GUIDE Classification and Regression Trees*
User Manual for Version 19.0

Wei-Yin Loh
Department of Statistics
University of Wisconsin–Madison
April 16, 2015

Contents

1 Warranty disclaimer ................................................. 2

2 Introduction .......................................................... 3
  2.1 Installation ...................................................... 6
  2.2 LATEX .............................................................. 6

3 Program operation .................................................... 7
  3.1 Required files .................................................... 7
  3.2 Input file creation ............................................... 9

4 Classification .......................................................... 10
  4.1 Default: univariate splits ..................................... 10
    4.1.1 Input file creation with default options ............... 10
    4.1.2 Contents of input.txt .................................. 12
    4.1.3 Executing the program and interpreting the output ... 13
  4.2 Non-default options .......................................... 18
    4.2.1 Linear splits ............................................. 18
    4.2.2 Equal priors ............................................. 20

*Based upon work partially supported by grants from the U.S. Army Research Office, the National Science Foundation and the National Institutes of Health.
4.2.3 Unequal misclassification costs: hepatitis data 21
4.2.4 Nearest-neighbor estimates: car data 23
4.2.5 Kernel density estimates: car data 32

5 Regression 45
5.1 Stepwise least-squares 45
5.2 Least-squares simple polynomial 51
5.3 ANCOVA models 55
5.4 Quantile regression 62
5.5 Least median of squares 68
5.6 Poisson regression with offset 76
5.7 Censored response data 85
5.8 Multi-response data 94
5.9 Longitudinal data 99
5.10 Subgroup identification 111
5.11 Differential item functioning 117

6 Tree ensembles 124
6.1 Bagged GUIDE 124
6.2 GUIDE forest 127

7 Importance scores 130
7.1 Baseball data example 130

8 Other features 134
8.1 Pruning with test samples 134
8.2 Prediction of test samples 134
8.3 GUIDE in R and in simulations 134
8.4 Generation of powers and products 135
8.5 Data formatting functions 136

1 Warranty disclaimer

Redistribution and use in binary forms, with or without modification, are permitted provided that the following condition is met:

Redistributions in binary form must reproduce the above copyright notice, this condition and the following disclaimer in the documentation and/or other materials provided with the distribution.
2 Introduction

GUIDE stands for Generalized, Unbiased, Interaction Detection and Estimation. It is the only classification and regression tree algorithm with all these features:

1. Unbiased variable selection.
2. Kernel and nearest-neighbor node models for classification trees.
3. Weighted least squares, least median of squares, quantile, Poisson, and relative risk (proportional hazards) regression models.
4. Univariate, multivariate, and longitudinal response variables.
5. Pairwise interaction detection at each node.
6. Linear splits on two variables at a time for classification trees.
7. Categorical variables for splitting only, or for both splitting and fitting (via 0-1 dummy variables), in regression tree models.
8. Ranking and scoring of predictor variables.
9. Tree ensembles (bagging and forests).
Table 1: Comparison of GUIDE, QUEST, CRUISE, CART, and C4.5 classification tree algorithms. Node models: S = simple, K = kernel, L = linear discriminant, N = nearest-neighbor.

<table>
<thead>
<tr>
<th></th>
<th>GUIDE</th>
<th>QUEST</th>
<th>CRUISE</th>
<th>CART</th>
<th>C4.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unbiased splits</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Splits per node</td>
<td>2</td>
<td>2</td>
<td>≥ 2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Interaction</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Importance</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ranking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class priors</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Misclassification</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear splits</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Categorical</td>
<td>Subsets</td>
<td>Subsets</td>
<td>Subsets</td>
<td>Subsets</td>
<td>Atoms</td>
</tr>
<tr>
<td>splits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node models</td>
<td>S, K, N</td>
<td>S</td>
<td>S, L</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Missing values</td>
<td>Special</td>
<td>Imputation</td>
<td>Surrogate</td>
<td>Surrogate</td>
<td>Weights</td>
</tr>
<tr>
<td>Tree diagrams</td>
<td>Text and \LaTeX</td>
<td>Proprietary</td>
<td>Text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bagging</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Forests</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Tables 1 and 2 compare the features of GUIDE with CRUISE (Kim and Loh, 2001, 2003), QUEST (Loh and Shih, 1997), C4.5 (Quinlan, 1993), RPART\(^1\), and M5’ (Quinlan, 1992; Witten and Frank, 2000).


This manual illustrates the use of the program and interpretation of the output.

\(^1\)RPART is an implementation of CART (Breiman et al., 1984) in R. CART is a registered trademark of California Statistical Software, Inc.
Table 2: Comparison of GUIDE, CART and M5’ regression tree algorithms

<table>
<thead>
<tr>
<th></th>
<th>GUIDE</th>
<th>CART</th>
<th>M5’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unbiased splits</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Pairwise interaction detection</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Importance scores</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Loss functions</td>
<td>Weighted least squares, least median of squares, quantile, Poisson, proportional hazards</td>
<td>Least squares, least absolute deviations</td>
<td>Least squares only</td>
</tr>
<tr>
<td>Survival, longitudinal and multi-response data</td>
<td>Yes, yes, yes</td>
<td>No, no, no</td>
<td>No, no, no</td>
</tr>
<tr>
<td>Node models</td>
<td>Constant, multiple, stepwise linear, polynomial, ANCOVA</td>
<td>Constant only</td>
<td>Constant and stepwise</td>
</tr>
<tr>
<td>Linear models</td>
<td>Multiple or stepwise (forward and forward-backward)</td>
<td>N/A</td>
<td>Stepwise</td>
</tr>
<tr>
<td>Variable roles</td>
<td>Split only, fit only, both, neither, weight, censored, offset</td>
<td>Split only</td>
<td>Split and fit</td>
</tr>
<tr>
<td>Categorical variable splits</td>
<td>Subsets of categorical values</td>
<td>Subsets</td>
<td>0-1 variables</td>
</tr>
<tr>
<td>Tree selection</td>
<td>Pruning or stopping rules</td>
<td>Pruning only</td>
<td>Pruning only</td>
</tr>
<tr>
<td>Tree diagrams</td>
<td>Text and \LaTeX</td>
<td>Proprietary</td>
<td>PostScript</td>
</tr>
<tr>
<td>Operation modes</td>
<td>Interactive and batch</td>
<td>Interactive and batch</td>
<td>Interactive</td>
</tr>
<tr>
<td>Case weights</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Transformations</td>
<td>Powers and products</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Missing values in split variables</td>
<td>Missing values treated as a special category</td>
<td>Surrogate splits</td>
<td>Imputation</td>
</tr>
<tr>
<td>Missing values in linear predictors</td>
<td>Choice of separate constant models or mean imputation</td>
<td>N/A</td>
<td>Imputation</td>
</tr>
<tr>
<td>Bagging &amp; forests</td>
<td>Yes &amp; yes</td>
<td>No &amp; no</td>
<td>No &amp; no</td>
</tr>
<tr>
<td>Data conversions</td>
<td>ARFF, C4.5, Minitab, R, SAS, Statistica, Systat, CSV</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
2.1 Installation

GUIDE is available free from www.stat.wisc.edu/~loh/guide.html in the form of compiled 32- and 64-bit executables for Linux, Mac OS X, and Windows on Intel and compatible processors.

**Mac OS X and Linux:** Make the unzipped file executable by issuing this command in a Terminal application in the folder where the file is located: `chmod a+x guide`

**Mac OS X only:** The Mac OS X version requires `Xcode` and `gfortran` to be installed. To ensure that the gfortran libraries are placed in the right place, follow the steps:

2. Go to http://hpc.sourceforge.net and download file gcc-4.9-bin.tar.gz to your Downloads folder. The direct link to the file is http://prdownloads.sourceforge.net/hpc/gcc-4.9-bin.tar.gz?download
3. Open a Terminal window and type (or copy and paste):
   
   `(a) cd ~/Downloads
   (b) gunzip gcc-4.9-bin.tar.gz
   (c) sudo tar -xvf gcc-4.9-bin.tar -C /

2.2 \LaTeX

GUIDE uses the public-domain software \LaTeX (http://www.ctan.org) to produce tree diagrams. The specific locations are:

**Linux:** TeX Live http://www.tug.org/texlive/

**Mac:** MacTeX http://tug.org/mactex/

**Windows:** proTeXt http://www.tug.org/protext/

After \LaTeX is installed, a pdf file of a \LaTeX file, called `diagram.tex` say, produced by GUIDE can be obtained by typing these three commands in a terminal window:

1. `latex diagram`
2. `dvips diagram`
3. ps2pdf diagram.ps

The first command produces a file called diagram.dvi which the second command uses to create a postscript file called diagram.ps. The latter can be viewed and printed if a postscript viewer (such as Preview for the Mac) is installed. If no postscript viewer is available, the last command can be used to convert the postscript file into a pdf file, which can be viewed and printed with Adobe Reader.

The file diagram.tex can be edited to change colors, node sizes, etc. See, e.g., http://tug.org/PSTricks/main.cgi/.

3 Program operation

3.1 Required files

The GUIDE program requires two text files for input.

**Data file:** This file contains the training sample. Each file record consists of observations on the response (i.e., dependent) variable, the predictor (i.e., X or independent) variables, and optional weight and time variables. Entries in each record are comma, space, or tab delimited (multiple spaces are treated as one space, but not for commas). A record can occupy more than one line in the file, but each record must begin on a new line.

Values of categorical variables can contain any ascii character except single and double quotation marks, which are used to enclose values that contain spaces and commas. Values can be up to 60 characters long. Class labels are truncated to 10 characters in tabular displays.

**Description file:** This provides information about the name and location of the data file, names and column positions of the variables, and their roles in the analysis. This file permits different models to be fitted by changing the roles of the variables. We use the files irisdesc.txt and irisdata.txt (both obtainable from http://www.stat.wisc.edu/~loh/guide.html) to illustrate. The data give the sepal lengths and widths and the petal lengths and widths of 150 iris flowers. The response variable is the type of iris flower. The contents of irisdesc.txt are:

irisdata.txt
"?"
column, varname, vartype
1 sepallen n
2 sepalwid n
3 petallen n
4 petalwid n
5 class d

The first line of the file *irisdisc.txt* gives the name of the training sample file. If the data file *irisdata.txt* is not in the folder where GUIDE is installed, its full path (such as "c:\data\irisdata.txt") is needed. The second line gives the the missing value code, which can be up to 80 characters long. If it contains non alphanumerical characters, it must be surrounded by quotation marks. A missing value code must appear in the second line of the file even if there are no missing values in the data (in that case any character string not present among the data values can be used). The third line contains three character strings to indicate the column headers of the subsequent lines. The position, name and role of each variable comes next (in that order), with one line for each variable.

Variable names must begin with an alphabet and be not more than 60 characters long. If a name contains non-alphanumerical characters, it must be enclosed in matching single or double quotes. Spaces and the four characters #, %, {, and } are replaced by dots (periods) if they appear in a name. Variable names are truncated to 10 characters in tabular output. Leading and trailing spaces are dropped.

The following roles for the variables are permitted. Lower and upper case letters are accepted.

- **b** Categorical variable that is used both for splitting and for node modeling in regression. It is transformed to 0-1 dummy variables for node modeling. It is converted to **c** type for classification.

- **c** Categorical variable used for splitting only.

- **d** Dependent variable. Except for multi-response data (see Sec. 5.8), there can only be one such variable. In the case of relative risk models, this is the death indicator. The variable can take character string values for classification.

- **f** Numerical variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes and is disallowed in classification.
n Numerical variable used both for splitting the nodes and for fitting the node models. It is converted to type s in classification.

r Categorical treatment (Rx) variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes. If this variable is present, all n variables are automatically changed to s.

s Numerical-valued variable only used for splitting the nodes. It is not used as a regressor in the linear models. This role is suitable for ordinal categorical variables if they are given numerical values that reflect the orderings.

t Survival time (for proportional hazards models) or observation time (for longitudinal models) variable.

w Weight variable for weighted least squares regression or for excluding observations in the training sample from tree construction. See section 8.2 for the latter. Except for longitudinal models, a record with a missing value in a d, t, or z-variable is automatically assigned zero weight.

x Excluded variable. This allows models to be fitted to different subsets of the variables without reformatting the data file.

z Offset variable used only in Poisson regression.

GUIDE runs within a terminal window of the computer operating system.

Do not double-click its icon!

Linux. Any terminal program will do.

Mac OS X. The program is called Terminal; it is in the Applications Folder.

Windows. The terminal program is started from the Start button by choosing All Programs → Accessories → Command Prompt

3.2 Input file creation

GUIDE is started by typing its (lowercase) name in a terminal. The preferred way is to create an input file (option 1 below) for subsequent execution. The input file may be edited if you wish to change some input parameters later. In the following, the sign (>) is the terminal prompt (not to be typed!).

C:\Users\Weiyin> guide
GUIDE Classification and Regression Trees and Forests
Version 19.0 (build date: March 6, 2015)
Copyright (c) 1997-2015 Wei-Yin Loh. All rights reserved.
This software is based upon work supported by the U.S. Army Research Office, the National Science Foundation and the National Institutes of Health.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice:

The meanings of these options are:

0. Print the warranty disclaimer.

1. Create an input file for subsequent execution.

2. Run the program right away without creating an input file.

3. Convert the data file into a format suitable for importation into database, spreadsheet, or statistics software. See Table 2 for the statistical packages supported. Section 8.5 has an example.

4. Importance scoring of variables and identification of items with differential item functioning.

4 Classification

4.1 Default: univariate splits

We first show how to generate an input file to produce a classification tree from the data in the file irisdata.txt, using the default options. Whenever you are prompted for a selection, there is usually range of permissible values given within square brackets and a default choice (indicated by the symbol <cr>=). The default may be selected by pressing the ENTER or RETURN key. Annotations are printed in blue italics in this manual.

4.1.1 Input file creation with default options

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning
   Input your choice: 1
   Name of batch input file: irisin.txt
   This file will store your answers to the prompts.
   Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
   Press the ENTER or RETURN key to accept the default selection.
   Name of batch output file: irisout.txt
   This file will contain the results when you apply the input file to GUIDE later.
   Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
   Name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): irisdsc.txt
   Reading data description file ...
   Training sample file: irisdata.txt
   The name of the data set is read from the description file.
   Some information about the data are printed in the next few lines.
   Missing value code: ?
   Warning: N variables changed to S
   This warning is triggered because we are fitting a classification model.
   Dependent variable is class
   Length of longest data entry = 11
   Total number of cases = 150
   Number of classes = 3
   Checking data ...
   Class name       Num. cases  Proportion
   Setosa           50  0.33333333
   Versicolour      50  0.33333333
   Virginica        50  0.33333333
   Total            150  0.33333333
   #cases w/         #missing
   #cases miss. D   ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
   150  0  0  0  0  4  0  0  0
   No. cases used for training = 150
   Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
   Input 1, 2, or 3 ([1:3], <cr>=1):
   See below for examples of equal priors and specified priors.
   Choose 1 for unit misclassification costs, 2 to input costs from a file
   Input 1 or 2 ([1:2], <cr>=1):
   Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
   Choose option 2 if you do not want LaTeX code.
4.1 Default: univariate splits

Input file name to store LaTeX code (use .tex as suffix): iristree.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: irisfit.txt
This file will contain the node number and predicted class for each observation.
Input file is created!
Run GUIDE with the command: guide < irisin.txt
Press ENTER or RETURN to quit

4.1.2 Contents of input.txt

Here are the contents of the input file:

```
123321 (do not edit this file unless you know what you are doing)
  19.0 (version of GUIDE that generated this file)
  1 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"irisout.txt" (name of output file)
  1 (1=one tree, 2=ensemble)
  1 (1=classification, 2=regression)
  1 (1=simple model, 2=nearest-neighbor, 3=kernel)
  1 (0=linear first, 1=univariate first, 2=skip linear, 3=skip linear and interaction tests)
  1 (1=prune by CV, 2=by test sample, 3=no pruning)
"irisdsc.txt" (name of data description file)
  10 (number of cross-validations)
  0.500 (SE number for pruning)
  1 (1=estimated priors, 2=equal priors, 3=other priors)
  1 (1=unit misclassification costs, 2=other)
  2 (1=split point from quantiles, 2=use exhaustive search)
  1 (1=default max number of split levels, 2=specify no. in next line)
  1 (1=default min node size, 2=specify node size in next line)
  1 (1=write latex, 2=skip latex)
"iristree.tex" (latex file name)
  1 (1=vertical tree, 2=sideways tree)
  1 (1=include node numbers, 2=exclude)
  1 (1=number all nodes, 2=only terminal nodes)
  1 (1=color terminal nodes, 2=no colors)
  1 (0=#errors, 1=class sizes in nodes, 2=nothing)
  1 (1=no storage, 2=store fit and split variables, 3=store split variables and values)
  2 (1=do not save individual fitted values and node IDs, 2=save in a file)
"irisfit.txt" (file name for fitted values and node IDs)
  1 (1=do not save terminal node IDs for importance scoring in a file, 2=save them)
  1 (1=do not write R function, 2=write R function)
```

GUIDE reads only the first entry in each line; the remainder of the line is for human consumption. Because each question depends on the answers you have given to pre-
4.1 Default: univariate splits

4.1.3 Executing the program and interpreting the output

Once the input file is generated, GUIDE can be executed with the command:

```
guide < irisin.txt
```

Following is an annotated copy of the contents of the output file.

Classification tree
Pruning by cross-validation
Data description file: irisdsc.txt
Training sample file: irisdata.txt
Missing value code: ?
Warning: N variables changed to S
Dependent variable is class
Length of longest data entry = 11
Number of classes = 3

<table>
<thead>
<tr>
<th>Class name</th>
<th>Num. cases</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>50</td>
<td>0.33333333</td>
</tr>
<tr>
<td>Versicolor</td>
<td>50</td>
<td>0.33333333</td>
</tr>
<tr>
<td>Virginica</td>
<td>50</td>
<td>0.33333333</td>
</tr>
</tbody>
</table>

This gives the number of observations in each class.

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

For categorical variables, #categories include one for missing values

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Variable type</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Number of categories</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sepalen</td>
<td>s</td>
<td>4.3000E+00</td>
<td>7.9000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sepalwd</td>
<td>s</td>
<td>2.0000E+00</td>
<td>4.4000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>petallen</td>
<td>s</td>
<td>1.0000E+00</td>
<td>6.9000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>petalwd</td>
<td>s</td>
<td>1.0000E-01</td>
<td>2.5000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>class</td>
<td>d</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This shows the type and minimum and maximum values of each ordered variable.

Total #cases w/ #missing

<table>
<thead>
<tr>
<th>#cases</th>
<th>miss.</th>
<th>D</th>
<th>ord. vals</th>
<th>X-var</th>
<th>N-var</th>
<th>F-var</th>
<th>S-var</th>
<th>B-var</th>
<th>C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

No. cases used for training = 150

This shows the number of each type of variable.

Univariate split highest priority
4.1 Default: univariate splits

Interaction and linear splits 2nd and 3rd priorities
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Simple node models
Estimated priors
Unit misclassification costs
Split values for N and S variables based on exhaustive search
Max number of split levels = 10
Minimum node size = 3
Number of SE's for pruned tree = 5.0000E-01

Size and CV mean cost and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Nodes</th>
<th>Mean Cost</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Cost</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>6.000E-02</td>
<td>1.939E-02</td>
<td>6.091E-03</td>
<td>6.667E-02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>6.000E-02</td>
<td>1.939E-02</td>
<td>6.091E-03</td>
<td>6.667E-02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>3</td>
<td>4.667E-02</td>
<td>1.722E-02</td>
<td>3.333E-02</td>
<td>3.333E-02</td>
<td>3.111E-02</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3.333E-01</td>
<td>3.849E-02</td>
<td>0.000E+00</td>
<td>3.333E-01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>6.667E-01</td>
<td>3.849E-02</td>
<td>0.000E+00</td>
<td>6.667E-01</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++

* tree, ** tree, + tree, and ++ tree all the same

The tree with the smallest mean CV cost is marked with an asterisk.
The selected tree is marked with two asterisks; it is the smallest one
having mean CV cost within the specified standard error (SE) bounds.
The mean CV costs and SEs are given in the 3rd and 4th columns.
The other columns are bootstrap estimates used for experimental purposes.

