**INTRODUCTION TO RANDOM FORESTS**

by Simon Moss

|  |
| --- |
| **Introduction** |

Random forests extend decision trees. Therefore, before reading this document, you should, at least briefly, read the guidelines on decision trees. However, random forests circumvent a key limitation of decision trees. In particular, decision trees tend to explain your original data effectively but do not seem to predict future events well—a problem that random forests overcome.

|  |
| --- |
| **Rationale that underpins random forests** |

To conduct random forests, researchers invariably use software programs, such as R. The program typically implements four main steps.

**Generate a bootstrap sample**

First, the program first constructs a bootstrap sample. To illustrate, suppose researchers want to predict which candidates are likely to complete their PhD or Masters by Research. To achieve this goal, the researchers collect data about previous applicants. An extract of these data appear in the following table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| John | Yes | Research Masters | Yes | 6 | Male |
| Karen | Yes | Research  Masters | No | 6 | Female |
| Len | No | Honours | No | 5 | Female |
| Marsha | No | Honours | Yes | 6 | Male |
| Neil | Yes | Bachelor | No | 8 | Male |
| … | … | … | … | … | … |
| Olivia | No | Coursework Masters | No | 6 | Female |

Suppose this dataset actually comprises 50 rows or people. To construct a bootstrap sample, the program would first randomly choose one row of data, such as the data associated with Karen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| Karen | Yes | Research  Masters | No | 6 | Female |

The program would then randomly choose a second row or person. This second row could be Karen again or someone different, as shown below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| Karen | Yes | Research  Masters | No | 6 | Female |
| Neil | Yes | Bachelor | No | 8 | Male |

The program would continue this procedure, until 50 rows or people appear—equivalent to the total number of rows or people in the original sample. This updated dataset is called a bootstrap sample. The bootstrap sample can include duplicate entries or identical rows, as shown below, and thus differs from the original dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| Karen | Yes | Research  Masters | No | 6 | Female |
| Neil | Yes | Bachelor | No | 8 | Male |
| Karen | Yes | Research  Masters | No | 6 | Female |
| Olivia | No | Coursework Masters | No | 6 | Female |
| Neil | Yes | Bachelor | No | 8 | Male |
| … | … | … | … | … | … |
| Neil | Yes | Bachelor | No | 8 | Male |

**Construct a decision tree**

Next, the program subjects the bootstrap sample to a decision tree. The following diagram is an extract of the output.



Actually, the program utilizes a modified procedure to construct these decision trees. Specifically, at each step in the decision tree, the program considers only a subset of the variables or columns—such as two variables or columns. To clarify

* in the first step or split, a typical decision tree will analyze all four columns or variables—highest degree, published papers, IELTS score, and gender—and then split the data on the most informative of these characteristics
* however, this modified decision tree will analyze only two columns or variables—perhaps highest degree and IELTS score. The decision tree might then split the data according to highest degree
* in the second step or split, this modified decision tress will then analyze two of the remaining columns or variables—perhaps IELTS score and published papers, and so forth
* researchers can control whether the program should examine two, three, or more variables at each step or split

**Repeat this pair of procedures many times**

The program that repeats this pair of procedures 500 or so times. In particular, each time

* the program constructs a different bootstrap sample
* the program constructs a decision tree. At each step of each decision tree, the program randomly selects two variables or columns to analyze.

The following diagram schematically illustrates a subset of these decision trees. This set of decision trees is called a random forest.





**Application of a random forest**

So far, this document has clarified how the program generates a random forest. But, how is this random forest helpful? How can researchers utilize this random forest? To illustrate the utility of this random forest, consider someone who wants to enrol in a PhD. This person

* has completed an Honors degree
* has published no papers
* received an IELTS score of 7
* is female.

To predict whether this person is likely to complete her PhD, the program could subject these scores to all the decision trees. That is, as you might recall, each decision tree will convert this information about the person to a prediction as to whether she is more likely than not to complete her thesis. The following diagram illustrates these predictions.



