Report for Exercise 4 in WSI

Adam Szostek, Index Number: 331443

February 11, 2025

1 Description of the Implemented Algorithm

Support Vector Machine (SVM) is a classification algorithm that involves finding a hyperplane that best separates data from two classes. The goal of SVM is to maximize the margin—the distance between the hyperplane and the nearest points from each class, which are called support vectors.

1.1 Application of the Dual Problem

To solve the SVM optimization problem and utilize the kernel trick, the dual form of the optimization problem is formulated. Starting from the primal problem of maximizing the margin:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N.$$

To obtain the dual form, we introduce the Lagrange function:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{N} \beta_i \xi_i$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are the Lagrange multipliers. We minimize this with respect to $\mathbf{w}$, b, and ξ to obtain the dual form.

After these steps, we obtain the dual objective function:

$$\max_{\alpha} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right)$$

subject to:

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

where C is the regularization parameter.

1.2 Solving the Dual Problem

For implementation in Python, the `cvxopt` solver was used, which solves the optimization problem using the interior-point method in the form:

$$\min_{x} \quad \frac{1}{2} x^T P x + q^T x$$

subject to:

$$Gx \leq h, \quad Ax = b.$$

The dual objective function is presented in the minimization form:

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i$$

subject to:

$$\sum_{i=1}^{N} \alpha_i y_i = 0,$$
$$0 \leq \alpha_i \leq C \quad \forall i = 1, \ldots, N.$$

Thus, after converting it to the appropriate form for the chosen solver, we obtain:

$$P_{ij} = y_i y_j K(x_i, x_j),$$

$$q_i = -1,$$
$$G = \begin{bmatrix} -I \\ I \end{bmatrix},$$
$$h = \begin{bmatrix} 0 \\ C\mathbf{1} \end{bmatrix},$$
$$A = \begin{bmatrix} y_1 & y_2 & \ldots & y_N \end{bmatrix},$$
$$b = 0,$$

which represent:

- P: Gram matrix (kernel matrix), representing the inner product of vectors in feature space, taking class labels into account.
- q: Vector of coefficients in the objective function, being a vector of negative ones since we are maximizing the objective function.
- G: Matrix of inequality constraints, ensuring that $\alpha_i \geq 0$ and $\alpha_i \leq C$ for all samples.
- h: Vector of values for the constraints in matrix G, corresponding to $\alpha_i \geq 0$ (0) and $\alpha_i \leq C$ (C).
- A: Matrix of equality constraints, containing class labels y_i, ensuring that the weighted sum is equal to 0.
- b: Vector of equality constraints, containing the value 0, corresponding to the condition $\sum_i \alpha_i y_i = 0$.

The solver returns the solution α, which minimizes the given objective function while satisfying the above constraints.
After solving the dual problem, we obtain the values of the Lagrange multipliers α_i. The indices for which α_i are non-zero (with a tolerance of 10^{-5}) correspond to the support vectors.

1.3 Calculating the Bias b, Decision Function, and Prediction

The bias b is calculated as the average of the differences between the labels of the support vectors and their dot products with other support vectors:

$$b = \frac{1}{|\text{support vectors}|} \sum_i \left(y_i - \sum_j \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right)$$

The decision function takes the form:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$$

Based on the decision function, a new point $\mathbf{x}$ is classified using:

$$\text{prediction} = \text{sign}(f(\mathbf{x}))$$

1.4 Implemented Kernels

The RBF (Radial Basis Function) kernel is defined by the formula:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ is the squared Euclidean distance between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is the kernel parameter that controls the width of the function.

