Support Vector Machines Using Stochastic Gradient Descent

Mohamed Ahmed Abdulrahman Omar Yasmin Hegy Moamen Elsayed Abdulrahman Khalid

Abstract:

Support Vector Machines (SVMs) are widely used for classification and regression tasks in machine learning. This report presents a mathematical formulation of SVMs and their optimization using Stochastic Gradient Descent (SGD). The derivation of the SVM objective function, hinge loss, and SGD updates are provided in detail. Additionally, highlighting their respective advantages and limitations. The report concludes with insights into the suitability of each method for different types of datasets.

Keywords: Support Vector Machines, Stochastic Gradient Descent, Hinge Loss, Logistic Regression, Optimization, Classification.

1. Introduction

Support Vector Machines (SVMs) are a class of supervised learning algorithms that aim to find the optimal hyperplane separating data points of different classes[1]. While SVMs are powerful, their optimization can be computationally intensive, especially for large datasets[2]. Stochastic Gradient Descent (SGD) is an efficient optimization technique that updates model parameters incrementally, making it suitable for large-scale problems[1]. This report explores the mathematical foundations of SVMs and their optimization using SGD

3.Methodology

3.1 Soft Margin SVM

In situations where data cannot be perfectly separated, the Soft Margin SVM is used to permit some misclassifications [1]. Unlike the Hard Margin SVM, which insists that all data points lie on the correct side of the margin, the Soft Margin SVM introduces slack variables to account for errors. This method is especially beneficial for datasets that include noise or have overlapping classes [2],[3].

3.1.1 Mathematical Formulation

The Soft Margin SVM adjusts the primal optimization problem by introducing slack variables ξ_i for each data point. The objective function is modified as follows:

$$min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \quad (1)$$

$$\text{S.T.} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \forall i \quad (1a)$$

Where:

w is the weight vector defining the hyperplane

b is the bias term

ξ_i are the slack variables that measure the degree of misclassification for each data point

C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors.

3.1.2 Lagrangian Formulation

In order to solve the inequality constrained problem, lagrange multiplier must be used $\alpha_i \geq 0 \; and \; \mu_i \geq 0$:

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2}||w||^2 + c\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i\left[y_i(w \cdot x_i + b) - 1 + \xi_i\right] - \sum_{i=1}^{n} \mu_i \xi_i \quad (2)$$

3.1.3 The Karush-Kuhn-Tucker (KKT) conditions

1. $\nabla_w L = 0$ (3)
2. $\frac{\partial L}{\partial b} = 0$ (4)
3. $\frac{\partial L}{\partial \xi_i} = 0$ (5)
4. $\alpha_i\left[y_i(w \cdot x_i + b) - 1 + \xi_i\right] = 0$ (6)

3.1.4 Dual Problem

$$max \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (7)$$

S.T. $0 \leq \alpha_i \leq C \quad \forall i = 1, 2, \ldots, n$ (7a)

3.1.5 Role of the Regularization Parameter C

The parameter C plays a critical role in balancing the two objectives of the Soft Margin SVM:

A **large value of** C emphasizes minimizing classification errors, potentially leading to a smaller margin.

A **small value of** C prioritizes maximizing the margin, allowing for more misclassifications.

3.1.6 Hinge Loss Interpretation

The Soft Margin SVM can also be interpreted in terms of the **hinge loss function**, which is defined as:

$$max(0,\ 1 - y_i(w^T x_i + b)) \quad (8)$$

The hinge loss penalizes misclassified points and points within the margin. The objective function can thus be rewritten as:

$$min_{w,b} \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i + b)) \quad (9)$$

3.2 Stochastic Gradient Descent (SGD) in Support Vector Machines (SVM)

a simple yet very efficient approach to fitting linear classifiers under convex loss functions such as (linear) Support Vector Machines [10]. In some scenarios where datasets are large, Stochastic Gradient Descent (SGD) is employed to optimize Support Vector Machine (SVM) models by randomly selecting samples for each iteration. Unlike Batch Gradient Descent, which uses the entire dataset to compute gradients, SGD focuses on a single data point at each step, making it more efficient for large-scale problems [7] [11].