Suppose that 60% of the 500 trees predict this person will complete her thesis. The researcher would thus conclude the candidate is more likely than not to complete. This procedure, in which the program generates many bootstrap samples and then aggregates the predictions of all the decision trees is called bagging.

**Evaluation of a random forest**

Finally, researchers need to evaluate this random forest. To evaluate the random forest, we should first recognize the original dataset differed from the bootstrapped dataset. In particular

* recall that our bootstrapped dataset included some duplicate rows
* consequently, some of the rows or people in the original dataset were not included in the bootstrapped dataset
* indeed, usually about one third of the rows in the original dataset are not included in the bootstrapped dataset. These excluded rows or individuals are called the out-of-bag dataset

To evaluate the random forest, we can now subject each out-of-bag row or participant to the random forest. That is, in the previous page, we introduced a procedure that can be applied to predict the outcome—whether or not a candidate is likely to complete a thesis—from four characteristics: highest degree, number of published papers, IELTS score, and gender. We can now apply the same procedure to predict the outcome for every row or person in the out-of-bag dataset. To illustrate, in the following table, the first column presents the predicted outcome of each row or individual. The second column presents the actual outcome of each row or individual. The final column specifies whether or not the prediction was accurate.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted outcome | Actual outcome | Was the prediction correct? |
| Marsha | Will complete | Completed | Correct |
| Paul | Will not complete | Did not complete | Correct |
| Rebecca | Will complete | Did not complete | Incorrect |
| … | … | … | … |

Suppose that .20 or 20% of these predictions were incorrect. This number, .20, is called the out-of-bag error.

**Evaluating alternative parameters**

Researchers can modify some of the parameters. To illustrate, in the previous example the program generated 500 decision trees. But, the researcher could instead instruct the program to generate 1000 decision trees. So, how should researchers decide the appropriate number of decision trees. In essence, researchers could test a variety of options. For example

* they might discover the out-of-bag error is the same whether the program generated 500 decision trees or 100 decisions trees
* they would then conclude they have probably reached a plateau;
* more than 1000 decisions trees is unlikely to reduce the out-of-bag error.

Likewise, researchers might vary the number of columns or variables the program analyses at each step or split of the decision tree. To illustrate, at each step or split of the decision tree, the program analyzed only two of the columns or variables, such as highest degree and gender, at a time. However, the researcher could then

* instruct the program to analyze two of the predictors at each step of the decision tree
* instruct the program to analyze three of the predictors at each step of the decision tree
* instruct the program to analyze four of the predictors at each step of the decision tree and so forth
* choose the option that generates the lowest out-of-bag error

But, if the dataset comprises 100 predictors, should the researcher apply the same procedure? That is, should the researcher instruct the program to analyse two, three, and then four predictors at each step. Or should the research instruct the program to analyse 10, 11, and 12 predictors at each step. To answer this question, when the outcome is categorical, such as whether or not individuals complete their thesis

* calculate the square root of the number of predictors
* in this instance, the number of predictors is 4; the square root is 2
* begin with this number.

When the outcome is numerical, such as weight,

* calculate the number of predictors minus 3
* if the number of predictors is 10, 10 minus 3 is 7
* the researcher might then analyze about 6, 7, and 8 predictors at each step.

|  |
| --- |
| **Missing data** |

**Data missing from the original dataset**

When conducting random forests, you need to decide how to manage and accommodate missing data. In particular, you need to differentiate two variants of missing data. First, as the following table shows, some data might be missing from the original data set—the set that was utilized to generate the random forest. The red square in this table signifies missing data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| John | Yes | Research Masters | Yes | 6 | Male |
| Karen | Yes |  | No | 6 | Female |
| Len | No | Honours | No | 5 | Female |
| Marsha | No | Honours | Yes | 6 | Male |
| Neil | Yes | Honours | No | 8 | Male |
| … | … | … | … | … | … |
| Olivia | No | Coursework Masters | No | 6 | Female |

