Project 3: Support Vector Classifiers
COSC425, Fall 2020

Due: Nov 11 @ 11:59pm

In this project, you will apply support vector classification to three classification problems. You will not need to program your own SVC because you'll be using scikit-learn to implement your classifiers as exemplified in our Kernel Methods lecture. This assignment follows the specification in the "Project Guidelines" document on Canvas.

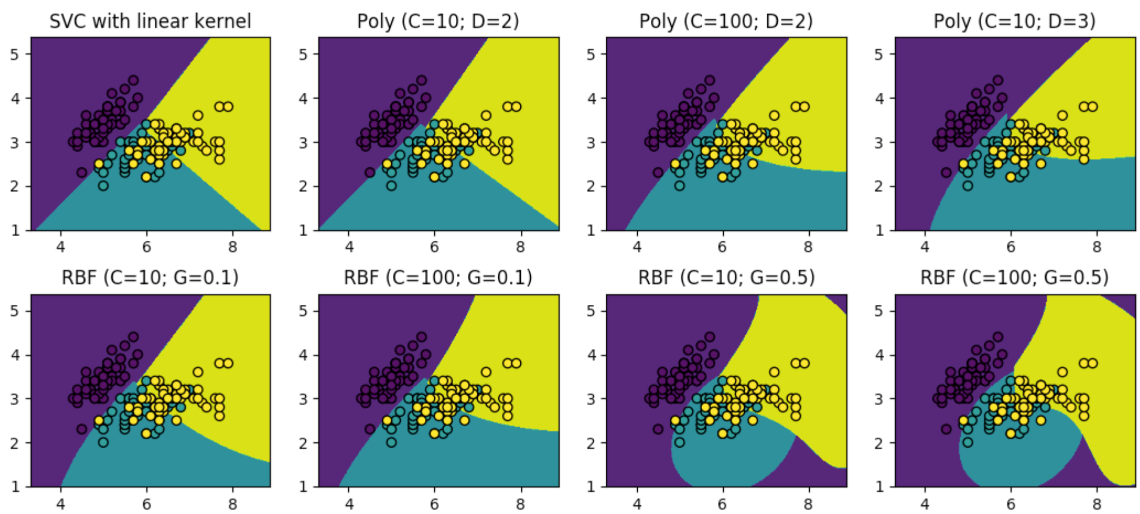

Figure 1: A visualization of SVCs with various parameters.

I. General Information

In this project, you will apply SVC to three different datasets. You should standardize your data. You should employ cross-validation where possible. The settings and datasets are as follows:

1. **Setting 1: Predicting Good/Bad Interactions of Radar Signals.** Your first setting is a binary classification problem: predicting good vs. bad interactions of radar signals with electrons in the ionosphere. The last attribute is the label (g or b) and the other 34 are measurements. There are 351 instances in the data. You will have to split them into training, validation, and testing subsets. The dataset is available on Canvas and online[1].
2. **Setting 2: Terrain Classification for Multispectrcal Pixel Values.** The second setting is a multiclass classification problem, with 7 classes and 36 features: determine the type of terrain from multispectral values of pixels in 3×3 neighborhoods in satellite images. Data has been separated into training and testing for you. There are 4,435 instances in the training set (`sat.trn`), and 2,000 instances in the test set (`sat.tst`). In the files, the last attribute is the label (17, but this sample contains no instances of class 6). The dataset is available on Canvas and online[2]. The file `sat.doc` explains the data format.

[1] https://archive.ics.uci.edu/ml/datasets/ionosphere
[2] https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

III. Implementation Requirements
You should implement SVCs for both settings using scikit-learn. You can use all of scikit-learn's utilities to assist you however you'd like, e.g. calculating accuracy, dividing test-train splits, etc. There are no other constraints for your implementation. (Use the `svm.py` script in the Kernel Methods module on Canvas as you wish.)

IV. Evaluation Requirements
The goal of this project is to identify the optimal set of hyper-parameters for SVCs in these two settings. You should explore the various kernel choices for SVCs (i.e. linear, polynomial, and RBF/Gaussian). Among these kernel choices, you should employ both (1) a coarse grid search and (2) a fine grid search to identify the optimal hyper-parameters for an SVC in a given setting. Report your results and classification performance for the hyper-parameters. Consider including a plot for that shows comparisons between various SVC configurations as shown in Figure 1 above. (Note: The `svm.py` script on Canvas contains the code to create this plot.)

Note: The goal of your report should be to convince the reader that you have explored the landscape of hyper-parameters at depth for each of the aforementioned settings.

V. Grading + Writing the Report
You will be graded entirely on your final report. Your report should be thorough in explaining how you've conducted your coarse grid search and fine grid search for each setting.

In addition, you should ensure that your report is structured appropriately. Your report should include five sections: (1) Introduction, (2) Pre-Processing, (3) Solution Description, (4) Evaluation and Analysis, and (5) Discussion. You should use headings that clearly separate these sections. If you do not format your document accordingly, you will lose points for formatting and readability.

Note: A submission without functional source code will result in a project submission of zero.