

Contents

Project 10 - Description	2
Introduction	3
SVR and Level Bundle Method	3
Support Vector for Regression (SVR)	3
Overview of the Level Bundle Method	4
Reformulating the Problem for <code>quadprog</code> in MATLAB	4
Quadratic Norm Expansion	5
Reformulation into <code>quadprog</code> Format	5
Constraints	6
Bundle Cuts Constraint	6
Equality Constraint	7
Bound Constraints	7
General Notes and Considerations	7
Conclusion	7

Project 10 - Description

(**M**) is a SVR-type approach of your choice (in particular, with one or more kernels of your choice).

(**A1**) is an algorithm of the class of level bundle methods, applied to either the primal or the dual formulation of the SVR.

(**A2**) is a general-purpose solver applied to an appropriate formulation of the problem.

Use of an off-the-shelf solver for the Master Problem of the bundle method is allowed.

Introduction

SVR and Level Bundle Method

This project focuses on the development and implementation of an SVR (Support Vector for Regression) capable of learning from a dataset in the form of “feature x target,” where “target” must be a vector of dimensions , in accordance with the definition of SVR. In addition to the basic implementation, a significant part of the SVR will leverage the Level Bundle Method for optimizing the dual function.

For the Master Problem of the SVR, the MATLAB function quadprog was used. This function is primarily designed for solving quadratic objective functions with linear terms.

Support Vector for Regression (SVR)

Support Vector Regression (SVR) aims to find a function $f(X)$ that approximates the training data while minimizing a given loss function. The primal optimization problem is formulated as follows:

$$\begin{aligned} \min_{w, b, \xi, t^*} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{s.t.:} \quad & y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i, \\ & \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}$$

where:

- w and b define the regression hyperplane
- ξ_i, ξ_i^* are slack variables that account for deviations beyond the margin ε
- C is a regularization parameter

By applying Lagrange multipliers and transforming the problem into its dual formulation, we obtain:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to:} \quad & \sum_{i=1}^n \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, n. \end{aligned}$$

where $K(x_i, x_j)$ is a kernel function that allows the method to handle non-linear relationships by implicitly mapping the input data to a higher-dimensional space.

Once the dual problem is solved, the **support vectors** are identified as the data points corresponding to nonzero Lagrange multipliers. Specifically, the support vectors are found by selecting the indices (i, j) that satisfy the condition:

$$\{(i, j) \mid |\alpha_i| > \tau\}$$

where τ is a small positive threshold to account for numerical precision. These support vectors are the most influential data points in defining the regression function, as they determine the final predictive model.

Overview of the Level Bundle Method

The Level Bundle Method (LBM) is an optimization approach that refines solutions iteratively by leveraging cutting-plane techniques and a level constraint. It is particularly useful in non-differentiable optimization problems, such as those encountered in support vector regression.

Reformulating the Problem for quadprog in MATLAB

Since the SVR function defined before is **non-differentiable**, we apply the **Level Bundle Method** to approximate it iteratively. Especially we find a new solution by solving the LBM objective function defined as follow:

$$\alpha_{k+1} = \arg \min_{\alpha} \left\{ \frac{1}{2} \|\alpha - \hat{\alpha}_k\|^2 \mid \hat{f}_k(\alpha) \leq f_k^{\text{level}}, \alpha \in X \right\}$$

where:

- α represents the vector of dual variables in the **Support Vector Regression (SVR)** problem.
- $\hat{\alpha}_k$ is the best solution found so far at iteration k .
- f_k^{level} is the current level used to restrict the search space within an acceptable region.
- X is the set of original constraints of the SVR dual problem.
- \hat{f}_k is defined as follow:

$$\hat{f}_k(x) := \max_{j \in \mathcal{B}_k} \{f(x_j) + \langle \xi_j, x - x_j \rangle\}$$

Where:

$$f(x) = \frac{1}{2}x^\top Kx + \varepsilon \sum_{i=1}^n |x_i| - y^\top x$$

$$\xi = Kx + \epsilon \cdot \text{sign}(x) - y$$

$$\text{sign}(x) = \begin{cases} \{1\}, & \text{if } x > 0, \\ \{-1\}, & \text{if } x < 0, \\ [-1, 1], & \text{if } x = 0. \end{cases}$$

However, from MATLAB documentaion, the **quadprog** general function is designed to solve quadratic optimization problems in the following standard form:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^\top Hx + f^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & A_{\text{eq}}x = b_{\text{eq}}, \\ & lb \leq x \leq ub. \end{aligned}$$

Therefore, it is necessary to reformulate the **Level Bundle Method** problem so that it is compatible with the form required by **quadprog**.

