Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Mathematics and Physics
Master in Data Science

A Generalized Lagrange Multiplier Method Support for Vector Regression Based

THESIS to obtain the **DEGREE** of
MASTER IN DATA SCIENCE

A thesis presented by: **Sara Eugenia Rodríguez Reyes**

Thesis Advisor: **Dr. Juan Diego Sánchez Torres**

Tlaquepaque, Jalisco, May, 2021

A Generalized Lagrange Multiplier Method Support for Vector Regression Based

Sara Eugenia Rodríguez Reyes

Abstract

This paper presents an approach to support vector regression based on the L_1^ϵ and L_2^ϵ formulations. Besides, unlike the standard architectures, we explore a new formulation where the dual optimization problem results from formulating an extended Lagrangian function, introducing additional terms to include a weighted elastic net regularization structure. Also, this paper shows the differences and similarities of this proposal with the classical support vector regression and the LASSO regression, aiming to compare with standard models. Finally, to demonstrate the capabilities of this approach, the document includes examples of predicting some benchmark functions.

Keywords— Kernel-based methods, Support vector regression,d Extended Lagrangian

Contents

Page

List of Figures

List of Tables

This thesis is dedicated to my amazing parents. To my beloved mother who, has always encouraged me with her fullest attention to accomplish not only my work but my whole life goals with truthful self-confidence, she have taught me to work hard for the things that I aspire to achieve, and my wise and loving father who, has always been a wonderful supporter and role model. Thank you both for your unconditional love and I am truly thankful for having you in my life.

1 Introduction

The underlying idea of this paper is to introduce a new type of Support Vector Regression model. We improve the ϵ-SVR adding an Elastic net regularization term based on the Generalized Lagrange Multiplier Method, which enables us to perform predictor selection and also reduce the influence of correlated predictors at once.

2 Support Vector Regression and Regularization

Contents

2.1 Regression and Regularization

For the case of the support vector machines for regression (in short support vector regression or SVR), let the set $D = (x_1, y_1), ..., (x_N, y_N)$, where $x_k \in \mathbb{R}^n$ and $y_k \in \mathbb{R}$. Let $\varphi : X \to \mathcal{F}$ be the function that makes each input point x correspond a point in the feature space $\mathcal{F}$, where $\mathcal{F}$ is a Hilbert space. This feature space can be of high dimension or even infinite. However, is common to define $X = \mathbb{R}^n$ and $\mathcal{F} = \mathbb{R}^m$. In this form, the approximating function, namely the model, has the form $\hat{y}_k = f(x_k) = w^T \varphi(x_k) + b$ with $w \in \mathbb{R}^m$ and $b \in \mathbb{R}$.

2.1.1 *p-Norms*

Given a vector space $\mathcal{V}$ over a subfield $\mathcal{J}$ of the complex numbers $\mathbb{C}$, a **norm** on $\mathcal{V}$ is a real-valued function $p : \mathcal{V} \to \mathbb{R}$ with the following properties, where $|s|$ denotes the usual absolute value of a scalar s:

1. Subadditivity/Triangle inequality: $p(x+y) \leq p(x) + p(y)$ for all $x, y \in \mathcal{V}$
2. Absolute homogeneity: $p(sx) = |s|p(x)$ for all $x \in \mathcal{V}$ and all scalars s.
3. Positive definiteness/Point-separating: for all $x \in \mathcal{V}$, if $p(x) = 0$ then $x = 0$.

Because property 2. implies $p(0) = 0$, some authors replace property 3. with the equivalent condition: for all $x \in \mathcal{V}, p(x) = 0$ if only if $x = 0$.

Considering $p \in \mathbb{N}$, $p \geq 1$, the *pth* root of the sum (or integral) of the *pth*-powers of the absolute values of the vector components gives the *p-norm* on suitable real vector spaces, defined as follows.[1]

$$\|\mathbf{x}\|_p := \left(\sum_{k=1}^{n} |x_k|^p \right)^{1/p} \tag{2.1}$$

For $p = 1$, the *p-norm* is the Absolute-value norm, which is a norm on the one-dimensional vector spaces formed by the real or complex numbers.

$$\|\mathbf{x}\|_1 := \sum_{k=1}^{n} |x_k| \tag{2.2}$$

For $p = 2$, the *p-norm* is the standard Euclidean norm, which gives the ordinary distance from the origin to the point x.[2]

$$\|\mathbf{x}\|_2 := \left(\sum_{k=1}^{n} |x_k|^2 \right)^{1/2} \tag{2.3}$$

2.1.2 *Multiple Regression*

Linear regression is a statistical method that attempts to model the relationship between a continuous variable and one or more independent variables by fitting a linear equation.

In most linear regression models, the objective is to minimize the sum of squared errors.

$$\sum_{k=1}^{n} \left(y_k - w^T \varphi(x_k) - b \right)^2 \tag{2.4}$$

[1] Also called ℓ_p-norm

[2] The Euclidean norm is also called the L^2 norm.

where y_k is a continuous target, w is the coefficient, x_k is the predictor and $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$.

Three of the limitations that appear in practice when trying to use this type of model are:

- They are adversely affected by the incorporation of correlated predictors.
- They do not select predictors; all predictors are incorporated into the model even if they do not provide relevant information.
- They cannot be adjusted when the number of predictors is greater than the number of observations.

One way to reduce the impact of these problems is to use regularization strategies such as Ridge, LASSO, or Elastic net, which force the coefficients of the model to tend to zero, thus minimizing the risk of overfitting, reducing variance, attenuating the effect of correlation between predictors and reducing the influence on the model of the less relevant predictors.

2.1.3 L_2 *Regularization*

Rigde regularization admits the following representation,

$$\min_{w} \sum_{k=1}^{n} \left(y_k - w^T \varphi(x_k) - b\right)^2 - \frac{\lambda}{2} \sum_{k=1}^{m} w_k^2 \tag{2.5}$$

Or in terms of the Euclidean norm

$$\min_{w} \sum_{k=1}^{n} \left(y_k - w^T \varphi(x_k) - b\right)^2 - \frac{\lambda}{2} \|w\|_2^2 \tag{2.6}$$

Which is equivalent to the following representation in terms of the internal product:

$$\min_{w} \sum_{k=1}^{n} \left(y_k - w^T \varphi(x_k) - b\right)^2 - \frac{\lambda}{2} w^T w \tag{2.7}$$

The above equations show different forms to represent Ridge regression, all composed of two expressions. The first one uses the square norm notation, and the second term is the regularization L_2 accompanied by the shrinkage quantity. The coefficients are estimated by minimizing this function.

Ridge regularization penalizes the sum of the squared coefficients. This penalty is known as L_2 and has the effect of proportionally reducing the value of all the coefficients in the model, but without them reaching zero. [3]

The main advantage of applying Ridge is the variance reduction without hardly increasing the bias, thus achieving a lower total error. The downside of the Ridge method is that the final model includes all the predictors. This is so because, although the penalty forces the coefficients to tend to zero, they never become precisely zero. This method manages to minimize the influence on the model of the predictors less related to the response variable, but, in the final model, they will continue to appear. Although this is not a problem for the accuracy of the model, it is for its interpretation.

2.1.4 L_1 Regularization

LASSO regularization admits the following representation,

$$\sum_{k=1}^{n}\left(y_k - w^T\varphi(x_k) - b\right)^2 - \lambda\sum_{k=1}^{m}|w_k| \tag{2.8}$$

Or in terms of the Taxicab norm

$$\sum_{k=1}^{n}\left(y_k - w^T\varphi(x_k) - b\right)^2 - \lambda\|w\|_1 \tag{2.9}$$

LASSO is another regularization variation where it only penalizes the high coefficients. It only uses $|w_k|$ (modulus) instead of squares of w, as its penalty, this is known as the L_1 norm and it has the effect of forcing the coefficients of the predictors to tend to zero.

2.1.5 Elastic net Regularization

Elastic net includes a regularization that combines the L_1 and the L_2 penalization $\sigma\lambda\|w\|_1 + \frac{1}{2}(1-\sigma)\|w\|_2^2$ With $0 < \sigma < 1$. The combination of both penalties usually leads to good results. A frequently used strategy is to assign almost all the weight to the L_1 penalty to be able to select predictors and a little to the L_2 to give some stability in the case that some predictors are correlated.

2.2 L_1^ϵ Formulation of the Support Vector Regression

Commonly, the first approach for solving the SVR is the L_1^ϵ formulation. The following problem statement considers such a regression problem as a convex optimization problem.

[3] The λ, is the tuning parameter that decides how much we want to penalize the flexibility of the model.

2.2.1 Problem Statement

The L_1 SVR admits the following optimization problem:

$$\begin{aligned}\min_{w,b,\xi,\xi^*} \mathcal{P}_\epsilon\left(w,b,\xi,\xi^*\right) =& \frac{1}{2}w^T w + C\sum_{k=1}^{N}\left(\xi_k+\xi_k^*\right)\\ \text{s.t. } & y_k - w^T\varphi\left(x_k\right) - b \leq \epsilon + \xi_k,\ k=1,\ldots,N\\ & w^T\varphi\left(x_k\right) + b - y_k \leq \epsilon + \xi_k^*,\ k=1,\ldots,N\\ & \xi_k,\xi_k^* \geq 0,\ k=1,\ldots,N\end{aligned} \tag{2.10}$$

where $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ and the regularization parameter $C > 0$ determines the balance between the regularity of f and the quantity up to which we tolerate deviations more significant than ϵ [4]. We will consider ξ_k and ξ_k^* as slack variables that control the error between the prediction $\hat{y}_k$ and the k-th sample y_k.

2.2.2 KKT Optimality Conditions and Dual Formulation

We proceed to pose the dual problem associated after obtaining the primal problem. The idea is to build a Lagrange function with the objective function and the corresponding constraints by introducing a set of dual variables. This function has a saddle point concerning the variables of the primal.

Let the Lagrangian function for the problem (2.10)

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*,\eta_k,\eta_k^*) =& \frac{1}{2}w^T w + C\sum_{k=1}^{N}\left(\xi_k+\xi_k^*\right)\\ &-\sum_{k=1}^{N}\alpha_k\left(\epsilon+\xi_k-y_k+w^T\ \varphi(x_k)+b\right)\\ &-\sum_{i=k}^{N}\alpha_k^*\left(\epsilon+\xi_k^*+y_k-w^T\varphi(x_k)-b\right)\\ &-\sum_{k=1}^{N}\eta_k\xi_k-\sum_{i=k}^{N}\eta_k^*\xi_k^*\end{aligned} \tag{2.11}$$

where w, b, ξ, ξ^* are the primal variables of the problem and the Lagrange Multipliers $\alpha, \alpha^*, \eta, \eta^*$ are the dual variables associated with the constraints.

First Order Conditions: For a nonlinear programming solution to be optimal, we use the KKT first-order necessary conditions, provided that some regularity conditions are satisfied. As the dual variables are positive, it follows from the saddle point condition that the partial

[4] A very large value of the constant C, in the case where $C(\to \infty)$ we would be considering that the set perfectly represents our hyperplane predictor $\xi_k \to 0$. By cons, too small a number for C would allow high values of ξ_k, that is, admitting a number very high number of poorly represented examples.

derivatives concerning the primal variables should be canceled for the optimal.

