

Assignment 6: Forest Cover-Type Prediction with Scikit-Learn

Apala Guha

CMPT 733

Spring 2017

1 References

- Support Vector Machines: Ch. 12, Statistical elements of machine learning, Hastie et al.
- Random Forest: Ch. 15.1, 15.2, Statistical elements of machine learning, Hastie et al.
- Coursera Predictive Analytics course: <https://www.coursera.org/learn/predictive-analytics/home/welcome>. Week 2, Lessons 7–9.
- Coursera Machine Learning course: <https://www.coursera.org/learn/machine-learning/home/welcome>. Week 7.

2 Dataset

You are provided with the file `forest_data.npz`. It contains data for the classification of forest covers. This file that can be loaded using `numpy.load` contains the following numpy variables:

- **data_training**: a matrix of size $25,000 \times 54$ which contains 25,000 training samples in each row. Each sample is represented with a 54 dimensional feature vector. Please refer to the description of the forest cover-type challenge for detailed information regarding these features at <https://www.kaggle.com/c/forest-cover-type-prediction>.
- **label_training**: a one-dimensional vector of class labels for the training samples. Therefore, its length is 25,000 and it contains labels in $\{1, 2, \dots, 7\}$
- **data_val**: a matrix of size 5000×54 which contains 5,000 samples that will be used for validation.
- **label_training** contains the class labels for the validation set.
- **data_test**: a matrix of size $550,000 \times 54$ which contains 550,000 samples used for test. This matrix can be ignored in this assignment.

3 Linear SVM

Train a linear SVM using C in $\{10^{-3}, 10^{-2}, 10^{10}\}$. Normalize the feature vectors. Run a 3-fold cross-validation for the given C values using `GridSearchCV` and `accuracy_score`. Plot the score of each parameter for each fold. Test the best model on the test data and report the accuracy score.

Submission Instructions. This code should be submitted as a single python file. It should use the `sklearn` library which depends on `scipy`. Use `pip install <library>` or `pip install --user <library>` to install libraries. It should produce as output the value of every datapoint that will be plotted and the final test accuracy. The datapoints should appear one on each line with their corresponding parameters. Plotting the datapoints in this code is optional.

The report must contain the plot. The y-axis should plot accuracy and the x-axis should plot the validation iteration number. There should be a separate line for each C value. The report should also contain the test accuracy.

4 Kernel SVM

For this part, use σ values of $\{10^0, 10^2, 10^4\}$ in combination with the C values above, and using 3-fold cross-validation. Plot the best accuracy of each C and σ combination in the training phase. Apply the best model to the test set and report the accuracy.

Submission Instructions. Refer to the submission instructions in Section 3. The plot x-axis should plot the σ values and there should be a separate line for each C value.

5 Random Forest

Normalize the feature vectors. Use the `RandomForestClassifier` to train decision trees. Use balanced sampling weights and bootstrapping. Use a maximum depth of 100. Use datapoint sampling rates of $\{0.2, 0.4, 0.6, 0.8\}$, and forest sizes of $\{10, 20, 50\}$. Use feature sampling rates of $\{0.2, 0.4, 0.6, 0.8\}$ and forest sizes of $\{10, 20, 50\}$. Plot the out-of-bag error estimates for these two experiments.

Submission Instructions. The code should output the plot values, one on each line, along with their corresponding parameters. Plotting code is optional. The report should contain two plots, one for datapoint sampling, another for feature sampling. The x-axis should plot the sampling rate and there should be a separate line for each forest size.