Kennard, *Ridge regression: Biased estimation for nonorthogonal*

problems, Technometrics, 12 (1970), pp. 55–67, https://doi.org/10.1080/00401706.1970.10488634.

[17] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, *A dual coordinate descent method for large-scale linear svm*, in Proceedings of the 25th International Conference on Machine Learning, ICML '08, New York, NY, USA, 2008, ACM, pp. 408–415, https://doi.org/10.1145/1390156.1390208.

[18] H. Jia and A. M. Martinez, *Support vector machines in face recognition with occlusions*, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, June 2009, pp. 136–141, https://doi.org/10.1109/CVPR.2009.5206862.

[19] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*, in Machine Learning: ECML-98, C. Nédellec and C. Rouveirol, eds., Berlin, Heidelberg, 1998, Springer Berlin Heidelberg, pp. 137–142.

[20] S. S. Keerthi and E. G. Gilbert, *Convergence of a generalized smo algorithm for svm classifier design*, Mach. Learn., 46 (2002), pp. 351–360, https://doi.org/10.1023/A:1012431217818.

[21] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, *Improvements to platt's smo algorithm for svm classifier design*, Neural Comput., 13 (2001), pp. 637–649, https://doi.org/10.1162/089976601300014493.

[22] F. Lauer and G. Bloch, *Incorporating prior knowledge in support vector regression*, Machine Learning, 70 (2008), https://doi.org/10.1007/s10994-007-5035-5.

[23] C. Lawson and R. Hanson, *Solving Least Squares Problems*, Society for Industrial and Applied Mathematics, 1995, https://doi.org/10.1137/1.9781611971217.

[24] C. K. Liew, *Inequality constrained least-squares estimation*, Journal of the American Statistical Association, 71 (1976), pp. 746–751, https://doi.org/10.1080/01621459.1976.10481560.

[25] C.-J. Lin, R. C. Weng, and S. S. Keerthi, *Trust region newton methods for large-scale logistic regression*, 9 (2008), pp. 627–650.

[26] J. Lopez and J. R. Dorronsoro, *Simple proof of convergence of the smo algorithm for different svm variants*, IEEE Transactions on Neural Networks and Learning Systems, 23 (2012), pp. 1142–1147, https://doi.org/10.1109/TNNLS.2012.2195198.

[27] S. Mohammadi, N. Zuckerman, A. Goldsmith, and A. Grama, *A critical survey of deconvolution methods for separating cell types in complex tissues*, Proceedings of the IEEE, 105 (2017), pp. 340–366, https://doi.org/10.1109/JPROC.2016.2607121.

[28] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362, https://doi.org/10.1137/100802001.

[29] A. M. Newman, C. Liu, M. R. Green, A. J. Gentles, W. Feng, Y. Xu, C. D. Hoang, M. Diehn, and A. A. Alizadeh, *Robust enumeration of cell subsets from tissue expression profiles.*, Nature methods, 12 (2015), pp. 453–457.

[30] J. Nutini, M. Schmidt, I. H. Laradji, M. Friedlander, and H. Koepke, *Coordinate descent converges faster with the gauss-southwell rule than random selection*, in Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, pp. 1632–1641.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[32] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, (1998), p. 21.

[33] W. Qiao, G. Quon, E. Csaszar, M. Yu, Q. Morris, and P. W. Zandstra, *Pert: A method for expression deconvolution of human blood samples from varied microenvironmental and developmental conditions*, PLOS Computational Biology, 8 (2012), pp. 1–14, https://doi.org/10.1371/journal.pcbi.1002838.

[34] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, *Shrinking the tube: A new support vector regression algorithm*, in Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II, Cambridge, MA, USA, 1999, MIT Press, pp. 330–336.

[35] J. She, *Linear convergence and support vector identifiation of sequential minimal optimization*, 2017.

[36] A. J. Smola and B. Schölkopf, *A tutorial on support vector regression*, Statistics and Computing, 14 (2004), pp. 199–222, https://doi.org/10.1023/B:STCO.0000035301.49549.88.

[37] J. Suykens and J. Vandewalle, *Least squares support vector machine classifiers*, Neural

Processing Letters, 9 (1999), pp. 293–300, https://doi.org/10.1023/A:1018628609742.

[38] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society: Series B (Methodological), 58 (1996), pp. 267–288, https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[39] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of Optimization Theory and Applications, 109 (2001), pp. 475–494, https://doi.org/10.1023/A:1017501703105.

[40] T. Van Gestel, J. A. K. Suykens, D. . Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, *Financial time series prediction using least squares support vector machines within the evidence framework*, IEEE Transactions on Neural Networks, 12 (2001), pp. 809–821, https://doi.org/10.1109/72.935093.

[41] S. J. Wright, *Coordinate descent algorithms*, Mathematical Programming, 151 (2015), pp. 3–34, https://doi.org/10.1007/s10107-015-0892-3.

[42] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.

Appendix A. Proof of Proposition 2.1.

Proof. We prove the part 2 and 3 of Proposition 2.1 starting by writing the Lagrangian associated to Problem (LSVR-P):

$$\begin{aligned} L =& \frac{1}{2}||\beta||^2 + C(\nu\epsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) + \sum_{i=1}^{n}\alpha_i(-\epsilon - \xi_i - y_i + \beta^T X_{i:} + \beta_0) \\ &+ \sum_{i=1}^{n}\alpha_i^*(-\epsilon - \xi_i^* + y_i - \beta^T X_{i:} - \beta_0) - \sum_{i=1}^{n}\lambda_i\xi_i + \lambda_i^*\xi_i^* - \eta\epsilon \\ &+ \gamma^T(A\beta - b) - \mu^T(\Gamma\beta - d) \end{aligned} \tag{A.1}$$

We will use the notation $x_i^{(*)}$ to denote x_i or x_i^*. From (A.1), we write the KKT conditions:

$$\nabla_\beta L = \beta + \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} + A^T\gamma - \Gamma^T\mu = 0 \tag{A.2a}$$

$$\nabla_{\beta_0} L = \sum_{i=1}^{n}\alpha_i - \alpha_i^* = 0 \tag{A.2b}$$

$$\nabla_{\xi_i^{(*)}} L = \frac{C}{n} - \alpha_i^{(*)} - \lambda_i^{(*)} = 0 \tag{A.2c}$$

$$\nabla_\epsilon L = C\nu - \sum_{i=1}^{n}\alpha_i + \alpha_i^* - \eta = 0 \tag{A.2d}$$

$$\alpha_i^{(*)} \geq 0 \tag{A.2e}$$

$$\eta \geq 0 \tag{A.2f}$$

$$\lambda_i^{(*)} \geq 0 \tag{A.2g}$$

$$\gamma_j \geq 0 \tag{A.2h}$$

$$\alpha_i(-\epsilon - \xi_i - y_i + \beta^T X_{i:} + \beta_0) = 0 \tag{A.2i}$$

$$\alpha_i^*(-\epsilon - \xi_i^* + y_i - \beta^T X_{i:} - \beta_0) = 0 \tag{A.2j}$$

$$\lambda_i^{(*)}\xi_i^{(*)} = 0 \tag{A.2k}$$

$$\eta\epsilon = 0 \tag{A.2l}$$

$$\gamma_j(A\beta - b)_j = 0. \tag{A.2m}$$

From (A.2c), we have that:

$$\lambda_i^{(*)} = \frac{C}{n} - \alpha_i^{(*)}. \tag{A.3}$$

From (A.2e) and (A.2g), we have that:

$$\frac{C}{n} \geq \alpha_i^* \geq 0. \tag{A.4}$$

From (A.2d), we have that:

$$\eta = C\nu - \sum_{i=1}^{n}(\alpha_i + \alpha_i^*). \tag{A.5}$$

From (A.2a),

$$\beta = -\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu. \tag{A.6}$$

From (A.2b),

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0. \tag{A.7}$$

From (A.2f),

$$C\nu \geq \sum_{i=1}^{n}(\alpha_i + \alpha_i^*). \tag{A.8}$$

Using (A.3), (A.5), (A.7), we obtain:

$$\begin{aligned} L =& \frac{1}{2}||\beta||^2 + C\nu\epsilon + \frac{C}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \epsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{n}\alpha_i\xi_i + \alpha_i^*\xi_i^* \\ &+ \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)(-y_i + \beta^T X_{i:}) - \sum_{i=1}^{n}(\frac{C}{n} - \alpha_i)\xi_i + (\frac{C}{n} - \alpha_i^*)\xi_i^* \\ &-(C\nu - \sum_{i=1}^{n}(\alpha_i + \alpha_i^*))\epsilon + \gamma^T(A\beta - b) - \mu^T(\Gamma\beta - d), \end{aligned}$$

and

$$L = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)y_i + \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\beta^T X_{i:} + \gamma^T(A\beta - b) - \mu^T(\Gamma\beta - d).$$

Replacing β by the expression obtained in (A.6) yields to:

$$\begin{aligned} L =& \frac{1}{2}\langle -\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu, -\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu\rangle \\ &- \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)y_i + \langle -\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu, \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:}\rangle \\ &+\gamma^T(A(-\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu) - b) \\ &-\mu^T(\Gamma(-\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu) - d), \end{aligned}$$

$$
\begin{aligned}
L =& \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle X_{i:}, X_{j:}\rangle + \frac{1}{2}\gamma^T AA^T\gamma + \frac{1}{2}\mu^T\Gamma\Gamma^T\mu \\
&+\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} - \gamma^T A\Gamma^T\mu - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)y_i \\
&-\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle X_{i:}, X_{j:}\rangle - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} \\
&+\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} - \gamma^T AA^T\gamma + \gamma^T A\Gamma^T\mu - \gamma^T b \\
&+\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} - \mu^T\Gamma\Gamma^T\mu + \gamma^T A\Gamma^T\mu + \mu^T d,
\end{aligned}
$$

and finally

(A.9)

$$
\begin{aligned}
L =& -\frac{1}{2}(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) - \frac{1}{2}\gamma^T AA^T\gamma - \frac{1}{2}\mu^T\Gamma\Gamma^T\mu - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} \\
&+\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} + \gamma^T A\Gamma^T\mu - \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)y_i - \gamma^T b + \mu^T d.
\end{aligned}
$$

Using the constraints derived from (A.2h), (A.7), (A.8), (A.4) and the expression of the Lagragian (A.9), the dual problem is as follows:

$$
\begin{aligned}
\min_{\alpha,\alpha^*,\gamma,\mu} \quad & \frac{1}{2}((\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + \gamma^T AA^T\gamma + \mu^T\Gamma\Gamma^T\mu + 2\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\gamma^T AX_{i:} \\
& -2\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\mu^T\Gamma X_{i:} - 2\gamma^T A\Gamma^T\mu) + \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)y_i + \gamma^T b - \mu^T d \\
\text{subject to} \quad & 0 \le \alpha_i^{(*)} \le \frac{C}{n} \\
& \sum_{i=1}^{n}\alpha_i + \alpha_i^* \le C\nu \\
& \sum_{i=1}^{n}\alpha_i - \alpha_i^* = 0 \\
& \gamma_j \ge 0.
\end{aligned} \tag{A.10}
$$

The equation linking the primal and the dual optimization problems is given by (A.6) which finishes the proof. ∎

Appendix B. Proof of Proposition 2.2.

Proof. To prove part 1., let's recall that α_i, α_i^* are the lagrange multipliers associated to the optimization problem (LSVR-P) constraints:

$$
\begin{aligned}
\beta^T X_{i:} + \beta_0 - y_i &\le \epsilon + \xi_i \\
y_i - \beta^T X_{i:} - \beta_0 &\le \epsilon + \xi_i^*.
\end{aligned} \tag{B.1}
$$

The complementary optimality conditions leads to

$$\alpha_i(\beta^T X_{i:} + \beta_0 - y_i - \epsilon - \xi_i) = 0$$
$$\alpha_i^*(y_i - \beta^T X_{i:} - \beta_0 - \epsilon - \xi_i^*) = 0.$$

Let's now suppose that $\alpha_i > 0$ and $\alpha_i^* > 0$ which implies that

$$\beta^T X_{i:} + \beta_0 - y_i - \epsilon - \xi_i = 0$$
$$y_i - \beta^T X_{i:} - \beta_0 - \epsilon - \xi_i^* = 0.$$

It follows that $-2\epsilon = \xi_i + \xi_i^*$ and $\xi_i, \xi_i^* \geq 0$ which implies $\xi_i = \xi_i^* = \epsilon = 0$. This goes against our condition $\epsilon > 0$.

To prove part 2., we need to remind the optimality conditions given in Appendix A, (A.2l) and (A.5) leads to

$$(C\nu - \sum_{i=1}^{l} \alpha_i + \alpha_i^*)\epsilon = 0.$$

Thus, if $\epsilon > 0$ we have that $\sum_{i=1}^{n} \alpha_i + \alpha_i^* = C\nu$. ∎

Appendix C. Proof of Proposition 3.6.

We start by giving a lemma that will be usefull to prove the proposition for the blocks α and α^*.

Lemma C.1. *If the update between iteration k and $k+1$ happens in the block α (or α^*) and that (i,j) is the most violating pair of variables then*

$$\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = \nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) + t^*(Q_{ii} + Q_{jj} - 2Q_{ij})$$

Proof. Let's recall that the update in the block α (or α^*) has the following form

$$\alpha_i^{k+1} = \alpha_i^k + t^*$$
$$\alpha_j^{k+1} = \alpha_j^k - t^*,$$

with t^* as defined in Definition 3.5. In a stacked form we have that

$$\begin{aligned}
\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) &= (Q\theta^{k+1})_i + l_i - (Q\theta^{k+1})_j - l_j \\
&= \sum_{s=1}^{2n+k_1+k_2} Q_{is}\theta_s^{k+1} + l_i - \sum_{s=1}^{2n+k_1+k_2} Q_{js}\theta_s^{k+1} - l_j \\
&= \nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) + t^*(Q_{ii} - Q_{ij}) + t^*(Q_{jj} - Q_{ij}) \\
&= \nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) + t^*(Q_{ii} + Q_{jj} - 2Q_{ij}).
\end{aligned}$$

∎

This lemma is helpful for the proof the blocks γ and μ.

