Challenge 5 Report

Kiyeob Lee

I. Introduction

Support Vector Machine(SVM) is a supervised learning model that analyze the data for classification. Given a set of training instances, goal of the model is to represent the training instances as points in space, mapped so that the instances of difference classes are as widely divided as possible so that when new instance comes in, it is mapped into the same space and predicted to belong to a class.

II. Application

For our purpose, we have chosen to use 'radial basis function' kernel due to the non-linearity of the dataset. There may be other kernels with hyperparameters that might work better than what is used in this report, but without exhaustively trying all possible kernels, we focused on understanding the natures of SVM. The purpose of this exercise is to understand how SVM works, when it possibly works very well and when it works miserably.

Fig 1. shows that 1)training data with true class and trend lines of each class and 2)testing data points with predicted classes. We say that a dataset is **linearly separable** if there are two half-spaces that each half-space has points only from a single class. Clearly from Fig 1., the training dataset is not linearly separable that there is no half-space that can separate the dataset into two disjoint spaces. In addition to that, it is almost uniformly random nearby where both trend lines intersect. Nevertheless, we can loosely divide the whole dataset into three regions: 1)elements in x-axis that is greater than 1 mostly belong to 'class 1', 2)elements in x-axis that is less than -1, largely belong to 'class 2', and 3)the middle region is, loosely speaking, uniformly distributed although upper block is largely blue and lower block is mostly yellow. Clearly, if there should be a good model for the dataset, it is reasonable to expect that the same analogue should be applied to the testing dataset under an assumption that both datasets are from the same distribution.

To be a bit more specific, Fig 2. shows some extra bits of information in addition to Fig 1. There is a convex hull(black line) of 'class 2' and a polygon(red line) in Fig 2. To give a bit of explanation with respect to black line, it is more likely that elements belongs to 'class 2' than 'class 1' outside black convex hull. To give a further refinement, in terms of red polygon, 1)left hand side of the red polygon contains more yellow elements, 2)right hand side of the red polygon contains more blue elements, and 3)inside the polygon, it is uniformly distributed. I've seen a probabilistic SVM model that assign classes probabilistically, not deterministically unlike used in this report. Probabilistic SVM can be useful considering it in other cases, but it at least won't help within red polygon due to the randomness.

Having said that, prediction generally follows the arguments given by training dataset that 1)most points, values in x-axis that is greater than 1, belong to 'class 1', 2)most points(values in x-axis that is less than -1) belong to 'class 2', and 3)points in the middle sector is somewhat well divided by both trend lines. That is, above the lines, elements are mostly blue and, below the lines, elements are dominantly yellow. Thus, there is a half-space that separates classes 1 and 2 within red polygon.

III. Conclusion

In the report, We have seen that when SVM can be a good classifier that if there is a sufficient separable 'gap' between classes, then it works fairly well. Also there can be methods that 'gap' can be defined loosely so that even if 'a few points' across the 'gap', there are still ways we can take advantage of SVM such as 'soft margin SVM'. Although it won't work well within the red polygon that data instances are uniformly randomly distributed, there is not much hope for any other models that would work well in that region.

