Classification of LandSat Images

Project McNulty - Weeks 4, 5, and 6 of the Metis Data Science Bootcamp

Steven Bierer November 1, 2018

Introduction

The third project, focused on supervised learning algorithms, encouraged us to try out a variety of classification models on a labeled data set. I chose to classify different types of terrain from satellite images, as the data features were strictly quantitative and the labels had a straight-forward interpretation. The multi-class nature of the data (as opposed to binary "yes/no" labels) added some challenges in training and assessing the results of the model. But the draw of working on an imaging data set and potentially applying image processing techniques was a plus, as I have a keen interest in that class of problems.

The U.S. LandSat Program was initiated by NASA in 1972 with the purpose of imaging the Earth's surface for scientific, military, and other purposes. The program was transferred to the scientific agency NOAA in 1979 and is currently managed by a private contractor. To date, seven satellites have recorded millions of images for a variety of applications and a new satellite, LandSat 9, is projected to launch in 2020.[1]

The images used for the current project, scanned by LandSat 4 and dating to the early 1990s, were initially collected for an agricultural study. Each data point in the set is a 3x3 grid of pixels extracted from a larger scene, with each pixel representing an 80m x 80m area of land surface. (The grids were randomly added to the set, so reconstruction of the scene is not possible.) Each pixel is described by four spectral components, corresponding to a subset of spectral bands sensed by the satellite's Multi-spectral Scanner (MSS). The components and their wavelengths are green (0.5-0.6 μm), red (0.6-0.7 μm), near infrared (0.7-0.8 μm, called "near IR" in this document), and a higher near infrared (0.8-1.1 μm, "IR"). A human observer assigned one of six terrain types to the grids, according to features at the grid center. The class labels (expressed in the data file and coded as the numbers 1-6) are "red soil", "cotton crop", "grey soil", "damp grey soil", "soil with vegetation stubble", and "very damp grey soil". I've abbreviated these labels below for brevity.[2]

The LandSat data was available from the UCI Machine Learning Repository[2] as a .csv file with 1 label column and 36 feature columns, expressed as the four spectral components for each pixel and with pixels ordered from top left to bottom right (e.g. features 17-20 are the four spectral values for the center pixel of a grid). Each row represents a different 3x3 image. The data came divided into separate training and test sets of 4435 and 2000 images, respectively.

Approach

My approach to analyzing the LandSat images was to train a variety of different classifier archetypes on the labeled data, optimizing select fitting parameters to increase classification performance. I chose K-Nearest Neighbors, Logistic Regression, and Random Forests, models which employ very different strategies to classify data. Because of the multi-class nature of the images (labeled with six terrain types), I chose mean accuracy as the scoring metric to maximize during model training (i.e. unlike the binary case, there would be no benefit if the model simply chose one class all the time). However, I also wanted cotton crop in particular to be classified with a low false positive rate, simulating the economics of knowing what farmers in the imaged region are growing this important plant. Thus, after choosing the best model, I anticipated that some cost-benefit refinement would be appropriate.

The following tools and sources were used in the analysis:

Sources: Machine Learning Repository, Univ. of California at Irvine (https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29).

Tools: 1) Python and the standard libraries *numpy*, *pandas*, *matplotlib*, and *seaboard* for routine data manipulation and graphical display.

2) Python modules *sklearn* for classification modeling and statistics generation.

3) The code editor Spyder in conjunction with Jupyter Notebooks was used to write the code.

Concepts learned and applied: Supervised classification (KNN, Decision Trees, Random Forest, Gradient Boost), confusion matrix, error statistics (accuracy, precision, etc), receiver-operator characteristics (ROC), cost-benefit analysis, web application.

Data Assessment

I first looked at the percentage occurrence of each label in the test and training sets (the UCI data was already divided into these groups). This is shown as a bar graph in Figure 1. By inspection, the two sets are comparable, so I can use them as is without reshuffling the date. Additionally, comparison across classes indicates a modest imbalance of labels. The targeted cotton crop, for example, occurs less than half as much as very damp or red soil. So care will be taken to account for this imbalance.