Lemma C.2. *If θ_i is the updated variable at iteration k, then the following holds:*

$$\nabla_i f(\theta^{k+1}) = \bar{Q}_{ii}(\theta_i^{k+1} - \theta_i^k) + \nabla_i f(\theta^k)$$

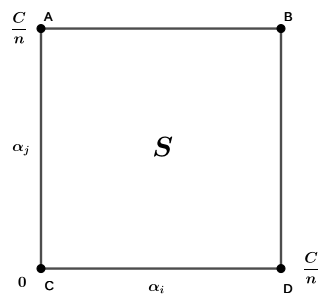


Fig. C.1: Possible update for the block α or α^*

Proof. The proof is straightforward,

$$\begin{aligned}\nabla_i f(\theta^{k+1}) =& (\bar{Q}\theta_i^{k+1})_i + l_i \\ =& \sum_{s=1}^{2n+k_1+k_2} \bar{Q}_{is}\theta_s^{k+1} + l_i \\ =& \sum_{s\neq i}^{2n+k_1+k_2} \bar{Q}_{is}\theta_s^{k+1} + l_i + \bar{Q}_{ii}\theta_i^{k+1} \\ =& \nabla_i f(\theta_k) + \bar{Q}_{ii}\theta_i^{k+1} - \bar{Q}_{ii}\theta_i^k \\ =& \bar{Q}_{ii}(\theta_i^{k+1} - \theta_i^k) + \nabla_i f(\theta^k).\end{aligned}$$

□

Let's now give the proof of Proposition 3.6.

Proof. Let's consider that the update between iteration k and $k+1$ takes place in the block α. We will define (i,j) as the most violating pair of variables as defined in section 3. From the discussion in subsection 3.3, we know that minimizing the objective function of (LSVR-D) considering that only the parameter t is a variable leads to minimizing the following function:

$$\psi(t) = \frac{1}{2}t^2(\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij}) + t(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)) + K \tag{C.1}$$

We recall that t is the parameter that will be used for the update of α_i and α_j and K is a constant term. We also have the following result from Lemma C.1:

$$\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = \nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) + (\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij})t^* \tag{C.2}$$

The minimization update takes place in the square $S = [0, \frac{C}{n}] \times [0, \frac{C}{n}]$ illustrated in Figure C.1.

At points B and C of the square S, (i,j) cannot be a τ-violating pair of variables because they belong to the same set of indices I_{up} (or I_{low}). Everywhere else, violation can take place.

- On $]CA]$, $\alpha_i = 0$ and $\alpha_j > 0$ so $i \in I_{\text{up}}$ and $j \in I_{\text{low}}$ which means that by definition of τ-violating pair of variable

$$\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) < -\tau < 0$$

which means $t_q = \frac{-(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k))}{(\bar{Q}_{ii} + \bar{Q}_{jj} - 2\bar{Q}_{ij})} > 0$. Let's remind that :

$$\max(-\alpha_i, \alpha_j - \frac{C}{n}) \leq t^* \leq \min(\frac{C}{n} - \alpha_i, \alpha_j) \tag{C.3}$$

It means that on $]CA]$, (C.3) becomes : $0 \leq t^* \leq \alpha_j$ There are then two possibilities:

- if $t_q \geq \alpha_j$, it implies because of the constraints on t^*, that $t^* = \alpha_j$. The update becomes then $\alpha_i^{k+1} = \alpha_i^k + \alpha_j^k$ and $\alpha_j^{k+1} = 0$. Then j belongs to the set of indices I_{up} and i belongs to I_{low}. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) \leq 0$ which proves that (i, j) is not a violating pair of variable anymore and that $\alpha^{k+1} \neq \alpha^k$
- Second possibility is that $t_q \leq \alpha_j$ then $t^* = t_q$, then $(\alpha_i^{k+1}, \alpha_j^{k+1})$ belongs to $\text{int}(S)$. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = 0$, (i, j) is not a τ-violating pair of variables anymore and $\alpha^{k+1} \neq \alpha^k$.

- On $]CD]$, $\alpha_i > 0$ and $\alpha_j = 0$ so $i \in I_{\text{low}}$ and $j \in I_{\text{up}}$ which means that by definition of τ-violating pair of variable

$$\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) > \tau$$

which yields to $t_q < 0$. It means that on $]CD]$, (C.3) becomes : $-\alpha_i \leq t^* \leq 0$ There are then two possibilities:

- $t_q \leq -\alpha_i$, it implies because of the constraints on t^*, that $t^* = -\alpha_i$. The update becomes then $\alpha_i^{k+1} = 0$ and $\alpha_j^{k+1} = \alpha_i^k$. Then j belongs to the set of indices I_{low} and i belongs to I_{up} at α^{k+1}. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) \geq 0$ which proves that (i, j) is not a violating pair of variable anymore and that $\alpha^{k+1} \neq \alpha^k$
- Second possibility is that $t_q \geq -\alpha_i$ then $t^* = t_q$. Implying that $(\alpha_i^{k+1}, \alpha_j^{k+1})$ belongs to $\text{int}(S)$. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = 0$, (i, j) is not a τ-violating pair of variables anymore and $\alpha^{k+1} \neq \alpha^k$.

- On $[AB[$, $0 \leq \alpha_i < \frac{C}{n}$ and $\alpha_j = \frac{C}{n}$ so $i \in I_{\text{up}}$ and $j \in I_{\text{low}}$ which means that by definition of τ-violating pair of variable

$$\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) < -\tau < 0,$$

which implies $t_q > 0$.
It means that on $[AB[$, (C.3) becomes $0 \leq t^* \leq \frac{C}{l} - \alpha_i$. There are then two possibilities:

- if $t_q \geq \frac{C}{l} - \alpha_i$, it implies, because of the constraints on t^*, that $t^* = \frac{C}{n} - \alpha_i$. The update is $\alpha_i^{k+1} = \frac{C}{n}$ and $\alpha_j^{k+1} = \alpha_i^k$. Then j belongs to the set of indices I_{up} and i belongs to I_{low}. From (C.2) we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) \leq 0$ which proves that (i, j) is not a violating pair of variable anymore and that $\alpha^{k+1} \neq \alpha^k$.
- Second possibility is that $t_q \leq \frac{C}{n} - \alpha_i$. Thus $t^* = t_q$, then $(\alpha_i^{k+1}, \alpha_j^{k+1})$ belongs to $\text{int}(S)$. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = 0$, (i, j) is not a τ-violating pair of variables anymore and $\alpha^{k+1} \neq \alpha^k$.

- On $]BD]$, $\alpha_i = \frac{C}{n}$ and $0 \leq \alpha_j < \frac{C}{l}$. Thus we have that $i \in I_{\text{low}}$ and $j \in I_{\text{up}}$ which means that by definition of τ-violating pair of variable

$$\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k) > \tau,$$

which yields to $t_q < 0$. It means that on $]BD]$, (C.3) becomes $\alpha_j - \frac{C}{n} \leq t^* \leq 0$. There are then two possibilities:

- if $t_q \leq \alpha_j - \frac{C}{n}$, it implies that $t^* = \alpha_j - \frac{C}{n}$. The update becomes $\alpha_i^{k+1} = \alpha_j^k$ and $\alpha_j^{k+1} = \frac{C}{n}$. Then j belongs to the set of indices I_{low} and i belongs to I_{up} at α^{k+1}. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) \geq 0$ which proves that (i, j) is not a violating pair of variable anymore and that $\alpha^{k+1} \neq \alpha^k$.
- Second possibility is that $t_q \geq \alpha_j - \frac{C}{n}$. Then $t^* = t_q$ and $(\alpha_i^{k+1}, \alpha_j^{k+1}) \in int(S)$. From (C.2), we deduce that $\nabla_{\alpha_i} f(\theta^{k+1}) - \nabla_{\alpha_j} f(\theta^{k+1}) = 0$, (i, j) is not a τ-violating pair of variables anymore and $\alpha^{k+1} \neq \alpha^k$.