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of training cases

<table>
<thead>
<tr>
<th>Node</th>
<th>Total cases</th>
<th>Train cases</th>
<th>Predicted class</th>
<th>Node cost</th>
<th>Split variable</th>
<th>Interacting variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150</td>
<td>150</td>
<td>Setosa</td>
<td>6.667E-01</td>
<td>petalwid</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>50</td>
<td>Setosa</td>
<td>0.000E+00</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>100</td>
<td>Versicolour</td>
<td>5.000E-01</td>
<td>petalwid</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>54</td>
<td>Versicolour</td>
<td>9.259E-02</td>
<td>sepalwid :petalwid</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>46</td>
<td>46</td>
<td>Virginica</td>
<td>2.174E-02</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

This shows the tree structure in tabular form. A node with label k has its left
and right child nodes are labeled 2k and 2k+1, respectively. Terminal nodes are
indicated with the symbol T. The notation "petalwid" in node 6 indicates that

Wei-Yin Loh
the variable petalwid has an interaction with the split variable sepalwid.

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5

Classification tree:
The tree structure is shown next in indented text form.
Node 1: petalwid <= 0.80000
   Node 2: Setosa
   Node 1: petalwid > 0.80000 or ?
      Node 3: petalwid <= 1.75000 or ?
         Node 6: Versicolour
         Node 3: petalwid > 1.75000 and not ?
         Node 7: Virginica

***************************************************************
Node 1: Intermediate node
A case goes into Node 2 if petalwid <= 8.000000E-01
petalwid mean = 1.1987E+00
ClassName Number ClassPrior
Setosa 50 0.3333
Versicolour 50 0.3333
Virginica 50 0.3333
Number of training cases misclassified = 100
Predicted class is Setosa

----------------------------
Node 2: Terminal node
ClassName Number ClassPrior
Setosa 50 1.0000
Versicolour 0 0.0000
Virginica 0 0.0000
Number of training cases misclassified = 0
Predicted class is Setosa

----------------------------
Node 3: Intermediate node
A case goes into Node 6 if petalwid <= 1.7500000E+00 or ?
petalwid mean = 1.6760E+00
ClassName Number ClassPrior
Setosa 0 0.0000
Versicolour 50 0.5000
Virginica 50 0.5000
Number of training cases misclassified = 50
Predicted class is Versicolour

----------------------------
Node 6: Terminal node
4.1 Default: univariate splits

<table>
<thead>
<tr>
<th>ClassName</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>Versicolou</td>
<td>49</td>
<td>0.9074</td>
</tr>
<tr>
<td>Virginica</td>
<td>5</td>
<td>0.0926</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 5
Predicted class is Versicolour

---

Node 7: Terminal node

<table>
<thead>
<tr>
<th>ClassName</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>Versicolou</td>
<td>1</td>
<td>0.0217</td>
</tr>
<tr>
<td>Virginica</td>
<td>45</td>
<td>0.9783</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 1
Predicted class is Virginica

---

Classification matrix for training sample:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>True class</th>
<th>Setosa</th>
<th>Versicolou</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setosa</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Versicolou</td>
<td>0</td>
<td>49</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Virginica</td>
<td>0</td>
<td>1</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Number of cases used for tree construction = 150
Number misclassified = 6
Resubstitution est. of mean misclassification cost = 4.0000000000000001E-002

This is the mean misclassification cost estimated from the training sample.

Observed and fitted values are stored in irisfit.txt
LaTeX code for tree is in iristree.tex
Elapsed time in seconds: 2.25169994E-02

The left side of Figure 1 shows the classification tree drawn by LaTeX using the file iristree.tex and the top lines of the file irisfit.txt are shown below. The order of the lines correspond to the order of the observations in the training sample file. The first column (labeled train) indicates whether the observation is used (“y”) or not used (“n”) to fit the model. Since we used the entire data set to fit the model here, all the entries in the first column are y.

<table>
<thead>
<tr>
<th>train</th>
<th>node</th>
<th>observed</th>
<th>predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
<tr>
<td>y</td>
<td>2</td>
<td>“Setosa”</td>
<td>“Setosa”</td>
</tr>
</tbody>
</table>
4.1 Default: univariate splits

Figure 1: GUIDE classification trees and plots of the data partitions for the iris data using estimated priors and unit misclassification costs. The tree on the left uses univariate splits. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. The symbol $\leq_*$ denotes $\leq$ or missing. The tree on the right uses linear splits on two variables at a time. Predicted classes (based on estimated misclassification cost) below terminal nodes; sample sizes for Setosa, Versicolour, and Virginica, respectively, beside nodes.
4.2 Non-default options

4.2.1 Linear splits

The above example uses the default options for classification trees. Other features are available with non-default options. We show how to obtain the linear splits shown in Figure 1 here.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: input2.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: output2.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2

Choosing 2 opens up the other options.

Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1):

Options 2 and 3 yield nearest-neighbor and kernel discriminant node models.

Input 0 for linear, interaction and univariate splits (in this order),
1 for univariate, linear and interaction splits (in this order),
2 to skip linear splits,
3 to skip linear and interaction splits:

Input your choice ([0:3], <cr>=1): 0

Option 1 is the default.

Input 1 to prune by CV, 2 by test sample, 3 for no pruning ([1:3], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): irisdsc.txt

Reading data description file ...

Training sample file: irisdata.txt

Missing value code: ?

Warning: N variables changed to S

Dependent variable is class

Length of longest data entry = 11

Total number of cases = 150

Number of classes = 3

Checking data ...
Class name  Num. cases  Proportion
Setosa 50 0.33333333
Versicolour 50 0.33333333
Virginica 50 0.33333333

Total  #cases w/  #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
150 0 0 0 4 0 0 0 0

No. cases used for training = 150
Default number of cross-validations = 10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=1):
Default max number of split levels = 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): tree2.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1): 2
Choosing 2 will give a tree with no node labels.
Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <cr>=1):
Choose 2 for very large trees.
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split variables and their values
Input your choice ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: pred.r
Input file is created!

Running the program with this input file yields the following results and the \LaTeX tree diagram and partitions on the right side of Figure 1.

Node 1: $7.0205078\times 10^{-01} \times petallen + petalwid \leq 2.4700244E+00$ or ?

Wei-Yin Loh 19  GUIDE manual
4.2 Non-default options

Node 2: Setosa
Node 1: 7.0205078E-01 * petallen + petalwid > 2.4700244E+00
Node 3: 3.2242660E-01 * petallen + petalwid <= 3.0960117E+00 or ?
Node 6: Versicolour
Node 7: Virginica

The R file `pred.r` contains this function

```r
predicted <- function(){
  if(is.na(petalwid) | is.na(petallen) | 0.70205077945722660 * petallen + petalwid <= 2.4700244096702053){
    nodeid <- 2
    predict <- "Setosa"
  } else {
    if(is.na(petalwid) | is.na(petallen) | 0.32242659679310148 * petallen + petalwid <= 3.0960116541258524){
      nodeid <- 6
      predict <- "Versicolour"
    } else {
      nodeid <- 7
      predict <- "Virginica"
    }
  }
  return(c(nodeid,predict))
}
```

4.2.2 Equal priors

If a data set has one dominant class, a classification tree will often be null after pruning, because it is hard to beat the classifier that predicts every observation to belong to the dominant class. One way to obtain a non-null tree is to specify equal priors. We illustrate this with the hepatitis data set from [http://archive.ics.uci.edu/ml/datasets/Hepatitis](http://archive.ics.uci.edu/ml/datasets/Hepatitis). The files `hepdsc.txt` and `hepdat.txt` are obtainable from [http://www.stat.wisc.edu/~loh/guide.html](http://www.stat.wisc.edu/~loh/guide.html). The data consist of observations from 155 individuals, of whom 32 are labeled “die” and 123 labeled “live”. The contents of the description file `hepdsc.txt` are:

```
hepdat.txt
"?"
column, var, type
1 CLASS d
2 AGE n
3 SEX c
4 STEROID c
5 ANTIVIRALS c
6 FATIGUE c
```

Wei-Yin Loh 20 GUIDE manual
4.2 Non-default options

Using the default estimated priors yields a null tree with no splits. To obtain a nonnull tree, choose “2” for equal priors in this dialog step:

Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1): 2

The resulting tree in text form is:

Node 1: ASCITES = "yes"
  Node 2: die
Node 1: ASCITES /= "yes"
  Node 3: SPIDERS = "?", "yes"
    Node 6: die
  Node 3: SPIDERS /= "?", "yes"
    Node 7: live

The tree drawn by \LaTeX is shown on the left of Figure 2. Nodes that predict the same class have the same color. Since the ratio of “die” to “live” classes is 32:123, the effect of equal priors is to treat one “die” observation as equivalent to \( r = 123/32 = 3.84375 \) “live” observations. Therefore a terminal node is classified as “die” if its ratio of “live” to “die” observations is less than \( r \).

4.2.3 Unequal misclassification costs: hepatitis data

So far, we have assumed that the cost of misclassifying a “die” observation as “live” is the same as the opposite. Another way to obtain a nonnull tree for the hepatitis data is to use unequal misclassification costs. For example, if we think that the cost
of misclassifying a “die” observation as “live” is four times that of the opposite, we will use the misclassification cost matrix

\[
C = \begin{pmatrix}
0 & 1 \\
4 & 0 \\
\end{pmatrix}
\]

where \(C(i, j)\) denotes the cost of classifying an observation as class \(i\) given that it belongs to class \(j\). Note that GUIDE sorts the class values in alphabetical order, so that “die” is treated as class 1 and “live” as class 2 here. This matrix is saved in the text file `cost.txt` which has these two lines:

```
0 1
4 0
```

The following lines in the input file generation step shows where this file is used:

```
Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1): 2
Input the name of a file containing the cost matrix \(C(i|j)\),
where \(C(i|j)\) is the cost of classifying class \(j\) as class \(i\)
The rows of the matrix must be in alphabetical order of the class names
Input name of file: cost.txt
```

The resulting tree is shown on the right of Figure 2.
4.2 Non-default options

4.2.4 Nearest-neighbor estimates: car data

The data file drive.txt gives the specifications and prices of 428 new cars in the 2004 model year. The data come from the J. Statistics Education website http://www.amstat.org/publications/jse/jse_data_archive.htm. Using the description file drivedsc.txt whose contents follow, the tree model for predicting Drive type misclassifies 90 of the 428 observations.

```
drive.txt
"*
c1 c2 c3
1 Region x
2 Import x
3 Make c
4 Model x
5 Type c
6 Drive d
7 SC x
8 SUV x
9 Wagon x
10 Minivan x
11 Pickup x
12 Allwheel x
13 Rearwheel x
14 Rprice n
15 Dcost n
16 Enginsz n
17 Cylin n
18 Hp n
19 City n
20 Hwy n
21 Weight n
22 Whlbase n
23 Length n
24 Width n
```

In the examples so far, the observations in each terminal node of a classification tree are all predicted to belong to the class that minimizes the node misclassification cost. This method can be inefficient if the data are difficult to classify with a small number of splits. Alternatively, we can fit a classification model to the data in each node and use it to classify individual observations in the node. GUIDE has two means to achieve this: nearest-neighbor and kernel discrimination. For nearest-neighbor, an observation in a node is classified to the plurality class among observations within its neighborhood. The neighborhood is defined to be the whole node if the split variable is categorical. We illustrate this for the car data with the following input file generation log.

Wei-Yin Loh

GUIDE manual
4.2 Non-default options

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: drive.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: drive.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1): 2
Choose nearest-neighbor method here.
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=1):
Default is univariate kernels.
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 by test sample, 3 for no pruning ([1:3], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): drivedsc.txt
Reading data description file ...
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Total number of cases = 428
Number of classes = 3
Col. no. Categorical variable #levels #missing values
  3 Make 38 0
  5 Type 6 0
Checking data ...
Class name Num. cases Proportion
4wd 94 0.21962617
fwd 224 0.52336449
rwd 110 0.25700935
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
428 0 0 10 0 0 11 0 2
No. cases used for training = 428
Default number of cross-validations = 10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):

Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):

Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):

Default max number of split levels = 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):

Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):

Input file name to store LaTeX code (use .tex as suffix): drive.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <cr>=1):

You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):

Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
Input file is created!

Results

Classification tree
Pruning by cross-validation
Data description file: drivedsc.txt
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Number of classes = 3

<table>
<thead>
<tr>
<th>Class</th>
<th>#Cases</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>94</td>
<td>0.21962617</td>
</tr>
<tr>
<td>fwd</td>
<td>224</td>
<td>0.52336449</td>
</tr>
<tr>
<td>rwd</td>
<td>110</td>
<td>0.25700935</td>
</tr>
</tbody>
</table>

Summary information (without x variables)
4.2 Non-default options

- **d**=dependent, **b**=split and fit cat variable using 0-1 dummies, **c**=split-only categorical, 
- **n**=split and fit numerical, **f**=fit-only numerical, **s**=split-only numerical, **w**=weight

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Make</td>
<td>c</td>
<td>3</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Type</td>
<td>c</td>
<td></td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Drive</td>
<td>d</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>Rprice</td>
<td>s</td>
<td>1.0280E+04</td>
<td>1.9246E+05</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>Dcost</td>
<td>s</td>
<td>9.8750E+03</td>
<td>1.7356E+05</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>Enginsz</td>
<td>s</td>
<td>1.3000E+00</td>
<td>8.3000E+00</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>Cylin</td>
<td>s</td>
<td>-1.0000E+00</td>
<td>1.2000E+01</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>Hp</td>
<td>s</td>
<td>7.3000E+01</td>
<td>5.0000E+02</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>City</td>
<td>s</td>
<td>1.0000E+01</td>
<td>6.0000E+01</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>Hwy</td>
<td>s</td>
<td>1.2000E+01</td>
<td>6.6000E+01</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>Weight</td>
<td>s</td>
<td>1.8500E+03</td>
<td>7.1900E+03</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>Whlbase</td>
<td>s</td>
<td>8.9000E+01</td>
<td>1.4400E+02</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>Length</td>
<td>s</td>
<td>1.4300E+02</td>
<td>2.2800E+02</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>Width</td>
<td>s</td>
<td>6.4000E+01</td>
<td>8.1000E+01</td>
<td>0</td>
</tr>
</tbody>
</table>

Total #cases w/ #missing D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
428 0 0 10 0 0 11 0 2

No. cases used for training: 428

Univariate split highest priority
Interaction splits 2nd priority; no linear splits
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Nearest-neighbor node models
Univariate preference
Estimated priors
Unit misclassification costs
Split values for N and S variables based on exhaustive search
Max number of split levels = 10
Minimum node size = 10
Number of SE's for pruned tree = 5.0000E-01

Size and CV mean cost and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Nodes</th>
<th>Mean Cost</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Cost</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
</tbody>
</table>
4.2 Non-default options

9  11  2.243E-01  2.016E-02  1.076E-02  2.209E-01  1.614E-02
10  10  2.243E-01  2.016E-02  1.076E-02  2.209E-01  1.614E-02
11  9   2.243E-01  2.016E-02  1.076E-02  2.209E-01  1.614E-02
12** 7  2.196E-01  2.001E-02  1.175E-02  2.118E-01  1.284E-02
13  6   2.453E-01  2.080E-02  9.596E-03  2.442E-01  1.558E-02
14  5   2.453E-01  2.080E-02  9.596E-03  2.442E-01  1.558E-02
15  3   2.547E-01  2.106E-02  1.652E-02  2.558E-01  2.811E-02
16  2   2.640E-01  2.131E-02  1.735E-02  2.674E-01  2.446E-02
17  1   3.808E-01  2.347E-02  1.384E-02  3.721E-01  1.732E-02

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of training cases

<table>
<thead>
<tr>
<th>Node</th>
<th>Total cases</th>
<th>Train cases</th>
<th>Predicted class</th>
<th>Node</th>
<th>Split variable followed by (+)fit variable(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>428</td>
<td>428</td>
<td>fwd</td>
<td></td>
<td>3.832E-01 Type +Type</td>
</tr>
<tr>
<td>2</td>
<td>73</td>
<td>73</td>
<td>fwd</td>
<td></td>
<td>3.425E-01 Type +Type</td>
</tr>
<tr>
<td>4T</td>
<td>24</td>
<td>24</td>
<td>4wd</td>
<td></td>
<td>2.083E-01 City +City</td>
</tr>
<tr>
<td>5T</td>
<td>49</td>
<td>49</td>
<td>4wd</td>
<td></td>
<td>1.020E-01 Make +Make</td>
</tr>
<tr>
<td>3</td>
<td>355</td>
<td>355</td>
<td>fwd</td>
<td></td>
<td>2.225E-01 Make +Make</td>
</tr>
<tr>
<td>6</td>
<td>105</td>
<td>105</td>
<td>fwd</td>
<td></td>
<td>2.762E-01 Make +Make</td>
</tr>
<tr>
<td>12T</td>
<td>33</td>
<td>33</td>
<td>4wd</td>
<td></td>
<td>2.121E-01 Width +Width</td>
</tr>
<tr>
<td>13T</td>
<td>72</td>
<td>72</td>
<td>fwd</td>
<td></td>
<td>1.944E-01 Type +Type</td>
</tr>
<tr>
<td>7</td>
<td>250</td>
<td>250</td>
<td>fwd</td>
<td></td>
<td>1.760E-01 Type +Type</td>
</tr>
<tr>
<td>14T</td>
<td>41</td>
<td>41</td>
<td>4wd</td>
<td></td>
<td>2.439E-01 City +City</td>
</tr>
<tr>
<td>15</td>
<td>209</td>
<td>209</td>
<td>fwd</td>
<td></td>
<td>1.053E-01 Hwy +Hwy</td>
</tr>
<tr>
<td>30T</td>
<td>32</td>
<td>32</td>
<td>fwd</td>
<td></td>
<td>2.500E-01 Length +Length</td>
</tr>
<tr>
<td>31T</td>
<td>177</td>
<td>177</td>
<td>fwd</td>
<td></td>
<td>2.825E-02 Cylin +Cylin</td>
</tr>
</tbody>
</table>

The variables preceded with a + sign are those used in the nearest neighbor models.

Number of terminal nodes of final tree: 7
Total number of nodes of final tree: 13

Classification tree:

Node 1: Type = "pickup", "sports"
Node 2: Type = "pickup"
Node 4: Mean cost = 2.08333E-01
Node 2: Type /= "pickup"
Node 5: Mean cost = 1.02041E-01
Node 1: Type /= "pickup", "sports"
Node 12: Mean cost = 2.12121E-01
Node 13: Mean cost = 1.94444E-01
Node 7: Type = "suv"
Node 14: Mean cost = 2.43902E-01
Node 7: Type /= "suv"
Node 15: Hwy <= 25.50000
Node 30: Mean cost = 2.50000E-01
Node 15: Hwy > 25.50000 or *
Node 31: Mean cost = 2.82486E-02

**************************************************************

Node 1: Intermediate node
A case goes into Node 2 if Type = "pickup", "sports"
Nearest-neighbor K = 7
Type mode = car
Fit variable
Class Number ClassPrior Type
4wd 94 0.21963
fwd 224 0.52336
rwd 110 0.25701
Number of training cases misclassified = 164
If node model is inapplicable due to missing values, predicted class = fwd
Although the number of nearest neighbors is 7 in this node, the neighborhood is the entire node because the fit variable, Type, is categorical.