Quadratic Norm Expansion

The objective function of the Level Bundle Method can be rewritten explicitly as follows:

$$\frac{1}{2}\|\alpha - \hat{\alpha}_k\|^2 = \frac{1}{2}(\alpha^\top \alpha - 2\hat{\alpha}_k^\top \alpha + \hat{\alpha}_k^\top \hat{\alpha}_k) = \frac{1}{2}\alpha^\top \alpha - \hat{\alpha}_k^\top \alpha + \frac{1}{2}\hat{\alpha}_k^\top \hat{\alpha}_k.$$

Since the constant term $\frac{1}{2}\hat{\alpha}_k^\top \hat{\alpha}_k$ does not affect the minimization, we can omit it. The function to minimize then becomes:

$$\frac{1}{2}\alpha^\top \alpha - \hat{\alpha}_k^\top \alpha.$$

Reformulation into quadprog Format

By comparing this expression with the standard objective function solved by **quadprog**:

$$\frac{1}{2}x^\top Hx + f^\top x,$$

we obtain the following parameters:

- $H = I$ (identity matrix, since the quadratic term is $\frac{1}{2}x^\top x$).
- $f = -\hat{x}_k$ (since the linear term is $-\hat{\alpha}_k^\top \alpha$).

Substituting these values, we obtain:

$$\frac{1}{2}x^\top Hx + f^\top x = \frac{1}{2}x^\top x - \hat{x}_k^\top x.$$

That is exactly the function that we want to minimize.

Constraints

The Level Bundle Method algorithm is subject to the following constraints:

s.t. :

$$\begin{aligned} \hat{f}_k(\alpha) &\leq f_k^{\text{level}} \\ \sum_{i=1}^n \alpha_i &= 0 \\ 0 &\leq \alpha_i \leq C \quad \forall i = 1, \dots, n \end{aligned}$$

Thus, they must be rewritten into a form solvable by `quadprog`, just as we did for the objective function.

Bundle Cuts Constraint

Given the following constraint:

$$\hat{f}_k(\alpha) \leq f_k^{\text{level}}$$

We can implement it through the cutting plane approach, where the function is approximated via a collection of tangent hyperplanes. These hyperplanes are defined by the subgradients of the function and translate into linear constraints of the form:

$$f(\alpha_k) + \langle \xi_k, \alpha - \alpha_k \rangle \leq f^{\text{level}}$$

Where ξ_j are the subgradients computed at those points and $f(\hat{\alpha}_k)$ is the real objective value evaluated.

This formulation must be rewritten in the form $Ax \leq b$. Rewriting the constraint:

$$f(\alpha_k) + \langle \xi_k, \alpha - \alpha_k \rangle \leq f^{\text{level}} \iff \langle \xi_k, \alpha \rangle - f^{\text{level}} \leq \langle \xi_k, \alpha_k \rangle - f(\alpha_k).$$

With:

$$A = [\xi_k^\top \quad -1], \quad b = \langle \xi_k, \alpha_k \rangle - f(\alpha_k).$$

Note that the current level f_k^{level} is updated at each iteration using the following formula:

$$f_{\text{level}} = \theta f + (1 - \theta) f_{\text{best}}$$

Equality Constraint

$$\sum_{i=1}^n \alpha_i = 0.$$

This can be defined directly as:

$$A_{\text{eq}} = [\mathbf{1}^\top], \quad b_{\text{eq}} = 0.$$

Bound Constraints

$$0 \leq \alpha_i \leq C,$$

These can be defined as:

$$\begin{aligned} lb &= [0, -\infty], \\ ub &= [C, t_{\text{level}}]. \end{aligned}$$

General Notes and Considerations

As previously described, this project aims to implement an SVR that leverages the Level Bundle Method for optimization. However, the goal is not to achieve a fully optimized SVR in terms of hyperparameters or generalization performance on the dataset cause this would require additional tuning techniques such as grid search or k-fold cross-validation, which are beyond the scope of this project.

Nevertheless, we will present the error achieved using the Mean Squared Error (MSE) metric, along with the selected hyperparameters. Additionally, we will compare our implementation against MATLAB's native SVR solver, which will be used as an oracle. Although our goal is not to match the oracle's performance, this comparison will provide useful insights into the effectiveness of our approach.

Conclusion

We have demonstrated how the objective function of the **Level Bundle Method** can be rewritten in the standard form required by `quadprog`. In the following sections, we will present the implementation of the algorithm following the steps outlined so far.