- The first order condition on the parameter w, $\nabla_w \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $w = \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \varphi(x_k)$.
- The first order condition on the parameter b, $\frac{\partial}{\partial b} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\sum_{k=1}^{N} (\alpha_k^* - \alpha_k) = 0$.
- The first order condition on the parameter ξ_k, $\frac{\partial}{\partial \xi_k} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\alpha_k + \eta_k = C$
- The first order condition on the parameter ξ_k^*, $\frac{\partial}{\partial \xi_k^*} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\alpha_k^* + \eta_k^* = C$

Replacing in the Lagrangian:

$$\begin{aligned}
&\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = \frac{1}{2} w^T w + C \sum_{k=1}^{N} (\xi_k + \xi_k^*) \\
&- \sum_{k=1}^{N} \alpha_k \left(\epsilon + \xi_k - y_k + \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) \varphi^T(x_l) \varphi(x_k) + b \right) \\
&- \sum_{k=1}^{N} \alpha_k^* \left(\epsilon + \xi_k^* + y_k - \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) \varphi^T(x_l) \varphi(x_k) - b \right) \\
&- \sum_{k=1}^{N} (C + \alpha_k) \xi_k - \sum_{i=k}^{N} (C + \alpha_k^*) \xi_k^*
\end{aligned} \tag{2.12}$$

Grouping variables:

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = & \\
& - \frac{1}{2} \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) \varphi^T(x_l) \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \varphi(x_k) \\
& - \epsilon \sum_{k=1}^{N} (\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k (\alpha_k - \alpha k^*) \\
& - \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) \varphi^T(x_l) \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \varphi(x_k)
\end{aligned} \tag{2.13}$$

Reducing terms:

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = & \\
& - \frac{1}{2} \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) \varphi^T(x_l) \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \varphi(x_k) \\
& - \epsilon \sum_{k=1}^{N} (\alpha_k + \alpha_k^*) + y_k \sum_{k=1}^{N} (\alpha_k - \alpha_k^*)
\end{aligned} \tag{2.14}$$

Primal Feasibility Conditions: Recalling the primal constraints

$$\begin{aligned} y_k - w^T\varphi(x_k) - b \leq \epsilon + \xi_k,\ k = 1, \dots, N \\ w^T\varphi(x_k) + b - y_k \leq \epsilon + \xi_k^*,\ k = 1, \dots, N \\ \xi_k \geq 0,\ \xi_k^* \geq 0 \end{aligned} \tag{2.15}$$

Dual Feasibility Conditions: Due to the Non-Negative Lagrange Multipliers, we find the following deductions:

$$\begin{aligned} &\alpha_k \geq 0,\ k = 1, \dots, N \\ &\alpha_k^* \geq 0,\ k = 1, \dots, N \\ &\eta_k \geq 0,\ k = 1, \dots, N \\ &\eta_k^* \geq 0,\ k = 1, \dots, N \\ &\text{ It follows} \\ &C - \alpha_k = \eta_k \geq 0 \ \rightarrow\ C - \alpha_k \geq 0 \ \text{ Therefore } \ 0 \leq \alpha_k \leq C \\ &C - \alpha_k^* = \eta_k^* \geq 0 \ \rightarrow\ C - \alpha_k^* \geq 0 \ \text{ Therefore } \ 0 \leq \alpha_k^* \leq C \end{aligned} \tag{2.16}$$

We obtain the following dual problem:

$$\begin{aligned} \max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\ -\epsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ \text{s.t.} \ \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ 0 \leq \alpha_k \leq C,\ k = 1, \dots, N \\ 0 \leq \alpha_k^* \leq C,\ k = 1, \dots, N \end{aligned} \tag{2.17}$$

Complementary Slackness Conditions: The Duality Theorem implies a relationship between the primal and dual that is known as complementary slackness. The number of variables in the dual is equal to the number of constraints in the primal, and the number of constraints in the dual is equal to the number of variables in the primal. This correspondence suggests that variables in one problem are complementary to constraints in the other.

For an inequality constraint, the constraint has slack if the slack variable is positive. For a variable constrained to be non-negative, there is slack if the variable is positive. The term complementary slackness refers to a relationship between the slackness in a primal constraint and the slackness (positivity) of the associated dual variable.

The optimal solution must satisfy the KKT Complementary Slackness Condition:

$$\alpha_k \left(\epsilon + \xi_k - y_k + w^T \varphi(x_k) + b\right) = 0 \tag{2.18}$$

$$\alpha_k^* \left(\epsilon + \xi_k^* + y_k - w^T \varphi(x_k) - b\right) = 0 \tag{2.19}$$

$$\eta_k \xi_k = (C - \alpha_k)\xi_k = 0 \tag{2.20}$$

$$\eta_k^* \xi_k^* = (C - \alpha_k^*)\xi_k^* = 0 \tag{2.21}$$

Analyzing the possible values for α_k:

1. For the case $\alpha_k = 0$, from (2.18) we have $y_k - w^T \varphi(x_k) - b - \epsilon - \xi_k \leq 0$. Besides, from $\eta_k = C$ and $\eta_k \xi_k = 0$, it follows $\xi_k = 0$. Therefore, $y_k - w^T \varphi(x_k) - b - \epsilon \leq 0$. In conclusion, when α_k is zero, $|y_k - f(x_k)| < \epsilon$ is satisfied.

2. For the case $0 < \alpha_k < C$, we have $y_k - w^T \varphi(x_k) - b - \epsilon - \xi_k = 0$. From the first order condition on the parameter ξ_k, where $\eta_k = C - \alpha_k$, therefore $\eta_k > 0$. Besides, from $\eta_k \xi_k = 0$, then $\xi_k = 0$ holds. When it is satisfied, in (2.18) the equation in the parenthesis vanishes and the following equation holds: $y_k - w^T \varphi(x_k) - b - \epsilon = 0$

3. For the case $\alpha_k = C$, we have $y_k - w^T \varphi(x_k) - b - \epsilon - \xi_k = 0$. Besides, from $\eta_k = C - \alpha_k$, where $\alpha_k = C$ and $\eta_k = 0$ and $\eta_k \xi_k = 0$, then $\eta_k = 0$ and $\xi_k \geq 0$ holds. In conclusion, when $\alpha_k = C$, the following equation holds: $y_k - w^T \varphi(x_k) - b \leq \epsilon + \xi_k$

On the other hand, analyzing the possible values for α_k^*:

1. For the case $\alpha_k^* = 0$, from (2.19) we have $w^T \varphi(x_k) + b - y_k - \epsilon - \xi_k^* \leq 0$. Besides, from $\eta_k^* = C$ and $\eta_k^* \xi_k^* = 0$, it follows $\xi_k^* = 0$. Therefore, $w^T \varphi(x_k) + b - y_k - \epsilon \leq 0$. In conclusion, when α_k is zero, $|y_k - f(x_k)| < \epsilon$ is satisfied.

2. For the case $0 < \alpha_k^* < C$, we have $w^T \varphi(x_k) + b - y_k - \epsilon - \xi_k^* = 0$. From the first order condition on the parameter ξ_k^*, where $\eta_k^* = C - \alpha_k^*$, therefore $\eta_k^* > 0$. Besides, from $\eta_k^* \xi_k^* = 0$, then $\xi_k^* = 0$ holds. When it is satisfied, in (2.19) the equation in the parenthesis vanishes and the following equation holds: $w^T \varphi(x_k) + b - y_k - \epsilon = 0$

3. For the case $\alpha_k^* = C$, we have $w^T \varphi(x_k) + b - y_k - \epsilon - \xi_k^* = 0$. Besides, from $\eta_k^* = C - \alpha_k^*$, where $\alpha_k^* = C$ and $\eta_k^* = 0$ and $\eta_k^* \xi_k^* = 0$, then $\eta_k^* = 0$ and $\xi_k^* \geq 0$ holds. In conclusion, when $\alpha_k^* = C$, the following equation holds: $w^T \varphi(x_k) + b - y_k \leq \epsilon + \xi_k^*$

Both α_k and α_k^* variables can not be larger than zero at the same time.

From (2.18) and (2.19). If $0 < \alpha_k < C$, it follows $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k = 0$, then $\xi_k = 0$.

Therefore, $y_k - w^T\varphi(x_k) - b = \epsilon$. Replacing in (2.19) we get: $\alpha_k^* \left(w^T\varphi(x_k) + b - y_k + w^T\varphi(x_k) + b - y_k - \xi_k^*\right) = 0$. Grouping terms $\alpha_k^* \left(2\left[w^T\varphi(x_k) + b - y_k\right] - \xi_k^*\right) = 0$ and replacing the ϵ value $\alpha_k^* \left(-2\epsilon - \xi_k^*\right) = 0$, then $\alpha_k^* = 0$ holds. Therefore $\left(-2\epsilon - \xi_k^*\right) < 0$

From these expressions, we would also be extracting an expression of our prediction function [5]:

$$f(x) = \textstyle\sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\varphi^T(x_k)\varphi(x_l) + b$$

To complete the regression function, we should calculate b. Using the slack complementary conditions, which say that the optimal solution of the product between the slack variables and the dual constraints must cancel out.

Deductions:

- $\alpha_k\alpha_k^* = 0$ the two dual variables associated with the same example cannot be positive at the same time.
- Only the examples (x_k, y_k) that $\alpha_k = 0$ or $\alpha_k^* = 0$ would be inside the ϵ tube. And these data do not contribute on constructing the prediction function. Inside the ϵ tube, x_k is not a support vector.
- In the cases where $\alpha_k, \alpha_k^* \in (0, C)$ we would have that the corresponding variable ξ_k, ξ_k^* must be canceled. x_k are not bounded support vectors and the data sample is outside the ϵ-tube. α_k, α_k^* equals C when the data sample is under the tube, so clearing the value of b

$$\begin{gathered} b = y_k - w^T\varphi(x_k) - \epsilon, \text{ such that} \\ \alpha_k \in (0, C) \\ b = y_k - w^T\varphi(x_k) + \epsilon, \text{ such that} \\ \alpha_k^* \in (0, C) \end{gathered} \tag{2.22}$$

[6]

2.2.3 *Reformulation of the L_1^ϵ Support Vector Regression*

The solution of an SVR is globally optimal considering that a quadratic optimization problem expresses it. Since we usually use nonlinear kernels, we need to solve the dual optimization problem whose number of

[5] Obtaining the desired function without depending on the resolution of the problem of the dimension in which our examples of input variables are, it would only depend on the support vectors.

[6] It is possible to calculate the value of b for each support vector, (2.22). In order to maintain numerical stability, b is the average of the set of b's associated with the support values.

variables is twice the training data. Therefore, if the number of training data is vast, training becomes difficult.

Looking at the dual formulation of the L_1^{ϵ} support vector regressor (2.17). The non-negative variables α_k and α_k^* appear in the forms of $\alpha_k - \alpha_k^*$ and $\alpha_k + \alpha_k^*$. Since both α_k and α_k^* are not positive at the same time, the number of variables can be reduced to half by replacing $\alpha_k - \alpha_k^*$ with β_k and $\alpha_k + \alpha_k^*$ with $|\beta_k|$.

Demonstration:

- Having $\beta_k = \alpha_k - \alpha_k^*$, where β_k is defined on the interval $\mathcal{I}$. Defining $\mathcal{P} = \{x \in I : \beta_k \geq 0\}$ and $\mathcal{N} = \{x \in I : \beta_k \leq 0\}$. Then $I = \mathcal{P} \cup \mathcal{N}$. On $\mathcal{P}$ we have that $\beta_k = \alpha_k$ and $\alpha_k = 0$ on $\mathcal{N}$. Furthermore, on $\mathcal{N}$ we have that $\beta_k = -\alpha_k^*$ and $-\alpha_k^* = 0$ on $\mathcal{P}$. So on all of I we define that $\beta_k = \alpha_k - \alpha_k^*$.
- On the other hand, having $|\beta_k| = \alpha_k + \alpha_k^*$, where β_k is defined on the interval $\mathcal{I}$. Let β_k be defined on $\mathcal{I}$ and let $\mathcal{P}$ and $\mathcal{N}$ be as above. Then $|\beta_k| = \alpha_k$ on $\mathcal{P}$ and $\alpha_k = 0$ on $\mathcal{N}$. Furthermore, $|\beta_k| = \alpha_k^*$ on $\mathcal{N}$ and $\alpha_k^* = 0$ on $\mathcal{P}$. So on all of $\mathcal{I}$ we have that $|\beta_k| = \alpha_k + \alpha_k^*$.

Rewriting the L_1^{ϵ} support vector regressor:

$$\begin{aligned}
&\max_{\beta} -\frac{1}{2}\sum_{k,l=1}^{N}\beta_k\beta_l\varphi^T(x_k)\varphi(x_l) - \epsilon\sum_{k=1}^{N}|\beta_k| + \sum_{k=1}^{N}y_k\beta_k \\
&\text{s.t.} \sum_{k=1}^{N}\beta_k = 0 \\
&\qquad -C \leq \beta_k \leq C,\ k = 1,\dots,N
\end{aligned} \tag{2.23}$$

Defining $k(x_k, x_l) = \varphi^T(x_k)\varphi(x_l)$, $\beta = \begin{bmatrix} \beta_1 & \dots & \beta_N \end{bmatrix}^T$, $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$, $y = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$ and $1_v = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$.