----------------------
Node 2: Intermediate node
4.2 Non-default options

A case goes into Node 4 if Type = "pickup"
Nearest-neighbor K = 5
Type mode = sports

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>17</td>
<td>0.23288</td>
</tr>
<tr>
<td>fwd</td>
<td>8</td>
<td>0.10959</td>
</tr>
<tr>
<td>rwd</td>
<td>48</td>
<td>0.65753</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 25
If node model is inapplicable due to missing values, predicted class = rwd

------------------------

Node 4: Terminal node
Nearest-neighbor K = 4
City mean = 1.6458E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>12</td>
<td>0.50000</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>0</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.50000</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 5
If node model is inapplicable due to missing values, predicted class = 4wd

------------------------

Node 5: Terminal node
Nearest-neighbor K = 4
Make mode = Porsche

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>5</td>
<td>0.10204</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>8</td>
<td>0.16327</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>36</td>
<td>0.73469</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 5
If node model is inapplicable due to missing values, predicted class = rwd

------------------------

Node 3: Intermediate node
Nearest-neighbor K = 6
Make mode = Toyota

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>77</td>
<td>0.21690</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>216</td>
<td>0.60845</td>
<td></td>
</tr>
</tbody>
</table>

Wei-Yin Loh 29 GUIDE manual
4.2 Non-default options

rwd 62 0.17465
Number of training cases misclassified = 79
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------
Node 6: Intermediate node
A case goes into Node 12 if Make =
Nearest-neighbor K = 5
Make mode = Mercedes

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>45</td>
<td>0.42857</td>
</tr>
<tr>
<td>fwd</td>
<td>10</td>
<td>0.09524</td>
</tr>
<tr>
<td>rwd</td>
<td>50</td>
<td>0.47619</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 29
If node model is inapplicable due to missing values, predicted class = rwd

-----------------------------
Node 12: Terminal node
Nearest-neighbor K = 4
Width mean = 7.1091E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>26</td>
<td>0.78788</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>7</td>
<td>0.21212</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 7
If node model is inapplicable due to missing values, predicted class = 4wd

-----------------------------
Node 13: Terminal node
Nearest-neighbor K = 5
Type mode = car

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>19</td>
<td>0.26389</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>3</td>
<td>0.04167</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>50</td>
<td>0.69444</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 14
If node model is inapplicable due to missing values, predicted class = rwd

-----------------------------
Node 7: Intermediate node
A case goes into Node 14 if Type = "suv"
Nearest-neighbor K = 6
4.2 Non-default options

Type mode = car

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>32</td>
<td>0.12800</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>206</td>
<td>0.82400</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.04800</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 44
If node model is inapplicable due to missing values, predicted class = fwd

Node 14: Terminal node
Nearest-neighbor K = 4
City mean = 1.6707E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>22</td>
<td>0.53659</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>19</td>
<td>0.46341</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 10
If node model is inapplicable due to missing values, predicted class = 4wd

Node 15: Intermediate node
A case goes into Node 30 if Hwy <= 2.5500000E+01
Nearest-neighbor K = 6
Hwy mean = 3.0148E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Hwy</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>10</td>
<td>0.04785</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>187</td>
<td>0.89474</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.05742</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 22
If node model is inapplicable due to missing values, predicted class = fwd

Node 30: Terminal node
Nearest-neighbor K = 4
Length mean = 1.9938E+02

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>6</td>
<td>0.18750</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>15</td>
<td>0.46875</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>11</td>
<td>0.34375</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 8
If node model is inapplicable due to missing values, predicted class = fwd
4.2 Non-default options 4 CLASSIFICATION

-----------------------------
Node 31: Terminal node
Nearest-neighbor K = 6
Cylin mean = 4.8192E+00

Fit variable

Class Number ClassPrior Cylin
4wd 4 0.02260
fwd 172 0.97175
rwd 1 0.00565

Number of training cases misclassified = 5
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------

Classification matrix for training sample:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>True class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4wd</td>
</tr>
<tr>
<td>4wd</td>
<td>67</td>
</tr>
<tr>
<td>fwd</td>
<td>12</td>
</tr>
<tr>
<td>rwd</td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td>94</td>
</tr>
</tbody>
</table>

Number of cases used for tree construction = 428
Number misclassified = 54
Resubstitution est. of mean misclassification cost = 0.12616822429906541

LaTeX code for tree is in drivenn.tex

Figure 3 shows the tree model, which misclassifies 54 observations. Figure 4 shows the observed and predicted values of Drive in node 30 of the tree.

4.2.5 Kernel density estimates: car data

An alternative to nearest-neighbor models is kernel discrimination models, where classification is based on maximum likelihood with class densities estimated by the kernel method. Unlike nearest-neighbor, however, this option also yields an estimated class probability vector for each observation. Therefore it can serve as a nonparametric alternative to multinomial logistic regression. Empirical evidence indicates that the nearest-neighbor and kernel methods possess similar prediction accuracy. See Loh (2009) for more details. Following is a log of the input file generation step for the kernel method.

0. Read the warranty disclaimer
1. Create an input file for batch run
4.2 Non-default options

Figure 3: GUIDE 0.50-SE classification tree for predicting `drive` with univariate nearest-neighbor node models, estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set \( S_1 = \{\text{pickup, sports}\} \). Set \( S_2 = \{\text{Audi, BMW, Hummer, Infiniti, Isuzu, Jaguar, Jeep, Land-Rover, Lexus, Lincoln, Mercedes, Porsche, Subaru}\} \). Set \( S_3 = \{\text{Audi, Hummer, Isuzu, Jeep, Land-Rover, Porsche, Subaru}\} \). Predicted classes (based on estimated misclassification cost) printed below terminal nodes; sample sizes for 4wd, fwd, and rwd, respectively, beside nodes.
Figure 4: Observed and predicted values of drive type in node 30 of tree in Figure 3
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: driveker.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: driveker.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1): 3

This is where you choose kernel density estimation.
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=1):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 by test sample, 3 for no pruning ([1:3], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): drivedsc.txt
Reading data description file ...
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Total number of cases = 428
Number of classes = 3
Col. no. Categorical variable #levels #missing values
  3 Make 38 0
  5 Type 6 0
Checking data ...
Class name Num. cases Proportion
  4wd  94 0.21962617
  fwd 224 0.52336449
  rwd 110 0.25700935
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
 428 0 0 10 0 0 11 0 2
No. cases used for training = 428
Default number of cross-validations = 10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file

Wei-Yin Loh 35 GUIDE manual
4.2 Non-default options  

Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max number of split levels = 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): driveker.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <cr>=1):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input name of file to store node IDs and fitted values: driveker.fit

This file contains the predicted class and terminal node label for each observation.
Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
Input name of file to store predicted class and probability: driveker.pro

This file contains the estimated class probabilities for each observation.
Input file is created!
Run GUIDE with the command: guide < driveker.in

The results in the output file are given next.

Classification tree
Pruning by cross-validation
Data description file: drivedsc.txt
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Number of classes = 3
Class  #Cases  Proportion
4wd  94  0.21962617
fwd  224  0.52336449
rwd  110  0.25700935

Wei-Yin Loh 36  GUIDE manual
Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Make</td>
<td>c</td>
<td></td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Type</td>
<td>c</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Drive</td>
<td>d</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Rprice</td>
<td>s</td>
<td>1.0280E+04</td>
<td>1.9246E+05</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Dcost</td>
<td>s</td>
<td>9.8750E+03</td>
<td>1.7356E+05</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Enginsz</td>
<td>s</td>
<td>1.3000E+00</td>
<td>8.3000E+00</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Cylin</td>
<td>s</td>
<td>-1.0000E+00</td>
<td>1.2000E+01</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Hps</td>
<td>s</td>
<td>7.3000E+01</td>
<td>5.0000E+02</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>City</td>
<td>s</td>
<td>1.0000E+01</td>
<td>6.0000E+01</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Hwy</td>
<td>s</td>
<td>1.2000E+01</td>
<td>6.6000E+01</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Weight</td>
<td>s</td>
<td>1.8500E+03</td>
<td>7.1900E+03</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Whlbase</td>
<td>s</td>
<td>8.9000E+01</td>
<td>1.4400E+02</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Length</td>
<td>s</td>
<td>1.4300E+02</td>
<td>2.2800E+02</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Width</td>
<td>s</td>
<td>6.4000E+01</td>
<td>8.1000E+01</td>
<td></td>
</tr>
</tbody>
</table>

Total #cases w/ #missing: 428 0 0 10 0 0 11 0 2

No. cases used for training: 428

Univariate split highest priority
Interaction splits 2nd priority; no linear splits
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Kernel density node models
Univariate preference
Estimated priors
Unit misclassification costs
Split values for N and S variables based on exhaustive search
Max number of split levels = 10
Minimum node size = 10
Number of SE's for pruned tree = 5.0000E-01

Size and CV mean cost and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Nodes</th>
<th>Mean Cost</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Cost</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>2.243E-01</td>
<td>2.016E-02</td>
<td>1.076E-02</td>
<td>2.209E-01</td>
<td>1.614E-02</td>
</tr>
</tbody>
</table>
4.2 Non-default options

<table>
<thead>
<tr>
<th>Node</th>
<th>Total cases</th>
<th>Train cases</th>
<th>Predicted class</th>
<th>Node</th>
<th>Split variable followed by (+) fit variable(s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>428</td>
<td>428</td>
<td>fwd</td>
<td>2</td>
<td>3.832E-01 Type +Type</td>
<td>1.762E-01 Type</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>21</td>
<td>fwd</td>
<td>3</td>
<td>2.762E-01 Type +Make</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>3</td>
<td>120</td>
<td>120</td>
<td>fwd</td>
<td>5</td>
<td>1.069E-01 City +City</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>48</td>
<td>fwd</td>
<td>7</td>
<td>1.760E-01 City +Type</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>5</td>
<td>209</td>
<td>209</td>
<td>fwd</td>
<td>14</td>
<td>3.902E-01 City +City</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>6</td>
<td>776</td>
<td>776</td>
<td>fwd</td>
<td>28</td>
<td>3.125E-01 -</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>7</td>
<td>776</td>
<td>776</td>
<td>fwd</td>
<td>29</td>
<td>2.800E-01 Whlbase +Whlbase</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>8</td>
<td>776</td>
<td>776</td>
<td>fwd</td>
<td>30</td>
<td>3.125E-01 Length +Length</td>
<td>1.944E-01 Type</td>
</tr>
<tr>
<td>9</td>
<td>776</td>
<td>776</td>
<td>fwd</td>
<td>31</td>
<td>2.825E-02 Cylin +Cylin</td>
<td>1.944E-01 Type</td>
</tr>
</tbody>
</table>

In the above, "split variable" refers to the variable selected to split the node and "fit variable(s)" refers to the one(s) used to estimate the class kernel densities. Fit variables are indicated with a preceding + sign. In this example,
the split and fit variables are the same in every node. If a categorical variable (e.g., Type) is selected for fitting, discrete kernel density estimates are used. A dash (-) indicates that a node is not split, usually due to sample size being too small, in which case all the observations in the node are predicted as belonging to the class that minimizes the misclassification cost.

Number of terminal nodes of final tree: 8
Total number of nodes of final tree: 15

Classification tree:

Node 1: Type = "pickup", "sports"
   Node 2: Type = "pickup"
      Node 4: Mean cost = 3.75000E-01
      Node 5: Mean cost = 1.02041E-01
      Node 1: Type /= "pickup", "sports"
      Node 12: Mean cost = 2.12121E-01
      Node 13: Mean cost = 1.94444E-01
      Node 7: Type = "suv"
      Node 14: City <= 15.50000
         Node 28: Mean cost = 3.12500E-01
         Node 14: City > 15.50000 or *
            Node 29: Mean cost = 2.80000E-01
      Node 7: Type /= "suv"
      Node 15: Hwy <= 25.50000
         Node 30: Mean cost = 3.12500E-01
         Node 15: Hwy > 25.50000 or *
            Node 31: Mean cost = 2.82486E-02

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if Type = "pickup", "sports"
4.2 Non-default options

Type mode = car

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>94</td>
<td>0.21963</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>224</td>
<td>0.52336</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>110</td>
<td>0.25701</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 164
If node model is inapplicable due to missing values, predicted class = fwd

*Categorical variables, such as Type, do not have bandwidths. Their kernel density estimates are the sample cell frequencies.*

-----------------------------

Node 2: Intermediate node
A case goes into Node 4 if Type = "pickup"

Type mode = sports

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>17</td>
<td>0.23288</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>8</td>
<td>0.10959</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>48</td>
<td>0.65753</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 25
If node model is inapplicable due to missing values, predicted class = rwd

-----------------------------

Node 4: Terminal node

City mean = 1.6458E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>12</td>
<td>0.50000</td>
<td>3.3823E+00</td>
</tr>
<tr>
<td>fwd</td>
<td>0</td>
<td>0.00000</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.50000</td>
<td>5.1881E+00</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 9
If node model is inapplicable due to missing values, predicted class = 4wd

*The numbers in the last column give the kernel density bandwidth for each class.*

-----------------------------

Node 5: Terminal node

Make mode = Porsche

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>5</td>
<td>0.10204</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>8</td>
<td>0.16327</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>36</td>
<td>0.73469</td>
<td></td>
</tr>
</tbody>
</table>

Wei-Yin Loh 40 GUIDE manual
Number of training cases misclassified = 5
If node model is inapplicable due to missing values, predicted class = rwd

----------------------------
Node 3: Intermediate node
A case goes into Node 6 if Make =
"Land-Rover", "Lexus", "Lincoln", "Mercedes", "Porsche", "Subaru"
Make mode = Toyota

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>77</td>
<td>0.21690</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>216</td>
<td>0.60845</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>62</td>
<td>0.17465</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 79
If node model is inapplicable due to missing values, predicted class = fwd

----------------------------
Node 6: Intermediate node
A case goes into Node 12 if Make =
Make mode = Mercedes

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Make</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>45</td>
<td>0.42857</td>
<td></td>
</tr>
<tr>
<td>fwd</td>
<td>10</td>
<td>0.09524</td>
<td></td>
</tr>
<tr>
<td>rwd</td>
<td>50</td>
<td>0.47619</td>
<td></td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 29
If node model is inapplicable due to missing values, predicted class = rwd

----------------------------
Node 12: Terminal node
Width mean = 7.1091E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>26</td>
<td>0.788788</td>
<td>3.3807E+00</td>
</tr>
<tr>
<td>fwd</td>
<td>7</td>
<td>0.21212</td>
<td>1.2558E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 7
If node model is inapplicable due to missing values, predicted class = 4wd

----------------------------
Node 13: Terminal node
Type mode = car

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2 Non-default options

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>19</td>
<td>0.26389</td>
</tr>
<tr>
<td>fwd</td>
<td>3</td>
<td>0.04167</td>
</tr>
<tr>
<td>rwd</td>
<td>50</td>
<td>0.69444</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 14
If node model is inapplicable due to missing values, predicted class = rwd

----------------------------

Node 7: Intermediate node
A case goes into Node 14 if Type = "suv"
Type mode = car

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>32</td>
<td>0.12800</td>
</tr>
<tr>
<td>fwd</td>
<td>206</td>
<td>0.82400</td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.04800</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 44
If node model is inapplicable due to missing values, predicted class = fwd

----------------------------

Node 14: Intermediate node
A case goes into Node 28 if City <= 1.5500000E+01
City mean = 1.6707E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>22</td>
<td>0.53659</td>
<td>3.9949E+00</td>
</tr>
<tr>
<td>fwd</td>
<td>19</td>
<td>0.46341</td>
<td>3.0269E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 16
If node model is inapplicable due to missing values, predicted class = 4wd

----------------------------

Node 28: Terminal node

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>11</td>
<td>0.68750</td>
</tr>
<tr>
<td>fwd</td>
<td>5</td>
<td>0.31250</td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 5
Predicted class is 4wd

----------------------------

Node 29: Terminal node
Whlbase mean = 1.0644E+02

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Whlbase</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>11</td>
<td>0.44000</td>
<td>5.6690E+00</td>
</tr>
<tr>
<td>fwd</td>
<td>14</td>
<td>0.56000</td>
<td>7.9672E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>0</td>
<td>0.00000</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Wei-Yin Loh 42 GUIDE manual
4.2 Non-default options

Number of training cases misclassified = 7
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------
Node 15: Intermediate node
A case goes into Node 30 if Hwy $\leq 2.550000E+01$
Hwy mean = 3.0148E+01

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Hwy</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>10</td>
<td>0.04785</td>
<td>5.8465E+00</td>
</tr>
<tr>
<td>fwd</td>
<td>187</td>
<td>0.89474</td>
<td>3.9058E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>12</td>
<td>0.05742</td>
<td>5.6846E+00</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 22
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------
Node 30: Terminal node
Length mean = 1.9938E+02

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>6</td>
<td>0.18750</td>
<td>1.1008E+01</td>
</tr>
<tr>
<td>fwd</td>
<td>15</td>
<td>0.46875</td>
<td>7.0085E+00</td>
</tr>
<tr>
<td>rwd</td>
<td>11</td>
<td>0.34375</td>
<td>1.6353E+01</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 10
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------
Node 31: Terminal node
Cylin mean = 4.8192E+00

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>ClassPrior</th>
<th>Cylin</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>4</td>
<td>0.02260</td>
<td>9.4732E-01</td>
</tr>
<tr>
<td>fwd</td>
<td>172</td>
<td>0.97175</td>
<td>9.4317E-01</td>
</tr>
<tr>
<td>rwd</td>
<td>1</td>
<td>0.00565</td>
<td>1.9453E-01</td>
</tr>
</tbody>
</table>

Number of training cases misclassified = 5
If node model is inapplicable due to missing values, predicted class = fwd

-----------------------------

Classification matrix for training sample:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>4wd</th>
<th>fwd</th>
<th>rwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>4wd</td>
<td>68</td>
<td>19</td>
<td>8</td>
</tr>
<tr>
<td>fwd</td>
<td>13</td>
<td>201</td>
<td>5</td>
</tr>
<tr>
<td>rwd</td>
<td>13</td>
<td>4</td>
<td>97</td>
</tr>
</tbody>
</table>

Wei-Yin Loh

43

GUIDE manual
4.2 Non-default options

Figure 5: GUIDE 0.50-SE classification tree for predicting Drive with univariate kernel discrimination node models, estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set \( S_1 = \{ \text{pickup, sports} \} \). Set \( S_2 = \{ \text{Audi, BMW, Hummer, Infiniti, Isuzu, Jaguar, Jeep, Land-Rover, Lexus, Lincoln, Mercedes, Porsche, Subaru} \} \). Set \( S_3 = \{ \text{Audi, Hummer, Isuzu, Jeep, Land-Rover, Porsche, Subaru} \} \). Predicted classes (based on estimated misclassification cost) printed below terminal nodes; sample sizes for 4wd, fwd, and rwd, respectively, beside nodes.

Total 94 224 110

Number of cases used for tree construction = 428
Number misclassified = 62
Resubstitution est. of mean misclassification cost = 0.14485981308411214

Predicted class probability estimates are stored in driveker.pro
Observed and fitted values are stored in driveker.fit
LaTeX code for tree is in driveker.tex
Elapsed time in seconds: 0.266952991

Figure 5 shows the LaTeX tree diagram. The top several lines of the file driveker.fit, which contains the terminal node label and predicted class for each observation in the training sample, are:

Wei-Yin Loh 44  GUIDE manual
5 Regression

5.1 Stepwise least-squares

We use the baseball dataset bbdat.txt to show the results for regression trees when there are no missing values. The data give the log-salary and performance measures of 263 professional baseball players (Hoaglin and Velleman, 1995). The response variable is the logarithm of salary (Logsalary). The data description file bbdsc.txt consists of the following lines:

```
bbdat.txt
NA
column, varname, vartype
1 Id x
2 Name x
3 Bat86 n
4 Hit86 n
5 Hr86 n
6 Run86 n
7 Rb86 n
8 Wlk86 n
```
Notice that there are four variables having the "b" variable type. This means that 0-1 dummy variables will be created for them in fitting the node linear models. The following shows how the input file is created.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: stepin.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: stepout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): bbdsc.txt
5.1 Stepwise least-squares

Reading data description file ...
Training sample file: bbdat.txt
Missing value code: NA
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases = 263

<table>
<thead>
<tr>
<th>Col. no.</th>
<th>Categorical variable</th>
<th>#levels</th>
<th>#missing values</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Leag86</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>Div86</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>Team86</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>Pos86</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>Leag87</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>Team87</td>
<td>24</td>
<td>0</td>
</tr>
</tbody>
</table>

Checking data ...
The program will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...

Number of dummy variables created: 25

<table>
<thead>
<tr>
<th>Total #cases w/ #missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#cases miss. D ord. vals</td>
</tr>
<tr>
<td>263</td>
</tr>
</tbody>
</table>

No weight variable in data file
No. cases used for training = 263

Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): step.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: step.fit
Input file is created!

The contents from the file stepout.txt follow. They show a tree with two terminal nodes and give the regression coefficients, sample means of the dependent and predictor variables, MSE and $R^2$ values, and names of the split variables in each node.

