Project 1

CIS 4930 — Alan Kuhnle — Fall 2019

Julian Sweatt

November 6, 2019

1 SVM

1.1 SVM Creation

SVM is implemented in the *svm.py* file. The SVM optimization itself follows the pattern:

$$\min_{w,b,\epsilon} \frac{1}{2}||w||^2 + C \sum_{i=1}^{m} \epsilon_i$$

subject to $Y_i(w \cdot X_i + b) \geq 1 - \epsilon_i$ and $\epsilon_i \geq 0$ where $i = 0, m$[1].

1.2 SVM Results

1.2.1 C=1

[1]Course Lecture 5, Slide 18

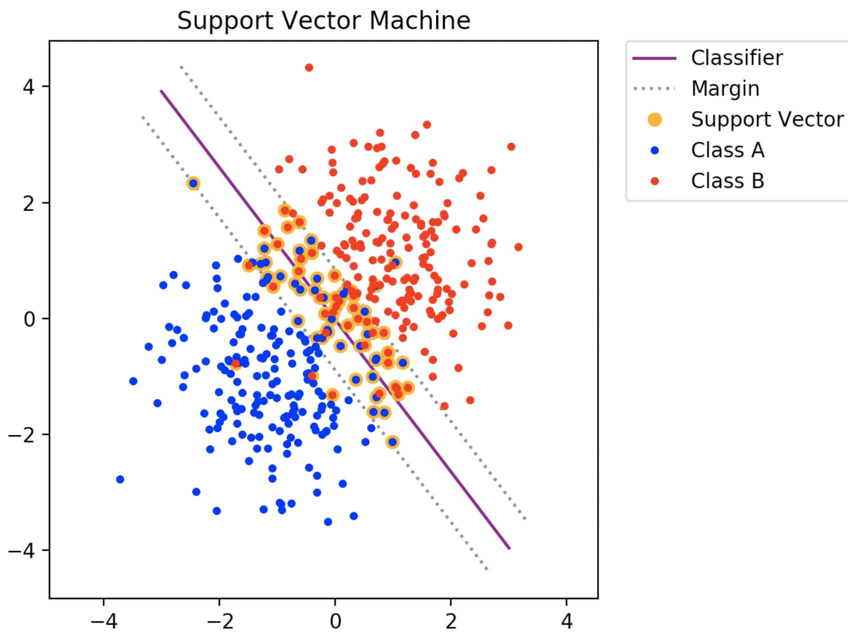

Figure 1: SVM Implementation Where $C = 1$

The Soft Margin SVM displayed in $Figure$ 1 meets the specifications listed, with the appropriate random sets described in the assignment.[2] The leave-one-out error was calculated by re-training the SVM with a missing point, then testing if the point could be accurately classified, repeated for all points. Leave-One-Out Error is defined by $100 * \frac{Incorrect\ Classifications}{Number\ of\ Total\ Vectors}$. The leave-one-out error is 13.75%. The margin displayed in $Figure$ 1 can be calculated using $\frac{1}{\epsilon_i}$, the margin is 0.5260689542315653. The support vectors are indicated in $Figure$ 1. The Support Vector Ratio, defined $100 * \frac{Number\ of\ Support\ Vectors}{Number\ of\ Total\ Vectors}$, is 18.5%.

1.2.2 C=0

[2] The last 5 digits of my library number begin with a 0, so 4 digits were used.