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \dots & k(x_N,x_N) \end{bmatrix}$$

We can write the (2.23) formulation in a matrix form:

$$\begin{aligned}
&\max_{\beta} -\frac{1}{2}\beta^T K\beta - \epsilon\|\beta\|_1 + y^T\beta \\
&\text{s.t. } \beta^T 1_v = 0 \\
&\qquad C \preceq |\beta|
\end{aligned} \tag{2.24}$$

Or equivalently,

$$\begin{aligned}\min_{\beta} \frac{1}{2}\beta^T K\beta + \epsilon\|\beta\|_1 - y^T\beta \\ \text{s.t. } \beta^T 1_v = 0 \\ C \preceq |\beta|\end{aligned} \tag{2.25}$$

In this expression the Hessian is K. Since the Hessian is positive semi-definite, the problem is convex and therefore the solution is global.

Additionally, we show in (2.25) the connection between the LASSO and the L_1^ϵ-SVR due to the appearance of a term with the L_1 norm. This is enough for us to show that the L_1^ϵ-SVR is in nature a LASSO problem. [7]

The KKT complementary conditions become:

$$\begin{aligned}\beta_k\left(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right) = 0 \\ \beta_k\left(\epsilon + \xi_k + y_k - w^T\varphi(x_k) - b\right) = 0 \\ \eta_k\xi_k = (C - |\beta_k|)\,\xi_k = 0\end{aligned} \tag{2.26}$$

The b is obtained by:

$$\begin{aligned}b = y_k - w^T\varphi(x_k) - \epsilon, \text{ for } C > \beta_k > 0 \\ b = y_k - w^T\varphi(x_k) + \epsilon, \text{ for } -C < \beta_k < 0\end{aligned} \tag{2.27}$$

By the reformulation in terms of β, the number of variables is reduced from 2N to N, and the obtained support vector regressors are very similar to support vector machines.

2.3 L_2^ϵ Formulation of the Support Vector Regression

In the same way that we derived the dual problem of the L_1^ϵ support vector regressor, we will do it for the L_2^ϵ case.

2.3.1 Problem Statement

L_2^ϵ SVR use the square sum of the slack variables ξ_k in the objective function instead of the linear sum of the slack variables. Thus the L_2^ϵ SVR admits the following optimization problem:

[7] In the feature space, the LASSO selects features. Likewise, the ϵ-SVR selects training samples as support vectors in the observation space. To put it another way, solving in the same way that LASSO selects features, the ϵ-SVR selects support vectors using an L_1-norm.

$$\min_{w,b,\xi,\xi^*} \mathcal{P}_\epsilon\left(w,b,\xi,\xi^*\right) = \frac{1}{2}w^T w + \frac{C}{2}\sum_{k=1}^{N}\left(\xi_k^2 + \xi_k^{*2}\right)$$
$$\text{s.t. } y_k - w^T\varphi\left(x_k\right) - b \leq \epsilon + \xi_k,\ k = 1,\ldots,N$$
$$w^T\varphi\left(x_k\right) + b - y_k \leq \epsilon + \xi_k^*,\ k = 1,\ldots,N \tag{2.28}$$

where $\varphi(\cdot) : \mathbb{R}^N \to \mathbb{R}^m$. [8]

2.3.2 KKT Optimality Conditions and Dual Formulation

We proceed to pose the dual problem associated after obtaining the primal problem.

Let the Lagrangian function for the problem (2.28)

$$\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = \frac{1}{2}w^T w + \frac{C}{2}\sum_{k=1}^{N}\left(\xi_k^2 + \xi_k^{*2}\right)$$
$$- \sum_{k=1}^{N}\alpha_k\left(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right) \tag{2.29}$$
$$- \sum_{k=1}^{N}\alpha_k^*\left(\epsilon + \xi_k^* + y_k - w^T\varphi(x_k) - b\right)$$

where w, b, ξ, ξ^* are the primal variables of the problem and the Lagrange Multipliers α, α^* are the dual variables associated with the constraints.

***First Order Conditions**:* The solution of the constrained optimization problem is determined by the saddle point of the Lagrangian function. The partial derivatives for the variables of the primal should be canceled for optimal.

- The first order condition on the parameter w, $\nabla_w\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = 0$, implies $w = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\varphi(x_k)$.
- The first order condition on the parameter b, $\frac{\partial}{\partial b}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = 0$, implies $\sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0$.
- The first order condition on the parameter ξ_k, $\frac{\partial}{\partial \xi_k}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = 0$, implies $C\xi_k - \alpha_k = 0$
- The first order condition on the parameter ξ_k^*, $\frac{\partial}{\partial \xi_k^*}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = 0$, implies $C\xi_k^* - \alpha_k^* = 0$

[8] For the L_2^ϵ SVR, the positivity constraints on the slack variables ξ_k and ξ_k^* are not necessary since by squaring the terms they preserve the positivity of $\mathcal{P}_\epsilon\left(w,b,\xi,\xi^*\right)$

Replacing the first order conditions in the Lagrangian (2.29):

$$\begin{aligned}\mathcal{L}\left(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*\right) &= \frac{1}{2}w^Tw + \frac{C}{2}\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C^2}+\frac{\alpha_k^{*2}}{C^2}\right)\\ &-\sum_{k=1}^{N}\alpha_k\left(\epsilon+\frac{\alpha_k}{C}-y_k+w^T\varphi(x_k)\right)\\ &-\sum_{k=1}^{N}\alpha_k^*\left(\epsilon+\frac{\alpha_k^*}{C}+y_k-w^T\varphi(x_k)\right)-\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)b\end{aligned} \tag{2.30}$$

Grouping Variables:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) &= \frac{1}{2}w^Tw + \frac{C}{2}\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C^2}+\frac{\alpha_k^{*2}}{C^2}\right)\\ &-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\\ &-\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C}+\frac{\alpha_k^{*2}}{C}\right)-\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)w^T\varphi(x_k)\end{aligned} \tag{2.31}$$

Simplifying the squared terms:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) &= \frac{1}{2}w^Tw + \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2+\alpha_k^{*2}\right)\\ &-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\\ &-\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C}+\frac{\alpha_k^{*2}}{C}\right)-\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)w^T\varphi(x_k)\end{aligned} \tag{2.32}$$

Grouping squared terms:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) &= -\frac{1}{2}w^Tw - \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2+\alpha_k^{*2}\right)\\ &-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\end{aligned} \tag{2.33}$$

Replacing the w term:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) &= -\frac{1}{2}\sum_{k=l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\varphi^T(x_k)\varphi(x_l)\\ -\frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2+\alpha_k^{*2}\right) &-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\end{aligned} \tag{2.34}$$

Considering the case where $k = l$:

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) &= -\frac{1}{2}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right)\varphi^T(x_k)\varphi(x_k) \\
&- \frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\
&- \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right) - \epsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*)
\end{aligned} \tag{2.35}$$

Grouping terms:

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) &= -\frac{1}{2}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right)\left(\varphi^T(x_k)\varphi(x_k) + \frac{1}{C}\right) \\
&- \frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\
&- \epsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*)
\end{aligned} \tag{2.36}$$

Rearranging terms

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) &= -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\left[\varphi^T(x_k)\varphi(x_l) + \frac{\delta_{k,l}}{C}\right] \\
&- \epsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*)
\end{aligned} \tag{2.37}$$

where $\delta_{k,l}$ is Kronecker's delta function. This is a function of two variables, usually just non-negative integers. The function is 1 if the variables are equal, and 0 otherwise:

$$\delta_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq \text{l} \end{cases} \tag{2.38}$$

Primal Feasibility Conditions: Recalling the Primal Constraints:

$$\begin{aligned}
y_k - w^T\varphi(x_k) - b &\leq \epsilon + \xi_k,\ k = 1, \ldots, N \\
w^T\varphi(x_k) + b - y_k &\leq \epsilon + \xi_k^*,\ k = 1, \ldots, N
\end{aligned} \tag{2.39}$$

Dual Feasibility Conditions: Due to the Non-Negative Lagrange Multipliers, we find the following deductions:

$$\begin{aligned}
&\alpha_k \geq 0,\ k = 1,\ldots,N \\
&\alpha_k^* \geq 0,\ k = 1,\ldots,N \\
&\text{It follows} \\
&C\xi_k = \alpha_k \\
&C\xi_k^* = \alpha_k^*
\end{aligned} \tag{2.40}$$

We obtain the following dual problem:

$$\begin{aligned}
&\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\left(\varphi^T(x_k)\varphi(x_l)+\frac{\delta_{kl}}{C}\right) \\
&\quad -\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+y_k\sum_{k=1}^{N}(\alpha_k-\alpha_k^*) \\
&\quad \text{s.t. } \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)=0 \\
&\quad \alpha_k \geq 0,\ k=1,\ldots,N \\
&\quad \alpha_k^* \geq 0,\ k=1,\ldots,N
\end{aligned} \tag{2.41}$$

Complementary Slackness Conditions: The optimal solution must satisfy the following KKT Complementary Slackness Conditions:

$$\alpha_k\left(\epsilon+\xi_k-y_k+w^T\varphi(x_k)+b\right)=0 \tag{2.42}$$

$$\alpha_k^*\left(\epsilon+\xi_k^*+y_k-w^T\varphi(x_k)-b\right)=0 \tag{2.43}$$

$$C\xi_k=\alpha_k,\ k=1,\ldots,N \tag{2.44}$$

$$C\xi_k^*=\alpha_k^*,\ k=1,\ldots,N \tag{2.45}$$

Analyzing the possible values for α_k:

1. For the case $\alpha_k = 0$, from (2.42) we have $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k \leq 0$. Besides, from $C\xi_k = 0$, it follows $\xi_k = 0$. Therefore, $y_k - w^T\varphi(x_k) - b - \epsilon \leq 0$.

2. For the case $\alpha_k > 0$, we have $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k = 0$. From the first order condition on the parameter ξ_k, where $C\xi_k - \alpha_k = 0$, therefore $\xi_k \geq 0$. When it is satisfied, the following equation holds: $y_k - w^T\varphi(x_k) - b \leq \epsilon + \xi_k$

On the other hand, analyzing the possible values for α_k^*:

1. For the case $\alpha_k^* = 0$, from (2.43) we have $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k^* \leq 0$. Besides, from $C\xi_k^* = 0$, it follows $\xi_k^* = 0$. Therefore, $y_k - w^T\varphi(x_k) - b - \epsilon \leq 0$.

2. For the case $\alpha_k^* > 0$, we have $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k^* = 0$. From the first order condition on the parameter ξ_k^*, where $C\xi_k^* = \alpha_k^*$, therefore $\xi_k^* \geq 0$. When it is satisfied, the following equation holds: $y_k - w^T\varphi(x_k) - b \leq \epsilon + \xi_k^*$

Note that the L_2^ϵ support vector regression does not have bounded support vectors.

Both α_k and α_k^* variables can not be larger than zero at the same time.

From (2.42) and (2.43). If $\alpha_k > 0$, it follows $y_k - w^T\varphi(x_k) - b - \epsilon - \xi_k = 0$, then $\xi_k = 0$.

Therefore, $y_k - w^T\varphi(x_k) - b = \epsilon$. Replacing in (2.43) we get: $\alpha_k^* \left(w^T\varphi(x_k) + b - y_k + w^T\varphi(x_k) + b - y_k - \xi_k^*\right) = 0$. Grouping terms $\alpha_k^* \left(2\left[w^T\varphi(x_k) + b - y_k\right] - \xi_k^*\right) = 0$ and replacing the ϵ value $\alpha_k^* \left(-2\epsilon - \xi_k^*\right) = 0$, then $\alpha_k^* = 0$ holds. Therefore $\left(-2\epsilon - \xi_k^*\right) < 0$

From the first order condition on the parameter w, where $w = \sum_{k=1}^N (\alpha_k - \alpha_k^*)\varphi(x_k)$, and using α_k and α_k^*, the prediction function $f(x)$ is expressed as:

$$f(x) = \sum_{i=1}^{N} (\alpha_k - \alpha_k^*)\varphi^T(x_k)\varphi(x_l) + b \qquad (2.46)$$

Obtaining the desired function without depending on the resolution of the problem of the dimension in which our examples of input variables are would only depend on the support vectors.

