Random Forests : An algorithm for image classification and generation of continuous fields data sets

Ned Horning[1]

[1]American Museum of Natural History
Center for Biodiversity and Conservation
Central Park West at 79th Street
New York, NY 10024 USA
Email: horning@amnh.org

ABSTRACT

Random forests is a classification and regression algorithm originally designed for the machine learning community. This algorithm is increasingly being applied to satellite and aerial image classification and the creation of continuous fields data sets, such as, percent tree cover and biomass. Random forests has several advantages when compared with other image classification methods. It is non-parametric, capable of using continuous and categorical data sets, easy to parametrize, not sensitive to over-fitting, good at dealing with outliers in training data, and it calculates ancillary information such as classification error and variable importance. This paper provides an overview of the random forests algorithm including how it works, and advantages and limitations.

1. INTRODUCTION

The conversion of radiance data, recorded by a sensor, to other variables, such as land cover type, forest biomass, or percent tree cover, is one of the most common remote sensing tasks. Before the satellite era of remote sensing aerial photographs were interpreted manually. As the digital age and satellite remote sensing began, so did an effort, that continues today, to develop robust and efficient computer algorithms to process remotely sensed digital imagery.

In recent years a number of algorithms developed for machine learning have been adopted for remote sensing applications. These include neural networks, support vector machines, boosting, and random forests. Traditionally, remote sensing classification methods rely on statistical models to determine how radiance values recorded by a sensor should be grouped into a number of categories or classes (e.g., land cover type). These statistical approaches work on the assumption that an appropriate data model is being used and parameters for the model can be approximated from the data (Elith et al. 2008). For example, when using the maximum likelihood classifier the model assumes that the image data for each class and therefore the training data used to parameterize the model are normally distributed. A machine learning approach, on the other hand, does not start with a data model but instead learns the relationship between predictor and response data (L. Breiman 2001). By removing the need for data to fit a specific model, machine learning algorithms offer the opportunity to incorporate a diverse variety of data layers in addition to image data (e.g., digital elevation models, soil type, and climate data) into the classification algorithm.

2. RANDOM FORESTS

The random forests algorithm is a machine learning technique that is increasingly being used for image classification and creation of continuous variables such as percent tree cover

and forest biomass. Random forests is an ensemble model which means that it uses the results from many different models to calculate a response. In most cases the result from an ensemble model will be better than the result from any one of the individual models (Dahinden 2009). In the case of random forests, several decision trees are created (grown) and the response is calculated based on the outcome of all of the decision trees.

2.1 Decision trees

In order to understand how random forests works it is necessary to become familiar with decision trees. Decision trees are predictive models that use a set of binary rules to calculate a target value. Two types of decision trees are classification trees and regression trees. Classification trees are used to create categorical data sets such as land cover classification and regression trees are used to create continuous data sets such as biomass and percent tree cover.

Figure 1 shows a very simple classification tree and the resulting land cover map creating using the red (band 3) and near-infrared (band 4) bands from the Landsat Enhanced Thematic Mapper Plus satellite sensor as the predictor variables. The tree is a set of binary decisions and terminal nodes connected by branches (lines in the figure). The first (top) decision node, also called the root node, evaluates the rule "is band 4 less than or equal to 46". If it is then a terminal node is reached and that node is assigned to the class "water". If band 4 is greater than 46 then another binary decision node is evaluated; "is band 3 less than or equal to 102". The tree continues to grow until all branches end with terminal nodes.