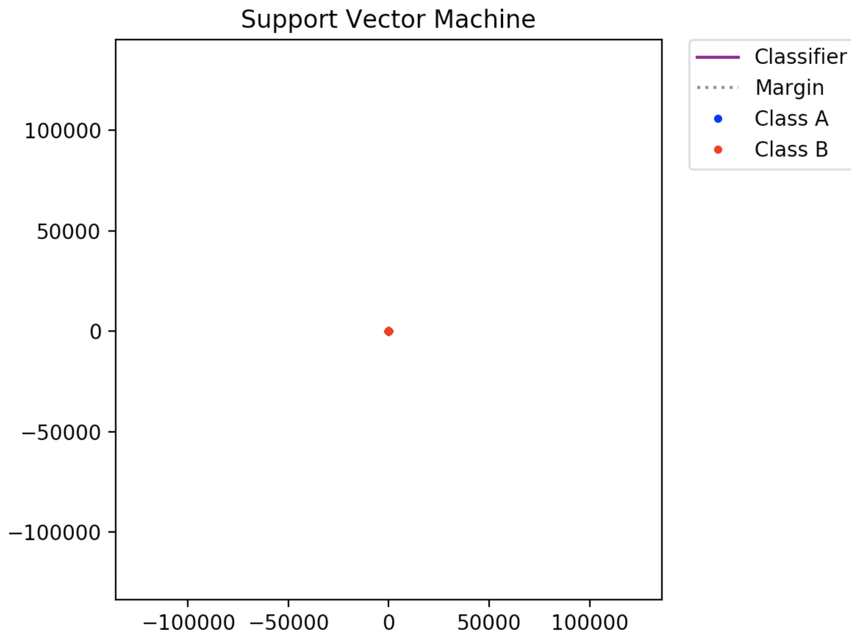

Figure 2: SVM Implementation Where $C = O$

$$Support\ Vector\ Ratio\ 0.0\%$$

$$Margin : 179475.5702426267$$

Setting C to 0 will completely remove the loss portion of the optimization problem. This causes the margins to become unbound. No vectors can be accurately identified.

1.2.3 C=10

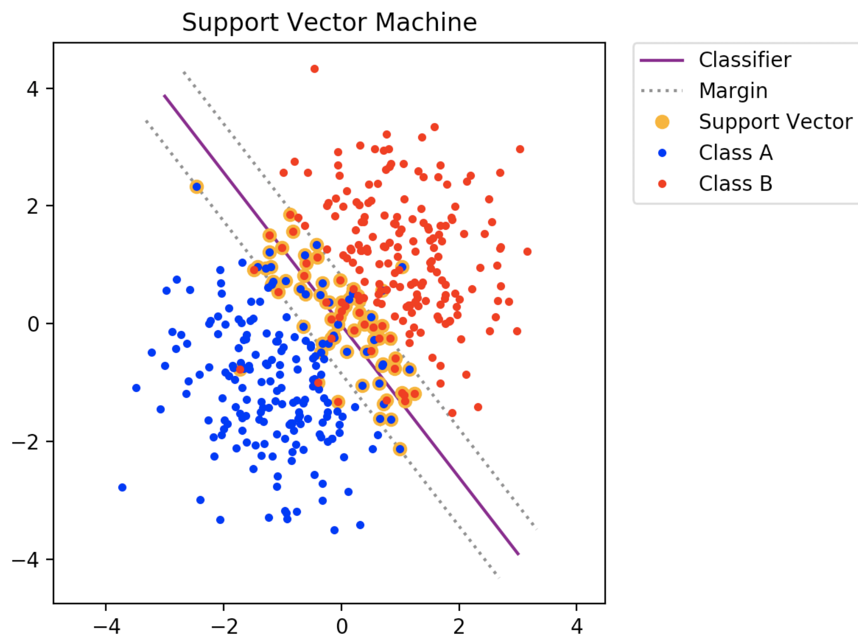

Figure 3: SVM Implementation Where $C = 10$

$$\textit{Support Vector Ratio:}\ 19.25\%$$

$$Margin : 0.505399047804732$$

When increasing the constant from 1 to 10, there decrease in the margin size from approximately 0.52 to approximately 0.50. There is a small increase in the support vector ratio (increase in support vectors) from 18.5% to 19.25%.