Least squares regression tree
Predictions truncated at global min and max of D sample values
The predicted values are truncated at the minimum and maximum values of the training sample by default.

Pruning by cross-validation
Data description file: bbdsc.txt
Training sample file: bbdat.txt
Missing value code: NA
Dependent variable is Logsalary
Piecewise forward and backward stepwise regression
F-to-enter and F-to-delete = 4.0000000000000000 3.9900000000000002
These default F values are the same as those used in SAS.
5.1 Stepwise least-squares

Using as many variables as needed
Length of longest data entry = 17
Number of dummy variables created = 25

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
For categorical variables, #categories include one for missing values

<table>
<thead>
<tr>
<th>Column number</th>
<th>Variable name</th>
<th>Variable type</th>
<th>Minimum value</th>
<th>Maximum value</th>
<th>Number of categories</th>
<th>Number of missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Bat86</td>
<td>n</td>
<td>1.2700E+02</td>
<td>6.8700E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Hit86</td>
<td>n</td>
<td>3.2000E+01</td>
<td>2.3800E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Hr86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>4.0000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Run86</td>
<td>n</td>
<td>1.3000E+01</td>
<td>1.3000E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Rb86</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.2100E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Wlk86</td>
<td>n</td>
<td>3.0000E+00</td>
<td>1.0500E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Yrs</td>
<td>n</td>
<td>1.0000E+00</td>
<td>2.4000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Batcr</td>
<td>n</td>
<td>1.8100E+02</td>
<td>1.4053E+04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Hitcr</td>
<td>n</td>
<td>4.2000E+01</td>
<td>4.2560E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Hrcr</td>
<td>n</td>
<td>0.0000E+00</td>
<td>5.4800E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Runcr</td>
<td>n</td>
<td>1.8000E+01</td>
<td>2.1650E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Rbcr</td>
<td>n</td>
<td>9.0000E+00</td>
<td>1.6590E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Wlkcr</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.5660E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Leag86</td>
<td>b</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Div86</td>
<td>b</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Team86</td>
<td>c</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Pos86</td>
<td>b</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Put86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>1.3770E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Aas86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>4.9200E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Err86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>3.2000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Leag87</td>
<td>b</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Team87</td>
<td>c</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Logsalary</td>
<td>d</td>
<td>4.2121E+00</td>
<td>7.8079E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The F variables below this line are dummy variables constructed from the B variables.

<table>
<thead>
<tr>
<th>Column number</th>
<th>Variable name</th>
<th>Variable type</th>
<th>Minimum value</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Leag86.N</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>28</td>
<td>Div86.W</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>29</td>
<td>Pos86.10</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>30</td>
<td>Pos86.23</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>31</td>
<td>Pos86.2B</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>32</td>
<td>Pos86.2S</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>33</td>
<td>Pos86.32</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>34</td>
<td>Pos86.3B</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>35</td>
<td>Pos86.3O</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>36</td>
<td>Pos86.3S</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>37</td>
<td>Pos86.C</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
</tbody>
</table>
5.1 Stepwise least-squares

<table>
<thead>
<tr>
<th></th>
<th>Pos86.CD</th>
<th>Pos86.CF</th>
<th>Pos86.DH</th>
<th>Pos86.DO</th>
<th>Pos86.LF</th>
<th>Pos86.01</th>
<th>Pos86.DO</th>
<th>Pos86.DD</th>
<th>Pos86.DF</th>
<th>Pos86.DS</th>
<th>Pos86.RF</th>
<th>Pos86.S3</th>
<th>Pos86.SS</th>
<th>Pos86.UT</th>
<th>Leag8.N</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Total #cases w/ #missing

<table>
<thead>
<tr>
<th>#cases</th>
<th>miss.</th>
<th>D ord.</th>
<th>vals</th>
<th>#X-var</th>
<th>#N-var</th>
<th>#F-var</th>
<th>#S-var</th>
<th>#B-var</th>
<th>#C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>263</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

No weight variable in data file
No. cases used for training = 263

Missing N and F values imputed with node means

*The default method of handling missing values is node-mean imputation.*

Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000
Max number of split levels = 10
Minimum node size = 13
Number of SE's for pruned tree = 5.0000E-01

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean MSE</th>
<th>SE(Mean)</th>
<th>BSE(Means)</th>
<th>Median MSE</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>2.001E-01</td>
<td>2.068E-02</td>
<td>2.408E-02</td>
<td>1.866E-01</td>
<td>2.982E-02</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>2.001E-01</td>
<td>2.068E-02</td>
<td>2.408E-02</td>
<td>1.866E-01</td>
<td>2.982E-02</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>2.001E-01</td>
<td>2.068E-02</td>
<td>2.408E-02</td>
<td>1.866E-01</td>
<td>2.982E-02</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>1.768E-01</td>
<td>1.920E-02</td>
<td>2.206E-02</td>
<td>1.440E-01</td>
<td>3.984E-02</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1.744E-01</td>
<td>1.935E-02</td>
<td>2.221E-02</td>
<td>1.440E-01</td>
<td>4.151E-02</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1.790E-01</td>
<td>1.937E-02</td>
<td>2.107E-02</td>
<td>1.486E-01</td>
<td>3.760E-02</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1.824E-01</td>
<td>1.979E-02</td>
<td>2.121E-02</td>
<td>1.492E-01</td>
<td>3.492E-02</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1.574E-01</td>
<td>1.882E-02</td>
<td>1.869E-02</td>
<td>1.309E-01</td>
<td>3.241E-02</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1.518E-01</td>
<td>1.765E-02</td>
<td>1.752E-02</td>
<td>1.296E-01</td>
<td>3.044E-02</td>
</tr>
<tr>
<td>10**</td>
<td>2</td>
<td>1.208E-01</td>
<td>1.439E-02</td>
<td>1.378E-02</td>
<td>1.166E-01</td>
<td>1.884E-02</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>3.469E-01</td>
<td>2.575E-02</td>
<td>2.224E-02</td>
<td>3.456E-01</td>
<td>3.877E-02</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
5.1 Stepwise least-squares

O-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of Logsalary in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node

<table>
<thead>
<tr>
<th>Node label</th>
<th>Total cases</th>
<th>Cases fit</th>
<th>Rank</th>
<th>D-mean</th>
<th>MSE</th>
<th>R^2</th>
<th>variable</th>
<th>variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>263</td>
<td>263</td>
<td>9</td>
<td>5.945E+00</td>
<td>2.907E-01</td>
<td>0.6391</td>
<td>Yrs</td>
<td></td>
</tr>
<tr>
<td>2T</td>
<td>143</td>
<td>143</td>
<td>7</td>
<td>5.506E+00</td>
<td>8.336E-02</td>
<td>0.8907</td>
<td>Yrs</td>
<td></td>
</tr>
<tr>
<td>3T</td>
<td>120</td>
<td>120</td>
<td>6</td>
<td>6.469E+00</td>
<td>1.258E-01</td>
<td>0.6456</td>
<td>Bat86</td>
<td></td>
</tr>
</tbody>
</table>

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3

Regression tree:

Node 1: Yrs <= 6.00000 or NA
   Node 2: Logsalary-mean = 5.50632
Node 1: Yrs > 6.00000 and not NA
   Node 3: Logsalary-mean = 6.46866

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if Yrs <= 6.0000000E+00 or NA
Yrs mean = 7.3802E+00

Coefficients of least squares regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>4.1321E+00</td>
<td>28.07</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bat86</td>
<td>-2.4528E-03</td>
<td>-2.73</td>
<td>0.0068</td>
<td>1.2700E+02</td>
<td>4.0829E+02</td>
<td>6.8700E+02</td>
</tr>
<tr>
<td>Hit86</td>
<td>1.4558E-02</td>
<td>5.16</td>
<td>0.0000</td>
<td>3.2000E+01</td>
<td>1.0916E+02</td>
<td>2.3800E+02</td>
</tr>
<tr>
<td>Wlk86</td>
<td>1.0020E-02</td>
<td>3.82</td>
<td>0.0002</td>
<td>3.0000E+00</td>
<td>4.1722E+01</td>
<td>1.0500E+02</td>
</tr>
<tr>
<td>Yrs</td>
<td>7.0500E-02</td>
<td>4.24</td>
<td>0.0000</td>
<td>1.0000E+00</td>
<td>7.3802E+00</td>
<td>2.4000E+01</td>
</tr>
<tr>
<td>Runcr</td>
<td>1.1939E-03</td>
<td>3.11</td>
<td>0.0021</td>
<td>1.8000E+01</td>
<td>3.6808E+02</td>
<td>2.1650E+03</td>
</tr>
<tr>
<td>Wlkcr</td>
<td>-9.6467E-04</td>
<td>-2.20</td>
<td>0.0291</td>
<td>8.0000E+00</td>
<td>2.6655E+02</td>
<td>1.5660E+03</td>
</tr>
<tr>
<td>Leag8.N</td>
<td>1.4081E-01</td>
<td>2.09</td>
<td>0.0373</td>
<td>0.0000E+00</td>
<td>4.7148E+01</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>Pos86.C</td>
<td>3.3729E-01</td>
<td>3.05</td>
<td>0.0025</td>
<td>0.0000E+00</td>
<td>1.1407E+01</td>
<td>1.0000E+00</td>
</tr>
</tbody>
</table>

Wei-Yin Loh  50  GUIDE manual
5.2 Least-squares simple polynomial

Often it is useful to be able to visualize the fitted regression function and the data simultaneously. This can be accomplished by fitting a piecewise simple linear model, where the best single regressor is selected to fit a straight line in each node, as follows.
5.2 Least-squares simple polynomial 5 REGRESSION

Figure 6: GUIDE piecewise linear least-squares regression tree with stepwise variable selection for predicting Logsalary. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘≤∗’ stands for ‘≤ or missing’. Sample sizes and means of Logsalary printed below nodes.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: linin.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: linout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0): 2

The default degree of the polynomial is 1.
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): bbdsc.txt
Reading data description file ...
Training sample file: bbdat.txt
Missing value code: NA
Warning: B variables changed to C
This warning is triggered because the description file contains some variables with the B designation, which is not allowed in piecewise polynomial regression.
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases = 263

Col. no. Categorical variable  #levels  #missing values
  16 Leag86                 2       0
  17 Div86                 2       0
  18 Team86               24       0
  19 Pos86                23       0
  24 Leag87                2       0
  25 Team87               24       0

Checking data ...

Total #cases w/ #missing
#cases      miss. D ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
263          0 0 3 16 0 0 0 6

No weight variable in data file
No. cases used for training = 263

Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
A file by that name already exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: lin.fit
Input file is created!

Partial contents of linout.txt

Size and CV MSE and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean MSE</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median MSE</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>1.816E-01</td>
<td>2.760E-02</td>
<td>2.322E-02</td>
<td>1.594E-01</td>
<td>2.882E-02</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>1.816E-01</td>
<td>2.760E-02</td>
<td>2.322E-02</td>
<td>1.594E-01</td>
<td>2.882E-02</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>1.809E-01</td>
<td>2.761E-02</td>
<td>2.336E-02</td>
<td>1.594E-01</td>
<td>2.990E-02</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>1.806E-01</td>
<td>2.763E-02</td>
<td>2.347E-02</td>
<td>1.599E-01</td>
<td>3.034E-02</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>1.787E-01</td>
<td>2.757E-02</td>
<td>2.241E-02</td>
<td>1.599E-01</td>
<td>3.108E-02</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1.790E-01</td>
<td>2.757E-02</td>
<td>2.256E-02</td>
<td>1.599E-01</td>
<td>3.108E-02</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1.803E-01</td>
<td>2.754E-02</td>
<td>2.351E-02</td>
<td>1.811E-01</td>
<td>3.253E-02</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>1.709E-01</td>
<td>2.654E-02</td>
<td>2.325E-02</td>
<td>1.610E-01</td>
<td>2.358E-02</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1.782E-01</td>
<td>2.683E-02</td>
<td>2.331E-02</td>
<td>1.651E-01</td>
<td>2.792E-02</td>
</tr>
<tr>
<td>10+</td>
<td>4</td>
<td>1.729E-01</td>
<td>2.115E-02</td>
<td>2.233E-02</td>
<td>1.575E-01</td>
<td>2.575E-02</td>
</tr>
<tr>
<td>11**</td>
<td>3</td>
<td>1.677E-01</td>
<td>2.150E-02</td>
<td>2.285E-02</td>
<td>1.643E-01</td>
<td>3.291E-02</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1.887E-01</td>
<td>2.313E-02</td>
<td>2.491E-02</td>
<td>1.733E-01</td>
<td>4.050E-02</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4.436E-01</td>
<td>3.247E-02</td>
<td>3.208E-02</td>
<td>4.574E-01</td>
<td>4.953E-02</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
* tree same as ** tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of Logsalary in the node
Cases fit give the number of cases used to fit node
MSE and R\(^2\) are based on all cases in node

<table>
<thead>
<tr>
<th>Node</th>
<th>Total cases</th>
<th>Cases fit</th>
<th>Rank</th>
<th>D-mean</th>
<th>MSE</th>
<th>R(^2)</th>
<th>Variable</th>
<th>Other variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>263</td>
<td>263</td>
<td>2</td>
<td>5.945E+00</td>
<td>4.502E-01</td>
<td>0.4257</td>
<td>Yrs +Hitcr</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>143</td>
<td>143</td>
<td>2</td>
<td>5.506E+00</td>
<td>1.284E-01</td>
<td>0.8254</td>
<td>Hitcr +Batcr</td>
<td></td>
</tr>
<tr>
<td>4T</td>
<td>110</td>
<td>110</td>
<td>2</td>
<td>5.146E+00</td>
<td>9.210E-02</td>
<td>0.7371</td>
<td>Wlkcr +Hitcr</td>
<td></td>
</tr>
<tr>
<td>5T</td>
<td>33</td>
<td>33</td>
<td>2</td>
<td>6.706E+00</td>
<td>7.500E-02</td>
<td>0.4395</td>
<td>Wlk86 +Rbcr</td>
<td></td>
</tr>
<tr>
<td>3T</td>
<td>120</td>
<td>120</td>
<td>2</td>
<td>6.469E+00</td>
<td>1.943E-01</td>
<td>0.4331</td>
<td>Wlkcr +Hit86</td>
<td></td>
</tr>
</tbody>
</table>

The last column, labeled ‘Fit variables’, give the regressor variable names and the signs of their regression coefficients.

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5

Regression tree:

Node 1: Yrs <= 6.00000 or NA
   Node 2: Hitcr <= 4.59500E+02 or NA
       Node 4: Logsalary-mean = 5.14642
   Node 2: Hitcr > 4.59500E+02 and not NA
       Node 5: Logsalary-mean = 6.70595
   Node 1: Yrs > 6.00000 and not NA
       Node 3: Logsalary-mean = 6.46866

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if Yrs <= 6.0000000E+00 or NA
Yrs mean = 7.3802E+00

Coefficients of least squares regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>5.2933E+00</td>
<td>84.65</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hitcr</td>
<td>8.8852E-04</td>
<td>13.91</td>
<td>0.0000</td>
<td>4.2000E+01</td>
<td>7.3392E+02</td>
<td>4.2560E+03</td>
</tr>
</tbody>
</table>

Wei-Yin Loh

GUIDE manual
5.3 ANCOVA models

Besides, multiple linear, stepwise linear, and best simple polynomial regression, GUIDE can also fit a best ANCOVA model in each node. The ANCOVA model
5.3 ANCOVA models

![GUIDE piecewise simple linear least-squares regression tree for predicting Logsalary. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘≤∗’ stands for ‘≤ or missing’. Sample sizes, means of Logsalary, and signs and names of regressor variable printed below nodes. Terminal nodes with negative, zero, and positive slopes are colored red, yellow, and green, respectively.

uses stepwise regression to find the best single linear regressor and the best subset of dummy (indicator) variables constructed from any B variables.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: ancova.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: ancova.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
   1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
   5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
   0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
   4: stepwise simple ANCOVA ([0:4], <cr>=0): 4
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Wei-Yin Loh 56 GUIDE manual
Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): bbdsc.txt
Reading data description file ...
Training sample file: bbdat.txt
Missing value code: NA
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases = 263

<table>
<thead>
<tr>
<th>Col. no.</th>
<th>Categorical variable</th>
<th>#levels</th>
<th>#missing values</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Leag86</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>Div86</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>Team86</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>Pos86</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>Leag87</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>Team87</td>
<td>24</td>
<td>0</td>
</tr>
</tbody>
</table>

Checking data ...
The program will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 25

| Total #cases w/ #missing
| #cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var |
|---------------------------|-----------------|-----|------|-----|-----|-----|-----|-----|
| 263                       | 0               | 0   | 3    | 16  | 0   | 0   | 4   | 2   |

No weight variable in data file
No. cases used for training = 263
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): ancova.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: ancova.fit
Input file is created!

Results

Least squares regression tree
Predictions truncated at global min and max of D sample values
Pruning by cross-validation
Data description file: bbdsc.txt
Training sample file: bbdat.txt
Missing value code: NA
Dependent variable is Logsalary
Piecewise simple linear ANCOVA model
F-to-enter and F-to-delete = 4.000 3.990
Length of longest data entry = 17
Number of dummy variables created = 25
### Summary information (without x variables)

- **d**: dependent, **b**: split and fit cat variable using 0-1 dummies,
- **c**: split-only categorical, **n**: split and fit numerical, **f**: fit-only numerical, **s**: split-only numerical, **w**: weight

For categorical variables, # categories include one for missing values.