- Inside the square S, if $i \in I_{\text{low}}$ and $j \in I_{\text{up}}$, we have that $t_q < 0$. Then there will be three possibilities for the update coming from this inequality $\max(-\alpha_i, \alpha_j - \frac{C}{n}) \leq t^* < 0$. The same discussion as the one we had for the edges of S gives the desired results, the only difference here is that there are three different possibilities: 2 clipped updates possibilities and the update using t_q. The same observation is true for the case where $i \in I_{\text{up}}$ and $j \in I_{\text{low}}$, it will only change the sign of t_q. Thus changing the 2 possible clipped update using the upper bound of (C.3) or the update using t_q. Everything leads to the conclusion that (i, j) cannot be a violating pair of variables at iteration $k + 1$ and that $\alpha^{k+1} \neq \alpha$.

The same arguments are used to prove the same for the block α^*, the proof is similar.

Let's now prove that when the update takes place at index i in the block γ then i is not violating variable at iteration $k+1$. Then we need to show that $\nabla_{\gamma_i} f(\theta^{k+1}) \geq 0$. Let's start with the case where the update $\gamma_i^{k+1} = \frac{\nabla_{\gamma_i} f(\theta^k)}{\bar{Q}_{ii}} - \gamma_i^k$. Using Lemma C.2, we have that $\nabla_{\gamma_i} f(\theta^{k+1}) = 0$. The second possible case is $\gamma_i^{k+1} = 0$ because $-\frac{\nabla_{\gamma_i} f(\theta^k)}{\bar{Q}_{ii}} + \gamma_i^k \leq 0$. If $\gamma_i^{k+1} = 0$ then $\nabla_{\gamma_i} f(\theta^{k+1}) = -\bar{Q}_{ii}\gamma_i^k + \nabla_{\gamma_i} f(\theta^k)$. $\bar{Q}_{ii}$ is positive because it is a diagonal element of a Gram matrix $(A^T A)$ thus we get that $\nabla_{\gamma_i} f(\theta^{k+1}) \geq 0$, which proves that i is not a violating variable anymore.

The proof for the block μ relies on the same idea except that it is simpler because there is no clipped updates possible so $\nabla_{\mu_i} f(\theta^{k+1}) = 0$ if the updates takes place at μ_i which also proves that i is not a violating variable for this block of variables anymore. ∎

Appendix D. Proof of Theorem 3.7.

We begin the proof of the theorem by giving several preliminary results that will be hepful for giving the final proof. The first result gives a bound for controlling the distance of the primal iterates generated by the algorithm and the solution of (LSVR-P).

LEMMA D.1. *For any SMO-LSSVR iterate* $\beta^k = -\sum_{i=1}^n (\alpha_i^k - (\alpha_i^*)^k) X_{i:} - A^T \gamma^k + \Gamma^T \mu^k$, β^{opt} *a solution of* (LSVR-P) *and* θ^{opt} *a solution of* (LSVR-D), *it holds that*

$$\frac{1}{2}||\beta^k - \beta^{opt}|| \leq f(\theta^k) - f(\theta^{opt}).$$

Proof. A first observation is that the relationship between the primal optimization problem and the dual leads to this equality

$$f(\theta^k) = \frac{1}{2}||\beta^k||^2 + l^T \theta^k. \tag{D.1}$$

Replacing β^k by $-\sum_{i=1}^n(\alpha_i^k - (\alpha_i^*)^k)X_{i:} - A^T\gamma^k + \Gamma^T\mu^k$ leads to (D.1). We have already seen that there is strong duality between both problems so the dual gap is zero at the solutions. Thus it means that for any primal optimal solution $(\beta^{\text{opt}}, \beta_0^{\text{opt}}, \xi^{\text{opt}}, \xi^{\text{opt}}, \epsilon^{\text{opt}})$ and any dual solution θ^{opt}, it holds true that

$$\frac{1}{2}||\beta^{\text{opt}}||^2 + C(\nu\epsilon^{\text{opt}} + \frac{1}{n}\sum_{i=1}^n \xi_i^{\text{opt}} + \xi_i^{\text{opt}}) = -f(\theta^{\text{opt}}) = -\frac{1}{2}||\beta^{\text{opt}}||^2 - l^T\theta^{\text{opt}}.$$

Using the equation link between primal and dual yields to

$$\begin{aligned}\langle\beta, \beta^{\text{opt}}\rangle &= \langle -\sum_{i=1}^n(\alpha_i - \alpha_i^*)X_{i:} - A^T\gamma + \Gamma^T\mu, \beta^{\text{opt}}\rangle \\ &= -\langle A^T\gamma, \beta^{\text{opt}}\rangle - \sum_{i=1}^n(\alpha_i - \alpha_i^*)\langle X_{i:}, \beta^{\text{opt}}\rangle + \langle\Gamma^T\mu, \beta^{\text{opt}}\rangle.\end{aligned}$$

Since $\sum_{i=1}^n(\alpha_i - \alpha_i^*) = 0$, we have that

$$\begin{aligned}\langle\beta, \beta^{\text{opt}}\rangle &= -\langle A^T\gamma, \beta^{\text{opt}}\rangle - \sum_{i=1}^n(\alpha_i - \alpha_i^*)\langle X_{i:}, \beta^{\text{opt}}\rangle + \langle\Gamma^T\mu, \beta^{\text{opt}}\rangle - \beta_0^{\text{opt}}\sum_{i=1}^n(\alpha_i - \alpha_i^*) \\ &= -\langle A^T\gamma, \beta^{\text{opt}}\rangle - \sum_{i=1}^n \alpha_i(\langle X_{i:}, \beta^{\text{opt}}\rangle + \beta_0^{\text{opt}}) + \sum_{i=1}^n \alpha_i^*(\langle X_{i:}, \beta^{\text{opt}}\rangle + \beta_0^{\text{opt}}) \\ &+ \langle\Gamma^T\mu, \beta^{\text{opt}}\rangle.\end{aligned}$$

Moreover, using the constraints of (LSVR-P) and the fact that $\alpha \geq 0$ and $\alpha^* \geq 0$ it holds that:

$$\begin{aligned}\langle\beta, \beta^{\text{opt}}\rangle &\geq -\langle A^T\gamma, \beta^{\text{opt}}\rangle + \sum_{i=1}^n \alpha_i(-y_i - \epsilon^{\text{opt}} - \xi_i^{\text{opt}}) + \sum_{i=1}^n \alpha_i^*(y_i - \epsilon^{\text{opt}} - (\xi_i^*)^{\text{opt}}) \\ &+ \langle\Gamma^T\mu, \beta^{\text{opt}}\rangle \\ &= -\langle A^T\gamma, \beta^{\text{opt}}\rangle - \sum_{i=1}^n(\alpha_i - \alpha_i^*)y_i - \epsilon^{\text{opt}}C\nu - \sum_{i=1}^n \alpha_i\xi_i^{\text{opt}} + \alpha_i^*(\xi_i^*)^{\text{opt}} \\ &+ \langle\Gamma^T\mu, \beta^{\text{opt}}\rangle\end{aligned}$$