3.2.1 Mathematical Formulation

Stochastic gradient descent is an optimization method for unconstrained optimization problems. In contrast to (batch) gradient descent, SGD approximates the true gradient, by considering a single training example at a time.The objective function for SVM can be expressed as:

$$w \leftarrow w - \eta \left[\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial L(w^T x_i + b, y_i)}{\partial w} \right] \quad (10)$$

The equation describes how to update the weights w by moving in the direction that reduces the total cost, which is a combination of the regularization and loss functions.

Where:

w: This is the weight vector of the model, which consists of the parameters that the model learns during training.

- η is the learning rate, controlling the step size taken towards the minimum of the cost function

α : a coefficient that balances the influence of the regularization term relative to the loss term.

$R(w)$: **regularization function**. Regularization techniques, like L1 or L2 regularization, are used to prevent overfitting by penalizing large weights.

$\partial L(w^T x_i + b, y_i)$: **loss function** that measures how well the model's predictions match the actual target values.

y_i : true label

b: bias term

$w^T x_i + b$: predicted value

$\frac{\partial R(w)}{\partial w}$, $\frac{\partial L(w^T x_i + b, y_i)}{\partial w}$: the **gradients** of the regularization and loss functions with respect to the weight vector w. They indicate the direction and rate of change of the functions relative to changes in the weights.

This stochastic process allows the algorithm to escape local minima and increases the chances of finding a more optimal solution.

3.2.2 SGD Steps

Initialize the Process

1. Start with initial guesses for the parameters w and b For simplicity, assume w=0, b=0.
2. Choose a learning rate, η, which controls how much the parameters change in each step. For example η=0.

Process Each Data Point

1. **Pick a Data Point**: Take one input (x) and its corresponding actual output (y) from the dataset.
2. **Make a Prediction**: Use the current values of $w$and b to calculate the predicted output:

$$\hat{y} = w \cdot x + b \quad (11)$$

3. **Calculate the Error**: Find the difference between the actual output and the predicted output:

   Error= y1 - ŷ1

4. **Compute Gradients**: Gradients tell you how to adjust w and b to reduce the error. Calculate them as:

$$\frac{\partial L(w^T x_i + b, y_i)}{\partial w} = -2.xi.\text{Error} \quad (12)$$

5. **Update Parameters**: Adjust w and b using the gradients and learning rate:

$$w \leftarrow w - \eta . \frac{\partial L(w^T x_i + b, y_i)}{\partial w} \quad (13)$$

$$b \leftarrow b - \eta . \frac{\partial L(w^T x_i + b, y_i)}{\partial w} \quad (14)$$

Repeat for All Data Points

Multiple Passes (Epochs)

- After processing all the data points once, repeat the process multiple times (**epochs**) to refine w and b

Steps for the implementation of SGD on SVM as a classifier problem from scratch:

- Using iris dataset 100 sample size (just two classes for simplification)
- Splitting the data into 80% train and 20% test
- Setting the learning to 0.001
- 2000 number of iteration
- Fit our model
- Predict the model
- Compare the built in function with ours