To manage missing data, the program will first identify the most common value for the corresponding variable or column. To illustrate, in this example, the most common highest degree may be Honours. Therefore, the program will initially substitute the missing data in this column with the label Honours—the mode of this column—as the following table shows. As an aside, when the variable is numeric, the program will initially substitute the missing data with the median of this column.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | Did this person complete | Highest degree | Published papers | IELTS score | Gender |
| John | Yes | Research Masters | Yes | 6 | Male |
| Karen | Yes | Honours | No | 6 | Female |
| Len | No | Honours | No | 5 | Female |
| Marsha | No | Honours | Yes | 6 | Male |
| Neil | Yes | Honours | No | 8 | Male |
| … | … | … | … | … | … |
| Olivia | No | Coursework Masters | No | 6 | Female |

The program will then gradually improve these estimates of the missing data. To achieve this goal, the program must next determine another row that is most similar to the row with missing data. To achieve this goal, the program applies an interesting and important procedure, generating a table of numbers that is called a proximity matrix. Specifically

* the program first subjects each row—that is, the data of each person—to the first tree of our random forest
* the program then determines which rows or participants ended at the same leaf of this tree

To illustrate, consider the following diagram. When the characteristics of John and Karen are entered into the first tree, they end on the left leaf. When the characteristics of Neil are entered into the first tree, he ends on the right leaf. Therefore, as summarized in the following matrix,

* John and Karen end on the same leaf—represented by the number 1 in bold
* John and Neil as well as Karen and Neil end on different leaves—represented by the 0s in bold



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Karen | Len | Marsha | Neil | … |
| John | **1** | 1 | 0 | **0** | … |
| Karen |  | 0 | 1 | **0** | … |
| Len |  |  | 1 | 1 | … |
| Marsha |  |  |  | 0 | … |
| Neil |  |  |  |  | … |
| … | … | … | … | … | … |

The program then repeats this procedure for every other tree in the random forest. Hence, the following matrix represents the number of trees in which each pair of individuals ended at the same leaf.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Karen | Len | Marsha | Neil | … |
| John | 192 | 142 | 323 | 412 | … |
| Karen |  | 142 | 247 | 291 | … |
| Len |  |  | 98 | 89 | … |
| Marsha |  |  |  | 78 | … |
| Neil |  |  |  |  | … |
| … | … | … | … | … | … |

Finally, the program divides these numbers by the number of trees. Therefore, in the following matrix, each number represents the proportion of trees in which each pair of individuals ended at the same leaf.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Karen | Len | Marsha | Neil | … |
| John | .34 | .23 | .65 | .83 | … |
| Karen |  | .23 | .52 | **.62** | … |
| Len |  |  | .19 | .18 | … |
| Marsha |  |  |  | .17 | … |
| Neil |  |  |  |  | … |
| … | … | … | … | … | … |

As this proximity matrix shows, the person who corresponded to the missing data—Karen—was more similar to Neil than to John, Len, and Marsha. So, how can we utilize this proximity data to improve our estimates of the missing data? To answer this question, consider this rationale:

* according to the proximity matrix, Karen is very different to Len
* therefore, because the highest degree for Len is an Honors, the highest degree for Karen is unlikely to be Honors
* similarly, according to the proximity matrix, Karen is more similar to Neil
* hence, because the highest degree for Neil is a Bachelor degree, the highest degree for Karen is more likely to be a Bachelor degree as well

To represent this rationale statistically, the program utilizes a variant of the following formula. The program then identifies the most likely of these degrees. The rationale is the same if the variable is numeric, but the formula needs to be adjusted slightly.

|  |  |
| --- | --- |
| Formula | |
| Probability the missing value is a Bachelor | Sum the proximities of each person with a Bachelor to Karen |
| Probability the missing value is an Honours | Sum the proximities of each person with an Honours to Karen. |
| Probability the missing value is a coursework Masters | Sum the proximities of each person with a coursework Masters to Karen. |
| Probability the missing value is a research Masters | Sum the proximities of each person with a research Masters to Karen. |

The program will then repeat this procedure many times until the estimates of missing data no longer seem to change.