Finally we have

$$\begin{aligned}\frac{1}{2}||\beta^k - \beta^{\text{opt}}||^2 &= \frac{1}{2}||\beta^k||^2 - \langle\beta^k, \beta^{\text{opt}}\rangle + \frac{1}{2}||\beta^{\text{opt}}||^2 \\ &\leq \frac{1}{2}||\beta^k||^2 + \langle A^T\gamma^k, \beta^{\text{opt}}\rangle + \sum_{i=1}^n(\alpha_i^k - (\alpha_i^*)^k)y_i + \epsilon^{\text{opt}}C\nu \\ &+ \sum_{i=1}^n \alpha_i^k\xi_i^{\text{opt}} + (\alpha_i^*)^k(\xi_i^*)^{\text{opt}} - \langle\Gamma^T\mu^k, \beta^{\text{opt}}\rangle + \frac{1}{2}||\beta^{\text{opt}}||^2\end{aligned}$$

Since β^{opt} statisfies the constraints of the primal optimization problem, it holds

that $\langle \Gamma^T \mu, \beta^{\text{opt}} \rangle = \mu^T d$ and since $\gamma \geq 0$ we have $\langle A^T \gamma, \beta^{\text{opt}} \rangle \leq \gamma^T b$, thus

$$\frac{1}{2}||\beta^k - \beta^{\text{opt}}||^2 \leq \frac{1}{2}||\beta^k||^2 + \gamma^T b + \sum_{i=1}^{n}(\alpha_i^k - (\alpha_i^*)^k)y_i + \epsilon^{\text{opt}} C\nu$$
$$+ \sum_{i=1}^{n} \alpha_i^k \xi_i^{\text{opt}} + (\alpha_i^*)^k (\xi_i^*)^{\text{opt}} - \mu^T d + \frac{1}{2}||\beta^{\text{opt}}||^2.$$

The linear term that we wrote l in the objective function of (LSVR-D) defines $l^T\theta = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)X_{i:} + \gamma^T b - \mu^T d$ which in combination with the equality (D.1) gives

$$\frac{1}{2}||\beta^k - \beta^{\text{opt}}||^2 \leq \frac{1}{2}f(\theta^k) + \epsilon^{\text{opt}} C\nu + \sum_{i=1}^{n} \alpha_i^k \xi_i^{\text{opt}} + (\alpha_i^*)^k (\xi_i^*)^{\text{opt}} + \frac{1}{2}||\beta^{\text{opt}}||^2.$$

Each α_i^k, $(\alpha_i^*)^k$ is bounded by $\frac{C}{n}$ which yields to

$$\frac{1}{2}||\beta^k - \beta^{\text{opt}}||^2 \leq f(\theta^k) + \epsilon^{\text{opt}} C\nu + \frac{C}{n}\sum_{i=1}^{n} \xi_i^{\text{opt}} + (\xi_i^*)^{\text{opt}} + \frac{1}{2}||\beta^{\text{opt}}||^2.$$

We recognize the objective function of the primal optimization problem and using that there is no dual gap at the optimum it follows that

$$\epsilon^{\text{opt}} C\nu + \frac{C}{n}\sum_{i=1}^{n} \xi_i^{\text{opt}} + (\xi_i^*)^{\text{opt}} + \frac{1}{2}||\beta^{\text{opt}}||^2 = -f(\theta^{\text{opt}}),$$

which finishes the proof. ∎

Before the next statement, we need to give a definition that we will use in the next proofs.

Definition D.2. *Let (i, j) ($i \in I_{low}$ and $j \in I_{up}$) be the most violating pair of variables in the block α, (i^*, j^*) ($i^* \in I_{low}^*$ and $j^* \in I_{up}^*$) for the block α^*. Let s_1 be the index of the most violating variable in the block γ and s_2 in the block μ. We will call "optimality score" at iteration k the quantity $\Delta^k = \max(\Delta_1^k, \Delta_2^k, \Delta_3^k, \Delta_4^k)$, where $\Delta_1^k = \max(\nabla_{\alpha_j} f(\theta^k) - \nabla_{\alpha_i} f(\theta^k), 0)$, $\Delta_2^k = \max(\nabla_{\alpha_{j^*}} f(\theta^k) - \nabla_{\alpha_{i^*}} f(\theta^k), 0)$, $\Delta_3^k = \max(-\nabla_{\gamma_{s_1}} f(\theta^k), 0)$ and $\Delta_4^k = \max(|\nabla_{\mu_{s_2}} f(\theta^k)|, 0)$.*

The next result states that the sequence $\{f(\theta^k)\}$ is a decreasing sequence. This result already states the convergence to a certain value $\bar{f}$ because we know that the sequence is bounded by the existing global minimum of the function since f is convex.

Lemma D.3. *The sequence generated by the Generalized SMO algorithm $\{f(\theta^k)\}$ is a decreasing sequence. This sequence converges to a value $\bar{f}$.*

Proof. We first prove that $f(\theta^k) - f(\theta^{k+1}) \geq 0$ when minimization takes place in the block α. Let (i, j) be the indices of the variables selected to be optimized and let $u \in \mathbb{R}^{2n+k_1+k_2}$ be the vector with only zeros except at the ith coordinate where it is equal to t^* as defined in Definition 3.5 and at the jth coordinate where it is equal to $-t^*$. We will also define $t_q = \frac{-(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k))}{Q_{ii} + Q_{jj} - 2Q_{ij}}$, the unconstrained minimum for

the update in α block. Let us compute

$$\begin{aligned}
f(\theta^k) - f(\theta^{k+1}) &= \frac{1}{2}(\theta^k)^T\bar{Q}\theta^k + l^T\theta^k - \frac{1}{2}(\theta^{k+1})^T\bar{Q}\theta^{k+1} + l^T\theta^{k+1} \\
&= \frac{1}{2}(\theta^k)^T\bar{Q}\theta^k + l^T\theta^k - \frac{1}{2}(\theta^k + U)^T\bar{Q}(\theta^k + u) + l^T(\theta^k + u) \\
&= -\frac{1}{2}u^T\bar{Q}u - u^T(Q\theta^k + l) \\
&= -\frac{1}{2}u^T\bar{Q}u - u^T(\nabla f(\theta^k)) \\
&= -\frac{(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) - t^*(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)).
\end{aligned}$$

We first study the case when there is no clipping which means that $t^* = t_q$

1. No clipping. Replacing t^* by its expression leads to the following result:

$$\begin{aligned}
f(\theta^k) - f(\theta^{k+1}) &= \frac{(\Delta_1^k)^2}{2(Q_{ii} + Q_{jj} - 2Q_{ij})} \\
&= \frac{(\Delta_1^k)^2}{2||X_{i:} - X_{j:}||^2} \geq 0.
\end{aligned}$$