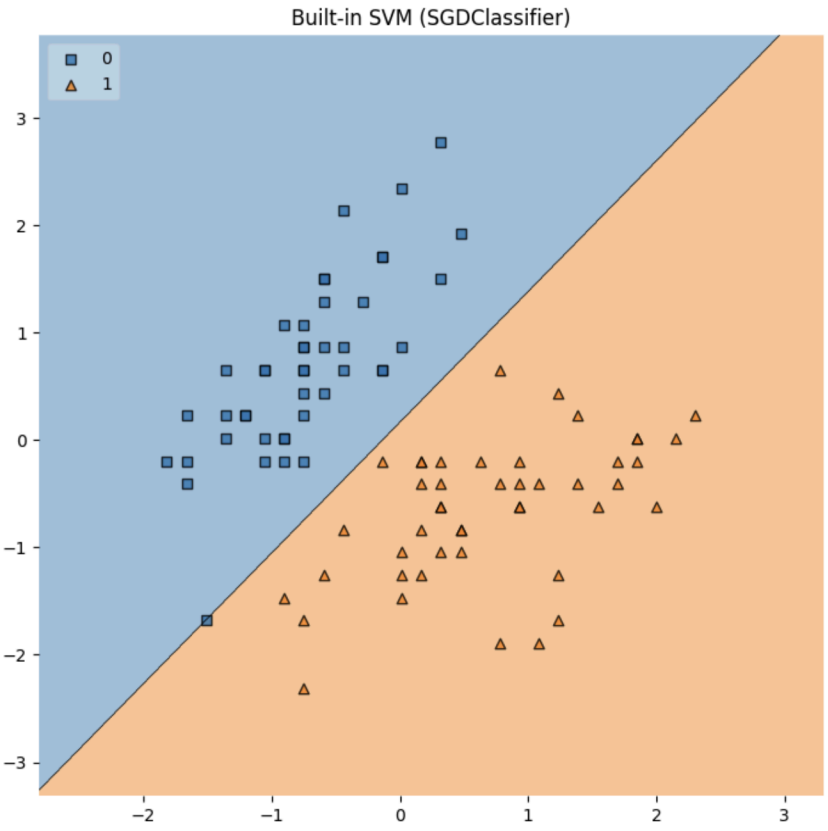

Figure 1. Represents the hyperplane that separates the two classes using built in functions.

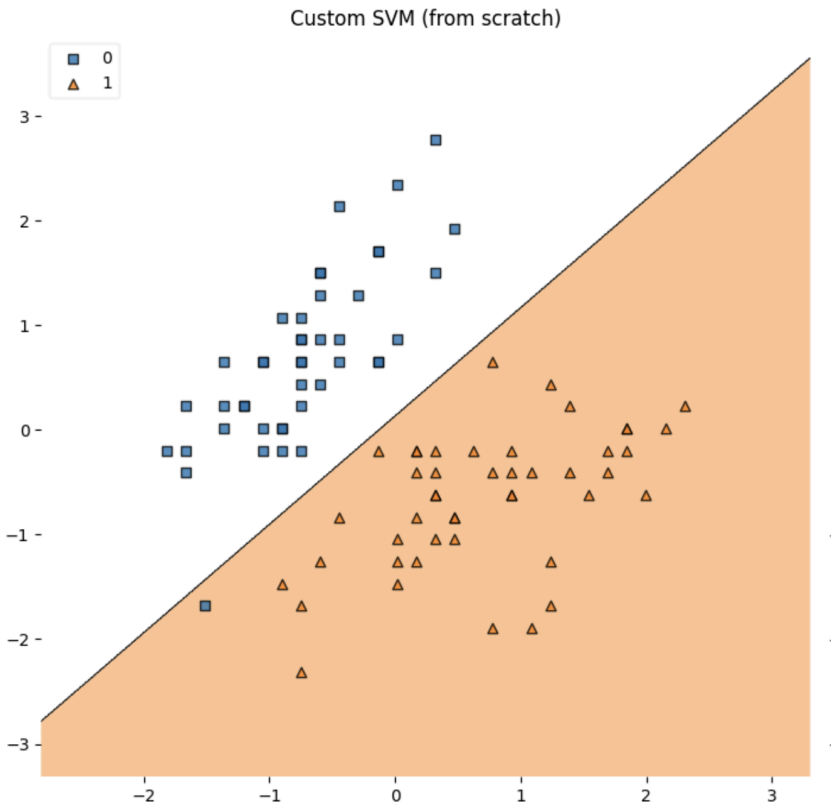

Figure 2. represent the hyperplane that separates the first and second classes from scratch

4.Results and Discussion

According to the implementation of this algorithm the result shows that:

4.1Accuracy Comparison

The two SVM models results are summarized below:

Model	Accuracy (%)
Custom SVM	100.0
Built-in SVM	100.0

The **Built-in SVM and Custom SVM** achieved perfect accuracy (100%)

4.2Confusion Matrices

The confusion matrices provide further insights into the classification performance:

4.2.1 Custom SVM:

- **True Positives (TP):** Correctly classified as class 1.
- **True Negatives (TN):** Correctly classified as class 0.
- **False Positives (FP):** Misclassified as class 1 instead of class 0.
- **False Negatives (FN):** Misclassified as class 0 instead of class 1.