The linear kernel is defined by the formula:

$$K(x_i, x_j) = x_i^\top x_j$$

2 Planned Numerical Experiments

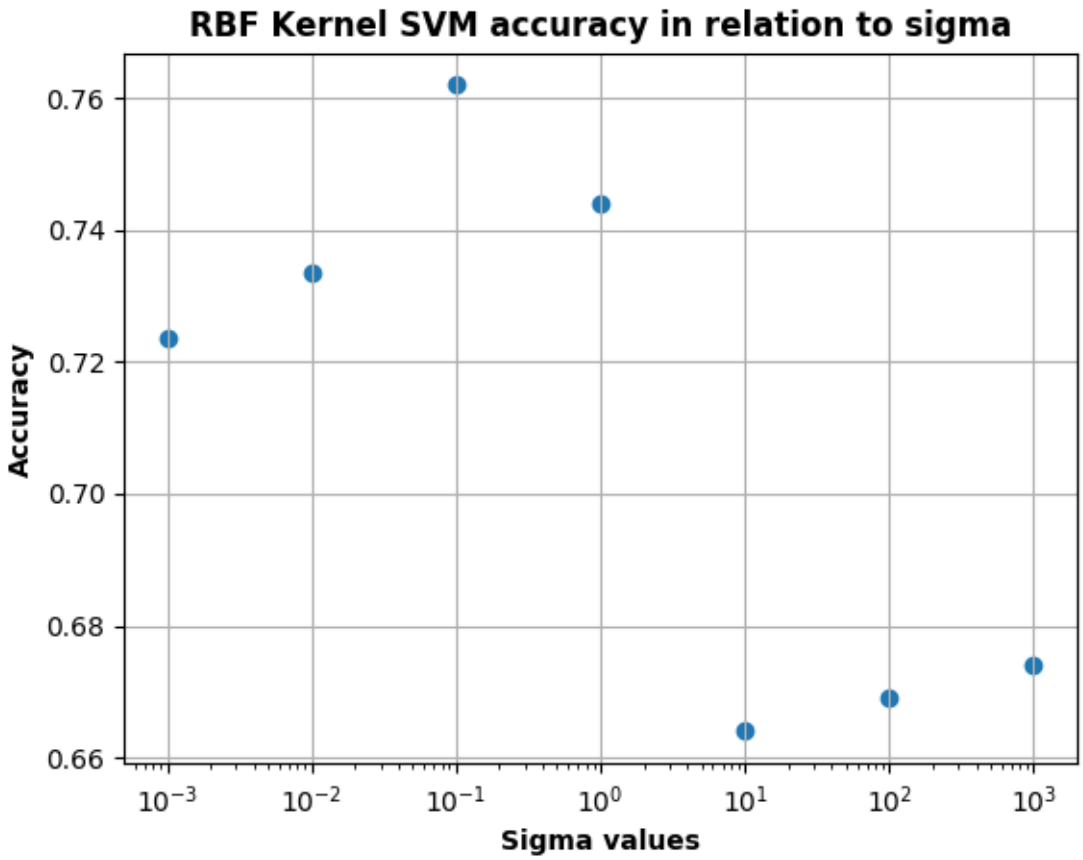
To test the implementation of the SVM algorithm, the "Wine Quality" dataset was used. To adapt the dataset to a binary classification problem, the target variable `quality` was discretized as follows for each y:

$$f(y) = \begin{cases} -1 & \text{for } y \in [1, 5], \\ 1 & \text{for } y \in [6, 10]. \end{cases}$$

The study results were averaged over five training model trials, with each trial using a randomly selected sample of 2000 training pairs. A train-test split was applied using the `train_test_split` function from the `sklearn` library. 80% of the data constitutes the training set, and 20% the testing set. Additionally, a study was conducted to find the best value of the sigma parameter for the RBF kernel, and then in a subsequent experiment concerning the parameter C, this value was used.