1.2.4 C=100

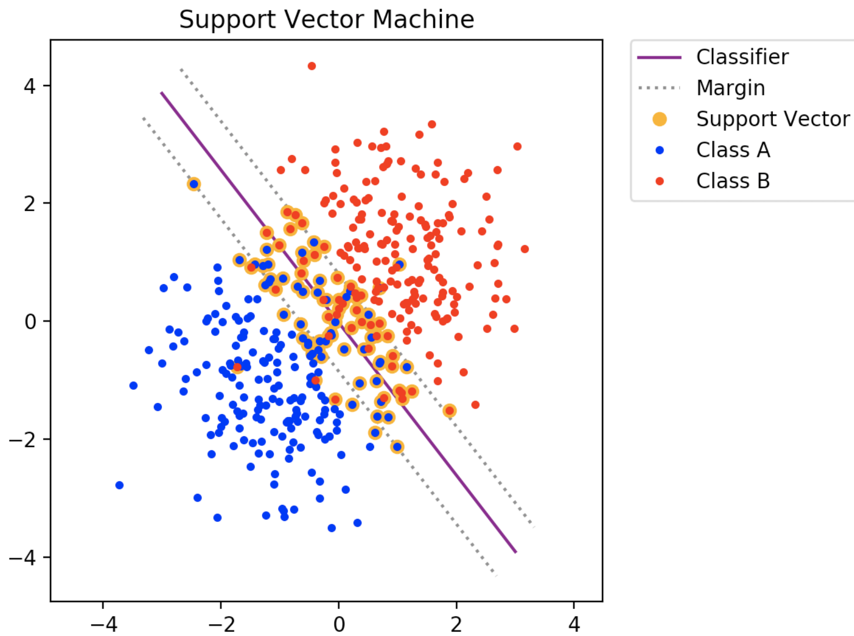

Figure 4: SVM Implementation Where $C = 100$

$$\textit{Support Vector Ratio:}\ 23.0\%$$

$$Margin : 0.5053990474598359$$

When increasing the constant from 10 to 100, the decrease in margin size is extremely small. There is a small increase in the support vector ratio (increase in support vectors).

1.2.5 C=1000

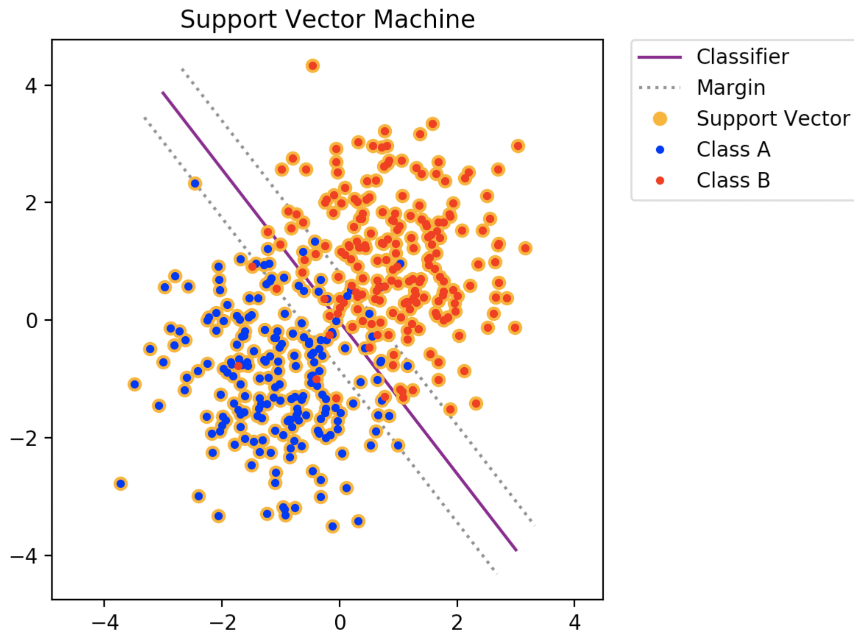

Figure 5: SVM Implementation Where $C = 1000$

$$\textit{Support Vector Ratio:}\ 100.0\%$$

$$Margin: 0.5053990455834894$$

When increasing the constant from 100 to 1000, the decrease in margin size is extremely small. The support vector ratio increases to 100% because all vectors become support vectors. This is likely due to the constraints on the definition of a support vector being lenient enough to produce no dual values equal to (or extremely close to) zero.

1.2.6 Summary: Changes to C

An increase in C values when $C > 0$ will cause small decrease in the margin length, and an increase in support vectors.

1.3 MNIST (EC)

Part C was not implemented (undergrad extra credit).