4.2.2 Built-in SVM:

- Achieved perfect classification, with no FP or FN.

Metric	Custom SVM	Built-in SVM
TP	12	12
TN	8	8
FP	0	0
FN	0	0

Visual Comparisons

Bar Plot of Accuracy

The bar plot comparing the accuracies highlights the superior performance of the built-in SVM:

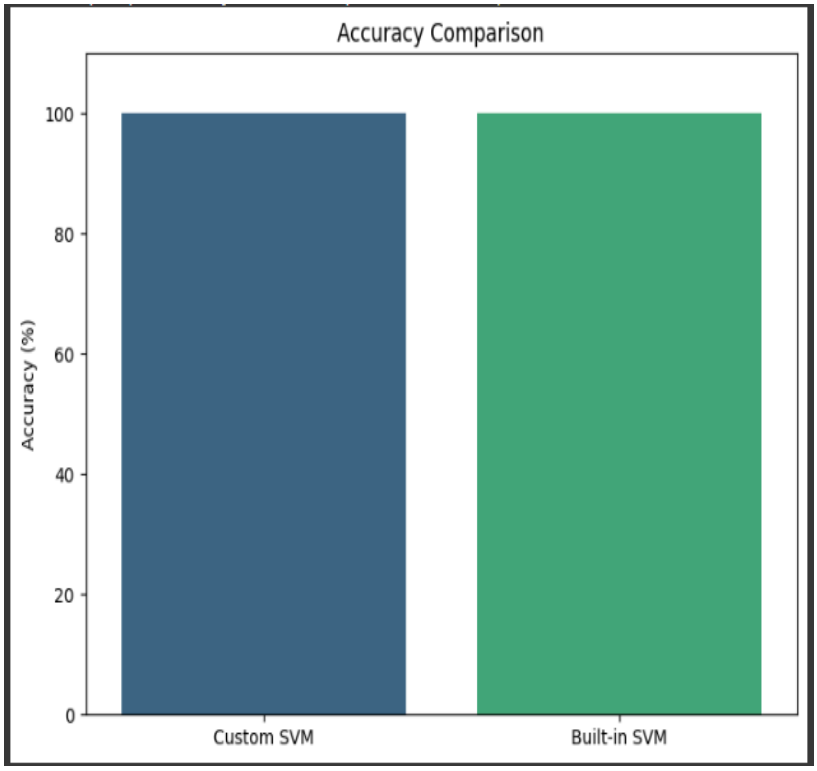

Figure 3.represents the accuracy for each methods

Confusion Matrices Visualization

The confusion matrices were displayed side by side:

1. **Custom SVM:** Shows slight misclassifications.