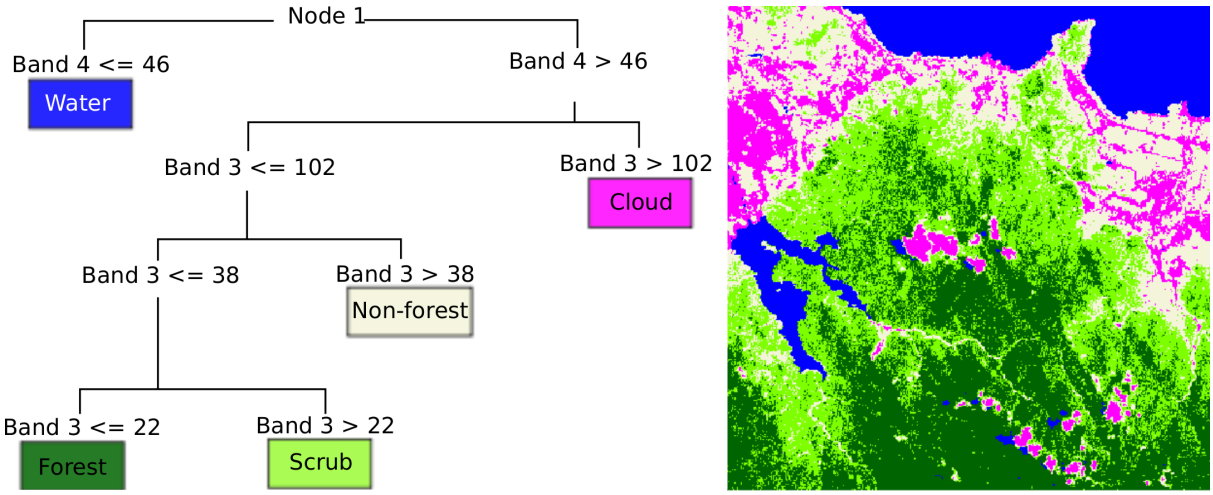

Figure 1. A classification tree (left) and the resulting land cover map (right)

If the tree in Figure 1 were a regression tree the terminal nodes would contain predicted function (modeled) values such as biomass or whatever continuous response variable is being modeled.

Using a binary decision tree to classify an image using greater than or less than rules with continuous data results in the partitioning of feature space using n-dimensional rectangles (Figure 2). In Figure 2 the two-dimensional feature space plot is partitioned using a set of two-dimensional rectangles based on the classification tree on the left. This is a very simplistic classification tree. When several predictor variables are used the tree and the resulting partitioning of the feature space would be significantly more complex with many more rectangles. Using categorical data, such as soil type, as a predictor variable would also

result in a more complex feature space plot. One of the drawbacks of using decision trees is that rectangles are rarely an effective way to partition feature space.

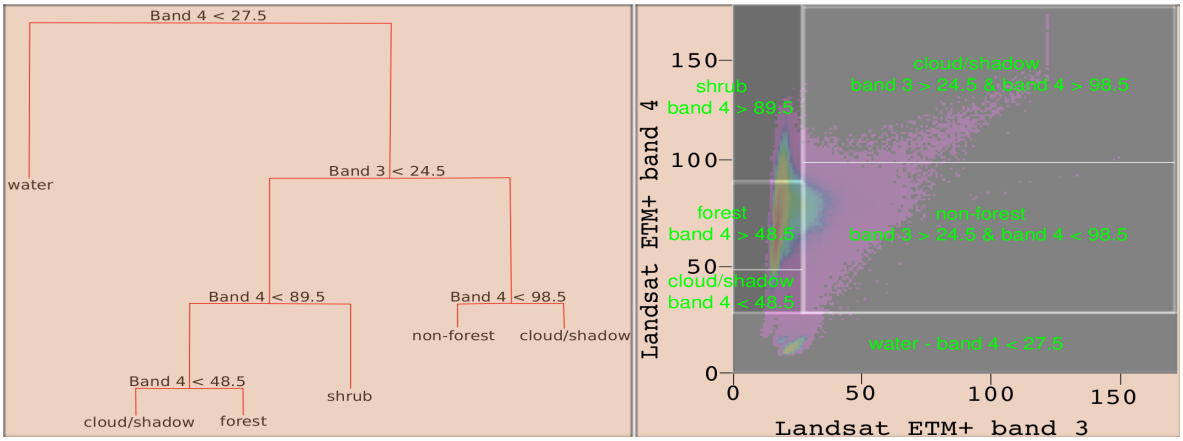

Figure 2. A classification tree (left) and resulting partitioned feature space plot (right)

Another drawback of decision trees is that they tend to over-fit the training data which can give poor results when the model is applied to the full data set. This happens when the tree grows too large with the terminal nodes representing very small subsets of the training data. To overcome this problem a process call "pruning" is used to simplify the tree by removing terminal nodes that could be linked to noise in the data. This will reduce the effects of over-fitting and improve the predictive power of the model. The problem is that the pruning process is somewhat subjective and it can be cumbersome to prune a tree so it gives the lowest error for a set of test data.