<table>
<thead>
<tr>
<th>Column number</th>
<th>Variable name</th>
<th>Type</th>
<th>Minimum value</th>
<th>Maximum value</th>
<th>Number of categories</th>
<th>Number of missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Bat86</td>
<td>n</td>
<td>1.2700E+02</td>
<td>6.8700E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Hit86</td>
<td>n</td>
<td>3.2000E+01</td>
<td>2.3800E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Hr86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>4.0000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Run86</td>
<td>n</td>
<td>1.3000E+01</td>
<td>1.3000E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Rb86</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.2100E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Wlk86</td>
<td>n</td>
<td>3.0000E+00</td>
<td>1.0500E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Yrs</td>
<td>n</td>
<td>1.0000E+00</td>
<td>2.4000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Batcr</td>
<td>n</td>
<td>1.8100E+02</td>
<td>1.4053E+04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Hitcr</td>
<td>n</td>
<td>4.2000E+01</td>
<td>4.2560E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Hrcr</td>
<td>n</td>
<td>0.0000E+00</td>
<td>5.4800E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Runcr</td>
<td>n</td>
<td>1.8000E+01</td>
<td>2.1650E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Rbcr</td>
<td>n</td>
<td>9.0000E+00</td>
<td>1.6590E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Wlkcr</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.5660E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Leag86</td>
<td>b</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Div86</td>
<td>b</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Team86</td>
<td>c</td>
<td></td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Pos86</td>
<td>b</td>
<td></td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Puto86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>1.3770E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Asst86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>4.9200E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Err86</td>
<td>n</td>
<td>0.0000E+00</td>
<td>3.2000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Leag87</td>
<td>b</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Team87</td>
<td>c</td>
<td></td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Logsalary</td>
<td>d</td>
<td>4.2121E+00</td>
<td>7.8079E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*************** Constructed variables ***************

<table>
<thead>
<tr>
<th>Column number</th>
<th>Variable name</th>
<th>Type</th>
<th>Minimum value</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Leag86.N</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>28</td>
<td>Div86.W</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>29</td>
<td>Pos86.10</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>30</td>
<td>Pos86.23</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>31</td>
<td>Pos86.2B</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>32</td>
<td>Pos86.2S</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>33</td>
<td>Pos86.32</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>34</td>
<td>Pos86.3B</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>35</td>
<td>Pos86.30</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>36</td>
<td>Pos86.3S</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>37</td>
<td>Pos86.C</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>38</td>
<td>Pos86.CD</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>39</td>
<td>Pos86.CF</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>40</td>
<td>Pos86.DH</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>41</td>
<td>Pos86.DO</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>42</td>
<td>Pos86.LF</td>
<td>f</td>
<td>0.0000E+00</td>
<td>1.0000E+00</td>
</tr>
</tbody>
</table>
5.3 ANCOVA models

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>Pos86.01</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>44</td>
<td>Pos86.0D</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>45</td>
<td>Pos86.0F</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>46</td>
<td>Pos86.0S</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>47</td>
<td>Pos86.RF</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>48</td>
<td>Pos86.S3</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>49</td>
<td>Pos86.SS</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>50</td>
<td>Pos86.UT</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>51</td>
<td>Leag8.N</td>
<td>f</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Total #cases w/ missing

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#cases</td>
<td>miss. D ord. vals</td>
<td>#X-var</td>
<td>#N-var</td>
<td>#F-var</td>
<td>#S-var</td>
<td>#B-var</td>
<td>#C-var</td>
</tr>
<tr>
<td>263</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

No weight variable in data file
No. cases used for training = 263

Missing N and F values imputed with node means
Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000
Max number of split levels = 10
Minimum node size = 13
Number of SE’s for pruned tree = 5.0000E-01

Size and CV MSE and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Nodes</th>
<th>Mean MSE</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median MSE</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>2.427E-01</td>
<td>2.723E-02</td>
<td>3.309E-02</td>
<td>2.140E-01</td>
<td>4.431E-02</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>2.427E-01</td>
<td>2.723E-02</td>
<td>3.309E-02</td>
<td>2.140E-01</td>
<td>4.431E-02</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>2.478E-01</td>
<td>2.765E-02</td>
<td>3.205E-02</td>
<td>2.218E-01</td>
<td>3.847E-02</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>2.409E-01</td>
<td>2.882E-02</td>
<td>3.097E-02</td>
<td>2.218E-01</td>
<td>3.172E-02</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>2.397E-01</td>
<td>2.885E-02</td>
<td>3.124E-02</td>
<td>2.212E-01</td>
<td>3.288E-02</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>2.391E-01</td>
<td>2.883E-02</td>
<td>3.119E-02</td>
<td>2.201E-01</td>
<td>3.471E-02</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>2.493E-01</td>
<td>3.000E-02</td>
<td>3.331E-02</td>
<td>2.201E-01</td>
<td>5.040E-02</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>2.280E-01</td>
<td>2.802E-02</td>
<td>3.414E-02</td>
<td>2.025E-01</td>
<td>4.236E-02</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>2.253E-01</td>
<td>2.788E-02</td>
<td>3.359E-02</td>
<td>2.025E-01</td>
<td>3.815E-02</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2.143E-01</td>
<td>2.656E-02</td>
<td>2.850E-02</td>
<td>2.021E-01</td>
<td>3.741E-02</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1.931E-01</td>
<td>2.503E-02</td>
<td>2.633E-02</td>
<td>1.959E-01</td>
<td>3.924E-02</td>
</tr>
<tr>
<td>12**</td>
<td>2</td>
<td>1.845E-01</td>
<td>2.378E-02</td>
<td>2.321E-02</td>
<td>1.658E-01</td>
<td>3.923E-02</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>4.259E-01</td>
<td>3.050E-02</td>
<td>3.442E-02</td>
<td>4.367E-01</td>
<td>5.145E-02</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
5.3 ANCOVA models

* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of Logsalary in the node
Cases fit give the number of cases used to fit node
MSE and $R^2$ are based on all cases in node

<table>
<thead>
<tr>
<th>Node label</th>
<th>Total cases</th>
<th>Cases fit</th>
<th>Node rank</th>
<th>D-mean</th>
<th>MSE</th>
<th>$R^2$</th>
<th>Variable</th>
<th>Other variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>263</td>
<td>263</td>
<td>5</td>
<td>5.945E+00</td>
<td>4.160E-01</td>
<td>0.4755</td>
<td>Yrs</td>
<td>+Hitcr</td>
</tr>
<tr>
<td>2T</td>
<td>143</td>
<td>143</td>
<td>4</td>
<td>5.506E+00</td>
<td>1.184E-01</td>
<td>0.8413</td>
<td>Hitcr</td>
<td>+Runcr</td>
</tr>
<tr>
<td>3T</td>
<td>120</td>
<td>120</td>
<td>4</td>
<td>6.469E+00</td>
<td>1.838E-01</td>
<td>0.4729</td>
<td>Wlk86</td>
<td>+Hit86</td>
</tr>
</tbody>
</table>

The linear predictor is the one with a + sign under the ‘‘Other variable’’ column.

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3

Regression tree:

Node 1: Yrs <= 6.00000 or NA
  Node 2: Logsalary-mean = 5.50632
  Node 1: Yrs > 6.00000 and not NA
  Node 3: Logsalary-mean = 6.46866

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if Yrs <= 6.0000000E+00 or NA
  Yrs mean = 7.3802E+00
Coefficients of least squares regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>5.3659E+00</td>
<td>72.24</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hitcr</td>
<td>8.9484E-04</td>
<td>14.45</td>
<td>0.0000</td>
<td>4.2000E+01</td>
<td>7.3392E+02</td>
<td>4.2560E+03</td>
</tr>
<tr>
<td>Div86.W</td>
<td>-1.8665E-01</td>
<td>-2.34</td>
<td>0.0000</td>
<td>0.0000E+00</td>
<td>5.0951E-01</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>Pos86.RF</td>
<td>4.6442E-01</td>
<td>3.22</td>
<td>0.0000</td>
<td>0.0000E+00</td>
<td>8.3650E-02</td>
<td>1.0000E+00</td>
</tr>
<tr>
<td>Pos86.UT</td>
<td>-5.5260E-01</td>
<td>-2.63</td>
<td>0.0000</td>
<td>0.0000E+00</td>
<td>3.8023E-02</td>
<td>1.0000E+00</td>
</tr>
</tbody>
</table>

Mean of Logsalary = 5.9454102795235446
Predicted values truncated at 4.2121275978784798 & 7.8079166289264101

***************************************************************

Node 2: Terminal node
Coefficients of least squares regression functions:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>4.4528E+00</td>
<td>89.49</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runcr</td>
<td>6.8396E-03</td>
<td>26.91</td>
<td>0.0000</td>
<td>1.8000E+01</td>
<td>1.6197E+02</td>
<td>5.2900E+02</td>
</tr>
</tbody>
</table>

Wei-Yin Loh 60 GUIDE manual
5.3 ANCOVA models

Figure 8: GUIDE piecewise ANCOVA regression tree for predicting Logsalary. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘≤’ stands for ‘≤ or missing’. Sample sizes, means of Logsalary, and signs and names of regressor variable printed below nodes. Terminal nodes with negative, zero, and positive slopes are colored red, yellow, and green, respectively.

Pos86.2B  -2.8035E-01  -2.72  0.0000  0.0000E+00  9.0909E-02  1.0000E+00
Pos86.CF  -2.7430E-01  -2.89  0.0000  0.0000E+00  1.0490E-01  1.0000E+00
Mean of Logsalary = 5.5063156054013094
Predicted values truncated at  4.2121275978784798  &  7.8079166289264101
--------------------------------------
Node 3: Terminal node
Coefficients of least squares regression functions:
Regressor  Coefficient  t-stat  p-val  Min  Mean  Max
Constant     5.5298E+00  46.67  0.0000
Hit86     9.2208E-03  9.08  0.0000  3.2000E+01  1.0759E+02  2.0000E+02
Pos86.2B  -2.7882E-01  -2.19  0.0000  0.0000E+00  1.0833E-01  1.0000E+00
Pos86.UT  -3.4458E-01  -2.11  0.0000  0.0000E+00  6.6667E-02  1.0000E+00
Mean of Logsalary = 6.4686647661858760
Predicted values truncated at  4.2121275978784798  &  7.8079166289264101
--------------------------------------
Proportion of variance (R-squared) explained by tree model = .8172

Observed and fitted values are stored in ancova.fit
LaTeX code for tree is in ancova.tex
Elapsed time in seconds:  3.97944403

The \LaTeX{} tree, shown in Figure 8.
5.4 Quantile regression

Instead of estimating the conditional mean, we can estimate conditional quantiles (Chaudhuri and Loh, 2002; Koenker and Bassett, 1978). We demonstrate this with the data set tuition.dat, which gives information on tuition and other variables for U.S. colleges. The data description file tuitiondsc.txt is:

\begin{verbatim}
tuition.dat
NA
col_num var_name var_type
1   FICE x
2  CollName x
3   State x
4 PubPriv c
5 MathSAT x
6 VerbSAT x
7 CombSAT n
8  ACT x
9  Q1MSAT x
10 Q3MSAT x
11 Q1VSAT x
12 Q3VSAT x
13  Q1ACT x
14  Q3ACT x
15 AppsRec n
16 AppsAcc n
17 NewEnrol n
18  Top10 n
19  Top25 n
20  FUgrad n
21  PUgrad x
22  InTuition x
23  OutTuition d
24  RnBcost n
25  RmCost x
26  BrdCost x
27  AddFees x
28  BookCost x
29  PerSpend x
30  PFacPhD n
31  PFacTerm x
32  StudFac n
33  PAIDonate x
34  InstExp n
35  GradRate n
36  Type c
37  FullPSal n
\end{verbatim}

Wei-Yin Loh
Following is a session log to create an input file for constructing a piecewise simple linear tree for the 90th percentile of out-of-state tuition (OutTuition).

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: quant.in
Name of batch output file: quant.out
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 2
Choose complexity of model to use at each node:
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1):
Option 2 allows simultaneous modeling of a pair of quantile values (e.g., 0.1 and 0.9)
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.9

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): tuitiondsc.txt
Reading data description file ...
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: B variables changed to C
Dependent variable is OutTuition
Length of longest data entry = 20
Total number of cases = 1134
Col. no. Categorical variable #levels #missing values
4 PubPriv 2 0
36 Type 3 0
Checking data ...
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
1134 13 621 32 14 0 0 0 2
No. cases used for training = 1121
No. cases excluded due to 0 weight or missing D = 13
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): quant.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: quant.fit
Input file is created!

Results

Quantile regression tree with quantile probability .9000
No truncation of predicted values
Pruning by cross-validation
Data description file: tuitiondsc.txt
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: B variables changed to C
Dependent variable is OutTuition
Piecewise simple linear or constant model
Length of longest data entry = 20

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
For categorical variables, #categories include one for missing values
Column Variable Variable Minimum Maximum Number of Number
number name type value value categories missing
4 PubPriv c 6.0000E+02 1.4100E+03 471
7 CombSAT n 5.7000E+01 4.8094E+03 9
15 AppsRec n 4.4000E+01 2.6330E+04 9
16 AppsAcc n 2.1000E+01 7.4250E+03 5
17 NewEnrol n 1.0000E+00 9.8000E+01 183
18 Top10 n 1.1000E+01 1.0000E+02 155
20 FUgrad n 1.1800E+02 3.1643E+04 3
23 OutTuition d 1.0440E+03 2.5750E+04 13

Wei-Yin Loh 64 GUIDE manual
### 5.4 Quantile regression

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>RnBcost</td>
<td>n</td>
<td>1.3060E+03</td>
<td>8.7000E+03</td>
</tr>
<tr>
<td>30</td>
<td>PFacPhD</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.0500E+02</td>
</tr>
<tr>
<td>32</td>
<td>StudFac</td>
<td>n</td>
<td>2.5000E+00</td>
<td>4.2600E+01</td>
</tr>
<tr>
<td>34</td>
<td>InstExp</td>
<td>n</td>
<td>1.8340E+03</td>
<td>6.2469E+04</td>
</tr>
<tr>
<td>35</td>
<td>GradRate</td>
<td>n</td>
<td>8.0000E+00</td>
<td>1.1800E+02</td>
</tr>
<tr>
<td>36</td>
<td>Type</td>
<td>c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>FullPSal</td>
<td>n</td>
<td>2.7000E+02</td>
<td>1.0090E+03</td>
</tr>
<tr>
<td>45</td>
<td>NFullProf</td>
<td>n</td>
<td>0.0000E+00</td>
<td>9.9700E+02</td>
</tr>
</tbody>
</table>

Total #cases w/ #missing

<table>
<thead>
<tr>
<th>#cases</th>
<th>miss. D</th>
<th>ord. vals</th>
<th>#X-var</th>
<th>#N-var</th>
<th>#F-var</th>
<th>#S-var</th>
<th>#B-var</th>
<th>#C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>1134</td>
<td>13</td>
<td>621</td>
<td>32</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

No. cases used for training = 1121
No. cases excluded due to 0 weight or missing D = 13

Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000
Max number of split levels = 11
Minimum node size = 55
Number of SE’s for pruned tree = 5.0000E-01

Size and CV’s Loss and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Nodes</th>
<th>Mean Loss</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Loss</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>3.637E+02</td>
<td>1.442E+01</td>
<td>1.879E+01</td>
<td>3.605E+02</td>
<td>1.738E+01</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>3.641E+02</td>
<td>1.443E+01</td>
<td>1.883E+01</td>
<td>3.605E+02</td>
<td>1.783E+01</td>
</tr>
<tr>
<td>3+</td>
<td>11</td>
<td>3.644E+02</td>
<td>1.443E+01</td>
<td>1.881E+01</td>
<td>3.588E+02</td>
<td>1.708E+01</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>3.648E+02</td>
<td>1.447E+01</td>
<td>1.882E+01</td>
<td>3.608E+02</td>
<td>1.719E+01</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>3.612E+02</td>
<td>1.400E+01</td>
<td>1.990E+01</td>
<td>3.608E+02</td>
<td>2.034E+01</td>
</tr>
<tr>
<td>6++</td>
<td>8</td>
<td>3.568E+02</td>
<td>1.393E+01</td>
<td>1.794E+01</td>
<td>3.594E+02</td>
<td>2.100E+01</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>3.667E+02</td>
<td>1.407E+01</td>
<td>1.598E+01</td>
<td>3.680E+02</td>
<td>2.113E+01</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>3.667E+02</td>
<td>1.397E+01</td>
<td>1.489E+01</td>
<td>3.680E+02</td>
<td>1.910E+01</td>
</tr>
<tr>
<td>9**</td>
<td>5</td>
<td>3.576E+02</td>
<td>1.311E+01</td>
<td>1.272E+01</td>
<td>3.718E+02</td>
<td>1.420E+01</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>3.830E+02</td>
<td>1.368E+01</td>
<td>6.555E+00</td>
<td>3.862E+02</td>
<td>8.025E+00</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>4.625E+02</td>
<td>1.439E+01</td>
<td>1.121E+01</td>
<td>4.616E+02</td>
<td>2.334E+01</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
* tree same as ++ tree

Following tree is based on mean CV with naive SE estimate (**).
5.4 Quantile regression

Structure of final tree. Each terminal node is marked with a T.

D-quant is quantile of OutTuition in the node
Cases fit give the number of cases used to fit node

<table>
<thead>
<tr>
<th>Node label</th>
<th>Total cases</th>
<th>Cases fit</th>
<th>rank</th>
<th>D-quant</th>
<th>variable</th>
<th>Other variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1121</td>
<td>1098</td>
<td>2</td>
<td>1.591E+04</td>
<td>PubPriv</td>
<td></td>
</tr>
<tr>
<td>2T</td>
<td>434</td>
<td>432</td>
<td>2</td>
<td>8.820E+03</td>
<td>RnBcost</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>687</td>
<td>672</td>
<td>2</td>
<td>1.745E+04</td>
<td>RnBcost</td>
<td></td>
</tr>
<tr>
<td>6T</td>
<td>209</td>
<td>203</td>
<td>2</td>
<td>1.160E+04</td>
<td>FUgrad</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>478</td>
<td>469</td>
<td>2</td>
<td>1.848E+04</td>
<td>AppsAcc</td>
<td></td>
</tr>
<tr>
<td>14T</td>
<td>220</td>
<td>214</td>
<td>2</td>
<td>1.468E+04</td>
<td>FullPSal</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>258</td>
<td>255</td>
<td>2</td>
<td>1.900E+04</td>
<td>StudFac</td>
<td></td>
</tr>
<tr>
<td>30T</td>
<td>108</td>
<td>105</td>
<td>2</td>
<td>1.970E+04</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>31T</td>
<td>150</td>
<td>147</td>
<td>2</td>
<td>1.615E+04</td>
<td>AppsRec</td>
<td></td>
</tr>
</tbody>
</table>

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9

Regression tree:

Node 1: PubPriv = "Public"
  Node 2: OutTuition sample quantile = 8.82000E+03
Node 1: PubPriv /= "Public"
  Node 3: RnBcost <= 3.89050E+03
   Node 6: OutTuition sample quantile = 1.16000E+04
  Node 3: RnBcost > 3.89050E+03 or NA
   Node 7: AppsAcc <= 9.26500E+02
    Node 14: OutTuition sample quantile = 1.46750E+04
   Node 7: AppsAcc > 9.26500E+02 or NA
    Node 15: StudFac <= 12.10000
     Node 30: OutTuition sample quantile = 1.97000E+04
    Node 15: StudFac > 12.10000 or NA
     Node 31: OutTuition sample quantile = 1.61545E+04

***********************************************************************
In the following the predictor node mean is mean of complete cases
Regression coefficients are computed from the complete cases

Node 1: Intermediate node
A case goes into Node 2 if PubPriv = "Public"
PubPriv mode = "Private"

Coefficients of quantile regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Min</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>3.7443E+03</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.4 Quantile regression

Regressor Coefficient Min Mean Max
Constant -8.9699E+02
FullPSal 1.6871E+01 3.5900E+02 5.4766E+02 8.9300E+02

Node 2: Terminal node

Node 3: Intermediate node
A case goes into Node 6 if RnBcost <= 3.8905000E+03
RnBcost mean = 4.5535E+03

Node 6: Terminal node

Node 7: Intermediate node
A case goes into Node 14 if AppsAcc <= 9.2650000E+02
AppsAcc mean = 1.4274E+03

Node 14: Terminal node

Node 15: Intermediate node
A case goes into Node 30 if StudFac <= 1.2100000E+01
StudFac mean = 1.2708E+01

Node 30: Terminal node

Wei-Yin Loh

GUIDE manual
5.5 Least median of squares

Although median regression may be preferred to least-squares regression if there are large outliers in a data set, an alternative that is even more robust to outliers is least median of squares regression (Rousseeuw and Leroy, 1987). GUIDE can construct tree models using this criterion. We use the college tuition data for illustration. A session log of the input file generation is below, followed by the results and the \LaTeX{} tree diagram in Figure 11.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning
Input your choice: 1
Name of batch input file: lms.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lms.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1): 2

The \LaTeX{} tree is shown in Figure 9 and plots of the data in the terminal nodes of the tree are given in Figure 10.
5.5 Least median of squares

Figure 9: GUIDE piecewise simple linear 0.900-quantile regression tree for predicting OutTuition. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. Sample sizes, 0.900-quantiles of OutTuition, and signs and names of best regressor printed below nodes. Terminal nodes with negative, zero, and positive slopes are colored red, yellow, and green, respectively.
5.5 Least median of squares

Figure 10: Plots of data and estimated linear 0.9-quantile fits in terminal nodes of tree in Figure 9.
This is where the option for least median of squares is selected.
Choose complexity of model to use at each node:
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=2):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes
if it contains spaces or non alphanumeric characters): tuitiondsc.txt
Reading data description file ...
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: B variables changed to C
Dependent variable is OutTuition
Length of longest data entry = 20
Total number of cases = 1134
Col. no. Categorical variable #levels #missing values
 4 PubPriv 2 0
36 Type 3 0
Checking data ...
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
1134 13 621 32 14 0 0 0 2
No weight variable in data file
No. cases used for training = 1121
No. cases excluded due to 0 weight or missing D = 13
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lms.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: lms.fit
Input file is created!