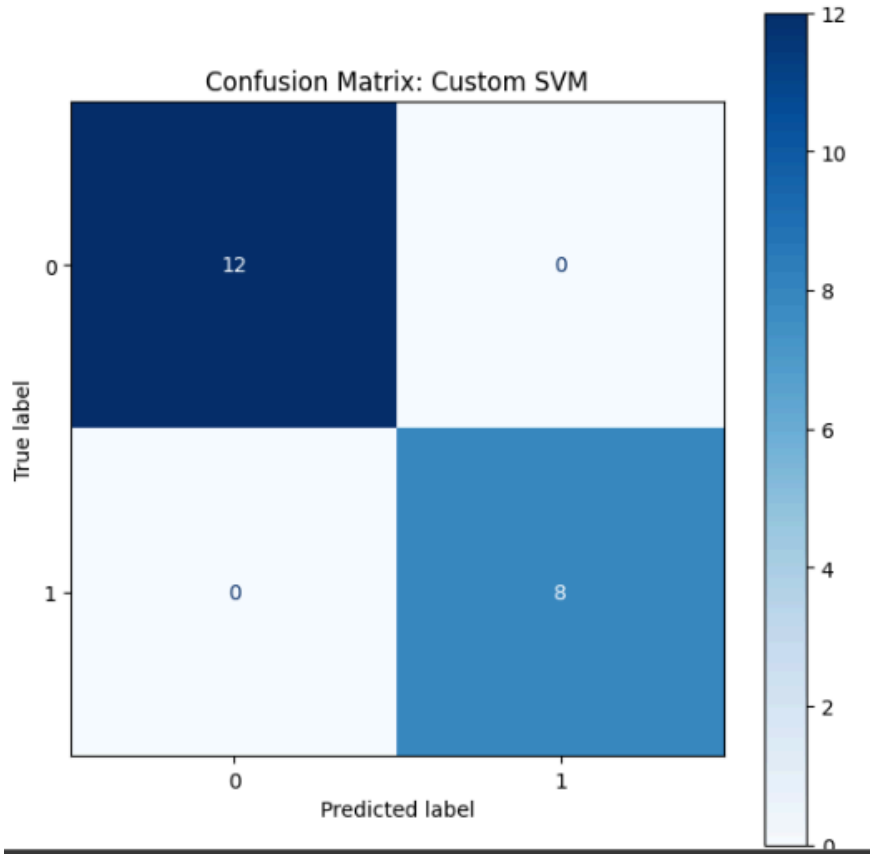

Figure 4. **Built-in SVM:** Perfect separation between classes.

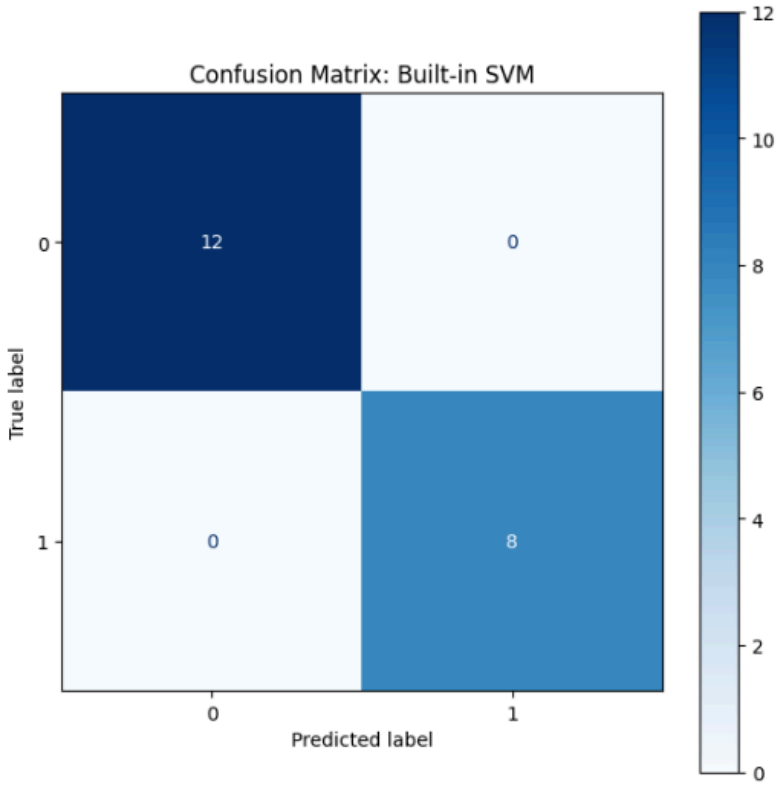

Figure 5. Represent the confusion matrix for built in functions

Implementation link : MATH-PRJ.ipynb

5.Conclusion

In conclusion, we have examined the theoretical foundations and practical applications of Support Vector Machines (SVMs) and their optimization using Stochastic Gradient Descent (SGD). The mathematical derivation of the SVM objective function, hinge loss, and the introduction of slack variables underscores the validity of SVMs in managing complex classification tasks, particularly in scenarios involving noisy or overlapping datasets. The soft margin SVM plays a pivotal role in balancing classification accuracy and margin maximization, as it allows for controlled misclassifications to improve generalization.The integration of SGD as an optimization technique further highlights its efficiency, especially for large-scale datasets, by