Some advantages of using decision for image classification or regression applications are:

- Easy to interpret the rules using a tree diagram;
- It is a nonparametric model and it is easy to incorporate a range of numeric or categorical data layers;
- No need to select training data with a unimodal distribution; and
- Classification is fast once the rules are developed.

2.2 How random forests works

Random forests, like decision trees, can be used to solve classification and regression problems but it is able to overcome the drawbacks associated with single decision trees while maintaining the benefits. The random forests model calculates a response variable (e.g., land cover, percent tree cover) by creating many (usually several hundred) different decision trees (the forest of trees) and then putting each object to be modeled (in our case the object is a multi-layered pixel) down each of the decision trees. The response is then determined by evaluating the responses from all of the trees. In the case of classification the class that is predicted most is the class that is assigned for that object (Leo Breiman & Cutler A.). In other words, if 500 trees are grown and 400 of them predict that a particular pixel is forest and 100 predict it is grass the predicted output for that pixel will be forest. In the case of regression the resulting value for an object is the mean of all of the predictions. Since predictions from random forests are derived using a forest of trees it is not possible to easily illustrate how the predictions are made. To illustrate the process it would be necessary to draw all of the trees for each prediction which would result in hundreds of decision tree diagrams for each model.

The key to the success of random forests is how it creates each of the decision trees that make up the forest. There are two steps involving random selection that are used when forming the trees in the forest. The first step involves randomly selecting, with replacement, data from supplied training areas to build each tree. For each tree a different subset of the training data are used to develop the decision tree model and the remaining one-third of the training data are used to test the accuracy of the model. The sample data used for testing are often called the "out-of-bag" samples. The second random sampling step is used to determine the split conditions for each node in the tree. At each node in the tree a subset of the predictor variables is randomly selected to create the binary rule. The number of predictor variables that are randomly selected can be set by the user or the choice can be left to the random forest algorithm. Using a randomly selected subset of the predictor variables to split each node results in less correlation among trees and as a result a lower error rate. If all of the variables were used for each tree the trees would be nearly identical (highly correlated), which would result in a higher error rate (L. Breiman 2001). Although smaller subsets of predictor variables will reduce correlation between trees it also results in trees with less predictive power than trees built using more predictor variables. It is important to select the number of variables that provides sufficiently low correlation with adequate predictive power. Fortunately, the optimum range for the subset of predictor variables is quite wide and there are easy tests that can be performed to select an optimum subset size (Pal 2005).

When running random forests there are a number of parameters that need to be specified. The most common parameters are (Liaw & Wiener 2002):

- Input training data including predictor variables such as image bands and digital elevation models and response variables such as land cover type and biomass;
- The number of trees that should be built;
- The the number of predictor variables to be used to create the binary rule for each split; and
- Parameters to calculate information related to error and variable significance.

2.3 Error reporting, variable significance, and outliers

One of the tremendous benefits of random forests is that it is able to calculate useful information about errors, variable importance, and data outliers. This information can be used to evaluate the performance of the model and make changes to the training data if necessary.

Since random forests only uses roughly two-thirds of the training data to build the random forests model the remaining one-third of the training data (the out-of-bag samples) can be used to estimate the error of the predictions. This greatly simplifies the accuracy assessment portion of each analysis. Another related feature is the ability to plot the error rate vs. the number of trees in a forest. This plot can give an indication if the number of trees in the forest is sufficient. If the error rate is stable the number of trees should be sufficient.

Variable importance is also provided by the random forests algorithm. By identifying variables that contribute little information to the analysis, it is possible to re-run the model without those variables. This can be particularly important if several dozens of variables are used. The determination of variable importance is accomplished for each tree by randomly reordering the values of a single predictor variable in the out-of-bag samples and then putting the samples down each tree and repeating this process for each predictor variable. In effect, this reordering substitutes a predictor variable for a particular pixel with a predictor variable

from another pixel. For example, if there are 100 out-of bag samples and each sample has 6 predictor variables, (e.g., satellite image bands) randomly reorder all values of the first predictor variable (e.g., band one) and then run those samples through the tree. This will create the case where the values for the band one variable is mixed up so, for example, an out-of-bag sample from a forest pixel could have a band one value from a water pixel. Repeat this for all of the predictor variables (e.g., all bands) for that tree and then repeat the whole process for all of the trees in the forest. By measuring how much the model prediction changes, it is possible to estimate the importance of that variable (L. Breiman 2001). In other words, if the model prediction is greatly impacted by replacing a valid variable with one from another pixel then it can be assumed that the variable is quite important but if the prediction is not effected much then the variable has little impact on the predicted value.