Results

Least median of squares regression tree
Predictions truncated at global min and max of D sample values
Pruning by cross-validation
Data description file: tuitiondsc.txt
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: B variables changed to C
Dependent variable is OutTuition
Piecewise simple linear or constant model
Length of longest data entry = 20

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
### 5.5 Least median of squares

n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>PubPriv</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>CombSAT</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>AppsRec</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>AppsAcc</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>NewEnrol</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>Top10</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>Top25</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>FUgrad</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>OutTuition</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>RnBcost</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>PFacPhD</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>StudFac</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>InstExp</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>GradRate</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>Type</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>FullPSal</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
<tr>
<td>NFullProf</td>
<td>0.0000E+00</td>
<td>1.4100E+03</td>
<td>621</td>
<td>0</td>
</tr>
</tbody>
</table>

**Total cases w/ missing**

<table>
<thead>
<tr>
<th>#cases</th>
<th>#cases miss.</th>
<th>D ord. vals</th>
<th>#X-var</th>
<th>#N-var</th>
<th>#F-var</th>
<th>#S-var</th>
<th>#B-var</th>
<th>#C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>1134</td>
<td>13</td>
<td>621</td>
<td>32</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

No weight variable in data file

No. cases used for training: 1121

No. cases excluded due to 0 weight or missing D: 13

Missing N and F values imputed with node means

Interaction tests on all variables

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Fraction of cases used for splitting each node = 1.0000

Max number of split levels = 11

Minimum node size = 55

Number of SE's for pruned tree = 5.0000E-01

Size and CV median absolute residual (MAR) and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean MAR</th>
<th>BSE(Mean)</th>
<th>Median MAR</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>1.344E+05</td>
<td>5.452E+01</td>
<td>1.176E+03</td>
<td>8.878E+01</td>
</tr>
<tr>
<td>2+</td>
<td>13</td>
<td>1.326E+05</td>
<td>5.288E+01</td>
<td>1.148E+03</td>
<td>7.454E+01</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>1.333E+05</td>
<td>5.296E+01</td>
<td>1.161E+03</td>
<td>6.814E+01</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>1.353E+05</td>
<td>5.546E+01</td>
<td>1.161E+03</td>
<td>8.151E+01</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>1.367E+05</td>
<td>5.932E+01</td>
<td>1.222E+03</td>
<td>6.692E+01</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1.364E+05</td>
<td>4.972E+01</td>
<td>1.230E+03</td>
<td>7.063E+01</td>
</tr>
<tr>
<td>7--</td>
<td>7</td>
<td>1.313E+05</td>
<td>2.490E+01</td>
<td>1.181E+03</td>
<td>4.526E+01</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>1.340E+05</td>
<td>2.932E+01</td>
<td>1.223E+03</td>
<td>3.248E+01</td>
</tr>
</tbody>
</table>

Wei-Yin Loh

GUIDE manual
5.5 Least median of squares

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>5</td>
<td>1.388E+05</td>
<td>2.292E+01</td>
<td>1.248E+03</td>
<td>2.001E+01</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1.406E+05</td>
<td>3.343E+01</td>
<td>1.264E+03</td>
<td>3.146E+01</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1.476E+05</td>
<td>5.183E+01</td>
<td>1.266E+03</td>
<td>7.042E+01</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1.752E+05</td>
<td>8.364E+01</td>
<td>1.553E+03</td>
<td>1.203E+02</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1.901E+05</td>
<td>8.738E+01</td>
<td>1.716E+03</td>
<td>9.536E+01</td>
</tr>
</tbody>
</table>

The selected tree is marked by two dashes.
0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
++ tree same as -- tree
* tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with bootstrap SE estimate (--).

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of OutTuition in the node
Cases fit give the number of cases used to fit node
MAR is median of absolute residuals

<table>
<thead>
<tr>
<th>Node</th>
<th>Total cases</th>
<th>Matrix Cases</th>
<th>D-mean</th>
<th>MAR</th>
<th>Variable variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1121</td>
<td>1121</td>
<td>8.820E+03</td>
<td>1.687E+03</td>
<td>PubPriv +InstExp</td>
</tr>
<tr>
<td>2</td>
<td>434</td>
<td>434</td>
<td>6.114E+03</td>
<td>9.854E+02</td>
<td>NFullProf +RnBcost</td>
</tr>
<tr>
<td>4T</td>
<td>198</td>
<td>198</td>
<td>5.130E+03</td>
<td>7.377E+02</td>
<td>StudFac +RnBcost</td>
</tr>
<tr>
<td>5T</td>
<td>236</td>
<td>236</td>
<td>6.832E+03</td>
<td>9.981E+02</td>
<td>InstExp +FullPSal</td>
</tr>
<tr>
<td>3</td>
<td>687</td>
<td>687</td>
<td>1.096E+04</td>
<td>1.485E+03</td>
<td>InstExp +InstExp</td>
</tr>
<tr>
<td>6</td>
<td>577</td>
<td>577</td>
<td>1.042E+04</td>
<td>1.314E+03</td>
<td>RnBcost +InstExp</td>
</tr>
<tr>
<td>12</td>
<td>242</td>
<td>242</td>
<td>8.444E+03</td>
<td>1.268E+03</td>
<td>InstExp +InstExp</td>
</tr>
<tr>
<td>24T</td>
<td>179</td>
<td>179</td>
<td>7.896E+03</td>
<td>1.178E+03</td>
<td>RnBcost +InstExp</td>
</tr>
<tr>
<td>25T</td>
<td>63</td>
<td>63</td>
<td>9.950E+03</td>
<td>7.434E+02</td>
<td>- +FUgrad</td>
</tr>
<tr>
<td>13</td>
<td>335</td>
<td>335</td>
<td>1.155E+04</td>
<td>1.072E+03</td>
<td>NewEnrol +InstExp</td>
</tr>
<tr>
<td>26T</td>
<td>157</td>
<td>157</td>
<td>1.086E+04</td>
<td>8.482E+02</td>
<td>RnBcost +InstExp</td>
</tr>
<tr>
<td>27T</td>
<td>178</td>
<td>178</td>
<td>1.259E+04</td>
<td>1.006E+03</td>
<td>Type +InstExp</td>
</tr>
<tr>
<td>7T</td>
<td>110</td>
<td>110</td>
<td>1.788E+04</td>
<td>8.988E+02</td>
<td>- +GradRate</td>
</tr>
</tbody>
</table>

Number of terminal nodes of final tree: 7
Total number of nodes of final tree: 13

Regression tree:

Node 1: PubPriv = "Public"
Node 2: NFullProf <= 93.00000
Node 4: OutTuition-mean = 5.13000E+03

Wei-Yin Loh 73 GUIDE manual
5.5 Least median of squares

Node 2: NFullProf > 93.00000 or NA
Node 5: OutTuition-mean = 6.83150E+03
Node 1: PubPriv /= "Public"
Node 3: InstExp <= 1.38525E+04 or NA
Node 6: RnBcost <= 3.99950E+03 or NA
  Node 12: InstExp <= 8.33000E+03 or NA
    Node 24: OutTuition-mean = 7.89600E+03
    Node 12: InstExp > 8.33000E+03 and not NA
      Node 25: OutTuition-mean = 9.95000E+03
    Node 6: RnBcost > 3.99950E+03 and not NA
  Node 13: NewEnrol <= 3.19500E+02
    Node 26: OutTuition-mean = 1.08600E+04
  Node 13: NewEnrol > 3.19500E+02 or NA
    Node 27: OutTuition-mean = 1.25860E+04
Node 3: InstExp > 1.38525E+04 and not NA
Node 7: OutTuition-mean = 1.78825E+04

*******************************************************************************
In the following the predictor node mean is mean of complete cases
Regression coefficients are computed from the complete cases

Node 1: Intermediate node
A case goes into Node 2 if PubPriv = "Public"
PubPriv mode = "Private"
Coefficients of least median of squares regression function:
Regressor Coefficient Minimum Mean Maximum
Constant 973.9978
InstExp 1.0571E+00 1834.00****** 6.2469E+04
Mean of OutTuition = 8820.0000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Node 2: Intermediate node
A case goes into Node 4 if NFullProf <= 9.3000000E+01
NFullProf mean = 1.7104E+02

Node 4: Terminal node
Coefficients of least median of squares regression function:
Regressor Coefficient Minimum Mean Maximum
Constant 1.3867E+03
RnBcost 1.2073E+00 1306.00****** 5.5440E+03
Mean of OutTuition = 5130.0000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Node 5: Terminal node
Coefficients of least median of squares regression function:
5.5 Least median of squares

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-330.8814</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FullPSal</td>
<td>1.2814E+01</td>
<td>440.00</td>
<td>8.9300</td>
<td></td>
</tr>
</tbody>
</table>

Mean of OutTuition = 6831.5000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Node 3: Intermediate node
A case goes into Node 6 if InstExp <= 1.3852500E+04 or NA
InstExp mean = 1.0340E+04

Node 6: Intermediate node
A case goes into Node 12 if RnBcost <= 3.9995000E+03 or NA
RnBcost mean = 4.3413E+03

Node 12: Intermediate node
A case goes into Node 24 if InstExp <= 8.3300000E+03 or NA
InstExp mean = 7.3339E+03

Node 24: Terminal node
Coefficients of least median of squares regression function:
<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-1.3502E+03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstExp</td>
<td>1.3199E+00</td>
<td>2589.00</td>
<td>8.3240</td>
<td></td>
</tr>
</tbody>
</table>

Mean of OutTuition = 7896.0000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Node 25: Terminal node
Coefficients of least median of squares regression function:
<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>8.1732E+03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUgrad</td>
<td>2.3909E+00</td>
<td>139.00</td>
<td>5.0640</td>
<td></td>
</tr>
</tbody>
</table>

Mean of OutTuition = 9950.0000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Node 26: Terminal node
Coefficients of least median of squares regression function:
<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>6.7469E+03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstExp</td>
<td>3.9716E-01</td>
<td>4054.00</td>
<td>1.3844</td>
<td></td>
</tr>
</tbody>
</table>

Mean of OutTuition = 10860.0000000000000
Predicted values truncated at 1044.0000000000000 & 25750.0000000000000

Wei-Yin Loh

GUIDE manual
5.6 Poisson regression with offset

We use a data set from www.statsci.org/data/general/motorins.html on motor insurance claims in Sweden for the year 1977 (Andrews and Herzberg, 1985; Hallin and Ingenbleek, 1983). The description and data files are swedendat.txt and swedendat.txt. The dependent variable is the number of claims. The other variables are mileagegp with ordered values 1–5, zone with 7 unordered values, bonus which is the number of years plus one since last claim, make of car with 9 unordered values, insured which is the number of insured in policy-years, and payment which is the total value of payments in Skr. We ignore insured and payments by giving them the x designation. To fit a Poisson regression model for the claim rate, we created an offset variable lninsured which is the log of insured and designate as z. Because mileagegp is an ordered categorical variable, we designate it as s to prevent it from being used as a linear predictor for fitting the Poisson node models. We designate bonus as f so that it is only used as a linear predictor and not for splitting the nodes.
5.6 Poisson regression with offset

REGRESSION

\[ \text{PubPriv} = \text{Public} \]

\[ N_{\text{FullProf}} \leq 93.00 \]

\[ + \text{InstExp} \leq \ast 8330.0 \]

\[ + \text{RnBcost} \leq \ast 999.5 \]

\[ + \text{NewEnrol} \leq \ast 319.50 \]

\[ + \text{FUgrad} \leq 319.50 \]

\[ + \text{InstExp} \leq 8330.0 \]

\[ + \text{InstExp} \leq 13852.5 \]

\[ + \text{InstExp} \leq 17882.0 \]

\[ + \text{InstExp} \leq 12586.0 \]

Figure 11: GUIDE 0.50-SE piecewise simple linear least-median-of-squares regression tree for predicting \( \text{OutTuition} \). At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘\( \leq \ast \)’ stands for ‘\( \leq \) or missing’. Sample sizes, means of \( \text{OutTuition} \), and signs and names of regressor variable are printed below nodes. Terminal nodes with negative, zero, and positive slopes are colored red, yellow, and green, respectively.
5.6 Poisson regression with offset

Since there is only one linear predictor, the multiple linear Poisson model is the same as the best simple linear Poisson model.

---

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1

Name of batch input file: input.txt

File input.txt exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):

Name of batch output file: output.txt

File output.txt exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2

Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 3

Choose complexity of model to use at each node:
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): swedendsc.txt

Reading data description file ...

Training sample file: swedendat.txt

Missing value code: ?
Dependent variable is claims
Length of longest data entry = 19
Total number of cases = 2182

Col. no. Categorical variable #levels #missing values
2 zone 7 0
4 make 9 0

Checking data ...
Number of cases with positive D values = 1797
5.6 Poisson regression with offset

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
2182 0 0 2 0 1 1 0 2
Offset variable in column: 6
No. cases used for training = 2182
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): sweden.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input file is created!

Results

Poisson regression tree
No truncation of predicted values
Pruning by cross-validation
Data description file: swedendsc.txt
Training sample file: swedendat.txt
Missing value code: ?
Dependent variable is claims
Piecewise linear model
Length of longest data entry = 19
Number of cases with positive D values: 1797

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight, z=offset variable

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mileagegp</td>
<td>1.0000E+00</td>
<td>5.0000E+00</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>zone</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>bonus</td>
<td>f</td>
<td>1.0000E+00</td>
<td>7.0000E+00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>make</td>
<td>c</td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>lninsured</td>
<td>-4.6052E+00</td>
<td>1.1757E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>claims</td>
<td>d</td>
<td>0.0000E+00</td>
<td>3.3380E+03</td>
<td></td>
</tr>
</tbody>
</table>

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
2182 0 0 2 0 1 1 0 2
Offset variable in column 6
No. cases used for training: 2182

Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000

Wei-Yin Loh 79 GUIDE manual
Max number of split levels = 12
Minimum node size = 109
100 bootstrap calibration replicates
Number of SE’s for pruned tree = 5.0000E-01

Size and CV Loss and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean Loss</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Loss</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1*</td>
<td>10</td>
<td>3.217E+00</td>
<td>2.507E-01</td>
<td>2.542E-01</td>
<td>2.870E+00</td>
<td>2.166E-01</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>3.229E+00</td>
<td>2.510E-01</td>
<td>2.538E-01</td>
<td>2.964E+00</td>
<td>2.098E-01</td>
</tr>
<tr>
<td>3**</td>
<td>7</td>
<td>3.301E+00</td>
<td>2.591E-01</td>
<td>2.974E-01</td>
<td>2.964E+00</td>
<td>2.109E-01</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3.458E+00</td>
<td>2.812E-01</td>
<td>2.940E-01</td>
<td>3.196E+00</td>
<td>2.070E-01</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3.584E+00</td>
<td>2.871E-01</td>
<td>2.760E-01</td>
<td>3.355E+00</td>
<td>2.192E-01</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>3.566E+00</td>
<td>2.974E-01</td>
<td>2.813E-01</td>
<td>3.523E+00</td>
<td>2.159E-01</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>4.243E+00</td>
<td>3.823E-01</td>
<td>3.722E-01</td>
<td>4.207E+00</td>
<td>5.342E-01</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>6.519E+00</td>
<td>6.365E-01</td>
<td>5.622E-01</td>
<td>6.609E+00</td>
<td>7.858E-01</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Rate is mean of $Y/\exp(\text{offset})$
Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node

<table>
<thead>
<tr>
<th>Node label</th>
<th>Total cases</th>
<th>Cases Matrix</th>
<th>Node rank</th>
<th>Rate</th>
<th>Deviance</th>
<th>Other variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2182</td>
<td>2182</td>
<td>2</td>
<td>4.749E-02</td>
<td>6.408E+00</td>
<td>make</td>
</tr>
<tr>
<td>2T</td>
<td>482</td>
<td>482</td>
<td>2</td>
<td>3.453E-02</td>
<td>2.253E+00</td>
<td>zone</td>
</tr>
<tr>
<td>3</td>
<td>1700</td>
<td>1700</td>
<td>2</td>
<td>4.864E-02</td>
<td>6.663E+00</td>
<td>zone</td>
</tr>
<tr>
<td>6</td>
<td>490</td>
<td>490</td>
<td>2</td>
<td>6.375E-02</td>
<td>6.030E+00</td>
<td>zone</td>
</tr>
<tr>
<td>12T</td>
<td>245</td>
<td>245</td>
<td>2</td>
<td>5.627E-02</td>
<td>4.450E+00</td>
<td></td>
</tr>
<tr>
<td>13T</td>
<td>245</td>
<td>245</td>
<td>2</td>
<td>7.258E-02</td>
<td>5.188E+00</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1210</td>
<td>1210</td>
<td>2</td>
<td>4.209E-02</td>
<td>3.927E+00</td>
<td>mileagegp</td>
</tr>
<tr>
<td>14T</td>
<td>243</td>
<td>243</td>
<td>2</td>
<td>3.613E-02</td>
<td>4.163E+00</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>967</td>
<td>967</td>
<td>2</td>
<td>4.497E-02</td>
<td>2.558E+00</td>
<td>mileagegp</td>
</tr>
<tr>
<td>30</td>
<td>490</td>
<td>490</td>
<td>2</td>
<td>4.368E-02</td>
<td>2.986E+00</td>
<td>zone</td>
</tr>
</tbody>
</table>
5.6 Poisson regression with offset

Regression tree:

Node 1: make = "4", "6"
  Node 2: claims sample rate = 3.45272E-02
Node 1: make /= "4", "6"
  Node 3: zone = "1", "2"
  Node 6: zone = "2"
    Node 12: claims sample rate = 5.62714E-02
    Node 6: zone /= "2"
    Node 13: claims sample rate = 7.25828E-02
  Node 3: zone /= "1", "2"
    Node 7: mileagegp <= 1.00000
      Node 14: claims sample rate = 3.61274E-02
    Node 7: mileagegp > 1.00000 or ?
      Node 15: mileagegp <= 3.00000 or ?
        Node 30: zone = "3", "5"
          Node 60: claims sample rate = 5.01083E-02
        Node 30: zone /= "3", "5"
          Node 61: claims sample rate = 4.05632E-02
      Node 15: mileagegp > 3.00000 and not ?
        Node 31: claims sample rate = 5.02372E-02

**************************************************************

Node 1: Intermediate node
A case goes into Node 2 if make = "4", "6"
make mode = "1"
Coefficients of loglinear regression function:
Regressor Coefficient t-stat p-val Minimum Mean Maximum
Constant -2.0572E+00 -305.02 0.0000
bonus -1.8651E-01 -148.88 0.0000 1.0000E+00 4.0151E+00 7.0000E+00
Node mean for offset variable = 4.4928E+00
If regression function is inapplicable due to missing values, predicted rate =
  4.7487588464521328E-002
-----------------------------

Node 2: Terminal node
Coefficients of loglinear regression function:
### 5.6 Poisson regression with offset

<table>
<thead>
<tr>
<th>Regressor Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-2.5849E+00</td>
<td>-104.45</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-1.6243E-01</td>
<td>-32.72</td>
<td>0.0000</td>
<td>4.0249E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 4.3366E+00

If regression function is inapplicable due to missing values, predicted rate = 3.4527165629107126E-002

---

Node 3: Intermediate node

A case goes into Node 6 if zone = "1", "2"
zone mode = "1"

---

Node 6: Intermediate node

A case goes into Node 12 if zone = "2"
zone mode = "1"

---

Node 12: Terminal node

Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-1.8885E+00</td>
<td>-117.98</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-1.8707E-01</td>
<td>-62.66</td>
<td>0.0000</td>
<td>4.0000E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 5.1038E+00

If regression function is inapplicable due to missing values, predicted rate = 5.6271427885877690E-002

---

Node 13: Terminal node

Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-1.5438E+00</td>
<td>-107.39</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-2.1116E-01</td>
<td>-75.60</td>
<td>0.0000</td>
<td>4.9811E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 4.9811E+00

If regression function is inapplicable due to missing values, predicted rate = 7.2582822726790452E-002

---

Node 7: Intermediate node

A case goes into Node 14 if mileagegp <= 1.0000000E+00
mileagegp mean = 2.9860E+00

---

Node 14: Terminal node

Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-2.3934E+00</td>
<td>-152.64</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-1.8117E-01</td>
<td>-60.43</td>
<td>0.0000</td>
<td>4.0123E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 4.6679E+00

If regression function is inapplicable due to missing values, predicted rate = 3.6127394092405472E-002
5.6 Poisson regression with offset

Node 15: Intermediate node
A case goes into Node 30 if mileagegp \leq 3.000000E+00 or ?

mileagegp mean = 3.4850E+00

Node 30: Intermediate node
A case goes into Node 60 if zone = "3", "5"
zone mode = "3"

Node 60: Terminal node
Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-1.9094E+00</td>
<td>-93.79</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-2.0022E-01</td>
<td>-53.60</td>
<td>0.0000</td>
<td>1.0000E+00</td>
<td>4.0000E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 5.1513E+00
If regression function is inapplicable due to missing values, predicted rate = 5.0108317504291920E-002

Node 61: Terminal node
Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-2.0908E+00</td>
<td>-131.72</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-2.0319E-01</td>
<td>-70.53</td>
<td>0.0000</td>
<td>1.0000E+00</td>
<td>4.0000E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 4.7057E+00
If regression function is inapplicable due to missing values, predicted rate = 4.0563186633902397E-002

Node 31: Terminal node
Coefficients of loglinear regression function:

<table>
<thead>
<tr>
<th>Regressor</th>
<th>Coefficient</th>
<th>t-stat</th>
<th>p-val</th>
<th>Minimum</th>
<th>Mean</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-2.0319E+00</td>
<td>-64.33</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bonus</td>
<td>-1.6109E-01</td>
<td>-30.64</td>
<td>0.0000</td>
<td>1.0000E+00</td>
<td>4.0377E+00</td>
<td>7.0000E+00</td>
</tr>
</tbody>
</table>

Node mean for offset variable = 3.5950E+00
If regression function is inapplicable due to missing values, predicted rate = 5.0237213688715164E-002

Latex code for tree is in sweden.tex
Elapsed time in seconds: 3.71040082

The tree is shown in Figure 12.
Figure 12: GUIDE 0.50-SE Poisson regression tree for predicting claims. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘\( \leq \ast \)’ stands for ‘\( \leq \) or missing’. Set \( S_1 = \{4, 6\} \). Set \( S_2 = \{1, 2\} \). Set \( S_3 = \{3, 5\} \). Sample sizes, sample rates, and names of regressor are printed below nodes.
5.7 Censored response data

GUIDE can fit a piecewise-constant, piecewise-simple linear, or piecewise multiple linear proportional hazards regression model to censored response data. Using usual notation, let \( \lambda(x, t) \) denote the hazard rate at time \( t \) for a subject with covariate vector \( x \). In a proportional hazards model, the hazard rate can be factored as

\[
\lambda(x, t) = \lambda_0(t) f(x, \beta),
\]

where \( \lambda_0(t) \) is a “baseline” hazard rate that is independent of the covariates and \( f(x, \beta) \) is a function of \( x \) and some coefficients \( \beta \), independent of \( t \). The Cox proportional hazards model uses \( \lambda(x, t) = \lambda_0(t) \exp(\beta'x) \). GUIDE fits the more general model

\[
\lambda(x, t) = \lambda_0(t) \sum_i I(x \in S_i) \exp(\beta'_i x),
\]

where \( S_i \) is a set corresponding node \( i \) and \( \beta_i \) is its associated coefficient vector. See Loh et al. (2015) for more details.