leveraging its incremental update mechanism. Experimental results demonstrated that both the from-scratch SVM implementation and the built-in library achieved high accuracy, with slight variations in misclassification rates attributable to the soft margin approach. These differences, as reflected in the confusion matrices, emphasize the balance between flexibility and precision inherent in SVMs.This comparison validates the reliability of SVMs as a classification tool and showcases the practicality of SGD in optimizing such models. Finally this report will be more powerful if more than classifiers such as naive and PCA, and compare them with our algorithm.

Acknowledgments

We extend our sincere appreciation to **Dr. Ahmed Abdelsamea** for his invaluable guidance and his effort during the whole semester.

We would also like to thank **Eng. Hossam Fathy** and **Eng. Youssef Mohamed** for all the knowledge we gained from them.

Additionally, we are grateful to the academic resources and Dr.Mayada Slides that provided the foundation for our work, including those that helped us understand the mathematics behind Support Vector Machines and Stochastic Gradient Descent.

Lastly, we acknowledge the contributions of our peers, **Mohamed Ahmed**,**Abdulrahman Omar, Momen Elsayed, Abdulrahman Khaled**, and **Yasmin Hegy**, for their collaboration and teamwork in bringing this project to completion.

Thank you all for your support

References

[1] G. J. Mahdi, “A Modified Support Vector Machine Classifiers Using Stochastic Gradient Descent with Application to Leukemia Cancer Type Dataset,” Baghdad Science Journal, vol. 17, no. 4. College of Science for Women, p. 1255, Dec. 01, 2020. doi:10.21123/bsj.2020.17.4.1255

[2] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). *A comprehensive survey on support vector machine classification: applications, challenges and trends.Neurocomputing.*doi:10.1016/j.neucom.2019.10.118.

[3] H. Wang, J. Xiong, Z. Yao, M. Lin, and J. Ren, “Research Survey on Support Vector Machine,” *Eudl*, Jan. 2017, doi: 10.4108/eai.13-7-2017.2270596.

[4]Y. Tian, Y. Shi, and X. Liu, “RECENT ADVANCES ON SUPPORT VECTOR MACHINES RESEARCH,” *Technological and Economic Development of Economy*, vol. 18, no. 1, pp. 5–33, Apr. 2012, doi: 10.3846/20294913.2012.661205.

[5] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, “An Overview on the advancements of support vector machine models in healthcare Applications: a review,” *Information*, vol. 15, no. 4, p. 235, Apr. 2024, doi: 10.3390/info15040235.

[6]GeeksforGeeks. (2023, February 2). *Introduction to Support Vector Machines (SVM)*. GeeksforGeeks.

[7]GeeksforGeeks. (2023b, August 1). *ML | NonLinear SVM*. GeeksforGeeks.

[8] www.naukri.com, “Code 360 by Coding Ninjas,” *2024 Naukri.com*. https://www.naukri.com/code360/library/combining-svm-sgd-in-machine-learning

[9]“1.5. Stochastic Gradient Descent,” *Scikit-learn*. https://scikit-learn.org/1.5/modules/sgd.html#sgd-mathematical-formulation

[10]“Breaking the Curse of Kernelization: Budgeted Stochastic Gradient Descent for Large-Scale SVM Training,” journal-article, Dec. 2012. [Online]. https://www.jmlr.org/papers/volume13/wang12b/wang12b.pdf

[11]S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[12]Ingoampt, “Day 5 _ Mathematical Explanation behind SGD Algorithm in Machine Learning,” *INGOAMPT*, Jul. 16, 2024.