A similar plot for the distribution of spectral magnitudes in the data (mean and spread), regardless of pixel location, indicated that the four spectra had comparable ranges. In Figure 2, the average component magnitudes are shown with respect to their assigned terrain class (labeled on the x-axis from 1 to 6). This graph gives an idea of the different "spectral signatures" that any classifying algorithm must decode in order to correctly classify the terrains.

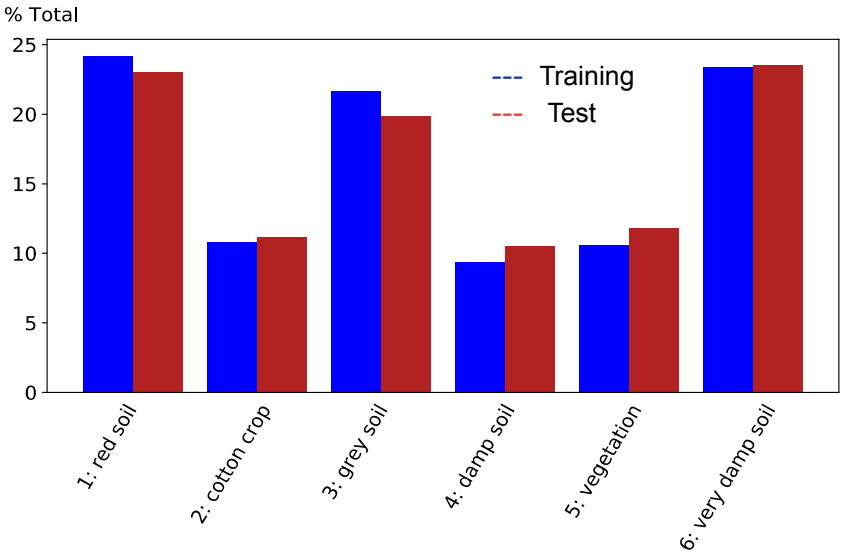

Figure 1. *Distribution of class labels in the training and test sets.*

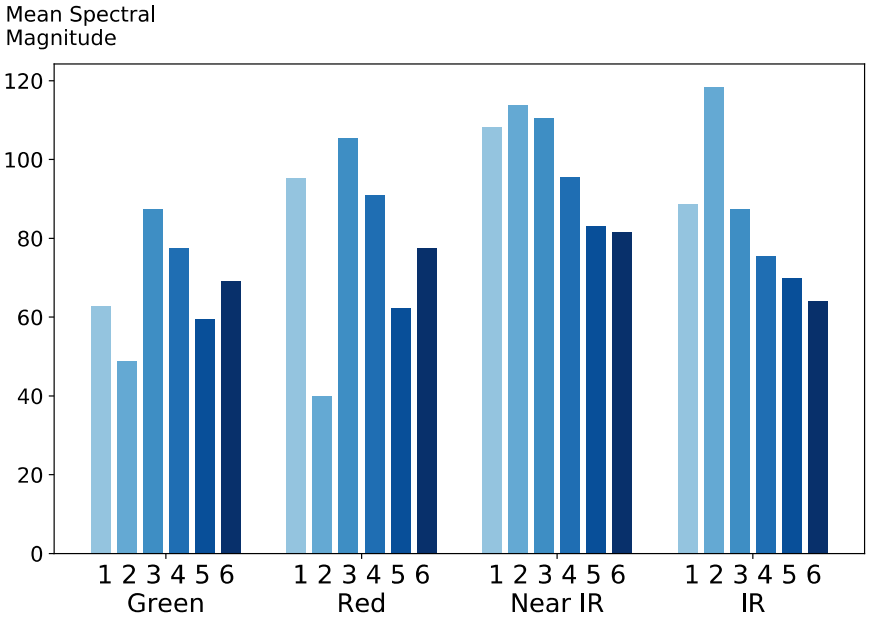

Figure 2. *Distribution of average spectral magnitudes (text labels) across the six terrain classes (numeric labels).*

Finally, I also looked at the correlation of spectral magnitudes across the pixels in a particular image. Figure 3 depicts two such correlations as scatter plots, with the center pixel of each image represented on each x-axis . As expected, the spectral

values are highly correlated. Such correlation can reduce performance of classifier types that expect features to be largely independent.