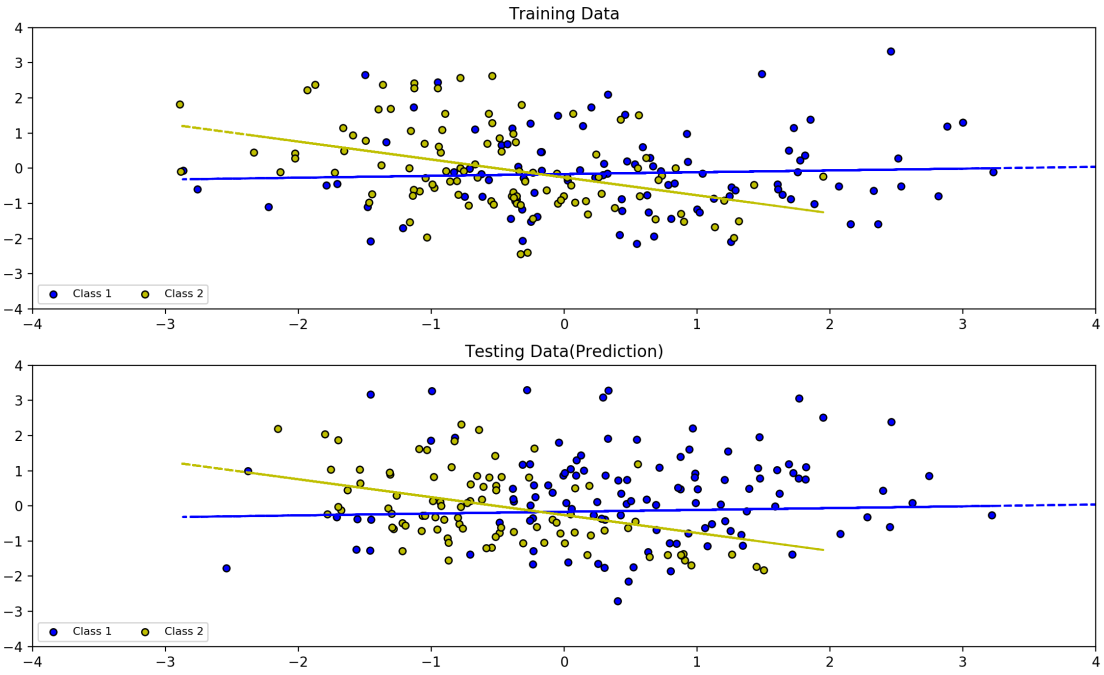

Fig. 1: Training and Testing(predicted) data

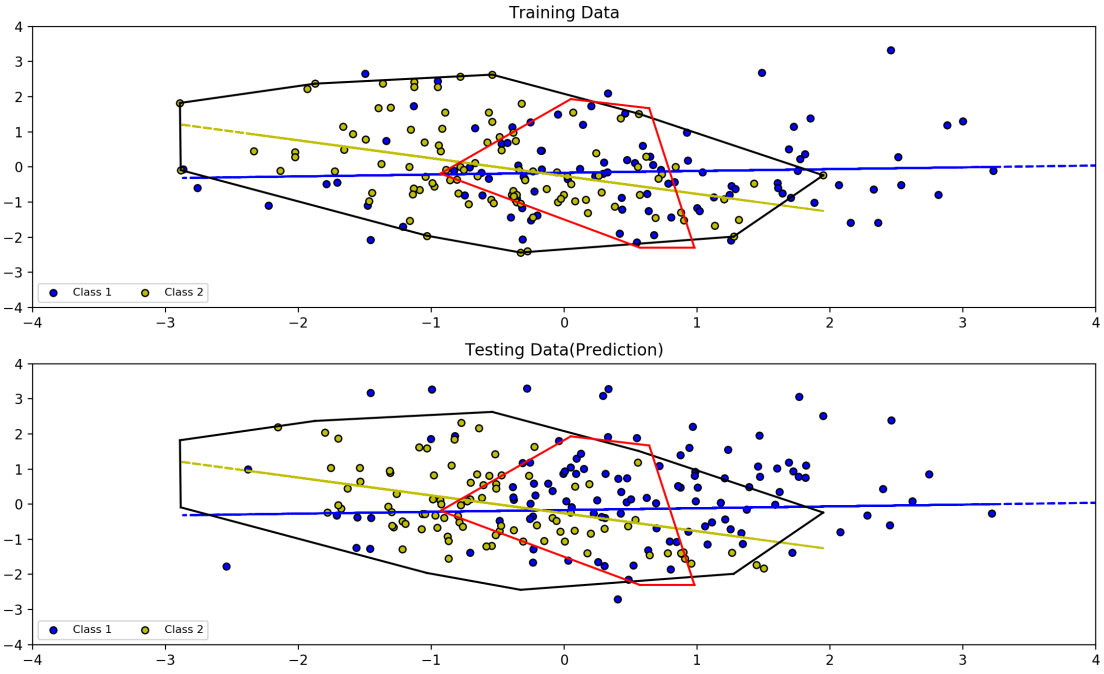

Fig. 2: Training and Testing data with refined interpretation