COSC 528 Project 5 Report

Support Vector Classification

J.R. Powers-Luhn
December 7th, 2018

Objective

The objective of this project was to explore off-the-shelf machine learning libraries (in this case the scikit-learn library) as they pertain to support vector classification. Three datasets were examined: a database of "good" and "bad" interactions of radar signals in the ionosphere, a multiclass problem to identify vowel sounds, and another multiclass problem identifying terrain features based on multispectral pixel values (4 features per pixel times 3x3 pixels per sample).

Preprocessing the data

Since support vector classifiers seek to maximize the margin between "support vectors" and the hyperplane separating the classes, scaling the data puts each dimension of the margin on the same footing. Therefore each of the real-valued measurements in all datasets were scaled to mean-center and unit variance.

Since k-fold (k=5) cross-validation was used to optimize the hyperparameters of each classifier, the data were split into training and testing sets. The training data was split into five segments and each model was trained successively on four of these and tested on the remaining one. This was repeated with each of the segments used to validate the training once. The performance metrics corresponding to each hyperparameter were calculated by averaging the performance on each of the five validation sets.

Ionosphere

The output classes for the ionosphere dataset were recorded as "b" (bad) or "g" (good). These values were re-encoded as a one-hot vector with "b" changed to 1 and "g" changed to 0.

Phonemes

Two columns in the dataset represented non continuously-valued information (gender of the participant and speaker number). These columns were not included in the model. This dataset also contained an assignment for each sample including it in either a training or test set. Performance of models trained with these preassigned splits was poor. The vectors were reassigned randomly and model performance improved significantly.

Terrain

Values in this training set were recorded as integers. These were converted to floating point form.

Implementation

For each of these datasets the data were MCUV scaled using the training data to determine appropriate mean and standard deviation values. An SVC classifier was trained to predict the class for each vector using a radial basis function kernel. Two hyperparameters were varied to optimize the model: the penalty parameter for violating the "soft boundary" near the dividing

hyperplane and the width of the radial basis function. For the second data set an evaluation of a linear kernel was also made (omitting the width parameter). For each hyperparameter five models were trained in accordance with the k-fold cross validation method described above. The hyperparameter set that gave the best mean performance was selected. Where appropriate, two hyperparameter searches were performed—a coarse search followed by a fine search.

Results

Ionosphere

This dataset proved easily classifiable using an RBF kernel. A fine parameter search resulted in a C-parameter (boundary softness) of 0.49 and a kernel bandwidth of 0.015. A total of 103 support vectors were selected for the decision boundary (45 from the 'b' class and 58 from the 'g' class). This predicted the training set perfectly and performed equivalently on the testing set (see Figure 1).

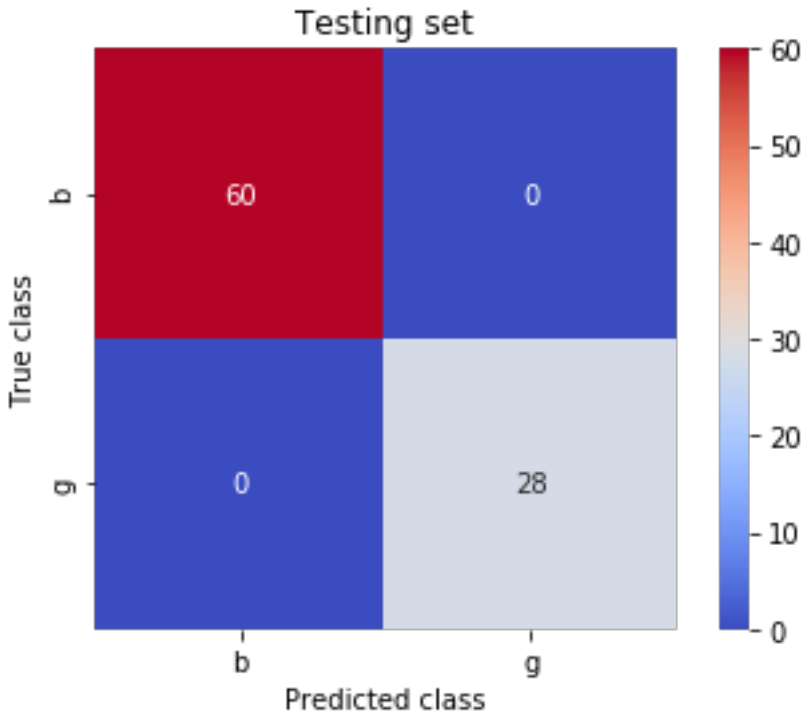

Figure 1: Test set performance for support vector classifier

Principal component analysis was performed on the dataset to determine if a more sparse solution could be achieved. Examination of the cumulative variance captured by the first *n* principal components, however, revealed that there was no obvious "elbow" in the variance captured graph to indicate an optimal number of principal components. Since the model performed very well without this step, PCA was omitted (and not performed for the other two datasets).