2. Clipping takes place because $t_q \leq t^* = \max(-\alpha_i, \alpha_j - \frac{C}{n})$

We notice that $t_q \leq \max(-\alpha_i, \alpha_j - \frac{C}{n}) \leq 0$ which implies that $i \in I_{\text{low}}$ and $j \in I_{\text{up}}$. In that case $\Delta_1^k = \nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)$. Replacing t_q by its expression leads to

$$\begin{aligned}
-(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)) &\leq t^*(Q_{ii} + Q_{jj} - 2Q_{ij}) \\
\frac{\Delta_1^k t^*}{2} &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) \\
\frac{\Delta_1^k t^*}{2} - t^*\Delta_1^k &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) - t^*(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k)) \\
-\frac{1}{2}\Delta_1^k t^* &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) - t^*(\nabla_{\alpha_i} f(\theta^k) - \nabla_{\alpha_j} f(\theta^k))
\end{aligned}$$

Thus we have that if $t^* = -\alpha_i$,

$$f(\theta^k) - f(\theta^{k+1}) \geq \frac{1}{2}\Delta_1^k \alpha_i \geq 0$$

and that if $t^* = \alpha_j - \frac{C}{n}$,

$$f(\theta^k) - f(\theta^{k+1}) \geq \frac{1}{2}\Delta_1^k(\frac{C}{n} - \alpha_j) \geq 0.$$

3. Clipping takes place because $t_q \geq t^* = \min(\frac{C}{n} - \alpha_i, \alpha_j)$.

This time $t_q \geq \min(\frac{C}{n} - \alpha_i, \alpha_j) \geq 0$ which also implies that $i \in I_{\text{up}}$ and $j \in I_{\text{low}}$ and that $\Delta_1^k = \nabla_{\alpha_j} f(\theta^k) - \nabla_{\alpha_i} f(\theta^k)$. The only difference here is that multiplying by $-t^*$ will imply a change in the inequality.

$$
\begin{aligned}
-(\nabla_i f(\theta^k) - \nabla_j f(\theta^k)) &\geq t^*(Q_{ii} + Q_{jj} - 2Q_{ij}) \\
\frac{-\Delta_1^k t^*}{2} &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) \\
\frac{-\Delta_1^k t^*}{2} + t^*\Delta_1^k &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) - t^*(\nabla_i f(\theta^k) - \nabla_j f(\theta^k)) \\
\frac{1}{2}\Delta_1^k t^* &\leq \frac{-(t^*)^2}{2}(Q_{ii} + Q_{jj} - 2Q_{ij}) - t^*(\nabla_i f(\theta^k) - \nabla_j f(\theta^k))
\end{aligned}
$$

Thus we have that if $t^* = \frac{C}{n} - \alpha_i$

$$f(\theta^k) - f(\theta^{k+1}) \geq \frac{1}{2}\Delta_1^k(\frac{C}{n} - \alpha_i) \geq 0,$$

and if $t^* = \alpha_j$,

$$f(\theta^k) - f(\theta^{k+1}) \geq \frac{1}{2}\Delta_1^k \alpha_j \geq 0.$$

To prove that $f(\theta^k) - f(\theta^{k+1}) \geq 0$ when the update takes place in the block γ and μ we first need to observe that when only one variable is updated between iteration k and $k+1$ it follows that

$$f(\theta^k) - f(\theta^{k+1}) = \frac{1}{2}\bar{Q}_{ii}(\theta_i^k - \theta_i^{k+1})^2.$$

Therefore, we now prove the result for the block γ. If the update is not a clipped update and i is the index of the updated variable, it holds that

$$\gamma_i^k - \gamma_i^{k+1} = \frac{\nabla_{\gamma_i} f(\theta^k)}{(AA^T)_{ii}},$$

which gives the following bound

$$f(\theta^k) - f(\theta^{k+1}) = \frac{1}{2(AA^T)_{ii}}(\nabla_{\gamma_i} f(\theta^k))^2 \geq 0. \tag{D.2}$$

Moreover, if a clipped update takes place in this block, we know that it happens when $0 \leq \gamma_i^k \leq \frac{\nabla_{\gamma_i} f(\theta^k)}{(AA^T)_{ii}}$. It yields to the following bound

$$f(\theta^k) - f(\theta^{k+1}) = \frac{1}{2}(AA^T)_{ii}(\gamma_i^k)^2 \geq 0.$$

The result for the block μ is obtained using the same arguments except that there is no clipped updates. ∎

LEMMA D.4. *There exists a subsequence $\{\theta^{k_j}\}$ of iterations generated by the generalized SMO where clipping does not take place.*

Proof. Let's suppose the contrary, which means that there exists an iteration K such that for all $k \geq K$ we only perform clipped updates. The number of variables N_B^k that belong to the boundary of its contraints (0 or $\frac{C}{n}$ for the blocks α or α^* and 0 for the block γ) is non-decreasing for all $k \geq K$ and it is bounded thus it must converge to another integer N^*.

This convergence implies that there exists k^* such that for all $k \geq k^*$, $N_B^k = N^*$ since N_B^k and N^* are integers. This observation allows us to conclude that for all $k \geq k^*$ clipped updates only take place in the blocks α or α^* since the updates in the block γ are made on only one variable and that the number of clipped variables has reached its maximum value. An update in the block γ would strictly increase the number of clipped variables which is not possible for all $k \geq k^*$ or the update would not change the value of θ and we showed before that this situation is not possible (Proposition 3.6).

For all $k \geq k^*$, we have that updates in the block α (resp. α^*) have this necessary scheme: α_i^k or α_j^k is equal to 0 or $\frac{C}{n}$ thus after the update, one of them will leave the boundary and the other one goes to it in order to keep the number of clipped variables equals to N^*. The different possibilities are then the following:

- if $\alpha_i^k = 0$ and $0 < \alpha_j^k \leq \frac{C}{l}$ the only possible update following the Definition 3.5 is
$$\alpha_i^{k+1} = \alpha_i^k + \alpha_j^k = \alpha_j^k$$
$$\alpha_j^{k+1} = \alpha_j^k - \alpha_j^k = 0.$$
- if $\alpha_j^k = \frac{C}{l}$ and $0 \leq \alpha_i^k < \frac{C}{l}$ the only possible update following the Definition 3.5 is
$$\alpha_i^{k+1} = \alpha_i^k + (\frac{C}{l} - \alpha_i^k) = \frac{C}{l}$$
$$\alpha_j^{k+1} = \alpha_j^k - (\frac{C}{l} - \alpha_i^k) = \alpha_i^k.$$

It stays true for the block α^* and the discussion is similar. It is clear that from the description of the updates made above that there is only a finite number of ways to shuffle the values which means that there exists $k_1, k_2 \geq k^*$ such as $\theta^{k_1} = \theta^{k_2}$ and with $k_1 < k_2$. Therefore $f(\theta^{k_1}) = f(\theta^{k_2})$ which contradicts the decrease of the sequence $f(\theta^k)$ (Lemma D.3).