2 Regression

2.1 Linear Regression

2.1.1 C=1000

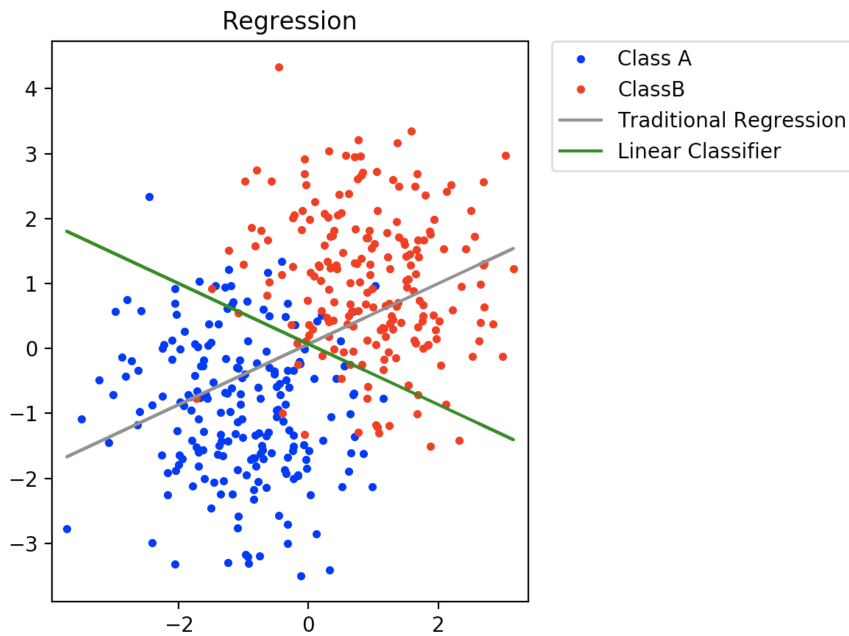

Figure 6: Linear Regression

Leave One Out Error: 13.5%

By calculating the traditional linear regression of the set as a whole, we can predict where new values belong. If this prediction is less than the threshold 0, it belongs to Class A. If the prediction is greater than the threshold 0, it belongs to Class B. This classification boundary and underlying regression can be seen in *Figure* 6.

The leave-one-out error is 13.5%. Theoretically, the leave-one-out error of the SVM should be lower than that of the linear regression because the linear regression is a less rigorous classification tool, however, they are equal. This is likely due to using the same random set centered at exactly diagonal points, therefore, the closer a classifier can get to a perfectly split classifier, the more accurate those predictions will be.

2.2 Logistic Regression

2.2.1 C=1000

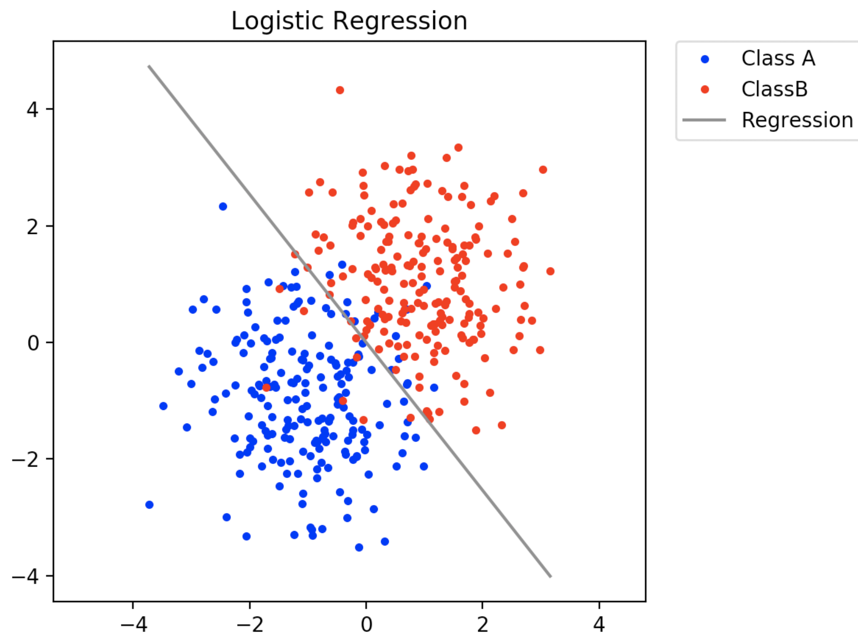

Figure 7: Logistic Regression

Leave One Out Error: 13.75%

The logistic regression classifier shown in *Figure* 7 can be used to separate Class A and Class B in the same manor explained in the linear regression, however, the hyperplane formed to classify the vectors is calculated logisticly rather than linearly.

The leave-one-out error is 13.75%. Normally, the logistic classifier should be more accurate than the linear alternative. However, the random set has an unusually linear relationship, explaining why a logistic classifier may not work as effectively in this case.

2.3 MNIST (EC)

Part C was not implemented (undergrad extra credit).