Phonemes

An analysis of the phonemes dataset produced an optimal model with a C value of 10.5 and a bandwidth of 0.176. These values were determined after a coarse search covering several decades for each of these parameters followed by a fine search within ±10% of the coarse search parameters. This produced a model that perfectly predicted the training dataset. It performed nearly perfectly on the test data set as well (Figure 2). The overall F1 score of the model was 0.99, with an F1 score of 0.96 on classes 5 and 10 (numbered from 0).

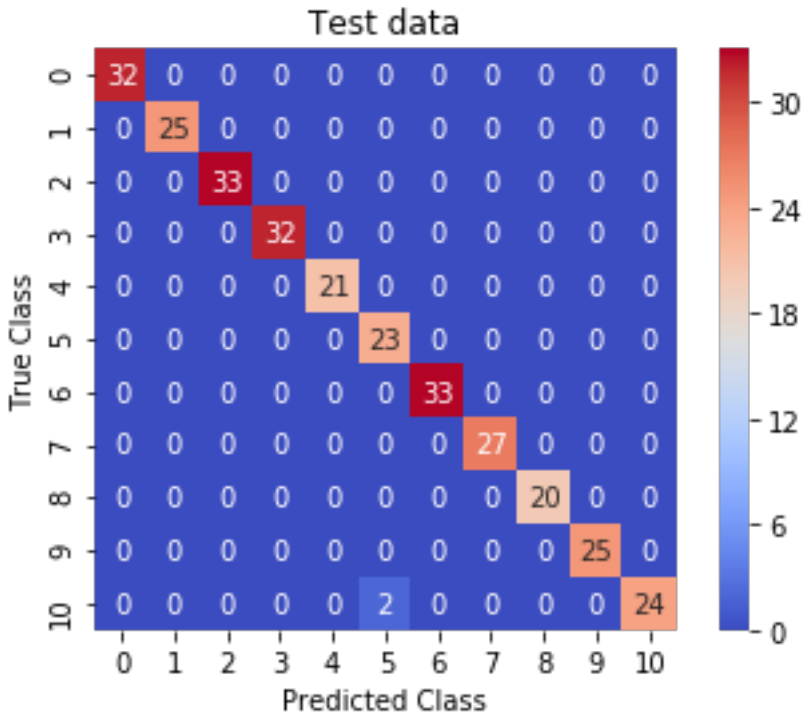

Figure 2: Performance of phoneme classification model on test data

Satellite data

These data proved more recalcitrant than the previous two datasets. While the performance overall seemed reasonable (92% precision, 91% recall, 0.91 F1), this obscures the performance on the worst class (class 4: 62%, 77% and 0.69). Since this was the class least represented in the testing set an attempt was made to correct for the undersampling; this did not significantly improve the results (77%, 63%, 0.69). The result of the best model is shown in Figure 3.

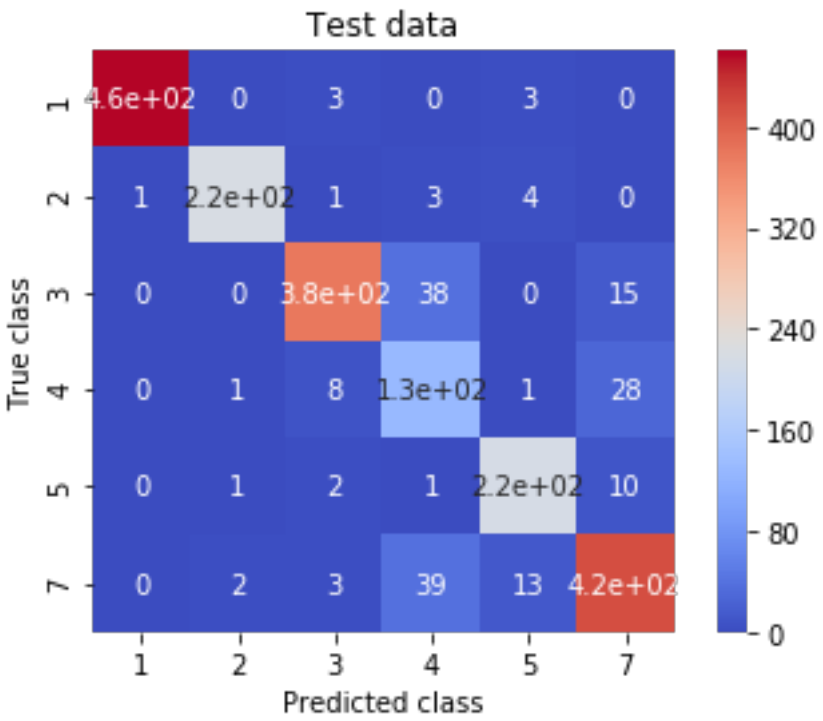

Figure 3: Optimal model for terrain classification

PCA on the training data revealed that 97% of the variance was captured in the first 8 principal components. Once again, however, this did not markedly affect the performance of the support vector classifier (test results on class 4 were 59%, 76% and 0.66).

Conclusions

Support vector machines were used to classify three datasets. The value of this classification technique for generalizing a model by maximizing the distance from each class to the classification boundary was shown by impressive performance on two of the three datasets. While the performance on the third set was not as good, it still gave recall and precision of ~90%, which might be sufficient (depending on the application). The value of pre-built libraries for optimizing model hyperparameters was also shown.