We illustrate the piecewise-constant model \( \lambda(x, t) = \lambda_0(t) \sum_i I(x \in S_i) \exp(\beta_0) \) with a data set from the Worcester Heart Attack Study analyzed in Hosmer et al. (2008). The data are in the file `whas500.csv` and the description file in `whas500.dsc` whose contents are repeated below.

whas500.csv
NA
NA
c1 c2 c3
1 id x
2 age n
3 gender c
4 hr n
5 sysbp n
6 diasbp n
7 bmi n
8 cvd c
9 afb c
10 sho c
11 chf c
12 av3 c
13 miord c
14 mitype c
15 year c
16 admitdate x
17 disdate x
18 fdate x
19 los n
20 dstat x
21 lenfol t

Wei-Yin Loh 85 GUIDE manual
The goal of the study is to observe survival rates following hospital admission for acute myocardial infarction. The response variable is lenfo1, which stands for total length of follow-up in days. Variable fstat is status at last follow-up (0=alive, 1=dead) and variable chf is congestive heart complications (0=no, 1=yes).

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: whas500.in
Input 1 for model fitting, 2 for importance or DIF scoring,
3 for data conversion ([1:3], <cr>=1):
Name of batch output file: whas500.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiproresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3): 3
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: whas500.dsc
Reading data description file ...
Training sample file: whas500.csv
Missing value code: NA
Warning: N variables changed to S
Dependent variable is fstat
Length of longest data entry = 10
Total number of cases: 500

Col. no. Categorical variable      #levels #missing values
3   gender       2                   0
8   cvd          2                   0
9   afb          2                   0
10  sho          2                   0
11  chf          2                   0
12  av3          2                   0
13  miord        2                   0
14  mitype       2                   0
15  year         3                   0
Checking data ...
Smallest uncensored T: 1.0000
No. complete cases excluding censored T < smallest uncensored T: 500
No. cases used to compute baseline hazard: 500
No. cases with D=1 and T >= smallest uncensored: 215
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
500 0 0 5 0 0 6 0 9
Survival time variable in column: 21
Censoring indicator variable in column: 22
Proportion of uncensored among nonmissing T and D variables = 0.430
No. cases used for training: 500
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): whas500.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node IDs and fitted values: whas500.fit
Input file is created!

Results

Proportional hazards regression with relative risk estimates
Pruning by cross-validation
Data description file: whas500.dsc
Training sample file: whas500.csv
Missing value code: NA
Warning: N variables changed to S
Dependent variable is fstat
Piecewise constant model
Length of longest data entry = 10
Smallest uncensored T: 1.0000
No. complete cases excluding censored T < smallest uncensored T: 500
No. cases used to compute baseline hazard: 500
No. cases with D=1 and T >= smallest uncensored: 215

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
T=survival time variable

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>age</td>
<td>3.0000E+01</td>
<td>1.0400E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>gender</td>
<td>c</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>hr</td>
<td>3.5000E+01</td>
<td>1.8600E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>sysbp</td>
<td>s</td>
<td>5.7000E+01</td>
<td>2.4400E+02</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>diastbp</td>
<td>s</td>
<td>6.0000E+00</td>
<td>1.9800E+02</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>bmi</td>
<td>1.3045E+01</td>
<td>4.4839E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>cvd</td>
<td>c</td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

Wei-Yin Loh
5.7 Censored response data

9 afb c 2
10 sho c 2
11 chf c 2
12 av3 c 2
13 miord c 2
14 mitype c 2
15 year c 3
16 los s 0.0000E+00 4.7000E+01
21 lenfol t 1.0000E+00 2.3580E+03
22 fstat d 0.0000E+00 1.0000E+00

=================== Constructed variables ===================
23 lnbasehaz z -4.1352E+00 9.7549E-01

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
500 0 0 5 0 0 6 0 9

Survival time variable in column: 21
Censoring indicator variable in column: 22
Proportion of uncensored among nonmissing T and D variables: 0.430
No. cases used for training: 500

Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000
Max number of split levels = 10
Minimum node size = 3
Number of iterations = 5
Number of SE’s for pruned tree = 5.0000E-01

Size and CV Loss and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean Loss</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Loss</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>1.579E+00</td>
<td>1.039E-01</td>
<td>7.117E-02</td>
<td>1.630E+00</td>
<td>8.633E-02</td>
</tr>
<tr>
<td>2</td>
<td>59</td>
<td>1.579E+00</td>
<td>1.039E-01</td>
<td>7.132E-02</td>
<td>1.632E+00</td>
<td>8.726E-02</td>
</tr>
<tr>
<td>3</td>
<td>58</td>
<td>1.574E+00</td>
<td>1.037E-01</td>
<td>7.088E-02</td>
<td>1.629E+00</td>
<td>8.560E-02</td>
</tr>
<tr>
<td>4</td>
<td>56</td>
<td>1.575E+00</td>
<td>1.037E-01</td>
<td>7.134E-02</td>
<td>1.629E+00</td>
<td>8.698E-02</td>
</tr>
<tr>
<td>5</td>
<td>55</td>
<td>1.576E+00</td>
<td>1.036E-01</td>
<td>7.085E-02</td>
<td>1.623E+00</td>
<td>8.370E-02</td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>1.565E+00</td>
<td>1.021E-01</td>
<td>7.149E-02</td>
<td>1.609E+00</td>
<td>8.418E-02</td>
</tr>
<tr>
<td>7</td>
<td>53</td>
<td>1.570E+00</td>
<td>1.021E-01</td>
<td>6.965E-02</td>
<td>1.609E+00</td>
<td>8.278E-02</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>1.573E+00</td>
<td>1.024E-01</td>
<td>6.999E-02</td>
<td>1.609E+00</td>
<td>8.370E-02</td>
</tr>
<tr>
<td>9</td>
<td>49</td>
<td>1.578E+00</td>
<td>1.023E-01</td>
<td>7.098E-02</td>
<td>1.609E+00</td>
<td>8.634E-02</td>
</tr>
<tr>
<td>10</td>
<td>48</td>
<td>1.574E+00</td>
<td>1.023E-01</td>
<td>7.246E-02</td>
<td>1.609E+00</td>
<td>8.744E-02</td>
</tr>
<tr>
<td>11</td>
<td>46</td>
<td>1.574E+00</td>
<td>1.023E-01</td>
<td>7.246E-02</td>
<td>1.609E+00</td>
<td>8.744E-02</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>1.569E+00</td>
<td>1.021E-01</td>
<td>7.195E-02</td>
<td>1.609E+00</td>
<td>8.400E-02</td>
</tr>
<tr>
<td>13</td>
<td>44</td>
<td>1.569E+00</td>
<td>1.018E-01</td>
<td>6.850E-02</td>
<td>1.609E+00</td>
<td>8.305E-02</td>
</tr>
<tr>
<td>14</td>
<td>43</td>
<td>1.561E+00</td>
<td>1.014E-01</td>
<td>6.161E-02</td>
<td>1.604E+00</td>
<td>8.005E-02</td>
</tr>
</tbody>
</table>

Wei-Yin Loh 88 GUIDE manual
### 5.7 Censored response data

#### REGRESSION

<table>
<thead>
<tr>
<th>Case</th>
<th>Value</th>
<th>Deviance</th>
<th>Rel. Risk</th>
<th>Split Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>42</td>
<td>1.560E+00</td>
<td>6.124E-02</td>
<td>1.604E+00</td>
</tr>
<tr>
<td>16</td>
<td>38</td>
<td>1.551E+00</td>
<td>6.017E-02</td>
<td>1.593E+00</td>
</tr>
<tr>
<td>17</td>
<td>37</td>
<td>1.556E+00</td>
<td>6.525E-02</td>
<td>1.558E+00</td>
</tr>
<tr>
<td>18</td>
<td>33</td>
<td>1.554E+00</td>
<td>6.493E-02</td>
<td>1.558E+00</td>
</tr>
<tr>
<td>19</td>
<td>30</td>
<td>1.553E+00</td>
<td>6.458E-02</td>
<td>1.558E+00</td>
</tr>
<tr>
<td>20</td>
<td>29</td>
<td>1.520E+00</td>
<td>9.861E-02</td>
<td>7.153E-02</td>
</tr>
<tr>
<td>21</td>
<td>28</td>
<td>1.508E+00</td>
<td>9.803E-02</td>
<td>7.192E-02</td>
</tr>
<tr>
<td>22</td>
<td>26</td>
<td>1.504E+00</td>
<td>9.723E-02</td>
<td>7.143E-02</td>
</tr>
<tr>
<td>23</td>
<td>25</td>
<td>1.501E+00</td>
<td>9.733E-02</td>
<td>7.324E-02</td>
</tr>
<tr>
<td>24</td>
<td>23</td>
<td>1.483E+00</td>
<td>9.532E-02</td>
<td>7.663E-02</td>
</tr>
<tr>
<td>25</td>
<td>21</td>
<td>1.478E+00</td>
<td>9.519E-02</td>
<td>7.655E-02</td>
</tr>
<tr>
<td>26</td>
<td>20</td>
<td>1.472E+00</td>
<td>9.488E-02</td>
<td>7.686E-02</td>
</tr>
<tr>
<td>27</td>
<td>19</td>
<td>1.437E+00</td>
<td>9.287E-02</td>
<td>8.809E-02</td>
</tr>
<tr>
<td>28</td>
<td>18</td>
<td>1.389E+00</td>
<td>8.985E-02</td>
<td>8.560E-02</td>
</tr>
<tr>
<td>29</td>
<td>12</td>
<td>1.382E+00</td>
<td>8.916E-02</td>
<td>8.459E-02</td>
</tr>
<tr>
<td>30</td>
<td>11</td>
<td>1.378E+00</td>
<td>9.014E-02</td>
<td>8.603E-02</td>
</tr>
<tr>
<td>31</td>
<td>9</td>
<td>1.329E+00</td>
<td>8.411E-02</td>
<td>6.501E-02</td>
</tr>
<tr>
<td>32</td>
<td>7</td>
<td>1.265E+00</td>
<td>7.544E-02</td>
<td>4.533E-02</td>
</tr>
<tr>
<td>33</td>
<td>6</td>
<td>1.238E+00</td>
<td>7.306E-02</td>
<td>3.864E-02</td>
</tr>
<tr>
<td>34</td>
<td>5</td>
<td>1.243E+00</td>
<td>7.171E-02</td>
<td>3.991E-02</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>1.281E+00</td>
<td>7.154E-02</td>
<td>4.064E-02</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>1.300E+00</td>
<td>7.015E-02</td>
<td>3.646E-02</td>
</tr>
<tr>
<td>37</td>
<td>2</td>
<td>1.325E+00</td>
<td>6.506E-02</td>
<td>2.839E-02</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>1.503E+00</td>
<td>5.698E-02</td>
<td>2.544E-02</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Rel. risk is mean risk relative to sample average ignoring covariates

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node

<table>
<thead>
<tr>
<th>Node</th>
<th>Total Cases</th>
<th>Cases Matrix</th>
<th>Node Split</th>
<th>Other Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>500</td>
<td>1</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>2</td>
<td>244</td>
<td>244</td>
<td>1</td>
<td>3.726E-01</td>
</tr>
</tbody>
</table>

Wei-Yin Loh 89 GUIDE manual
5.7 Censored response data

Regression tree:

Node 1: age <= 71.00000
   Node 2: chf = "1"
      Node 4: Risk relative to sample average ignoring covariates = 1.10956
      Node 2: chf /= "1"
      Node 5: Risk relative to sample average ignoring covariates = 0.21235
   Node 1: age > 71.00000 or NA
      Node 3: chf = "1"
      Node 6: Risk relative to sample average ignoring covariates = 3.02760
      Node 3: chf /= "1"
      Node 7: age <= 85.00000 or NA
         Node 14: Risk relative to sample average ignoring covariates = 1.06334
      Node 7: age > 85.00000 and not NA
         Node 15: Risk relative to sample average ignoring covariates = 3.32215

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if age <= 7.1000000E+01
   age mean = 6.9846E+01
Coefficients of log-relative risk function:
Regressors Coefficient t-stat p-val
Constant -3.5381E-02 -0.52 0.6041
Predicted relative risk = 1.0000000000000000

-----------------------------

Node 2: Intermediate node
A case goes into Node 4 if chf = "1"
   chf mode = "0"

-----------------------------

Node 4: Terminal node
Coefficients of log-relative risk function:
Regressors Coefficient t-stat p-val
Constant 6.8580E-02 0.34 0.7332
Predicted relative risk = 1.1095574995429367
----------------------------
Node 5: Terminal node
Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val
Constant \(-1.5849E+00\) \(-7.43\) 0.0000
Predicted relative risk = 0.21235168812700622
----------------------------
Node 3: Intermediate node
A case goes into Node 6 if chf = "1"
chf mode = "0"
----------------------------
Node 6: Terminal node
Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val
Constant \(1.0724E+00\) \(9.89\) 0.0000
Predicted relative risk = 3.0276015801608627
----------------------------
Node 7: Intermediate node
A case goes into Node 14 if age <= 8.5000000E+01 or NA
age mean = 8.0667E+01
----------------------------
Node 14: Terminal node
Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val
Constant \(2.6029E-02\) \(0.19\) 0.8459
Predicted relative risk = 1.0633351387096228
----------------------------
Node 15: Terminal node
Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val
Constant \(1.1652E+00\) \(6.05\) 0.0000
Predicted relative risk = 3.3221527399879980

Observed and fitted values are stored in whas500.fit
LaTeX code for tree is in whas500.tex
Elapsed time in seconds: 7.85750771

The tree model, given in Figure 13, shows that risk of death is lowest (0.21 relative to the sample average for the whole data set) for those younger than 72 with no congestive heart complications. The groups with the highest risks (3.0–3.32 relative to average) are those older than 85 and those between 72 and 85 with congestive heart complications.
5.7 Censored response data

Figure 13: GUIDE 0.50-SE piecewise constant relative risk regression tree for predicting \textit{fstat}. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘$\leq$’ stands for ‘$\leq$ or missing’. Sample sizes and mean relative risks (relative to sample average ignoring covariates) are printed below nodes.

The top 8 lines of the file \textit{whas500.fit} and its column definitions are:

\begin{verbatim}
train node survivaltime logbasecumhaz relativerisk survivalprob mediansurvtime
y 14 2.178000E+03 -7.667985E-02 1.063335E+00 3.865048E-01 1.553841E+03
y 5 2.172000E+03 -7.667985E-02 2.123517E-01 8.270912E-01 2.354932E+03
y 5 2.190000E+03 -7.667985E-02 2.123517E-01 8.270912E-01 2.354932E+03
y 4 2.970000E+02 -1.320296E+00 1.109557E+00 7.512523E-01 1.534972E+03
y 5 2.131000E+03 -2.213734E-01 2.123517E-01 8.485159E-01 2.354932E+03
y 5 1.000000E+00 -4.352824E+00 2.123517E-01 9.973654E-01 2.354932E+03
y 5 2.122000E+03 -2.213734E-01 2.123517E-01 8.485159E-01 2.354932E+03
y 5 1.496000E+03 -4.919833E-01 2.123517E-01 8.822135E-01 2.354932E+03
\end{verbatim}

\textit{train}: “y” if the observation is used for model fitting, “n” if not.

\textit{node}: terminal node label of observation.

\textit{survivaltime}: observed survival time \( t \).

\textit{logbasecumhaz}: \( \log \text{ of the estimated baseline cumulative hazard function } \log \Lambda_0(t) = \int_0^t \lambda_0(u) \, du \) at observed time \( t \).
relativerisk: $\exp(\mathbf{\beta}' \mathbf{x} - \mathbf{\beta}_s)$, risk of death relative to the average for the sample, where $\mathbf{x}$ is the covariate vector of the observation, $\mathbf{\beta}$ is the estimated regression coefficient vector in the node, and $\mathbf{\beta}_s$ is the coefficient in the constant model $\lambda_0(t) \exp(\mathbf{\beta}_s)$ fitted to all the training cases in the root node. Because a constant is fitted to each node here, $\mathbf{\beta}_s = -0.035381$ is the value of $\mathbf{\beta}$ at the root node. For example, the first subject, which is in node 14, has $\mathbf{\beta} = 0.026029$ and so $\text{relativerisk} = \exp(\mathbf{\beta} - \mathbf{\beta}_s) = \exp(0.026029 - 0.035381) = 1.063335$.

survivalprob: probability that the subject survives up to observed time $t$. For the first subject, this is

$$
\exp\{-\Lambda_0(t) \exp(\mathbf{\beta}' \mathbf{x})\} = \exp\{-\exp(\mathbf{\beta}_s + \log\text{basecumhaz}) \times \text{relativerisk}\} \\
= \exp(-\exp(-0.035381 - 0.07667985) \times 1.063335) \\
= 0.3865048.
$$

mediansurvtime: estimated median survival time $t$ such that $\exp\{-\Lambda_0(t) \exp(\mathbf{\beta}' \mathbf{x})\} = 0.5$, or, equivalently, $\Lambda_0(t) \exp(\mathbf{\beta}' \mathbf{x}) = -\log(0.5)$, or $\log(\text{basecumhaz}(t)) = \log \log(2) - \mathbf{\beta}' \mathbf{x}$, using linear interpolation of $\Lambda_0(t)$. Median survival times greater than the largest observed time have a trailing plus (+) sign. Figure 14 shows plots of $\log \Lambda_0(t)$ and $\Lambda_0(t)$ for this data set.