**Data missing from additional participants**

As discussed in a previous section, researchers apply random forests to predict the outcomes of additional participants or units, such as a future applicant. Sometimes, in these circumstances, some of the data are missing as well. For example, consider someone who wants to enrol in a PhD. This person

* has published no papers
* received an IELTS score of 7
* is female.
* but the highest degree of this individual was not specified.

How does the program manage the missing data in these circumstances? In essence, the

* program first assumes the individual had achieved one of the possible outcomes, such as completed the thesis.
* the program utilizes the previous iterative method to derive the best guess of the missing data, such as a Bachelor degree
* the program also computes the out-of-bag error, such as .23—as shown in the following table
* the program next assumes the individual had achieved the other possible outcome, such as not completed the thesis—and then again derives the best guess of the missing data and the out-of-bag error

The results appear in the following table. The out-of-bag error is lower when the person was assumed to complete the thesis. Consequently, the program will predict the person is likely to complete the thesis.

|  |  |
| --- | --- |
| Assume person had completed | Assume person had not completed |
| Missing data: most likely a Bachelor | Missing data: most likely an Honours |
| Out-of-bag error: 0.23 | Out-of-bag error: 0.35 |

|  |
| --- |
| **How to conduct a random forest in R** |

Unless you live to 1000 years, few researchers are able to conduct a random forest manually, but instead need to utilize a software program. Many researchers utilize R to conduct random forests. If you have not used R before, please first read the guidelines on R, available in the CDU webpage about “choosing your research methodology and methods”.

Several packages are available to conduct a random forest in R. The following table outlines some of the code you could utilize to achieve this goal.

**Generate the random forest**

|  |  |
| --- | --- |
| Sample R code | Details and explanation |
| * library(gglot2) * library(cowplot) * library(randomForest) | * ggplot and cowplot can be used to construct helpful plots later |
| * set.seed(42) | * Sets a seed for the random number generator, enabling researchers to reproduce their results |
| * datafile.imputed <- rfImpute(outcome~ ., data = datafile, iter=6) | This code is used to substitute the missing data with the best estimates. The code assumes that   * whether or not the person completed is labelled as “outcome” * the dataset is called “datafile” * you can increase the number of iterations from 6 to a higher number to increase the accuracy of these estimates. But, typically, 6 is enough |
| * model <- randomForest(outcome ~ ., data = datafile.imputed, ntree = 1000, mtry=3, proximity = TRUE) | This code generates the random forests. In particular   * the number of trees in this example is 1000; 500 is the default * the number of predictors the program analyses at each step or split is 3—but researchers can try different numbers * proximity = TRUE will generate the proximity matrix: this matrix can be subjected to further analyses, such as multidimensional scaling, to uncover clusters of participants and achieve other goals |
| * model | This command will then generate information about the random forest such as   * the out of bag error rate, called OOB * you could shift some of the parameters, such as the number of predictors the program analyses at each step, to attempt to minimize the OOB |

The output that is generated specifies

* the type of random forest—called "classification random forest” if the outcome is categorical and “regression random forest” if the outcome is numerical. The type is called “unsupervised” if the model includes no outcome variable
* specifies the number of trees
* specifies the number of predictors at each split or step
* reports the out-of-bag or OOB error rate
* presents something called the confusion matrix. The columns specify the actual outcomes and the rows specify the predicted outcomes.

**Apply the random forest to a specific case or person**

The previous R code will construct a random forest. However, this random forest is not especially helpful unless the model is utilized to predict the outcome of future cases—such as to predict whether a future application will complete the thesis or not. To achieve this goal, consider some of the following code.

|  |  |
| --- | --- |
| Sample R code | Details and explanation |
| * newdata1= c(“Bachelor”, “None”, 6, “Male”) | * Creates a vector that represents the predictors * You could construct a data frame if you want to include more than one person at a time |
| * predictedmodel <- predict(model, newdata1) |  |
| * predictedmode | * displays the prediction |

You may consult other sources to improve the output. For example, you might learn how to

* subject the proximity matrix to multidimensional scaling
* examine the association between the OOB error rate and various parameters, such as the number of trees