**Assignment 6 Group Report - SVM**

**Group Members:**

Che-Pai Kung 5999-9612-95

Chenqi Liu 2082-6026-02

Mengyu Zhang 3364-2309-80

# Part 1: Implementation

## Data structure

## Code-level optimizations

## Challenge

## Results

# Part 2: Software Familiarization

## Implementation

The library we used this time is scikit-svm.SVC, which stands for Support Vector Classification.

In the following images, ‘- - - ‘lines are the line whose distance to the optimal hyperplane is equal to one. Besides, each point is colored with its target value, while the color of region means the prediction of the model.

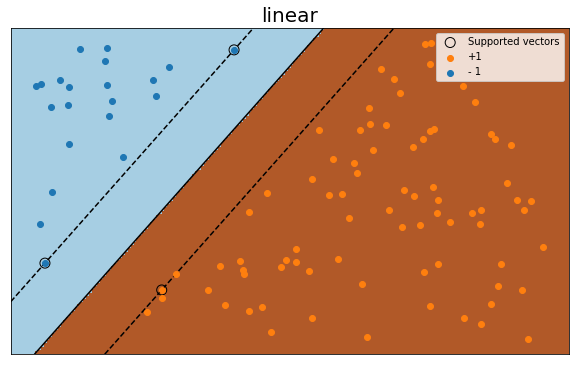
Linearly separable file

We ran the model with linear kernel and set C equal to 1000. The result is presented below, which are similar to what we got from our implementation.

Coefficient: [ 7.24837069, -3.86099178]

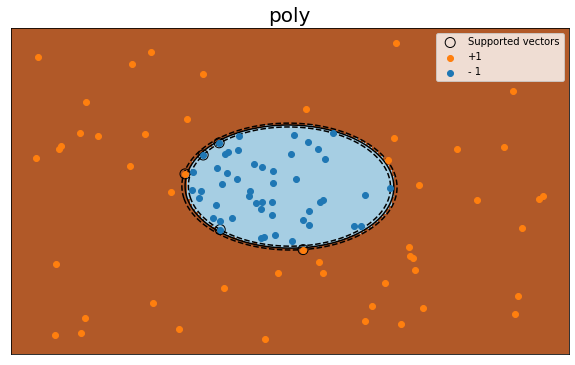
Intercept: [-0.10703977]

Support vectors: [[0.3917889 ，0.96675591], [0.02066458，0.27003158], [0.24979414，0.18230306]]

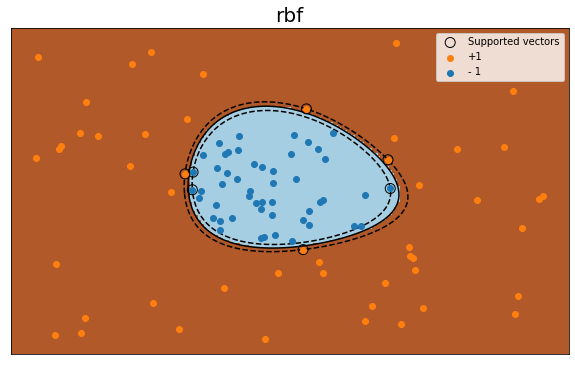


Nonlinearly separable file

We ran the model with ‘ploy’ kernel with degree equal to 2 and set C equal to 1000. The result is presented below, which are also like what we got from our implementation.

Support vectors: [[ -8.47422847, 5.15621613], [ -6.90647562, 7.14833849], [ -6.80002274, -7.02384335], [ -10.260969, 2.07391791], [ 1.3393313, -10.29098822]] 

We also ran the model with ‘rbf’ kernel. It also classifies the dataset accurately.

****

## Comparison and Inspiration

We think that the reason that the answers from sklean library and our implementation are slightly different is due to the different approach to solve quadratic programming problems. We use Quadratic Programming solver to solve the formulations, while it seems that sklearn uses other iterative ways to solve it.

In addition, for the ‘poly’ kernel, there are other parameters, gamma and coef0, in sklearn used to adjust the coefficient of polynomial functions. If we set gamma equal to 1, which is what we did in part1(b), then we got the same support vectors.

Regarding ‘poly’ kernel, sklearn library provides multiple parameters, such as degree, so that the dataset with various features has higher chance to be separated well. Therefore, it may be helpful to add these features into our implementation to generate more robust model.

Moreover, we can prevent the model from overfitting by setting parameter, C, in sklearn.svm.SVC. In this case, although, some of the training data is misclassified, it avoids overfitting and may have a better outcome in testing data if there is any. Hence, our next step will be adding this parameter into our model.

# Part 3: Applications

Support Vector Machine (SVM) is mainly applied on classification. It is a efficient algorithm, especially when training smaller dataset, so it is widely used in many fields.

One of the interesting applications we found is facial expression classification. With the help of SVM, it is possible to recognize whether a person is happy or sad based on the image or his/her face expression. This plays an important role when analyzing a communication between human.

Another impressive application is handwriting recognition. SVM is used to help recognizing the handwriting, and it is essential if we want to transferring paper documents into digital files.

# Part 4: Individual Contributions

● Model discussion: Che-Pai Kung, Chenqi Liu, Mengyu Zhang

● Model implementation: Chenqi Liu, Mengyu Zhang

● Model optimization: Chenqi Liu, Mengyu Zhang

● Software Familiarization: Che-Pai Kung

● Applications: Che-Pai Kung