To complete the regression function, we should calculate b. Using the slack complementary conditions. This says that the product's optimal solution between the slack variables and the dual constraints must cancel out.

Deductions:

- Only the examples (x_k, y_k) that $\alpha_k = 0$ or $\alpha_k^* = 0$ would be outside the ϵ tube

- $\alpha_k\alpha_k^* = 0$ the two dual variables associated with the same example cannot be activated at the same time

- Obtaining the value of b:

$$\begin{aligned} b &= y_k - w^T\varphi(x_k) - \epsilon - \frac{\alpha_k}{C}, \quad \text{for } \alpha_k > 0 \\ b &= y_k - w^T\varphi(x_k) + \epsilon + \frac{\alpha_k^*}{C}, \quad \text{for } \alpha_k^* > 0 \end{aligned} \tag{2.47}$$

9

2.3.3 Reformulation of the L_2^ϵ Support Vector Regression

Similarly, replacing $\alpha_k - \alpha_k^*$ with β_k and $\alpha_k + \alpha_k^*$ with $|\beta_k|$ in 2.41, we obtain the following dual problem for the L_2^ϵ SVR:

$$\begin{aligned} &\max_{\beta} -\frac{1}{2}\sum_{k,l=1}^{N}\beta_k\beta_l\left(\varphi^T(x_k)\varphi(x_l)+\frac{\delta_{kl}}{C}\right) - \epsilon\sum_{k=1}^{N}|\beta_k| + \sum_{k=1}^{N}y_k\beta_k \\ &\text{s.t. } \sum_{k=1}^{N}\beta_k = 0 \end{aligned} \tag{2.48}$$

Defining $k(x_k, x_l) = \varphi^T(x_k)\varphi(x_l)$, $\beta = \begin{bmatrix} \beta_1 & \dots & \beta_N \end{bmatrix}^T$, $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$, $y = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$ and $1_v = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$.

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \dots & k(x_N,x_N) \end{bmatrix}$$

We can write the (2.48) formulation in a matrix form:

$$\begin{aligned} &\max_{\beta} -\frac{1}{2}\beta^T\left(K+\frac{1}{C}I\right)\beta - \epsilon\|\beta\|_1 + y^T\beta \\ &\text{s.t. } \beta^T 1_v = 0 \end{aligned} \tag{2.49}$$

Or equivalently:

$$\begin{aligned} &\min_{\beta}\frac{1}{2}\beta^T\left(K+\frac{1}{C}I\right)\beta + \epsilon\|\beta\|_1 - y^T\beta \\ &\text{s.t. } \beta^T 1_v = 0 \end{aligned} \tag{2.50}$$

In this expression, the Hessian is $\left(K+\frac{1}{C}I\right)$, here the eigenvalues of the matrix are positive. As the Hessian is positive definite, the problem is strongly convex, and therefore the solution is global and unique.

We can express the formulation (2.50) as:

[9] It is possible to calculate the value of b for each support vector (2.47). In order to maintain numerical stability, b is the average of the set of b's associated with the support values

$$\begin{aligned}&\min_{\beta}\frac{1}{2}\beta^T K\beta + \frac{1}{2C}\|\beta\|_2^2 + \epsilon\|\beta\|_1 - y^T\beta\\&\text{s.t. } \beta^T 1_v = 0\end{aligned} \tag{2.51}$$

We show in this expression the connection between the LASSO, the Ridge and the L_2^ϵ-SVR due to the appearance of a term with the L_1 norm and a squared term with the L_2 norm. [10]

When the L_1 and the L_2 regularization appear together they are known as Elastic Net regularization.[11]

The KKT complementary conditions become

$$\begin{aligned}&\beta_k\left(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right) = 0\\&\beta_k\left(\epsilon + \xi_k + y_k - w^T\varphi(x_k) - b\right) = 0\\&C\xi_k = |\beta_k|,\ k = 1,\ldots,N\end{aligned} \tag{2.52}$$

Therefore, b is obtained by

$$\begin{aligned}b = y_k - w^T\varphi(x_k) - \epsilon - \frac{\beta_k}{C},\ \text{ for}\\ \beta_k > 0\\ b = y_k - w^T\varphi(x_k) + \epsilon - \frac{\beta_k}{C},\ \text{ for}\\ \beta_k < 0\end{aligned} \tag{2.53}$$

As in the case of the L_1^ϵ machine, with the reformulation in terms of β, the number of variables are reduced from 2N to N.

If the value of ϵ is very small, almost all training data becomes support vectors.

[10] The Ridge regularization helps the eigenvalues to be positive, making the problem strictly convex and also it creates an elastic net regularization structure.

[11] The combination of both penalties usually leads to good results. The L_1 penalty helps to select predictors and a the L_2 gives some stability in the case that some predictors are correlated.

3 Support Vector Regression Based on Generalized Lagrangian

Contents

3.1 Generalized Lagrange Multiplier Method

The Lagrange multiplier method helps to connect constrained optimization and saddle-point problems since saddle points of Lagrangians provide solutions to corresponding constrained optimization problems, as in the case of the Support Vector Regression, which is based on this saddle-point dynamics.

The Generalized Lagrange Multiplier Method (GLMM) [1] reduces the duality gap between primal and dual problems for non-convex optimization.

Considering a constrained optimization problem with one equality and one inequality constraints

$$\begin{aligned} &\inf_{x\in\mathbb{R}^n} f(x) \\ &\text{s.t. } g(x)\leq 0,\; h(x)=0 \end{aligned} \tag{3.1}$$

Denoting the feasibility set $\{x|g(x)\leq 0, h(x)=0\}$ as $\mathcal{F}$

Adopting the GLMM proposed in [2],

$$\mathcal{L}(x,\tau,\nu)=f(x)+\mathcal{G}(\tau,g(x))+\mathcal{H}(\nu,h(x)) \tag{3.2}$$

[1] S. Boyd and L.Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3

[2] Mengmou Li. Generalized lagrange multiplier method and kkt conditions with an application to distributed optimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66 (2):252–256, 2019. DOI: 10.1109/TCSII.2018.2842085

where $\mathcal{G}(\tau, g(x))$ is a function of $g(x)$ and τ, satisfying

1. Monotonically increasing with respect to $g(x)$:
2. Concave with respect to λ, if $g(x) \leq 0$:
3. $\sup_{\tau} \mathcal{G}(\tau, g(x)) = 0$, $\forall g(x) \leq 0$, and $\sup_{\tau} \mathcal{G}(\tau, g(x)) = +\infty$, $\forall g(x) > 0$;
   similarly, $\mathcal{H}(\nu, h(x))$ is a function of $h(x)$ and ν, satisfying
4. Concave with respect to ν, if $h(x) = 0$;
5. $\sup_{\nu} \mathcal{H}(\nu, 0) = 0$, and $\sup_{\nu} \mathcal{H}(\nu, h(x)) = +\infty$, $\forall h(x) \neq 0$.

The concavity of τ is relaxed; $\mathcal{G}(0, g(x))$, $\mathcal{G}(\tau, 0)$ are not required to be zero.
Assuming $\mathcal{G}(\tau, g(x))$ and $\mathcal{H}(\nu, h(x))$ are continuous and differentiable.

Theorem 1. *Assume the constraints are strictly feasible. If a generalized Lagrangian $\mathcal{L}(x, \tau, \nu)$ under conditions (1)-(5) is closed and proper, it satisfies strong duality between primal and dual problems, i.e.,*

$$\inf_x \sup_{\tau,\nu} \mathcal{L}(x, \tau, \nu) = \sup_{\tau,\nu} \inf_x \mathcal{L}(x, \tau, \nu) \tag{3.3}$$

Proof The primal function is $\mathcal{L}_p(x) = \sup_{\tau,\nu} \mathcal{L}(x, \tau, \nu)$, the primal problem $\inf_x \mathcal{L}_p(x)$ is equivalent to the original problem (3.2) by conditions (3) and (5),

$$\inf_x \mathcal{L}_P(x) = \inf\{f(x), +\infty\} = \inf_x f(x), \forall x \in \mathcal{F} \tag{3.4}$$

Since $\mathcal{L}(x, \tau, \nu)$ is closed, there exists (x^*, τ^*, ν^*) as a solution to (3.4). The dual function is $\mathcal{L}_D(\tau, \nu) = \inf_x \mathcal{L}(x, \tau, \nu) = \inf_x\{f(x) + \mathcal{G}(\tau, g(x)) + \mathcal{H}(\nu, h(x))\}$. Since $\mathcal{L}(x, \tau, \nu)$ is proper, $\mathcal{L}(x, \tau, \nu) > -\infty$, and x' minimizes $\mathcal{L}(x, \tau, \nu)$ for fixed τ and ν, then its gradient $\nabla_x \mathcal{L}(x, \tau, \nu)$ vanishes at x',

$$\nabla_x f\left(x'\right) + \nabla_x \mathcal{G}\left(\tau, g\left(x'\right)\right) + \nabla_x \mathcal{H}\left(\nu, h\left(x'\right)\right) = \mathbf{0}_n \tag{3.5}$$

where $x'(\tau, \nu)$ is a function of τ and ν. Equation (3.5) is guaranteed to have solution for any feasible (τ, ν) including (τ^*, ν^*). Therefore by (3.4) and (3.5),

$$\inf_x \sup_{\tau,\nu} \mathcal{L}(x, \tau, \nu) = \mathcal{L}\left(x'\left(\tau^*, \nu^*\right), \tau^*, \nu^*\right) \tag{3.6}$$

and due to max-min inequality [3]

$$\mathcal{L}\left(x'\left(\tau^*, \nu^*\right), \tau^*, \nu^*\right) \leq \sup_{\tau,\nu} \mathcal{L}_D(\tau, \nu) \leq \inf_x \sup_{\tau,\nu} \mathcal{L}(x, \tau, \nu) \tag{3.7}$$

$\inf_x \sup_{\tau,\nu} \mathcal{L}(x, \tau, \nu) = \sup_{\tau,\nu} \inf_x \mathcal{L}(x, \tau, \nu)$

[3] S. Boyd and L.Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3

***Remark* 1.** *Being closed and proper is a sufficient condition for strong duality. Besides, one major difference between the classic and generalized Lagrangians is that a dual function is concave with respect to its multipliers in the classic Lagrange multiplier method, while a generalized $\mathcal{L}_D(\tau, \nu)$ is not necessarily concave with respect to (τ, ν) but a function with upper bound according to the proof.*

KKT Conditions for Generalized Lagrangians The generalized KKT conditions of the GLMM for the optimization problem can be derived

$$g\left(x^*\right) \leq 0 \tag{3.8}$$

$$\mathcal{G}\left(\tau^*, g\left(x^*\right)\right) = 0 \tag{3.9}$$

$$h\left(x^*\right) = 0 \tag{3.10}$$

$$\mathcal{H}\left(\nu^*, h\left(x^*\right)\right) = 0 \tag{3.11}$$

$$\nabla_x f\left(x^*\right) + \nabla_x \mathcal{G}\left(\tau^*, g\left(x^*\right)\right) + \nabla_x \mathcal{H}\left(x^*, g\left(x^*\right)\right) = \mathbf{0}_n. \tag{3.12}$$

3.2 A GLMM Reformulation for the L_1^ϵ-SVR

We propose a new type of ϵ-SVR using a Generalized Lagrange Multiplier Method, motivated by the interest of adding an Elastic net regularization term to the L_1^ϵ-SVR.

$$\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = {} & \frac{1}{2} w^T w + C \sum_{k=1}^{N} (\xi_k + \xi_k^*) \\
& - \sum_{k=1}^{N} \alpha_k \left(\xi_k - y_k + w^T \ \varphi(x_k) + b \right) \\
& - \sum_{i=k}^{N} \alpha_k^* \left(\xi_k^* + y_k - w^T \varphi(x_k) - b \right) \\
& - \sum_{k=1}^{N} \eta_k \xi_k - \sum_{i=k}^{N} \eta_k^* \xi_k^* \\
& - \lambda \left[(1 - \epsilon) \sum_{k=1}^{N} (\alpha_k + \alpha_k^*) + \frac{\epsilon}{2} (\alpha_k + \alpha_k^*)^2 \right]
\end{aligned} \tag{3.13}$$

In order to continue with the development of this Lagrangian, we first need to check if it meets the definition of GLMM. [4]

[4] Since there are no equality constraints, only conditions (1)-(3) will be analyzed.

