COSC 528: Project 5

Ian Lumsden

December 7, 2018

1 Objective

The goal of this project was to use support vector classifiers (SVCs) to solve three classification problems. The first problem was predicting *good* vs *bad* interactions of radar signals with electrons in the ionosphere. The second was classifying vowel sounds. The third and final was classifying the type of terrain from flattened pixel data from several multispectral satellite images. All of these problems were implemented using *SVC*, *StandardScaler*, *train_test_split*, and *GridSearchCV* from scikit-learn.

2 General Algorithm

The code for each of the three solutions followed the same general algorithm, described below. All deviations from this algorithm are stated in the section for the particular problem.

1. Read the dataset from file
2. Clean the data so there are no invalid values
3. Split the data into data and labels
4. Standardize the data using *StandardScaler*
5. Split the data into training and testing sets
6. Run a *SVC* model through a course Grid Search (3-Fold Cross Validation)
7. Run a *SVC* model through a fine Grid Search (3-Fold Cross Validation) based on the outcome of the last step

All problems used the same set of potential hyperparameters for the coarse Grid Search:

Linear
C
0.01
0.1
1
10
100
1000

Polynomial		
Degree	C	Gamma
2	0.01	1
3	0.1	0.1
4	1	0.01
	10	0.001
	100	0.0001
	1000	

RBF	
C	Gamma
0.01	1
0.1	0.1
1	0.01
10	0.001
100	0.0001
1000	

3 Problem 1

3.1 Data and Imputation

The data for problem 1 consisted of 34 continuous numbers representing the features, along with one discrete column of "g" or "b" representing the classes. Before using the data, the class data was altered so that "b" became 0 and "g" became 1. The general algorithm was followed for the rest of the problem. The Grid Search used F1 score as the scoring measure.

3.2 Results

After the coarse Grid Search, the best SVC model had $C = 1$, $Gamma = 0.1$, and a RBF kernel. During training, this model produced a F1 score of 0.94561929. Scikit-Learn was used to produce the following classification report for this model:

	Precision	Recall	F1-Score	support
Class 0	0.95	0.97	0.96	38
Class 1	0.98	0.96	0.97	50
Micro Avg	0.97	0.97	0.97	88
Macro Avg	0.96	0.97	0.97	88
Weighted Avg	0.97	0.97	0.97	88

For the fine Grid Search, the RBF kernel was used, with C values from 0.75 to 1.25 by 0.01 and Gamma values from 0.075 to 0.125 by 0.001. The fine Grid Search showed that the model with the same hyperparameters as the coarse Grid Search was best. After generating the classification report (which was the same as the course search), the F1 score for prediction was determined to be 0.96969697.

4 Problem 2

4.1 Data and Imputation

The data for problem 2 consisted of 12 features and one class feature with discrete values between 0 and 10 inclusive. The first three features represented the pre-determined train-test split, an ID for the speaker who provided the data, and the sex of the speaker. For the purposes of this classification, these three features were not needed, so they were removed. The general algorithm was followed for the rest of the problem. Because the classification was not binary, log loss was used as the scoring measure for the Grid Search.

4.2 Results

After the coarse Grid Search, the best SVC model had $C = 10$, $Gamma = 0.1$, and a RBF kernel. During training, this model produced a log loss score of 0.22487192. Scikit-Learn was used to produce the following classification report for this model:

	Precision	Recall	F1-Score	support
Class 0	1.00	1.00	1.00	27
Class 1	1.00	1.00	1.00	20
Class 2	1.00	1.00	1.00	22
Class 3	1.00	1.00	1.00	20
Class 4	0.96	1.00	0.98	25
Class 5	1.00	0.91	0.95	23
Class 6	1.00	1.00	1.00	23
Class 7	1.00	1.00	1.00	24
Class 8	1.00	1.00	1.00	17
Class 9	1.00	1.00	1.00	27
Class 10	0.95	1.00	0.98	20
Micro Avg	0.99	0.99	0.99	248
Macro Avg	0.99	0.99	0.97	248
Weighted Avg	0.99	0.99	0.99	248