It is also possible to evaluate outliers in the training data if a classification model is being used. Outliers are data that are not very similar to other data for a particular class. For example, with sample data for a forest class there might be some pixels that represent non-forest mixed in with the forest pixels. By identifying which classes have significant outliers it becomes possible to re-evaluate the training data to see if it is necessary to edit the samples.

2.4 Advantages and limitation

There are a number of advantages to using random forests. It has been found to be comparable to other machine learning algorithms such as boosting, and support vector machines but with the advantage that random forests is not very sensitive to the parameters used to run it and it is easy to determine which parameters to use (L. Breiman 2001). Overfitting is less of an issue than it is with individual decision trees and there is no need for the cumbersome task of pruning the trees. Lastly, the ability of automatically producing accuracy and variable importance and information about outliers makes random forests easier to use effectively.

There are some limitations when using random forests, especially when using it for regression. Due to the way regression trees are constructed it is not possible to predict beyond the range of the response values in the training data. For example, if the training data for a biomass model contain low and moderate biomass values but no high values it will not be possible to accurately predict high biomass values when the model is applied to the full data set. It is extremely important that the training data include samples that cover the entire range of response data values.

Another issue related to regression is that random forests tends to overestimate the low values and underestimate the high values. This is because the response from random forests in the case of regression is the average (mean) of all of the trees.

3. PRACTICAL RESOURCES

There are a number of freely available resources that are useful to implement the random forest algorithm. The most important is software to run random forests. There is a package called “randomForest” that can be run using the open source software “R” (R Development Core Team 2009). R is a software environment for statistical computing and graphics that runs on most computer platforms including Linux, Mac OS, and Microsoft Windows. The randomForest package (Liaw & Wiener 2002) available for R is a full-featured implementation of the random forests algorithm and it comes complete with

documentation and example data.

To facilitate using random forest for image classification and regression and number of guides have been developed and are available on the American Museum of Natural History's Center for Biodiversity and Conservation Biodiversity Informatics Facility web site: http://biodiversityinformatics.amnh.org/. These guides provide detailed information about how to use random forests for creating land cover, biomass, and percent cover maps using satellite imagery. Scripts are provided with the guides to make is easy to implement random forests for image classification and regression.

A good website to better understand the details of the random forests algorithm is the site developed by the man who developed the algorithm, Leo Breiman (http://www.stat.berkeley.edu/~breiman/RandomForests/). Another document written by Leo Breiman that describes random forests is the manual for setting up and usnig his original implementation (http://oz.berkeley.edu/users/*breiman*/*Using_random_forests_v4.0.pdf*).

4. CONCLUSION

Random forests is a robust algorithm that can be used for remotely sensed data classification and regression. Performance of random forests is on par with other machine learning algorithms but it is much easier to use and more forgiving with regard to over fitting and outliers than other algorithms. Some common applications of random forests classification include land cover and land cover change mapping and cloud and shadow detection. Regression applications include continuous filed mapping (e.g., percent tree cover, percent shrub cover, impervious surfaces) and biomass mapping. At this point in time, random forests is gaining in popularity but it is still not a common approach for image classification and regression largely because many remote sensing practitioners are unaware of the algorithm.

5. ACKNOWLEDGEMENTS

The author would like to thank the John D. and Catherine T. MacArthur Foundation for supporting the development of the random forest guides and this paper.

6. REFERENCES

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), pp.5–32.

Breiman, L. and Cutler, A., Random Forests. Available at: http://www.stat.berkeley.edu/~breiman/RandomForests/ [Accessed October 12, 2010].

Dahinden, C., 2009. An improved Random Forests approach with application to the performance prediction challenge datasets. *Hands on Pattern Recognition. Microtome*.

Elith, J., Leathwick, J.R. and Hastie, T., 2008. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4), pp.802–813.

Liaw, A. & Wiener, M., 2002. Classification and regression by randomForest. *R News*, 2(3), pp.18-22.

Pal, M., 2005. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1), pp.217–222.

R Development Core Team, 2009. *R: A Language and Environment for Statistical Computing*, Vienna, Austria. Available at: http://www.R-project.org.