1. Monotonically Decreasing with respect to $g(x)$ Defining:

- $g_a(x) = \left[\xi_k - y_k + w^T\varphi(x_k) + b\right]$
- $g_b(x) = \left[\xi_k^* + y_k - w^T\varphi(x_k) - b\right]$
- $g_c(x) = \xi_k$
- $g_d(x) = \xi_k^*$
- $g_e(x) = \lambda\left[(1-\epsilon)\sum_{k=1}^{N}\left(\alpha_k + \alpha_k^*\right) + \frac{\epsilon}{2}\sum_{k=1}^{N}\left(\alpha_k + \alpha_k^*\right)^2\right]$

As:

$g(x) = \langle g_a(x), g_b(x), g_c(x), g_d(x), g_e(x)\rangle$

And,

$$\nabla_{g(x)}\mathcal{G}(\tau, g(x)) = \begin{vmatrix} \nabla_{g_a(x)}\mathcal{G}(\tau, g(x)) \\ \nabla_{g_b(x)}\mathcal{G}(\tau, g(x)) \\ \nabla_{g_c(x)}\mathcal{G}(\tau, g(x)) \\ \nabla_{g_d(x)}\mathcal{G}(\tau, g(x)) \\ \nabla_{g_e(x)}\mathcal{G}(\tau, g(x)) \end{vmatrix} \tag{3.14}$$

Partially deriving each expression against their respective Lagrange multiplier:

- $\alpha g_a(x) = -\alpha_k\left[\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right]$
  $\nabla_{g_a(x)}\mathcal{G}(\alpha, g(x)) = -\alpha_k$
- $\alpha^* g_b(x) = -\alpha_k^*\left[\epsilon + \xi_k^* + y_k - w^T\varphi(x_k) - b\right]$
  $\nabla_{g_b(x)}\mathcal{G}(\alpha^*, g(x)) = -\alpha_k^*$
- $\eta g_c(x) = -\eta_k\xi_k$
  $\nabla_{g_c(x)}\mathcal{G}(\eta, g(x)) = -\eta_k$
- $\eta^* g_d(x) = -\eta_k^*\xi_k^*$
  $\nabla_{g_d(x)}\mathcal{G}(\eta^*, g(x)) = -\eta_k^*$

All Lagrange multipliers are negative, therefore the condition of monotonically decreasing with respect to $g(x)$ holds.

2. Concave with respect to $\alpha, \alpha^*, \eta, \eta^*$***, if*** $g(x) \leq 0$ Obtaining the second partial derivative for each expression:

- $\alpha g_i(x) = -\alpha_k\left[\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right]$
  $\nabla_\alpha^2\mathcal{G}(\alpha, g(x)) = 0$

- $\alpha^* g_j(x) = -\alpha_k^* \left[\epsilon + \xi_k^* + y_k - w^T \varphi(x_k) - b\right]$
  $\nabla^2_{\alpha^*} \mathcal{G}(\alpha^*, g(x)) = 0$
- $\alpha g_m(x) = -\lambda \left[(1-\epsilon)\left(\alpha_k + \alpha_k^*\right) + \frac{\epsilon}{2}\left(\alpha_k + \alpha_k^*\right)^2\right]$
  $\nabla^2_{\alpha} \mathcal{G}(\alpha, g(x)) = \left(\frac{\partial g_m(x)}{\partial \alpha_k}\right) = -\lambda\epsilon$
- $\alpha^* g_m(x) = -\lambda \left[(1-\epsilon)\left(\alpha_k + \alpha_k^*\right) + \frac{\epsilon}{2}\left(\alpha_k + \alpha_k^*\right)^2\right]$
  $\nabla^2_{\alpha^*} \mathcal{G}(\alpha^*, g(x)) = \left(\frac{\partial g_m(x)}{\partial \alpha_k^*}\right) = -\lambda\epsilon$
- $\eta g_l(x) = -\eta_k \xi_k$
  $\nabla^2_{\eta} \mathcal{G}(\eta, g(x)) = 0$
- $\eta^* g_k(x) = -\eta_k^* \xi_k^*$
  $\nabla^2_{\eta^*} \mathcal{G}(\eta^*, g(x)) = 0$

Since the results are either negative or zero, we can conclude that the equation is concave with respect to $\alpha, \alpha^*, \eta, \eta^*$, if $g(x) \leq 0$.

3. $\sup_\tau \mathcal{G}(\tau, g(x)) = 0, \forall g(x) \leq 0$***, and*** $\sup_\tau \mathcal{G}(\tau, g(x)) = \infty, \forall g(x) > 0$

- $\sup_\tau \mathcal{G}(\tau, g(x)) = 0, \forall g(x) \leq 0$
  Having $g(x) \leq 0$ and since Lagrange multipliers are non-negative, it is set bounded by zero. Therefore, the $\sup_\tau \mathcal{G}(\tau, g(x)) = 0$.
- $\sup_\tau \mathcal{G}(\tau, g(x)) = \infty, \forall g(x) > 0$
  Having $g(x) > 0$ and since Lagrange multipliers are non-negative, the set has no upper bound. Therefore, the $\sup_\lambda \mathcal{G}(\tau, g(x)) = \infty$.

Now that we have proved that the new Lagrangian proposal for the ϵ-SVR adding an Elastic net regularization term meets the characteristics of a GLMM, we continue to solve the optimization problem.

First Order Conditions*:*

- The first order condition on the parameter w, $\nabla_w \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $w = \sum_{k=1}^N (\alpha_k - \alpha_k^*) \varphi(x_k)$.
- The first order condition on the parameter b, $\frac{\partial}{\partial b} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\sum_{k=1}^N (\alpha_k - \alpha_k^*) = 0$.
- The first order condition on the parameter ξ_k, $\frac{\partial}{\partial \xi_k} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\alpha_k + \eta_k = C$
- The first order condition on the parameter ξ_k^*, $\frac{\partial}{\partial \xi_k^*} \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*, \eta_k, \eta_k^*) = 0$, implies $\alpha_k^* + \eta_k^* = C$

Replacing in the Lagrangian:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*,\eta_k,\eta_k^*) = {} & \frac{1}{2}w^T w + C\sum_{k=1}^{N}(\xi_k+\xi_k^*) \\ & - \sum_{k=1}^{N}\alpha_k\left(\xi_k - y_k + \sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)\varphi^T(x_k)\varphi(x_l) + b\right) \\ & - \sum_{k=1}^{N}\alpha_k^*\left(\xi_k^* + y_k - \sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)\varphi^T(x_k)\varphi(x_l) - b\right) \\ & - \sum_{k=1}^{N}(C-\alpha_k)\xi_k - \sum_{k=1}^{N}(C-\alpha_k^*)\xi_k^* \\ & - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)^2\right]\end{aligned} \tag{3.15}$$

Grouping variables:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*,\eta_k,\eta_k^*) = {} & \\ & - \frac{1}{2}\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)\varphi^T(x_k)\sum_{l=1}^{N}(\alpha_l-\alpha_l^*)\varphi(x_l) \\ & + C\sum_{k=1}^{N}\xi_k + C\sum_{k=1}^{N}\xi_k^* - \sum_{k=1}^{N}\alpha_k\xi_k + \sum_{k=1}^{N}\alpha_k y_k \\ & - b\sum_{k=1}^{N}\alpha_k - \sum_{k=1}^{N}\alpha_k^*\xi_k^* - \sum_{k=1}^{N}\alpha_k^* y_k + b\sum_{k=1}^{N}\alpha_k^* \\ & - C\sum_{k=1}^{N}\xi_k + \sum_{k=1}^{N}\alpha_k\xi_k - C\sum_{k=1}^{N}\xi_k^* + \sum_{k=1}^{N}\alpha_k^*\xi_k^* \\ & - \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)\varphi^T(x_k)\sum_{l=1}^{N}(\alpha_l-\alpha_l^*)\varphi(x_l) \\ & - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)^2\right]\end{aligned} \tag{3.16}$$

Reducing terms:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*,\eta_k,\eta_k^*) = {} & \\ & - \frac{1}{2}\sum_{k=1}^{N}(\alpha_k-\alpha_k^*)\varphi^T(x_k)\sum_{l=1}^{N}(\alpha_l-\alpha_l^*)\varphi(x_l) \\ & + y_k\sum_{k=1}^{N}(\alpha_k-\alpha_k^*) \\ & - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)^2\right]\end{aligned} \tag{3.17}$$

Primal Feasibility Conditions: Recalling the primal constraints

$$\begin{aligned} y_k - w^T\varphi(x_k) - b &\leq \epsilon + \xi_k,\ k = 1, \ldots, N \\ w^T\varphi(x_k) + b - y_k &\leq \epsilon + \xi_k^*,\ k = 1, \ldots, N \\ \xi_k \geq 0,\ \xi_k^* &\geq 0 \end{aligned} \tag{3.18}$$

Dual Feasibility Conditions: Due to the Non-Negative Lagrange Multipliers, we find the following deductions:

$$\begin{aligned} &\alpha_k \geq 0,\ k = 1, \ldots, N \\ &\alpha_k^* \geq 0,\ k = 1, \ldots, N \\ &\eta_k \geq 0,\ k = 1, \ldots, N \\ &\eta_k^* \geq 0,\ k = 1, \ldots, N \\ &\text{It follows} \\ &C - \alpha_k = \eta_k \geq 0 \ \rightarrow \ C - \alpha_k \geq 0 \ \text{ Therefore } \ 0 \leq \alpha_k \leq C \\ &C - \alpha_k^* = \eta_k^* \geq 0 \ \rightarrow \ C - \alpha_k^* \geq 0 \ \text{ Therefore } \ 0 \leq \alpha_k^* \leq C \end{aligned} \tag{3.19}$$

We obtain the following dual problem:

$$\begin{aligned} \max_{\alpha,\alpha^*} & -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\ & + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ & - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right] \\ & \text{s.t.} \ \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ & 0 \leq \alpha_k \leq C,\ k = 1, \ldots, N \\ & 0 \leq \alpha_k^* \leq C,\ k = 1, \ldots, N \end{aligned} \tag{3.20}$$

Complementary Slackness Conditions: The optimal solution must satisfy the KKT Complementary Slackness Condition:

$$\alpha_k\left(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right) = 0 \tag{3.21}$$

$$\alpha_k^*\left(\epsilon + \xi_k^* + y_k - w^T\varphi(x_k) - b\right) = 0 \tag{3.22}$$

$$\eta_k\xi_k = (C - \alpha_k)\xi_k = 0 \tag{3.23}$$

$$\eta_k^*\xi_k^* = (C - \alpha_k^*)\xi_k^* = 0 \tag{3.24}$$

To complete the regression function, we should calculate b. Using the slack complementary conditions.

$$\begin{aligned} b &= y_k - w^T\varphi(x_k) - \epsilon, \text{ such that} \\ &\alpha_k \in (0, C) \\ b &= y_k - w^T\varphi(x_k) + \epsilon, \text{ such that} \\ &\alpha_k^* \in (0, C) \end{aligned} \tag{3.25}$$

3.2.1 Reformulation of the GLMM L_1^ϵ SVR

Rewriting the GLMM-L_1^ϵ support vector regressor from (3.20):

$$\begin{aligned} \max_\beta & -\frac{1}{2}\sum_{k,l=1}^{N}\beta_k\beta_l\varphi^T(x_k)\varphi(x_l) + \sum_{k=1}^{N} y_k\beta_k \\ & -\lambda\left[(1-\epsilon)\sum_{k=1}^{N}|\beta_k| + \frac{\epsilon}{2}\sum_{k=1}^{N}\beta_k^2\right] \\ \text{s.t. } & \sum_{k=1}^{N}\beta_k = 0 \\ & -C \leq \beta_k \leq C, \; k = 1, \ldots, N \end{aligned} \tag{3.26}$$

Defining $k(x_k, x_l) = \varphi^T(x_k)\varphi(x_l)$, $\beta = \begin{bmatrix} \beta_1 & \ldots & \beta_N \end{bmatrix}^T$, $x = \begin{bmatrix} x_1 & \ldots & x_N \end{bmatrix}^T$, $y = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}^T$ and $1_v = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T$.