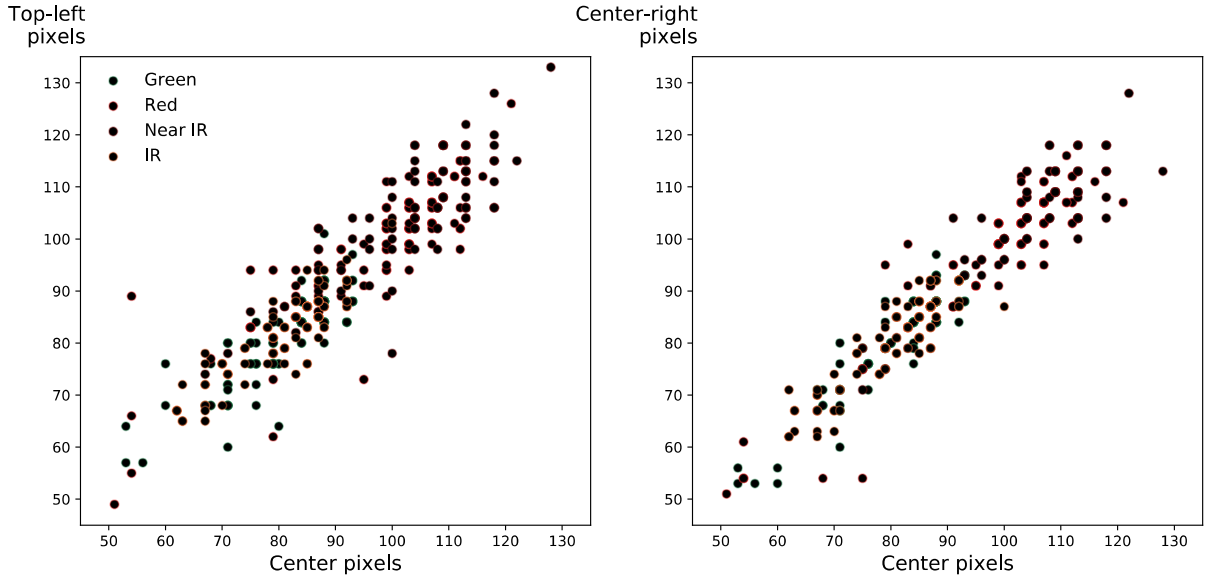

Figure 3. *Scatter plot of spectral magnitudes across all images for the top-left (left panel) and center-right (right panel) pixels compared to the center pixel. The four spectral bands are depicted with different colors.*

Initial Model Evaluation

For "K-folds" cross-validation training of the three models, the training data set (N=4435 was first split into 10 subgroups using a stratified method that sampled from each class according to their full-population proportions. Each model was then trained on these 10 subgroups, using out-of-sample points to score the predicted class with respect to the actual labeled class. Scoring was based on accuracy (the fraction of true positives or true negatives to all observations), using an average weighting across classes to account for their imbalance. During the cross-validation process, a range of fitting parameters was tested, and the combination of parameters giving the highest score was chosen to determine the best score for each classifier type. The parameters varied for each model were 1) number of neighbors and type of proximity weighting (uniform or distance) for K-Nearest Neighbors; 2) regularization constant, C, for Logistic Regression; and 3) number of estimations and maximum number of features for the Random Forest model. Figure 4 displays accuracy scores for the Random Forest parameters, indicating that 30% of features with 100 estimations was optimal.

