Support Vector Machines (SVMs)

Sujay Jakka, Shane Rivera, Amaan Haque, Michael Moskowitz

April 29 2024

1 Data Collection & Pre-processing

The first dataset, a banana quality dataset, is a publicly published dataset that can be found on Kaggle. The second dataset, a banknote authenticity dataset, is another public dataset hosted by the University of California Irvine Machine learning repository.

Both datasets were pre-processed by relabeling class labels to be compatible with our SVM input requirements. For the banana quality dataset, this involved changing the class labels "Good" and "Bad" to 1 and -1. For the banknote authenticity dataset, this involved changing the class labels "Legitimate" and "Fraudulent" to 1 and -1 respectively.

As part of testing our SVM's capability, we created a set of dummy test data using scikit-learn's "make_blobs" function.

2 Running the code

The source code is stored inside of a Google Colab workspace, and while this implementation can be run from this workspace, the source code could also be transferred to a file and run from the command line.

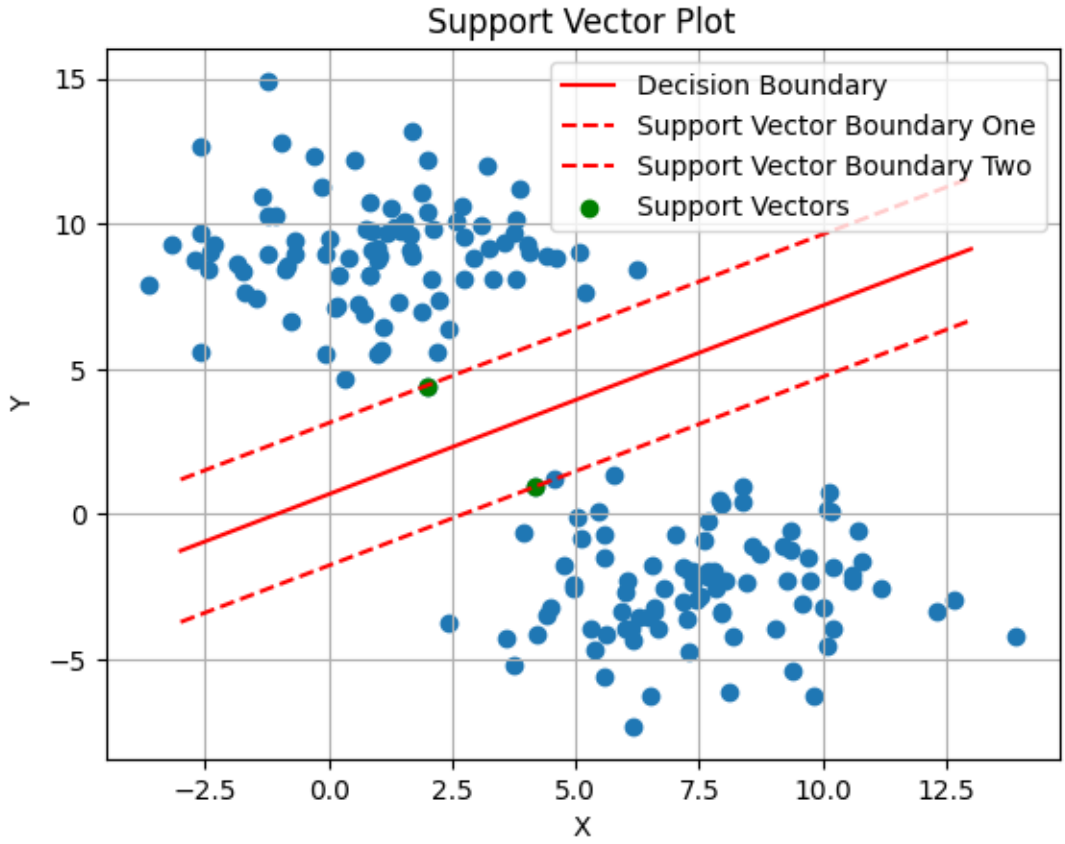
In order to run the code from the Google Colab, iteratively click through each start symbol next to the block of code that corresponds to the dataset you want to run through the SVM.

In order to run the code from it's own standalone script, assuming that the script has been populated with the source code from the Google Colab, all package dependencies have been satisfied, and the datasets are in the same directory as the script, the following command could be issued:

```
$ python3 ./SVM_assignment.py
```

3 Results

For the dummy data, the SVMs training accuracy was 1.0. The testing accuracy was also 1.0. The runtime of the SVM training and testing was 4.47 ms.



For the banana quality data, the SVMs training accuracy was 0.89125. The testing accuracy was 0.885. The runtime of the SVM training and testing was 11.87 ms.

For the banknote authenticity data, the SVMs training accuracy was 0.98875. The testing accuracy was 0.985. The runtime of the SVM training and testing was 10.28 ms.

4 Analysis and Conclusions

The results from the Support Vector Machines demonstrate the efficacy of this type of machine learning algorithm across diverse datasets. However, some datasets are definitely better suited for SVMs. Across all three of our scenarios—dummy data, banana quality data, and banknote authenticity data—the SVMs consistently produce high levels of accuracy in both training and testing phases. The dummy data experiment showcases perfect accuracy, indicating the SVM's robust capability to learn and classify data patterns effectively. As

the complexity of the datasets increases, as shown in the real-world scenarios with datasets of banana quality and banknote authenticity, the SVMs maintain strong performance. Although there were slight reductions in accuracy, these reductions are small and still yield reliable outcomes. This confirms the algorithm's adaptability and generalization across different data distributions and characteristics.

Furthermore, the runtime analyses reveal efficient computational performance of SVMs, with relatively low processing times across each of the datasets. This aspect is crucial for real-time or large-scale applications, where efficient model training and testing are a large factor is usability. In conclusion, the results outputted displayed SVMs' versatility and efficiency as a machine learning algorithm, able to effectively combat classification challenges across different domains.

5 Environment settings

5.a Hyperparameters

C, our regularization term, controls how much misclassifications are penalized. For the banana quality dataset, C is set to 0.01. For the banknote authenticity dataset, C is set to 1. For the dummy data, C is set to 1.

5.b OS & python version

The source code is os-agnostic, however, for the purposes of outlining all details regarding our environment for reliably reproducing results, this project uses the following OS and versions of said OS/python.

- OS platform: Linux
- OS release: 6.1.58+
- Python version: 3.10.12

5.c Package versions

- numpy: 1.25.2
- pandas: 2.0.3
- matplotlib: 3.7.1
- cvxopt: 1.3.2
- sklearn: 1.2.2