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \ldots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \ldots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \ldots & k(x_N,x_N) \end{bmatrix}$$

We can write the (3.26) formulation in a matrix form:

$$\begin{aligned} \max_\beta & -\frac{1}{2}\beta^T K\beta + y^T\beta - \lambda\left[(1-\epsilon)\|\beta\|_1 + \frac{\epsilon}{2}\|\beta\|_2^2\right] \\ & \text{s.t.} \\ & \beta^T 1_v = 0 \\ & |\beta| \preceq C \end{aligned} \tag{3.27}$$

Or equivalently,

$$
\begin{aligned}
&\min_{\beta} \frac{1}{2}\beta^T K\beta - y^T\beta + \lambda\left[(1-\epsilon)\|\beta\|_1 + \frac{\epsilon}{2}\|\beta\|_2^2\right] \\
&\quad \text{s.t.} \\
&\quad \beta^T 1_v = 0 \\
&\quad |\beta| \preceq C
\end{aligned} \tag{3.28}
$$

Conclusion We can observe that this new proposal of ϵ-SVR based on the L_1^ϵ-SVR offers a new structure that proposes an Elastic net regularization keeping the box constraints where $0 \leq \alpha_k, \alpha_k^* \leq C$ which makes easier to calculate the b parameter.

3.3 A GLMM Reformulation for the L_2^ϵ-SVR

In the same way, we proposed a new type of ϵ-SVR using a Generalized Lagrange Multiplier Method based on the L_1^ϵ-SVR: we propose a new type of ϵ-SVR based on the structure of the L_2^ϵ-SVR keeping in mind that the original L_2^ϵ-SVR already has an Elastic net regularization term in the dual form. The motivation of this new proposal is to see if we can keep the same Elastic net structure but adding the box constraints where $0 \leq \alpha_k, \alpha_k^* \leq C$.

$$
\begin{aligned}
\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = &\frac{1}{2}w^T w + \frac{C}{2}\sum_{k=1}^{N}\left(\xi_k^2 + \xi_k^{*2}\right) \\
&- \sum_{k=1}^{N}\alpha_k\left(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b\right) \\
&- \sum_{k=1}^{N}\alpha_k^*\left(\epsilon + \xi_k^* + y_k - w^T\varphi(x_k) - b\right) \\
&- \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]
\end{aligned} \tag{3.29}
$$

In order to continue with the development of this Lagrangian we first need to check if it meets the definition of GLMM. [5]

1. Monotonically Decreasing with respect to $g(x)$ Defining:

- $g_a(x) = [\xi_k - y_k + w^T\varphi(x_k) + b]$
- $g_b(x) = [\xi_k^* + y_k - w^T\varphi(x_k) - b]$
- $g_c(x) = \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]$

[5] Since there are no equality constraints, only conditions (1)-(3) will be analyzed.

As:

$g(x) = \langle g_a(x), g_b(x), g_c(x) \rangle$

And,

$$\nabla_{g(x)} \mathcal{G}(\tau, g(x)) = \begin{vmatrix} \nabla_{g_a(x)} \mathcal{G}(\tau, g(x)) \\ \nabla_{g_b(x)} \mathcal{G}(\tau, g(x)) \\ \nabla_{g_c(x)} \mathcal{G}(\tau, g(x)) \end{vmatrix} \tag{3.30}$$

Partially deriving each expression against their respective Lagrange multiplier:

- $\alpha g_a(x) = -\alpha_k \left[\epsilon + \xi_k - y_k + w^T \varphi(x_k) + b\right]$
  $\nabla_{g_a(x)} \mathcal{G}(\alpha, g(x)) = -\alpha_k$
- $\alpha^* g_b(x) = -\alpha_k^* \left[\epsilon + \xi_k^* + y_k - w^T \varphi(x_k) - b\right]$
  $\nabla_{g_b(x)} \mathcal{G}(\alpha^*, g(x)) = -\alpha_k^*$

All Lagrange multipliers are negative, therefore the condition of monotonically decreasing with respect to $g(x)$ holds.

2. *Concave with respect to τ, respectively, if* $g(x) \leq 0$ Obtaining the second partial derivative for each expression:

- $\alpha g_i(x) = -\alpha_k \left[\epsilon + \xi_k - y_k + w^T \varphi(x_k) + b\right]$
  $\nabla^2_{\alpha} \mathcal{G}(\alpha, g(x)) = 0$
- $\alpha^* g_j(x) = -\alpha_k^* \left[\epsilon + \xi_k^* + y_k - w^T \varphi(x_k) - b\right]$
  $\nabla^2_{\alpha^*} \mathcal{G}(\alpha^*, g(x)) = 0$
- $\alpha g_m(x) = \lambda \left[(1-\epsilon) \sum_{k=1}^{N} \left(\alpha_k + \alpha_k^*\right) + \frac{\epsilon}{2} \sum_{k=1}^{N} \left(\alpha_k + \alpha_k^*\right)^2\right]$
  $\nabla^2_{\alpha} \mathcal{G}(\alpha, g(x)) = \frac{\partial g_m(x)}{\partial \alpha_k} = -\lambda\epsilon$
- $\alpha^* g_m(x) = \lambda \left[(1-\epsilon) \sum_{k=1}^{N} \left(\alpha_k + \alpha_k^*\right) + \frac{\epsilon}{2} \sum_{k=1}^{N} \left(\alpha_k + \alpha_k^*\right)^2\right]$
  $\nabla^2_{\alpha^*} \mathcal{G}(\alpha, g(x)) = \frac{\partial g_m(x)}{\partial \alpha_k^*} = -\lambda\epsilon$

Since the results are either negative or zero, we can conclude that the equation is concave with respect to τ, if $g(x) \leq 0$.

3. $\sup_{\lambda} \mathcal{G}(\lambda, g(x)) = 0, \forall g(x) \leq 0$*, and* $\sup_{\lambda} \mathcal{G}(\lambda, g(x)) = \infty, \forall g(x) > 0$

- $\sup_{\tau} \mathcal{G}(\tau, g(x)) = 0, \forall g(x) \leq 0$
  Having $g(x) \leq 0$ and since Lagrange multipliers are non-negative, it is set bounded by zero. Therefore, the $\sup_{\tau} \mathcal{G}(\tau, g(x)) = 0$.

- $\sup_{\tau} \mathcal{G}(\tau, g(x)) = \infty, \forall g(x) > 0$
  Having $g(x) > 0$ and since Lagrange multipliers are non-negative, the set has no upper bound. Therefore, the $\sup_{\tau} \mathcal{G}(\tau, g(x)) = \infty$.

Now that we have proved that the new Lagrangian proposal for the ϵ-SVR adding an Elastic net regularization term meets the characteristics of a GLMM, we continue to solve the optimization problem.

***First Order Conditions**:*

- The first order condition on the parameter w, $\nabla_w \mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = 0$, implies $w = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\varphi(x_k)$.
- The first order condition on the parameter b, $\frac{\partial}{\partial b}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = 0$, implies $\sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0$.
- The first order condition on the parameter ξ_k, $\frac{\partial}{\partial \xi_k}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = 0$, implies $C\xi_k - \alpha_k = 0$
- The first order condition on the parameter ξ_k^*, $\frac{\partial}{\partial \xi_k^*}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = 0$, implies $C\xi_k^* - \alpha_k^* = 0$

Replacing the first order conditions in the Lagrangian (3.29):

$$
\begin{aligned}
\mathcal{L}\left(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*\right. &= \frac{1}{2}w^T w + \frac{C}{2}\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C^2} + \frac{\alpha_k^{*2}}{C^2}\right) \\
&- \sum_{k=1}^{N}\alpha_k\left(\frac{\alpha_k}{C} - y_k + w^T\varphi(x_k)\right) \\
&- \sum_{k=1}^{N}\alpha_k^*\left(\frac{\alpha_k^*}{C} + y_k - w^T\varphi(x_k)\right) - \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)b \\
&- \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]
\end{aligned} \tag{3.31}
$$

Grouping variables:

$$
\begin{aligned}
\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) &= \frac{1}{2}w^T w + \frac{C}{2}\sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C^2} + \frac{\alpha_k^{*2}}{C^2}\right) \\
&+ \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\
&- \sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C} + \frac{\alpha_k^{*2}}{C}\right) - \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)w^T\varphi(x_k) \\
&- \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]
\end{aligned} \tag{3.32}
$$

Simplifying the squared terms:

$$\begin{aligned}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = \frac{1}{2}w^T w + \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right) \\ + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ - \sum_{k=1}^{N}\left(\frac{\alpha_k^2}{C} + \frac{\alpha_k^{*2}}{C}\right) - \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)w^T\varphi(x_k) \\ - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]\end{aligned} \tag{3.33}$$

Grouping squared terms:

$$\begin{aligned}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = -\frac{1}{2}w^T w - \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right) \\ + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]\end{aligned} \tag{3.34}$$

Replacing the w term:

$$\begin{aligned}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = -\frac{1}{2}\sum_{k=l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\ - \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]\end{aligned} \tag{3.35}$$

Considering the case where $k = l$:

$$\begin{aligned}\mathcal{L}(w, b, \xi_k, \xi_k^*; \alpha_k, \alpha_k^*) = -\frac{1}{2}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right)\varphi^T(x_k)\varphi(x_k) \\ - \frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\varphi^T(x_k)\varphi(x_l) \\ - \frac{1}{2C}\sum_{k=1}^{N}\left(\alpha_k^2 + \alpha_k^{*2}\right) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ - \lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k + \alpha_k^*)^2\right]\end{aligned} \tag{3.36}$$

Grouping terms:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = &-\frac{1}{2}\sum_{k=1}^{N}\left(\alpha_k^2+\alpha_k^{*2}\right)\left(\varphi^T(x_k)\varphi(x_k)+\frac{1}{C}\right)\\ &-\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\varphi^T(x_k)\varphi(x_l)\\ &+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\\ &-\lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)^2\right]\end{aligned} \tag{3.37}$$

Rearranging terms:

$$\begin{aligned}\mathcal{L}(w,b,\xi_k,\xi_k^*;\alpha_k,\alpha_k^*) = &-\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\left[\varphi^T(x_k)\varphi(x_l)+\frac{\delta_{k,l}}{C}\right]\\ &+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)\\ &-\lambda\left[(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\frac{\epsilon}{2}\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)^2\right]\end{aligned} \tag{3.38}$$

where $\delta_{k,l}$ is Kronecker's delta function.