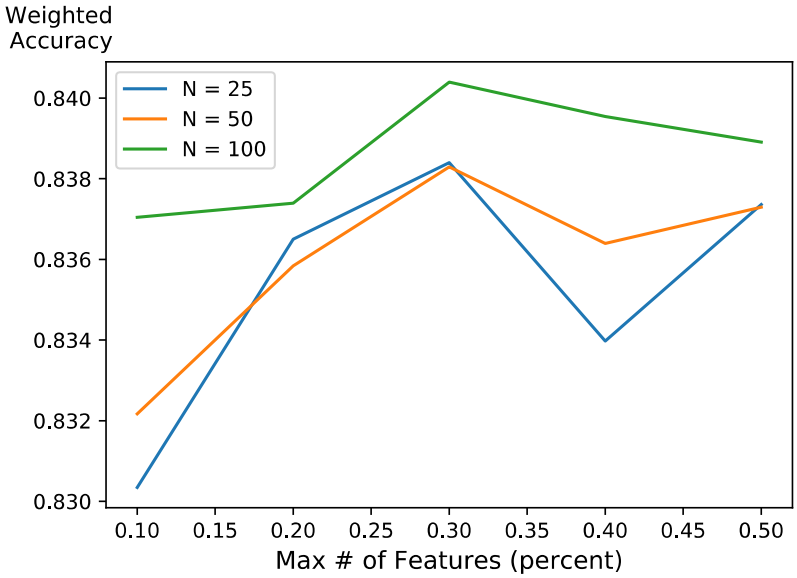

Figure 4. *Accuracy scores for different parameter combinations of the Random Forest model.*

Ultimately, the KNN and Random Forest classifiers gave similar performance, with weighted accuracy scores of 0.83 and 0.84, respectively. The Logistic Regression model, on the other hand, only performed with a 0.74 weighted accuracy. Before choosing the best model, however, another evaluation on a reduced feature set was next considered.

Principal Component Analysis

Principal component analysis was applied to reduce and effectively decorrelate the feature space. The first seven principal components explained 96.8% of the variance in the data, which reflects the high degree of correlation observed across pixels and spectral bands (Fig. 3). In the next analysis step, the same classifier models were applied to the reduced image feature set.

Final Model Fitting and Cost-Benefit Considerations

Following the cross-validation parameter-optimization using the principal component features, it was again found that the KNN and Random Forest models yielded higher prediction accuracies, at 0.82 and 0.83 respectively. Given the likely robustness of the Random Forest classifier, and its faster processing speed working on the reduced feature set, I chose it as the model for final application on the reserved test data set.

The Random Forest model, with parameters of 100 estimators and 30% maximum features, was fit to the full training set. Surprisingly, the result was perfect classification, yielding a rather trivial confusion matrix and 100% on all of the fitting statistics. Such is the nature of Decision Trees, and Random Forest algorithms in particular, when working on sufficiently large amounts of data.

It was anticipated that the Random Forest model thus produced would not give as accurate a fit to the test data, due to the expected overfitting to the training data. To improve its performance, especially on the identification of cotton crop, a cost-benefit refinement of the model predictions was imposed. Specifically, the following criteria were established: 1) minimize false identifications of cotton crop; 2) minimize confusions of cotton crop with vegetation; and 3) tolerate confusions of the different grey and damp soil types. To implement the adjustments the output of the classifier, expressed as six class probabilities for each row of data, was multiplied by a matrix of benefit scalings. The benefit matrix started as an identity matrix, with certain column/row combinations altered as needed to produce the above adjustments (e.g. a -2 for the cotton/vegetable combination, +0.2 or +0.3 to allow grey soil confusions). Similarly, the complement of the probabilities (1-p) was multiplied by a cost matrix to effect cost adjustments.

Final Model Evaluation

The confusion matrix for the final Random Forest model, applied to the principal component transformed data, is shown in Figure 5. The predictions were quite good, taking into account the tolerated misclassifications of the grey soil types, which had similar spectral content. The precision (rejection of false negatives) for the cotton crop was 1.00, an improvement over the 0.96 measured without the cost-benefit adjustment. Also, no actual cotton crop was confused for vegetation cover. This came at the cost of a lower accuracy for cotton, and lower accuracy and precision for the vegetation. The full statistics of the cost-benefit-adjusted classifier is shown below:

Class	Precision	Recall	F1-Score	Support
Red Soil	0.96	0.99	0.97	461
Cotton Crop	1.00	0.91	0.95	224
Grey Soil	0.81	0.95	0.87	397
Damp Soil	0.55	0.48	0.51	211
Vegetation	0.93	0.60	0.73	237
Very Damp Soil	0.78	0.86	0.82	470

The final unweighted accuracy and precision for the model, with all classes pooled, were both 0.84.