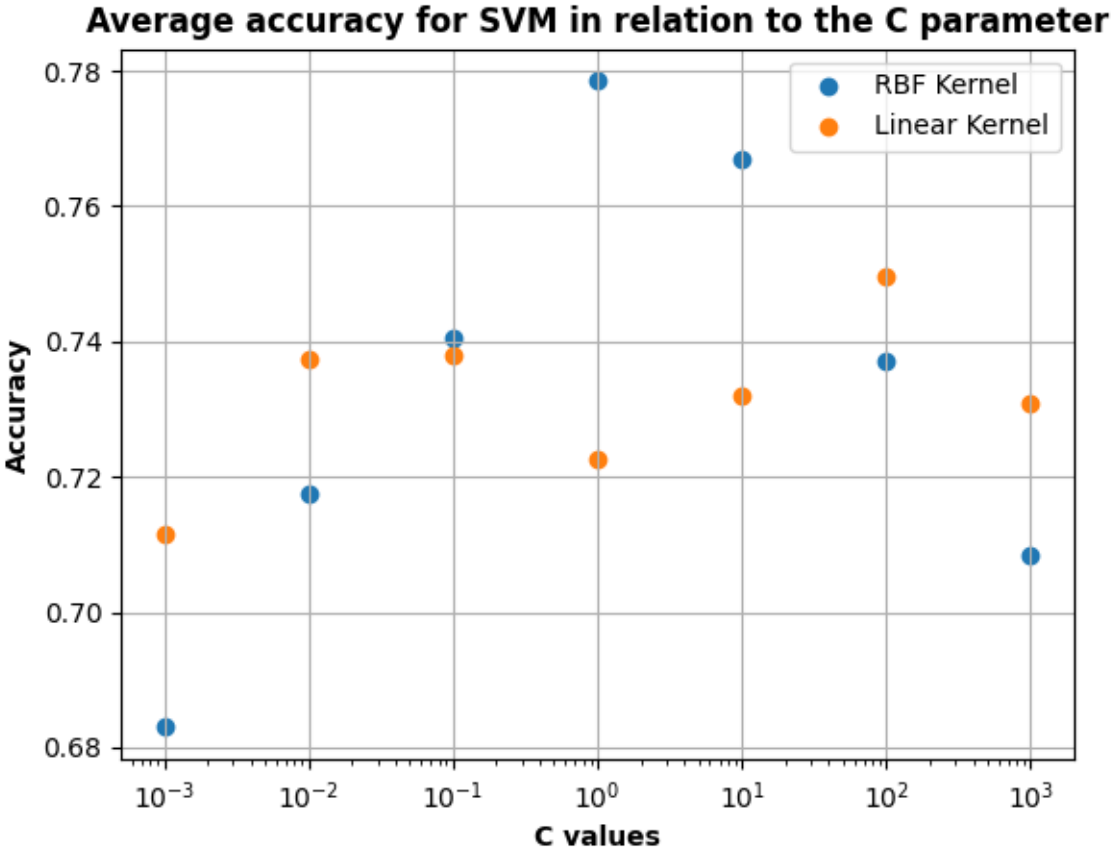
3 Obtained Results

3.1 Sigma Parameter in the RBF Kernel



Analyzing the influence of the sigma parameter value on the accuracy of SVM with the RBF kernel, it can be observed that the optimal range is $[10^{-3}, 10^{0}]$, with the highest results achieved for values 10^{-1} and 10^{0}. At the same time, it is easy to notice that from 10^{1}, the algorithm achieves significantly lower accuracy. In further studies for the RBF kernel, $\sigma = 10^{-1}$ was used.

3.2 Parameter C Study



When analyzing the accuracy of SVM for RBF and linear kernels, we see that the linear kernel behaves much more stably when changing C compared to RBF. However, ultimately the highest accuracy is achieved by RBF at $C = 10^0$. For $C = 10^1$, the result is similar, but other values of SVM with the RBF kernel are significantly worse.

4 Conclusions from the Conducted Experiments

Based on the observations from the studies, the following conclusions can be drawn. The selection of the σ parameter is crucial for the RBF kernel, and the optimal value may vary depending on the dataset. However, it should be limited to choosing σ values smaller than 10^0, as for larger values, the effectiveness of SVM with the RBF kernel significantly decreases.

For both kernels, the parameter C has a significant impact on the achieved accuracy. In the case of the linear kernel, the results are quite stable, but better values can be distinguished (e.g., 10^{-2}, 10^{-1}, 10^2). For the RBF kernel, C values less than or equal to 10^{-2} and greater than 10^3 provide worse results, which in the first range may be caused by excessive model simplification, and in the second range by the model overfitting to the training set. Ultimately, the RBF kernel achieves better accuracy than the linear one but is much more sensitive to changes in the parameter C. In summary, the RBF kernel is a better choice for more complex problems but requires more precise parameter tuning.