LEMMA D.5. *Let $\{\theta^{k_j}\}$ be a subsequence generated by the Generalized SMO algorithm where clipping does not take place. We then have that $\Delta^{k_j} \to 0$.*

Proof. We have that $f(\theta^{k_j}) - f(\theta^{k_j+1}) \geq \frac{(\nabla_{\alpha_i} f(\theta^{k_j}) - \nabla_{\alpha_j} f(\theta^{k_j}))^2}{2D^2} = \frac{(\Delta^{k_j})^2}{2D^2}$ where $D = \max_{p,q} ||X_{p:} - X_{q:}||$ when the update happens in the blocks α or α^*. When it happens in the block γ with no clipping we have the following inequality $f(\theta^{k_j}) - f(\theta^{k_j+1}) \geq \frac{(\nabla_{\gamma_i} f(\theta^{k_j}))^2}{2} = \frac{(-\Delta^{k_j})^2}{2} = \frac{(\Delta^{k_j})^2}{2}$. When the update takes place in the block μ, we have that $f(\theta^{k_j}) - f(\theta^{k_j+1}) \geq \frac{(\nabla_{\mu_i} f(\theta^{k_j}))^2}{2}) = \frac{(\Delta^{k_j})^2}{2}$. We then define a sequence

$$u^{k_j} = \begin{cases} \frac{1}{2D^2}(\Delta^{k_j})^2 & \text{if the update takes place in the blocks } \alpha \text{ or } \alpha^*. \\ \frac{1}{2}(\Delta^{k_j})^2 & \text{if the update takes place in the blocks } \gamma \text{ or } \mu. \end{cases}$$

The sequence $\{u^{k_j}\} \to 0$ because of the bound given above and the fact that $f(\theta^{k_j}) - f(\theta^{k_j+1}) \to 0$ too (Lemma D.3). This implies that $\Delta^{k_j} \to 0$ as well. ∎

A consequence of the lemma above is that $\Delta_1^{k_j} \to 0$, $\Delta_2^{k_j} \to 0$, $\Delta_3^{k_j} \to 0$ and $\Delta_4^{k_j} \to 0$ because Δ^{k_j} is defined as the maximum of those four positive values.

LEMMA D.6. *Let $\{\theta^{k_j}\}$ be a subsequence generated by the generalized SMO algorithm where clipping does not take place. This subsequence is bounded.*

Proof. To prove the statement, we will show that $||\theta^{k_j} - \theta^{\text{opt}}||^2$ is bounded where θ^{opt} belongs to the set of solution of (LSVR-D). Since each α_i and α_i^* is belongs to $[0, \frac{C}{n}]$, we have that

$$\begin{aligned}||\theta^{k_j+1} - \theta^{\text{opt}}||^2 &= ||\alpha^{k_j+1} - \alpha^{\text{opt}}||^2 + ||(\alpha^*)^{k_j+1} - (\alpha^*)^{\text{opt}}||^2 + ||\gamma^{k_j+1} - \gamma^{\text{opt}}||^2 \\ &\quad + ||\mu^{k_j+1} - \mu^{\text{opt}}||^2 \\ &\leq \frac{2C^2}{n} + ||\gamma^{k_j+1} - \gamma^{\text{opt}}||^2 + ||\mu^{k_j+1} - \mu^{\text{opt}}||^2.\end{aligned}$$

We will work on the bound for the quantity $||\mu^{k_j+1} - \mu^{\text{opt}}||^2$ first. If the update happens in the block μ at coordinate μ_j, we have the following

$$\begin{aligned}||\mu^{k_j+1} - \mu^{\text{opt}}||^2 &= ||\mu^{k_j} - e_j \frac{\nabla_{\mu_j} f(\theta^{k_j})}{(\Gamma\Gamma^T)_{jj}} - \mu^{\text{opt}}||^2 \\ &= ||\mu^{k_j} - \mu^{\text{opt}}||^2 - 2\langle \mu^{k_j} - \mu^{\text{opt}}, e_j \frac{\nabla_{\mu_j} f(\theta^{k_j})}{(\Gamma\Gamma^T)_{jj}} \rangle + ||e_j \frac{\nabla_{\mu_j} f(\theta^{k_j})}{(\Gamma\Gamma^T)_{jj}}||^2 \\ &= ||\mu^{k_j} - \mu^{\text{opt}}||^2 + \frac{\nabla_{\mu_j} f(\theta^{k_j})^2}{(\Gamma\Gamma^T)_{jj}^2} - 2\frac{\nabla_{\mu_j} f(\theta^{k_j})}{(\Gamma\Gamma^T)_{jj}}(\mu_j^{k_j} - \mu_j^{\text{opt}}).\end{aligned}$$

We then have that

$$\begin{aligned}-2\frac{\nabla_{\mu_j} f(\theta^{k_j})}{(\Gamma\Gamma^T)_{jj}}(\mu_j^{k_j} - \mu_j^{\text{opt}}) &= 2(\mu_j^{k_j+1} - \mu_j^{k_j})(\mu_j^{k_j} - \mu_j^{\text{opt}}) \\ &= 2\langle \mu^{k_j+1} - \mu^{k_j}, \mu^{k_j} - \mu^{\text{opt}} \rangle \\ &\leq 2||\mu^{k_j+1} - \mu^{k_j}|| \cdot ||\mu^{k_j} - \mu^{\text{opt}}|| \\ &\leq 2\frac{|\nabla_{\mu_j} f(\theta^{k_j})|}{(\Gamma\Gamma^T)_{jj}}||\mu^{k_j} - \mu^{\text{opt}}|| \\ &\leq 2\frac{\Delta_4^{k_j}}{(\Gamma\Gamma^T)_{jj}}||\mu^{k_j} - \mu^{\text{opt}}||\end{aligned}$$

From Lemma D.5, we have that $\Delta_4^{k_j} \to 0$ then it can be bounded by a constant M_0. We know from (D.2) that $\frac{\nabla_{\mu_j} f(\theta^{k_j})^2}{(\Gamma\Gamma^T)_{jj}^2} = \frac{2}{(\Gamma\Gamma^T)_{jj}}(f(\theta^{k_j}) - f(\theta^{k_j+1}))$. From Lemma D.3, we know that $f(\theta^{k_j}) - f(\theta^{k_j+1}) \to 0$ then it can be bounded by a constant M_1. Overall we have that

$$||\mu^{k_j+1} - \mu^{\text{opt}}||^2 \leq ||\mu^{k_j} - \mu^{\text{opt}}||^2 + 2\frac{M_0}{(\Gamma\Gamma^T)_{jj}}||\mu^{k_j} - \mu^{\text{opt}}|| + \frac{2}{(\Gamma\Gamma^T)_{jj}}M_1.$$

By recursion we have

$$||\mu^{k_j+1} - \mu^{\text{opt}}||^2 \leq ||\mu^0 - \mu^{\text{opt}}||^2 + 2\frac{M_0}{(\Gamma\Gamma^T)_{jj}}||\mu^0 - \mu^{\text{opt}}|| + \frac{2}{(\Gamma\Gamma^T)_{jj}}M_1 < \infty.$$

Since there is no clipped update on the subsequence $\{\theta^{k_j}\}$, the proof for the block γ is similar which proves that $||\theta^{k_j} - \theta^{\text{opt}}||$ is bounded. ∎

LEMMA D.7. *Let $\{\theta^{k_j}\}$ be a subsequence generated by the generalized SMO algorithm where clipping does not take place. There exists a sub-subsequence that converges to $\bar{\theta}$, with $\bar{\theta}$ being a solution of* (LSVR-D).