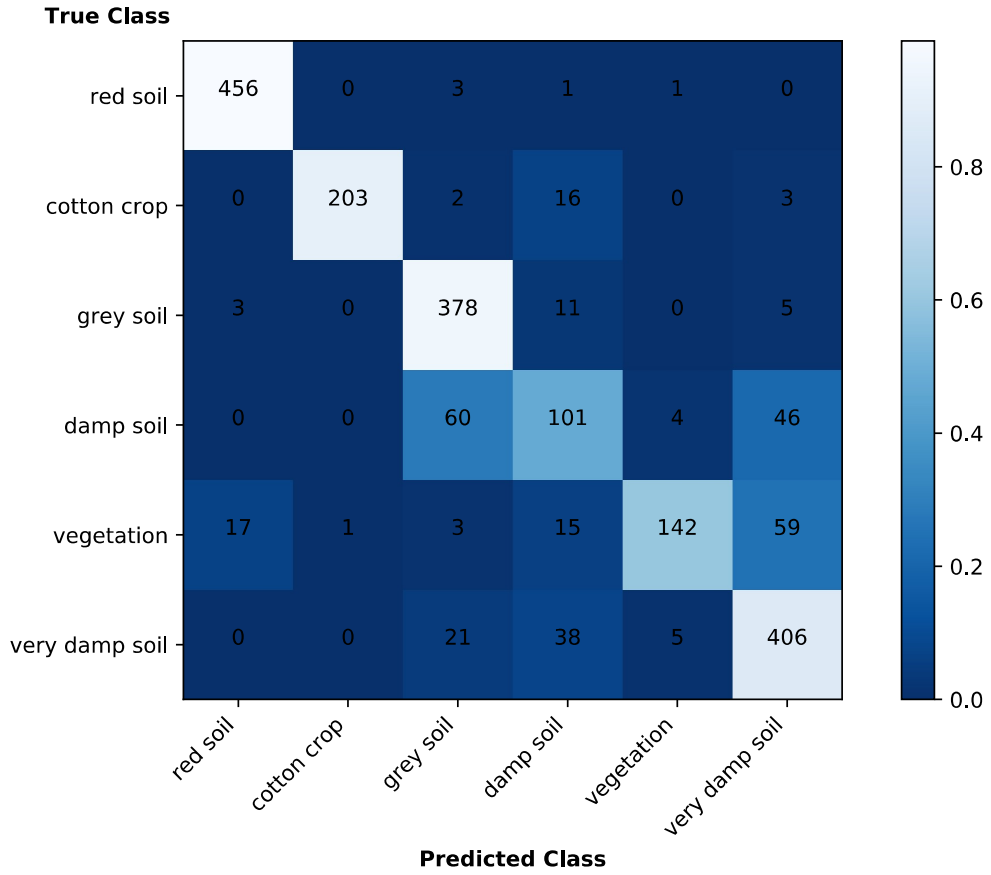

Figure 5. *Confusion matrix of the test data set using the Random Forest classifier on PC-transformed data (N = 2000). Numbers show the predicted counts for each class, while the color scale relates to the fraction of identifications.*

Conclusions

Overall, the Random Forest classifier trained to weighted accuracy score made very good predictions of the terrain imaged by the LandSat satellite. Implementation of cost-benefit considerations based on a variety of critieria, such as the rejection of false negatives or tolerance of confusions of certain classes, achieved the expected results.

Future refinements to the analysis might include the use of rotation-invariant principal components. The “Hu” image transform may also be a valuable pre-processing tool. Additionally, a deep learning network, widely used for supervised image classification problems, would likely outperform the Random Forest classifier utilized in the current study.

References

1. Wikipedia - LandSat: https://en.wikipedia.org/wiki/Landsat_program

2. UCI Repository: https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29