***Primal Feasibility Conditions**:* Recalling the primal constraints

$$\begin{aligned}&y_k - w^T\varphi(x_k) - b \le \epsilon + \xi_k,\ k=1,\dots,N\\ &w^T\varphi(x_k) + b - y_k \le \epsilon + \xi_k^*,\ k=1,\dots,N\end{aligned} \tag{3.39}$$

***Dual Feasibility Conditions**:* Due to the Non-Negative Lagrange Multipliers, we find the following deductions:

$$\begin{aligned}&\alpha_k \ge 0,\ k=1,\dots,N\\ &\alpha_k^* \ge 0,\ k=1,\dots,N\\ &\text{It follows}\\ &C\xi_k = \alpha_k\\ &C\xi_k^* = \alpha_k^*\end{aligned} \tag{3.40}$$

We obtain the following dual problem:

$$\begin{aligned}
&\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\left[\varphi^T(x_k)\varphi(x_l)+\delta_{k,l}\left(\frac{1}{C}-\frac{\lambda\epsilon}{2}\right)\right]\\
&\quad+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)-\lambda(1-\epsilon)\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)\\
&\quad\text{s.t.}\ \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)=0\\
&\quad\alpha_k\geq 0,\ k=1,\ldots,N\\
&\quad\alpha_k^*\geq 0,\ k=1,\ldots,N
\end{aligned}\tag{3.41}$$

Complementary Slackness Conditions: The optimal solution must satisfy the KKT Complementary Slackness Condition:

$$\alpha_k\left(\epsilon+\xi_k-y_k+w^T\varphi(x_k)+b\right)=0\tag{3.42}$$

$$\alpha_k^*\left(\epsilon+\xi_k^*+y_k-w^T\varphi(x_k)-b\right)=0\tag{3.43}$$

$$C\xi_k=\alpha_k,\ k=1,\ldots,N\tag{3.44}$$

$$C\xi_k^*=\alpha_k^*,\ k=1,\ldots,N\tag{3.45}$$

To complete the regression function, we should calculate b. Using the slack complementary conditions.

$$\begin{aligned}
b&=y_k-w^T\varphi(x_k)-\epsilon-\frac{\alpha_k}{C},\quad\text{for }\alpha_k>0\\
b&=y_k-w^T\varphi(x_k)+\epsilon+\frac{\alpha_k^*}{C},\quad\text{for }\alpha_k^*>0
\end{aligned}\tag{3.46}$$

3.3.1 Reformulation of the GLMM L_2^ϵ SVR

Rewriting the L_2^ϵ support vector regressor from (3.41):

$$\begin{aligned}
&\max_{\beta} -\frac{1}{2}\sum_{k,l=1}^{N}\beta_k\beta_l\left(\varphi^T(x_k)\varphi(x_l)+\frac{\delta_{kl}}{C}\right)+\sum_{k=1}^{N}y_k\beta_k\\
&\quad-\lambda\left[(1-\epsilon)\sum_{k=1}^{N}|\beta_k|+\frac{\epsilon}{2}\sum_{k=1}^{N}\|\beta_k\|^2\right]\\
&\quad\text{s.t.}\ \sum_{k=1}^{N}\beta_k=0
\end{aligned}\tag{3.47}$$

Defining $k(x_k,x_l)=\varphi^T(x_k)\varphi(x_l)$, $\beta=\begin{bmatrix}\beta_1 & \ldots & \beta_N\end{bmatrix}^T$, $x=\begin{bmatrix}x_1 & \ldots & x_N\end{bmatrix}^T$, $y=\begin{bmatrix}y_1 & \ldots & y_N\end{bmatrix}^T$ and $1_v=\begin{bmatrix}1 & \ldots & 1\end{bmatrix}^T$.

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix}$$

We can write the (3.47) formulation in a matrix form:

$$\begin{aligned} &\max_{\beta} -\frac{1}{2}\beta^T \left(K + \frac{1}{C} I\right) \beta + y^T \beta \\ &\text{s.t. } \beta^T 1_v = 0 \end{aligned} \tag{3.48}$$

Or equivalently,

$$\begin{aligned} &\min_{\beta} \frac{1}{2}\beta^T \left(K + \frac{1}{C} I\right) \beta - y^T \beta \\ &\text{s.t. } \beta^T 1_v = 0 \end{aligned} \tag{3.49}$$

Conclusion We can observe that this new proposal of ϵ-SVR based on the L_2^ϵ-SVR does not offer a new structure since it is another L_2^ϵ with the same Elastic net regularization and without the box constraints. Therefore, we will not use this new SVR proposal for further study.

4 Application

Contents

4.1 Types of Kernels

Kernel functions are helpful in various situations because they provide a simple bridge from linearity to non-linearity for algorithms that can be represented as dot products.

Kernel methods map data into higher-dimensional spaces to be easier to separate or better structure it in this higher-dimensional space. There are no constraints on the form of this mapping, which means it could lead to infinite-dimensional spaces.

The choice of a Kernel depends on the problem because it depends on what we are trying to model. For example, a polynomial kernel allows us to model feature conjunctions up to the polynomial's order. Radial basis functions allows to pick out circles (or hyper-spheres) – in contrast with the Linear kernel, which allow only to pick outlines (or hyper-planes).

A few kernel functions will be mentioned, along with some of their properties.

4.1.1 Linear Kernel

This one is the simplest kernel function. It is given by the inner product $\langle x, y \rangle$ plus an optional constant **c**.

$$k(x, y) = x^T y + c \tag{4.1}$$

4.1.2 Polynomial Kernel

The Polynomial kernel is a non-stationary kernel. These are well suited for problems where all the training data is normalized.

Adjustable parameters are the slope α, the constant term **c** and the polynomial degree **d**.

$$k(x, y) = \left(\alpha x^T y + c\right)^d \tag{4.2}$$

4.1.3 Gaussian Kernel

The Gaussian kernel is an example of a radial basis function kernel. Where the adjustable parameter sigma has a significant impact on the kernel's efficiency and should be fine-tuned to the specific problem at hand. When the exponential is overestimated, it behaves almost linearly, and the higher-dimensional projection loses its non-linear power. On the other hand, if the function is undervalued, it will lack regularization, and the decision boundary will be extremely sensitive.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{4.3}$$

4.1.4 Exponential Kernel

The exponential kernel is very similar to the Gaussian kernel, except for the square of the norm. It is also a kernel with a radial basis function.

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{2\sigma^2}\right) \tag{4.4}$$

4.1.5 Laplacian Kernel

Except for being less sensitive to changes in the sigma parameter, the Laplace Kernel is identical to the exponential kernel. It is also a radial basis function kernel.

$$k(x,y) = \exp\left(-\frac{\|x-y\|}{\sigma}\right) \tag{4.5}$$

4.1.6 Hyperbolic Tangent (Sigmoid) Kernel

The Hyperbolic Tangent Kernel is also known as the Sigmoid Kernel and as the Multilayer Perceptron (MLP) kernel. The Sigmoid Kernel is derived from Neural Networks, where the bipolar sigmoid function is often used as an artificial neuron activation function.

$$k(x,y) = \tanh\left(\alpha x^T y + c\right) \tag{4.6}$$

It is worth noting that a sigmoid kernel function SVM model is similar to a two-layer perceptron neural network. It has also been discovered that it performs well in practice despite being only conditionally positive definite.

The slope **alpha** and the intercept constant **c** are two customizable parameters in the sigmoid kernel. 1/N, where N is the data dimension, is a common value for alpha.

4.1.7 Kernel Implementation

Linear (4.1) and Gaussian (4.3) Kernels will be implemented in this paper in **Python**, where the free parameter in the model is **sigma**.

```python
import numpy as np

def kernel(x,xt,sigma=0.1,t='rbf'):
    if t == 'linear':
        return (np.dot(x, xt.T))
    else:
        n =x.shape[0]
        nt = xt.shape[0]
        K = np.zeros((n,nt))
        for i in range(n):
            for j in range(nt):
                K[i,j] = np.exp(-np.linalg.norm(x[i,:]-xt[j,:])**2 /(2*sigma
     **2))
        return (K)
```

4.2 Implementation Issues L_1-SVR

Proceeding with the L_1^ϵ-SVR Python implementation using the β reformulation in (2.25).

The ϵ, **c** and σ free hyper-parameters can be tuned using multiple optimization algorithms.

```
#Parameters
e = 0.8
c = 10
sigma = 2
n_train = x_train.shape[0]
onev = np.ones((n_train,1))
Ev = onev*e
error = 1E-5
K = kernel(x_train,x_train,sigma,’rbf’)
#Dual Problem
beta = cp.Variable((n_train,1))
problem = cp.Problem(cp.Minimize((1/2)*cp.quad_form(beta, K)  + (Ev.T) @ (cp
     .atoms.elementwise.abs.abs(beta)) - y_train.T @ (beta)), [onev.T @ (
     beta) == 0, beta >= -c, beta <= c])
problem.solve(solver=’ECOS’)
beta = np.matrix(beta.value)

#Support Vectors
sv = abs(beta) > error
beta_sv = beta[sv].T
n_sv = beta_sv.shape[0]
x_sv = x_train[np.repeat(sv,x_train.shape[1],axis=1)].reshape(n_sv,x_train.
     shape[1])
#Compute b
sb = np.logical_and (abs(beta)> error, abs(beta) < c)
beta_sb =beta[sb].T
n_sb = beta_sb.shape[0]
y_sb = y_train[sb].T
K_sb = K[sb*sb.T].reshape(n_sb,n_sb)
E_sb = np.sign(beta_sb)*e
b = np.mean(y_sb + E_sb - (K_sb*beta_sb))
#prediction model
K_pred_b1 = kernel(x_test,x_sv,sigma,’rbf’)
y_pred_b1= (sum(np.multiply(beta_sv,K_pred_b1.T))+b).T
```

Listing 4.1: L_1^ϵ-SVR

4.3 Implementation Issues L_2-SVR

Proceeding with the L_2^ϵ-SVR Python implementation using the β reformulation in (2.50).

The ϵ, **c** and σ free hyper-parameters can be tuned using multiple optimization algorithms.

```
n_train = x_train.shape[0]
#Parameters
e = 0.8
c = 10
sigma = 2
onev = np.ones((n_train,1))
Ev = onev*e
error = 1E-5
K = kernel(x_train,x_train,sigma,’rbf’)
```

```
#Dual Problem
beta = cp.Variable((n_train,1))
problem = cp.Problem(cp.Minimize((1/2)*cp.quad_form(beta, K+np.identity(
     n_train)/c) + Ev.T @ (cp.atoms.elementwise.abs.abs(beta)) - y_train.T @
     (beta)), [onev.T @ (beta) == 0])
problem.solve(solver='ECOS')
beta = np.matrix(beta.value)
#Support Vectors
alfa = alfa1 - alfa2
sv = abs(alfa) > error
alfa_sv = alfa[sv].T
n_sv = alfa_sv.shape[0]
x_sv = x_train[np.repeat(sv,x_train.shape[1],axis=1)].reshape(n_sv,x_train.
     shape[1])
#Compute b
sb = abs(alfa)> error
#sb = np.logical_and (abs(alfa)> error)
alfa_sb =alfa[sb].T
n_sb = alfa_sb.shape[0]
y_sb = y_train[sb].T
K_sb = K[sb*sb.T].reshape(n_sb,n_sb)
E_sb = np.sign(alfa_sb)*e
b = np.mean(y_sb + E_sb - (K_sb*alfa_sb)+alfa_sb/c)
#prediction model
K_pred = kernel(x_test,x_sv,sigma,'rbf')
y_pred= (sum(np.multiply(alfa_sv,K_pred.T))+b).T
```

Listing 4.2: L_2^{ϵ}-SVR

4.4 Implementation Issues New SVR

Proceeding with the New GLMM SVR Python implementation using the β reformulation in (3.28).

The free hyper-parameters ϵ, $\mathbf{c}$, σ and the new parameter λ can be tuned using multiple optimization algorithms.

```
n_train = x_train.shape[0]
#Parameters
e = 0.8
c = 10
sigma = 2
sigma = 2
lam = .2
onev = np.ones((n_train,1))
Ev = onev*e
error = 1E-5
K = kernel(x_train,x_train,sigma,'rbf')
#Dual Problem
beta = cp.Variable((n_train,1))
problema = cp.Problem(cp.Minimize((1/2)*cp.quad_form(beta, K) +lam*((1-Ev.T)
      @ (cp.atoms.elementwise.abs.abs(beta)) + (Ev.T)/2 @ (cp.atoms.
     elementwise.abs.abs(beta))**2) - y_train.T @ (beta)), [onev.T @ (beta)
     == 0, beta >= -c, beta <= c])
problema.solve(solver='ECOS')
beta = np.matrix(beta.value)
#SV
sv = abs(beta) > error
```

```
beta_sv = beta[sv].T
n_sv = beta_sv.shape[0]
x_sv = x_train[np.repeat(sv,x_train.shape[1],axis=1)].reshape(n_sv,x_train.
     shape[1])
#Compute b
sb = abs(beta)> error
beta_sb =beta[sb].T
n_sb = beta_sb.shape[0]
y_sb = y_train[sb].T
K_sb = K[sb*sb.T].reshape(n_sb,n_sb)
E_sb = np.sign(beta_sb)*e
b = np.mean(y_sb + E_sb - (K_sb*beta_sb)+beta_sb/c)
#prediction model
K_pred = kernel(x_test,x_sv,sigma,'rbf')
y_pred= (sum(np.multiply(beta_sv,K_pred.T))+b).T
```

Listing 4.3: New-SVR

4.5 Hyper-parameter Tuning

It is a difficult task the selection of optimal hyper-parameter values, e.g., C, ϵ and σ values for Radial Based Function (RBF) kernels and λ for the regularization.