For the fine Grid Search, the RBF kernel was used, with C values from 7.5 to 12.5 by 0.5 and Gamma values from 0.075 to 0.125 by 0.005. The step size was chosen to save time because using log loss as the error slows down the SVC model. The best model's hyperparameters were determined to be

$C = 7.5$, $Gamma = 0.115$, and a RBF kernel. The log loss of this model during training was 0.21502784. During testing, the log loss was 0.10493455. The classification report was the same as the course search, so it is not shown.

5 Problem 3

5.1 Data and Imputation

The data for problem 3 consisted of 36 continuous features with values between 0 and 255 and a classification with value 1 to 7 inclusive, although no instances of class 6 existed. Each feature represents a pixel in one of four spectral images. Unlike the other two problems, the data for this problem was split between two files: a training file, *sat.trn*, and a testing file, *sat.tst*. As a result, the data was not manually split with *train_test_split*. Additionally, the non-label data for both sets had to be standardized individually using σ and μ from the training set. The general algorithm was followed for the rest of the problem. Because the classification was not binary, log loss was used as the scoring measure for the Grid Search. Additionally, since the *sat.name* file said not to apply cross-validation to the data, the *cv* parameter of *GridSearchCV* was set to 2 instead of 3.

5.2 Results

After the coarse Grid Search, the best SVC model had $C = 0.1$, $Gamma = 0.1$, and a RBF kernel. During training, this model produced a log loss score of 0.46132665. Scikit-Learn was used to produce the following classification report for this model:

	Precision	Recall	F1-Score	support
Class 1	0.95	0.99	0.97	461
Class 2	0.93	0.99	0.96	224
Class 3	0.86	0.96	0.90	397
Class 4	0.74	0.52	0.61	211
Class 5	0.93	0.77	0.84	237
Class 7	0.83	0.87	0.85	470
Micro Avg	0.88	0.88	0.88	2000
Macro Avg	0.87	0.85	0.86	2000
Weighted Avg	0.88	0.88	0.87	2000

For the fine Grid Search, the RBF kernel was used, with C values from 0.075 to 0.125 by 0.005 and Gamma values from 0.075 to 0.125 by 0.005. The step size was chosen to save time because using log loss as the error slows down the SVC model. The best model hyperparameters were determined to be $C = 0.085$, $Gamma = 0.12$, and a RBF kernel. The log loss of this model during training was 0.46033781. During testing, the log loss was 0.30876334. The classification report was slightly different than the coarse grid and is shown below:

	Precision	Recall	F1-Score	support
Class 1	0.95	0.99	0.97	461
Class 2	0.89	0.99	0.94	224
Class 3	0.85	0.95	0.90	397
Class 4	0.75	0.52	0.61	211
Class 5	0.93	0.76	0.83	237
Class 7	0.84	0.87	0.85	470
Micro Avg	0.88	0.88	0.88	2000
Macro Avg	0.87	0.85	0.85	2000
Weighted Avg	0.87	0.88	0.87	2000

6 Conclusion

All three models performed very well at their respective tasks. For the first problem, the F1 testing score being very close to 1 means that there were almost no false positives or negatives, meaning the model was very accurate. For the second problem, the near-zero value of the log loss suggests that the model had little variance between the predicted values and the actual labels. This means that the model was very accurate as well. The third model was by far the worst, with its testing log loss score suggesting a decent, but not great level of accuracy. All three of these models could possibly have been improved by considering a sigmoid kernel or more hyperparameter values in the coarse Grid Search. However, all three models are quite good at their tasks, and any additional gains would likely be minimal.

7 Code

All code for this project can be found in *project5.ipynb*. The code uses Markdown cells to divide the different problems.