Proof. From Lemma D.6, we have that $\{\theta^{k_j}\}$ is a bounded sequence, it means that we can extract a converging subsequence that we will write $\{\theta^{k_j}\}$ not to complicate the notations. Since $\mathcal{F}$ is closed, $\bar{\theta}$ meets the constraints of the dual optimization problem and belongs to $\mathcal{F}$. We now want to prove that it belongs to the set of solution of (LSVR-D) by showing that $\bar{\Delta}_1(\bar{\theta}) \leq 0$, $\bar{\Delta}_2(\bar{\theta}) \leq 0$, $\bar{\Delta}_3(\bar{\theta}) \leq 0$ and $\bar{\Delta}_4(\bar{\theta}) \leq 0$. Let's make two observations that will be used for the following proof. The first one comes from the continuity of the gradient which implies that for all ϵ there exists K_1 such that for all $k_j \geq K_1$, $|\nabla_i f(\theta^{k_j}) - \nabla_i f(\bar{\theta})| < \epsilon$ for all i. The second observation is that it is possible too chose an ϵ small enough such that there exists K_2 such that for all $k_j \geq K_2$: if $\bar{\alpha_i} > 0$, we have $\alpha_i^{k_j} > 0$ and if $\bar{\alpha_i} < \frac{C}{n}$ we have $\alpha_i^{k_j} < \frac{C}{n}$. In other words, we say that all the indices in the set $I_{\text{low}}(\bar{\alpha})$(resp. I_{up}) are also in $I_{\text{low}}(\alpha_{k_j})$(resp. I_{up}). The same argument holds for indices in the block α^*.

Let's assume that $\bar{\Delta}_1 > 0$, it means that there exists at least one violating pair of variables that we will note $(\bar{i}, \bar{j})$ at $\bar{\theta}$. From the discussion above, we know that $\bar{i} \in I_{\text{low}}$ for all $k_j \geq K_2$ and that $\bar{j} \in I_{\text{up}}$ for all $k_j \geq K_2$. We then have that for all $\epsilon > 0$, there exists K_1 such as for all $k_j \geq \max(K_1, K_2)$,

$$\begin{aligned}
\Delta_1^{k_j} &= \min_{i \in I_{\text{up}}} \nabla_i f(\theta^{k_j}) - \max_{i \in I_{\text{low}}} \nabla_i f(\theta^{k_j}) \\
&\geq \nabla_{\bar{i}} f(\theta^{k_j}) - \nabla_{\bar{j}} f(\theta^{k_j}) \\
&\geq (\nabla_{\bar{i}} f(\bar{\theta}) - \epsilon) - (\nabla_{\bar{j}} f(\bar{\theta}) + \epsilon) \\
&= \bar{\Delta}_1 - 2\epsilon.
\end{aligned}$$

We choose $\epsilon = \frac{\bar{\Delta}_1}{2} - \epsilon'$ where $0 < \epsilon' < \frac{\bar{\Delta}_1}{2}$ which leads to :

$$\Delta_1^{k_j} \geq \bar{\Delta}_1 - 2\epsilon' = 2\epsilon' > 0.$$

This inequality is true for all $k_j \geq \max(K_1, K_2)$ which contradicts the fact that $\Delta_1^{k_j} \to 0$. The proof is similar to show that $\bar{\Delta}_2 \leq 0$.

Let's now suppose that $\bar{\Delta}_3 > 0$ it means that there exists an index $\bar{i}$ such that $\nabla_{\gamma_i} f(\bar{\theta}) < 0$. For all $\epsilon > 0$, there exists K_1, K_2 such as for all $k_j > \max(K_1, K_2)$

$$\begin{aligned}
\Delta_3^{k_j} &= -\min_{i \in \{1,\dots,k_1\}} \nabla_{\gamma_i} f(\theta^{k_j}) \\
&\geq -\nabla_{\gamma_{\bar{i}}} f(\theta^{k_j}) \\
&\geq -(\nabla_{\gamma_{\bar{i}}} f(\bar{\theta}) + \epsilon) \\
&= \bar{\Delta}_3 - \epsilon.
\end{aligned}$$

We choose $\epsilon = \bar{\Delta}_3 - \epsilon'$ where $0 < \epsilon' < \bar{\Delta}_3$ which leads to :

$$\Delta_3^{k_j} \geq \bar{\Delta}_3 - \epsilon = \epsilon' > 0.$$

This inequality is true for all $k_j \geq \max(K_1, K_2)$ which contradicts the fact that $\Delta_3^{k_j} \to 0$.

Finally let's assume that $\Delta_4^{k_j} > 0$, it means that $|\nabla_{\mu_i} f(\theta^{k_j})| \neq 0$. Using the continuity of the gradient we write that for all $\epsilon > 0$ there exists K_1 such that for all

$k_j \geq K_1$ we have $|\nabla_{\mu_i} f(\theta^{k_j}) - \nabla_{\mu_i} f(\bar{\theta}| < \epsilon$. Using triangle inequality we get that

$$\Big| |\nabla_{\mu_i} f(\theta^{k_j})| - |\nabla_{\mu_i} f(\bar{\theta})| \Big| \leq |\nabla_{\mu_i} f(\theta^{k_j}) - \nabla_{\mu_i} f(\bar{\theta})| < \epsilon.$$

Thus

$$-\epsilon \leq |\nabla_{\mu_i} f(\theta^{k_j})| - |\nabla_{\mu_i} f(\bar{\theta})| \leq \epsilon,$$

which means that

$$|\nabla_{\mu_i} f(\bar{\theta})| - \epsilon \leq |\nabla_{\mu_i} f(\theta^{k_j})|.$$

Then we have the following:

$$\begin{aligned} \Delta_4^{k_j} &= \max_{i \in \{1,\ldots,k_2\}} |\nabla_{\mu_i} f(\theta^{k_j})| \\ &\geq \nabla_{\mu_{\bar{i}}} f(\theta^{k_j})| \\ &\geq |\nabla_{\sigma} f(\bar{\theta})| - \epsilon \\ &= \bar{\Delta}_4 - \epsilon. \end{aligned}$$

We choose $\epsilon = \bar{\Delta}_4 - \epsilon'$ where $0 < \epsilon' < \bar{\Delta}_4$ which leads to

$$\Delta_4^{k_j} \geq \bar{\Delta}_4 - \epsilon = \epsilon' > 0.$$

This inequality is true for all $k_j \geq \max(K_1, K_2)$ which contradicts the fact that $\Delta_4^{k_j} \to 0$. ∎

Proof. We are now able to give the proof of the Theorem 3.7. From Lemma D.1, we have that $\frac{1}{2}||\beta^k - \beta^{\text{opt}}|| \leq f(\theta^k) - f(\theta^{\text{opt}})$. Moreover, from Lemma D.5 we know that there is a subsequence $\{\theta^{k_j}\}$ generated by the Generalized SMO algorithm where clipping does not take place and that converges to $\bar{\theta}$, with $\bar{\theta}$ a solution of (LSVR-D). The continuity of the objective function f allows us to say that $f(\theta^{k_j}) \to f(\bar{\theta})$. From Lemma D.3, we know that $\{f(\theta^{k_j})\}$ is decreasing and bounded so the monotone convergence theorem implies that the whole sequence $f(\theta^k) \to f(\bar{\theta})$ and it follows that $\frac{1}{2}||\beta^k - \beta^{\text{opt}}|| \to 0$ and finally that $\beta^k \to \beta^{\text{opt}}$. ∎