As it is a hyper-parameter, there is no way of knowing in advance which value is appropriate. A Bayesian Optimization method has been proposed for automatic parameter selection. With Python's **Bayesian Optimization** package, this can be achieved.[1]

4.5.1 Bayesian Optimization

Bayesian optimization generates a posterior distribution of functions (Gaussian process) that better describes the function we want to improve. The posterior distribution improves as the number of observations increase, and the algorithm becomes more confident in determining which regions of parameter space are worth exploring and which are not.

The algorithm balances its discovery and exploitation needs as we iterate, considering what we know about the target feature. A Gaussian Process is fitted to the known samples (points previously explored) at each stage, and the posterior distribution is combined with an exploration strategy (such as UCB (Upper Confidence Bound) or EI (Expected Improvement)) to decide the next point to be explored.

This procedure is designed to reduce the number of steps taken to find a parameter combination that is close to optimal. This approach accomplishes this by employing a proxy optimization problem (finding

[1] This is a constrained global optimization package built upon Bayesian inference and Gaussian process that attempts to find the maximum value of an unknown function in as few iterations as possible. This technique is particularly suited for optimization of high-cost functions, situations where the balance between exploration and exploitation is important.

the limit of the acquisition function), which, while still a complex problem, is less expensive (in terms of computation) and can be solved with common resources.

4.5.2 *Hyper-parameter Tuning Implementation*

We are implementing the Bayesian Optimization method in Python. The first thing to do is to create a function that uses the previously seen SVR models where the goal is to maximize the R^2 metric. After maximizing this metric, we apply the Bayesian Optimizer to fit and extract the best hyper-parameters that maximize the goal function.

The free hyper-parameters ϵ, $\mathbf{c}$, σ and the new parameter λ can be now tuned.

```python
#libraries
from bayes_opt import BayesianOptimization
import numpy as np
import pandas as pd
import cvxpy as cp
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.svm import SVR
import time

def estimador(c,sigma, e, lam):
    n_train = x_train.shape[0]
    onev = np.ones((n_train,1))
    Ev = onev*e
    error = 1E-5
    K = kernel(x_train,x_train,sigma,'rbf')
    #Dual Problem
    alfa1 = cp.Variable((n_train,1))
    alfa2 = cp.Variable((n_train,1))
    problem = cp.Problem(cp.Minimize((1/2)*cp.quad_form(alfa1-alfa2, K) +lam
     *((1-Ev.T) @ (alfa1+alfa2) + (Ev.T)/2 @ (alfa1+ alfa2)**2) - y_train.T
     @ (alfa1 - alfa2)), [onev.T @ (alfa1-alfa2) == 0, alfa1 >= 0, alfa1 <=
     c,alfa2 >= 0, alfa2 <= c])
    problem.solve(solver='ECOS')
    alfa1 = np.matrix(alfa1.value)
    alfa2 = np.matrix(alfa2.value)
    #Support vectors
    alfa = alfa1 - alfa2
    sv = abs(alfa) > error
    alfa_sv = alfa[sv].T
    n_sv = alfa_sv.shape[0]
    x_sv = x_train[np.repeat(sv,x_train.shape[1],axis=1)].reshape(n_sv,
     x_train.shape[1])
    #Compute b
    sb = np.logical_and (abs(alfa)> error, abs(alfa) < c)
    alfa_sb =alfa[sb].T
    n_sb = alfa_sb.shape[0]
    y_sb = y_train[sb].T
    K_sb = K[sb*sb.T].reshape(n_sb,n_sb)
    E_sb = np.sign(alfa_sb)*e
    b = np.mean(y_sb + E_sb - (K_sb*alfa_sb))
    K_pred = kernel(x_train,x_sv,sigma,'rbf')
    y_pred1= (sum(np.multiply(alfa_sv,K_pred.T))+b).T
    return metrics.r2_score(y_train, y_pred1 )

#Hyper-parameters
hparams = {"c": (0.1, 10), "sigma": (0.001, 1), "e":(0.1, 1), "lam":(0.2,1)}

# give model and hyperparameter to optmizer
bayes_svr = BayesianOptimization(estimador, hparams)
# maximize means optimization
start_time = time.time()
bayes_svr.maximize(init_points=10, n_iter=10)
tiempo = time.time() - start_time
print(’Time’, tiempo)
c = bayes_svr.max[’params’][’c’]
sigma = bayes_svr.max[’params’][’sigma’]
e = bayes_svr.max[’params’][’e’]
lam=bayes_svr.max[’params’][’lam’]
```

Listing 4.4: Hyper-parameter tuning

5 Examples

Contents

5.1 Linear Equation

Proposing the next equation:

$$y = 3x_1 - 2x_2 + 3x_3 - 10 \tag{5.1}$$

where $x_1 = 5\omega$, $x_2 = 10\omega$, $x_3 = 15\omega$, and $n = 1000$ records.

After using Bayesian Optimization to maximize the R^2 and finding the following hyper-parameters:

- L_1^ϵ-SVR and L_1^ϵ-SVR with Linear Kernel
    - $\epsilon = 0.1$
    - $c = 7.3439$
    - Time elapsed: $t = 63.6536$ minutes
- New GLMM SVR with Linear Kernel
    - $\epsilon = 1$
    - $c = 10$
    - $\lambda = 0.2$
    - Time elapsed: $t = 58.2876$ minutes
- L_1^ϵ-SVR and L_1^ϵ-SVR with RBF Kernel
    - $\epsilon = 0.1$
    - $c = 8.83349$
    - $\sigma = 7.8221$

  - Time elapsed: $t = 72.53$ minutes

- New GLMM SVR with RBF Kernel
  - $\epsilon = 1$
  - $c = 3.1517$
  - $\sigma = 8.7241$
  - $\lambda = 0.2$
  - Time elapsed: $t = 72.3878$ minutes

Using different seeds to partition the data with a training set of 80% and a test set of 20%, we obtain the following results.

Model/Metric	New SVR RBF	L_1^ϵ SVR RBF	L_2^ϵ SVR RBF	New SVR Linear	L_1^ϵ SVR Linear	L_2^ϵ SVR Linear
Avg. R^2 Test	0.986111	0.985626	0.958656	0.996797	0.996587	0.996599
Avg. MSE Test	3.217	3.3159	9.8694	0.744273	0.746314	0.742797

Table 5.1: Metric Performance Comparison for Linear Equation

The new SVR proposal has the highest R^2 metric. For this example, since it is a linear equation, the Linear Kernel outperforms the other models.

5.2 *Bicycle Sharing Demand*

Bike sharing systems are a form of bicycle rental service in which the process of obtaining a membership, renting a bike, and returning the bike is all automated via a network of kiosks located throughout a city. People can rent a bike from one location and return it to a different location as required using these systems. There are currently over 500 bike-sharing schemes operating around the world.

Since the length of flight, departure place, arrival location, and time elapsed are all clearly documented by these systems, they are appealing to researchers. As a result, bike-sharing networks serve as a sensor network that can be used to research urban mobility.

The data set has daily rental data spanning two years, containing 731 records and 6 variables. [1]

1. **season:** 1 = spring, 2 = summer, 3 = fall, 4 = winter
2. **weather:** 1: Clear, Few clouds, Partly cloudy, Partly cloudy
   2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
   3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light

[1] Data available in `https://www.kaggle.com/c/bike-sharing-demand`

Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

3. **temp:** temperature in Celsius

4. **humidity:** relative humidity

5. **windspeed:** wind speed

6. **count:** number of total rentals

After using Bayesian Optimization for hyper-parameter tuning (maximize R^2 metric), and standardizing the data, we predict how many bikes will be rented on a daily basis using the following hyper-parameters:

- L_1^ϵ-SVR and L_1^ϵ-SVR with Linear Kernel
  - $\epsilon = 0.569195$
  - $c = 8.1578$
  - Time elapsed: $t = 23.46$ seconds
- New GLMM SVR with Linear Kernel
  - $\epsilon = 0.22799$
  - $c = 3.94$
  - $\lambda = 0.56670$
  - Time elapsed: $t = 7.86$ seconds
- L_1^ϵ-SVR and L_1^ϵ-SVR with RBF Kernel
  - $\epsilon = 0.1$
  - $c = 9.3374$
  - $\sigma = 8.173$
  - Time elapsed: $t = 11.06$ minutes
- New GLMM SVR with RBF Kernel
  - $\epsilon = 0.1$
  - $c = 9.337443$
  - $\sigma = 8.173$
  - $\lambda = 0.4$
  - Time elapsed: $t = 14.71$ minutes

Using different seeds to partition the data with a training set of 80% and a test set of 20%, we obtain the following results.

Model/Metric	New SVR RBF	L_1^ϵ SVR RBF	L_2^ϵ SVR RBF	New SVR Linear	L_1^ϵ SVR Linear	L_2^ϵ SVR Linear
Avg. R^2 Test	0.59382	0.582765	0.533532	0.529133	0.531920	0.525974
Avg. MSE Test	0.402037	0.431260	0.481244	0.486925	0.483891	0.490282

Table 5.2: Metric Performance Comparison for Bike Sharing Demand

The new SVR proposal with the RBF Kernel has the highest R^2 metric. For this example, we can assume that the variables have a non-linear relationship, therefore the RBF Kernel outperforms the other models.

5.3 Dispersion Diagram

Setting the next equation:

$$y = 3x^2 + 2x + 1 + N(0, \gamma) \tag{5.2}$$

where $x = 5\omega$, and $n = 100$ records.

Using the following hyper-parameters and the RBF Kernel:

- $\epsilon = 0.8$
- $c = 10$
- $\sigma = 2$
- $\lambda = 0.1$

Partitioning the data with a training set of 80% and a test set of 20%, we obtain the following results using different sizes of the standard deviation for the random normal variable.

R^2/γ	0	10	50	100	1000
L_1^ϵ-SVR RBF	0.986816	0.832790	0.173668	-0.010470	-0.026809
L_2^ϵ-SVR RBF	0.997083	0.842714	0.275697	-0.264729	-0.028
New SVR	0.985309	0.832647	0.169981	-0.213016	-0.024490
Scikit-Learn	0.989723	0.834180	0.166574	-0.310142	-0.019583

Table 5.3: Metric Performance Comparison with different Standard Deviance for Quadratic Linear Equation

Since this is a quadratic equation, the L_2^ϵ-SVR performs better than the other models, even if the standard deviation increases.

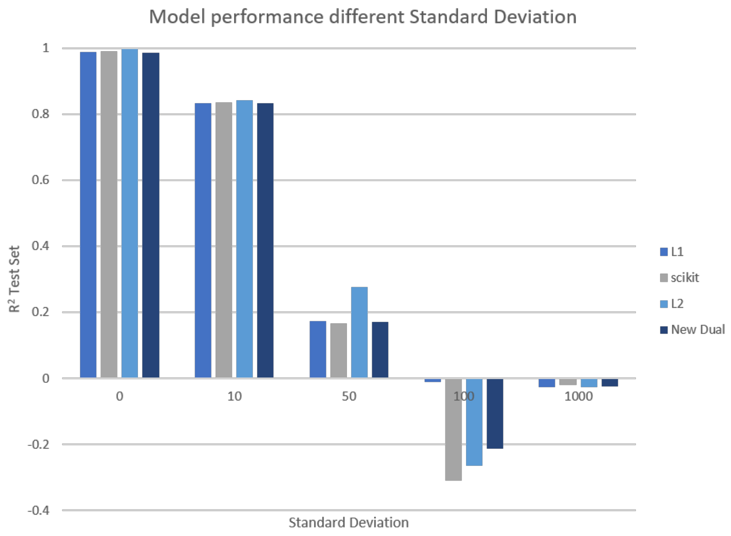

Figure 5.1: Metric performance comparison with different Standard Deviations

6 Conclusion

This paper has proposed a generalized Lagrange multiplier method and derived generalized KKT conditions for support vector regression based on the L_1^ϵ SVR formulation, which includes a weighted elastic net regularization structure. We showed that the extended Lagrange SVR models outperform the classic SVR models in predicting different use cases. A disadvantage of this new model proposal would be the increasing time of the optimization for the new hyper-parameter λ but on the other side, the advantage is that this new elastic net structure gives the possibility to reduce the number of support vectors used to create the model.

Bibliography

S. Boyd and L.Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3.

Mengmou Li. Generalized lagrange multiplier method and kkt conditions with an application to distributed optimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(2):252–256, 2019. DOI: 10.1109/TCSII.2018.2842085.