![Figure 14: Plots of estimated baseline cumulative hazard function](image)
5.8 Multi-response data

GUIDE can fit a piecewise-constant regression model for two or more dependent variables simultaneously. Following is an example from Loh and Zheng (2013) on estimating the strength and viscosity of concrete. The comma-delimited data file concrete.csv is from Yeh (2007). The data description file is below. Notice that there are three D variables. Our goal is to construct a single regression tree that predicts all three D variables simultaneously.

concrete.csv
NA
c1 c2 c3
1 No x
2 Cement n
3 Slag n
4 FlyAsh n
5 Water n
6 SP n
7 CoarseAggr n
8 FineAggr n
9 Slump d
10 Flow d
11 Strength d

Following is an annotated log of the input file creation.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning
Input your choice: 1
Name of batch input file: concrete.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: concrete.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Option 5 is for multiresponse data.
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): concretedsc.txt
Reading data description file...
Training data description file ...
5.8 Multi-response data

Missing value code: NA
Warning: N variables changed to S
Number of D variables = 3
D variables are:
Slump
Flow
Strength

Choose multivariate or univariate split variable selection:
Choose multivariate if fewer than 5 D variables, choose univariate otherwise
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1):
The D vector can be grouped into segments to look for patterns
Input 0 for no grouping, 1 for roughly equal groups, 2 for other choices
Input your selection ([0:2], <cr>=0):
Grouping is recommended if there are numerous D variables and they are clustered.
Here, Slump and Flow may be considered as belonging to one cluster and Strength to another, but in this illustration, we treat them as separate.
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):
Normalization means scaling each variable to have variance 1.
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Length of longest data entry = 6
Total number of cases = 103
Checking data ...

#cases w/ miss. D = number of cases with all D values missing
<table>
<thead>
<tr>
<th>Total</th>
<th>#cases w/</th>
<th>#missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#cases</td>
<td>miss. D</td>
<td>ord. vals</td>
</tr>
<tr>
<td>103</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

No. cases used for training = 103
Warning: interaction tests skipped
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): concrete.tex
Input 2 to save node IDs of cases, 1 otherwise ([1:2], <cr>=1): 1
Input 2 to save node fitted values; 1 otherwise ([1:2], <cr>=1): 1
Input file is created!

Results

Multi-response or longitudinal data without T variables
Pruning by cross-validation
Data description file: concretedsc.txt
Training sample file: concrete.csv
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 3
Multivariate split variable selection method
Missing D values treated as below average for variable selection
No grouping of D variables
Segment boundaries are:
1.500  2.500

*GUIDE* labels the D variables as $Y_1, Y_2, \ldots$ (in the order of their appearance in the data file). The segment boundaries refer to the grouping of the indices. In this case, since there is no grouping, each variable is its own group. As a result, the indices are grouped into intervals $(0.5, 1.5), (1.5, 2.5), (2.5, 3.5)$.

Mean-squared errors (MSE) are calculated from normalized D variables

This is a reminder that the D variables are normalized.

D variables equally weighted

Piecewise constant model

Length of longest data entry = 6

Summary information (without x variables)

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Type</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Number of</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>name</td>
<td></td>
<td>value</td>
<td>value</td>
<td>categories</td>
<td>missing</td>
</tr>
<tr>
<td>2</td>
<td>Cement</td>
<td>s</td>
<td>1.3700E+02</td>
<td>3.7400E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Slag</td>
<td>s</td>
<td>0.0000E+00</td>
<td>1.9300E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FlyAsh</td>
<td>s</td>
<td>0.0000E+00</td>
<td>2.6000E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Water</td>
<td>s</td>
<td>1.6000E+02</td>
<td>2.4000E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SP</td>
<td>s</td>
<td>4.4000E+00</td>
<td>1.9000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CoarseAggr</td>
<td>s</td>
<td>7.0800E+02</td>
<td>1.0499E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FineAggr</td>
<td>s</td>
<td>6.4060E+02</td>
<td>9.0200E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Slump</td>
<td>d</td>
<td>0.0000E+00</td>
<td>2.9000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Flow</td>
<td>d</td>
<td>2.0000E+01</td>
<td>7.8000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Strength</td>
<td>d</td>
<td>1.7190E+01</td>
<td>5.8530E+01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#cases w/ miss. D = number of cases with all D values missing

Total #cases w/ missing

<table>
<thead>
<tr>
<th>#cases</th>
<th>miss. D</th>
<th>ord. vals</th>
<th>X-var</th>
<th>N-var</th>
<th>F-var</th>
<th>S-var</th>
<th>B-var</th>
<th>C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

No. cases used for training = 103

Warning: interaction tests skipped

No interaction tests

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Split values for N and S variables based on exhaustive search

Max number of split levels = 10

Minimum node size = 10

Number of SE’s for pruned tree = 5.0000E-01

Size and CV Loss and SE of subtrees:
5.8 Multi-response data

Regression tree for multi-response data:

Node 1: Water <= 1.82250E+02
Node 2: Cement <= 1.61000E+02
Node 4: Mean cost = 4.15924E-01

Wei-Yin Loh

GUIDE manual
**Node 2:** Cement $> 1.61000E+02$ or NA  
**Node 5:** Mean cost = $9.95311E-01$

**Node 1:** Water $> 1.82250E+02$ or NA  
**Node 3:** Slag $\leq 1.35000E+02$ or NA  
**Node 6:** Cement $\leq 1.57100E+02$

- **Node 12:** Mean cost = $1.76946E-01$
- **Node 6:** Cement $> 1.57100E+02$ or NA
  - **Node 13:** FlyAsh $\leq 1.17500E+02$
    - **Node 26:** Mean cost = $3.07784E-01$
  - **Node 13:** FlyAsh $> 1.17500E+02$ or NA
    - **Node 27:** Mean cost = $4.31499E-01$

- **Node 3:** Slag $> 1.35000E+02$ and not NA
- **Node 7:** Mean cost = $9.61024E-01$

***************************************************************

**Node 1:** Intermediate node  
A case goes into Node 2 if Water $\leq 1.8225000E+02$  
Water mean = $1.9717E+02$

Estimated D values are:  
$1.8049E+01$  $4.9611E+01$  $3.6039E+01$

----------------------------

**Node 2:** Intermediate node  
A case goes into Node 4 if Cement $\leq 1.6100000E+02$  
Cement mean = $1.9275E+02$

----------------------------

**Node 4:** Terminal node  
Estimated D values are:  
$1.5286E+01$  $4.0536E+01$  $3.5470E+01$

*The three numbers are the sample mean values of Slump, Flow and Strength, in order of their appearance in the data file.*

----------------------------

**Node 5:** Terminal node  
Estimated D values are:  
$5.6667E+00$  $2.5347E+01$  $4.4189E+01$

----------------------------

**Node 3:** Intermediate node  
A case goes into Node 6 if Slag $\leq 1.3500000E+02$ or NA  
Slag mean = $7.8059E+01$

----------------------------

**Node 6:** Intermediate node  
A case goes into Node 12 if Cement $\leq 1.5710000E+02$  
Cement mean = $2.5277E+02$

----------------------------

**Node 12:** Terminal node  
Estimated D values are:
2.2083E+01  5.8389E+01  2.9007E+01
----------------------------
Node 13: Intermediate node
A case goes into Node 26 if FlyAsh <= 1.1750000E+02
FlyAsh mean = 1.1328E+02
----------------------------
Node 26: Terminal node
Estimated D values are:
   2.0950E+01  5.4325E+01  3.1665E+01
----------------------------
Node 27: Terminal node
Estimated D values are:
   2.2712E+01  6.1450E+01  4.2456E+01
----------------------------
Node 7: Terminal node
Estimated D values are:
   1.5300E+01  4.2700E+01  2.9337E+01
----------------------------
Case and node IDs are in file: node.txt
Node fitted values are in file: fit.txt
Observed and fitted values are stored in node.txt
LaTeX code for tree is in tree.tex
Elapsed time in seconds: 3.15099992E-02

The \LaTeX{} tree is shown in Figure 15.

5.9 Longitudinal data with irregular time points

The data come from a longitudinal study on the hourly wage of 888 male high-
school dropouts (246 black, 204 Hispanic, 438 white), where the observation time
points as well as their number (1–13) varied across individuals (Murnane et al., 1999;
Singer and Willett, 2003). An earlier version of GUIDE was used to analyze these

The response variable is hourly wage (in 1990 dollars) and the predictor variables
are $\text{hgc}$ (highest grade completed; 6–12), $\text{exper}$ (years in labor force; 0.001–12.7
yrs), and $\text{race}$ (Black, Hispanic, and White). The data file $\text{widewage.txt}$ is in wide
format, where each record refers to one individual. The description file $\text{wage.dsc}$ is
given below. Note that observation time points are marked as $t$.

\begin{verbatim}
widewage.txt
NA
c1 c2 c3
1 id x
\end{verbatim}

Wei-Yin Loh 99
GUIDE manual
Figure 15: GUIDE 0.50-SE regression tree for predicting response variables Slump, Flow, and Strength. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘≤∗’ stands for ‘≤ or missing’. Sample sizes and predicted values of Slump, Flow, and Strength are printed below nodes.
5.9 Longitudinal data

2 hgc n
3 exper1 t
4 exper2 t
5 exper3 t
6 exper4 t
7 exper5 t
8 exper6 t
9 exper7 t
10 exper8 t
11 exper9 t
12 exper10 t
13 exper11 t
14 exper12 t
15 exper13 t
16 postexp1 x
17 postexp2 x
18 postexp3 x
19 postexp4 x
20 postexp5 x
21 postexp6 x
22 postexp7 x
23 postexp8 x
24 postexp9 x
25 postexp10 x
26 postexp11 x
27 postexp12 x
28 postexp13 x
29 wage1 d
30 wage2 d
31 wage3 d
32 wage4 d
33 wage5 d
34 wage6 d
35 wage7 d
36 wage8 d
37 wage9 d
38 wage10 d
39 wage11 d
40 wage12 d
41 wage13 d
42 ged1 x
43 ged2 x
44 ged3 x
45 ged4 x
46 ged5 x
47 ged6 x
5.9 Longitudinal data

Because the default 0.5-SE rule yields a trivial tree with no splits, we show how
the options can be changed to produce a tree with the 0-SE rule. Following is a
session log.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning
Input your choice: 1
Name of batch input file: wage.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: wage.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiprogression or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 6
Input 1 for lowess smoothing, 2 for spline smoothing ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Choosing 1 will produce a 0.5-SE tree. We choose 2 to allow more options.
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):
Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): wage.dsc
Reading data description file ...
Training sample file: widewage.txt
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 13
Number of D variables = 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
The D variables can be grouped into segments to look for patterns
Input 1 for roughly equal-sized groups, 2 for customized groups ([1:2], <cr>=1):
Input number of roughly equal-sized groups ([2:9], <cr>=3):
Input number of interpolating points for prediction ([10:100], <cr>=31):
Length of longest data entry = 16
Total number of cases = 888
Col. no. Categorical variable #levels #missing values
  68 race 3 0
Checking data ...
#cases w/ miss. D = number of cases with all D values missing

<table>
<thead>
<tr>
<th></th>
<th>Total #cases w/ #missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#cases</td>
<td>888</td>
</tr>
<tr>
<td>miss. D</td>
<td>0</td>
</tr>
<tr>
<td>ord. vals</td>
<td>0</td>
</tr>
<tr>
<td>#X-var</td>
<td>40</td>
</tr>
<tr>
<td>#N-var</td>
<td>0</td>
</tr>
<tr>
<td>#F-var</td>
<td>0</td>
</tr>
<tr>
<td>#S-var</td>
<td>1</td>
</tr>
<tr>
<td>#B-var</td>
<td>0</td>
</tr>
<tr>
<td>#C-var</td>
<td>1</td>
</tr>
</tbody>
</table>

No. cases used for training = 888
No. cases excluded due to 0 weight or missing D = 0
Warning: interaction tests skipped
Default number of cross-validations = 10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50): 0
   This is where we choose the 0-SE pruning rule.
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max number of split levels = 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 44
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): wage.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Choose a color for the terminal nodes:
(1) white
(2) lightgray
(3) gray
(4) darkgray
(5) black
(6) yellow
(7) red
(8) blue
(9) green
(10) magenta
(11) cyan
Input your choice ([1:11], <cr>=6):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables, 3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save node IDs of cases, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store terminal node IDs: wage.nid
Input 2 to save node fitted values; 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node fitted values: wage.fit
Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!

Results

Lowess smoothing
Longitudinal data with T variables
Pruning by cross-validation
Data description file: wagedsc.txt
Training sample file: wagedat.txt
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 13
Number of D variables = 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
Length of longest data entry = 16
Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>#Categories</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>hgc</td>
<td>0.0000E+00</td>
<td>1.2000E+01</td>
<td>6.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>3</td>
<td>exper1</td>
<td>0.0000E+00</td>
<td>9.7770E+00</td>
<td>1.0000E+03</td>
<td>5.6370E+00</td>
</tr>
<tr>
<td>4</td>
<td>exper2</td>
<td>0.0000E+00</td>
<td>1.0815E+01</td>
<td>7.5840E+00</td>
<td>1.1777E+01</td>
</tr>
<tr>
<td>5</td>
<td>exper3</td>
<td>0.0000E+00</td>
<td>1.1777E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>6</td>
<td>exper4</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>7</td>
<td>exper5</td>
<td>0.0000E+00</td>
<td>1.5873E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>8</td>
<td>exper6</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>9</td>
<td>exper7</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>10</td>
<td>exper8</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>11</td>
<td>exper9</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>12</td>
<td>exper10</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>13</td>
<td>exper11</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>14</td>
<td>exper12</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>15</td>
<td>exper13</td>
<td>0.0000E+00</td>
<td>1.2799E+01</td>
<td>0.0000E+00</td>
<td>7.5840E+00</td>
</tr>
<tr>
<td>29</td>
<td>wage1</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.0299E+00</td>
<td>6.8649E+01</td>
</tr>
<tr>
<td>30</td>
<td>wage2</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.0689E+00</td>
<td>5.0400E+01</td>
</tr>
<tr>
<td>31</td>
<td>wage3</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.0462E+00</td>
<td>3.4501E+01</td>
</tr>
<tr>
<td>32</td>
<td>wage4</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1170E+00</td>
<td>3.3149E+01</td>
</tr>
<tr>
<td>33</td>
<td>wage5</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>34</td>
<td>wage6</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>35</td>
<td>wage7</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>36</td>
<td>wage8</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>37</td>
<td>wage9</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>38</td>
<td>wage10</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>39</td>
<td>wage11</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>40</td>
<td>wage12</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>41</td>
<td>wage13</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
<tr>
<td>68</td>
<td>race</td>
<td>0.0000E+00</td>
<td>1.2260E+01</td>
<td>2.1043E+00</td>
<td>4.9304E+01</td>
</tr>
</tbody>
</table>

Total cases w/ missing
<table>
<thead>
<tr>
<th>#cases</th>
<th>miss. D ord. vals</th>
<th>#X-var</th>
<th>#N-var</th>
<th>#F-var</th>
<th>#S-var</th>
<th>#B-var</th>
<th>#C-var</th>
</tr>
</thead>
<tbody>
<tr>
<td>888</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

No. cases used for training: 888
No. cases excluded due to 0 weight or missing D: 0

Warning: interaction tests skipped
No interaction tests
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Split values for N and S variables based on exhaustive search
Max number of split levels = 10
5.9 Longitudinal data

Minimum node size = 44
Number of SE’s for pruned tree = 0.0000E+00

Size and CV Loss and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean Loss</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Loss</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>1.257E+02</td>
<td>1.044E+01</td>
<td>8.502E+00</td>
<td>1.204E+02</td>
<td>1.525E+01</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>1.257E+02</td>
<td>1.044E+01</td>
<td>8.502E+00</td>
<td>1.204E+02</td>
<td>1.525E+01</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>1.257E+02</td>
<td>1.044E+01</td>
<td>8.502E+00</td>
<td>1.204E+02</td>
<td>1.525E+01</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1.242E+02</td>
<td>1.049E+01</td>
<td>8.463E+00</td>
<td>1.181E+02</td>
<td>1.537E+01</td>
</tr>
<tr>
<td>4**</td>
<td>5</td>
<td>1.238E+02</td>
<td>1.058E+01</td>
<td>8.434E+00</td>
<td>1.175E+02</td>
<td>1.530E+01</td>
</tr>
<tr>
<td>5++</td>
<td>1</td>
<td>1.244E+02</td>
<td>1.064E+01</td>
<td>8.700E+00</td>
<td>1.157E+02</td>
<td>1.577E+01</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
+ tree same as ++ tree
* tree same as ** tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node

<table>
<thead>
<tr>
<th>Node label</th>
<th>Total cases</th>
<th>Cases fit</th>
<th>MSE</th>
<th>variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>888</td>
<td>888</td>
<td>1.222E+02</td>
<td>race</td>
</tr>
<tr>
<td>2T</td>
<td>246</td>
<td>246</td>
<td>1.111E+02</td>
<td>hgc</td>
</tr>
<tr>
<td>3</td>
<td>642</td>
<td>642</td>
<td>1.259E+02</td>
<td>race</td>
</tr>
<tr>
<td>6</td>
<td>204</td>
<td>204</td>
<td>1.278E+02</td>
<td>hgc</td>
</tr>
<tr>
<td>12T</td>
<td>127</td>
<td>127</td>
<td>1.085E+02</td>
<td>-</td>
</tr>
<tr>
<td>13T</td>
<td>77</td>
<td>77</td>
<td>1.514E+02</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>438</td>
<td>438</td>
<td>1.252E+02</td>
<td>hgc</td>
</tr>
<tr>
<td>14T</td>
<td>299</td>
<td>299</td>
<td>9.813E+01</td>
<td>hgc</td>
</tr>
<tr>
<td>15T</td>
<td>139</td>
<td>139</td>
<td>1.777E+02</td>
<td>hgc</td>
</tr>
</tbody>
</table>

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9

Regression tree for longitudinal data:

Wei-Yin Loh

107

GUIDE manual
Node 1: race = "black"
    Node 2: Mean cost = 1.10602E+02
Node 1: race /= "black"
    Node 3: race = "hispanic"
        Node 6: hgc <= 9.50000 or NA
            Node 12: Mean cost = 1.07621E+02
        Node 6: hgc > 9.50000 and not NA
            Node 13: Mean cost = 1.49412E+02
    Node 3: race /= "hispanic"
        Node 7: hgc <= 9.50000 or NA
            Node 14: Mean cost = 9.78002E+01
        Node 7: hgc > 9.50000 and not NA
            Node 15: Mean cost = 1.76394E+02

***************************************************************

Node 1: Intermediate node
A case goes into Node 2 if race = "black"
    race node = "white"
                        -------------------------------
Node 2: Terminal node
                        -------------------------------
Node 3: Intermediate node
A case goes into Node 6 if race = "hispanic"
    race node = "white"
                        -------------------------------
Node 6: Intermediate node
A case goes into Node 12 if hgc <= 9.500000E+00 or NA
    hgc mean = 8.9118E+00
                        -------------------------------
Node 12: Terminal node
                        -------------------------------
Node 13: Terminal node
                        -------------------------------
Node 7: Intermediate node
A case goes into Node 14 if hgc <= 9.500000E+00 or NA
    hgc mean = 8.8973E+00
                        -------------------------------
Node 14: Terminal node
                        -------------------------------
Node 15: Terminal node
                        -------------------------------

Case and node IDs are in file: wage.nid
Node fitted values are in file: wage.fit

Wei-Yin Loh 108 GUIDE manual
Figure 16: GUIDE 0.00-SE regression tree for predicting longitudinal variables \(wage_1, wage_2, \ldots\). At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ‘\(\leq\)’ stands for ‘\(\leq\) or missing’. Sample sizes are printed below nodes.

Observed and fitted values are stored in wage.nid
LaTeX code for tree is in wage.tex
Elapsed time in seconds: 2.98005509

Figure 16 shows the tree and Figure 17 plots lowess-smoothed curves of mean wage in the two terminal nodes. The plotting values are obtained from the result file wage.fit whose contents are given below. The first column gives the node number and the next two columns the start and end of the times at which fitted values are computed. The other columns give the fitted values equally spaced between the start and end times.

node t.start t.end fitted1 fitted2 fitted3 fitted4 fitted5 fitted6 fitted7 fitted8 fitted9 fitted10
2 \(0.40000E-02\) \(0.12556E+02\) 0.50794E+01 0.52623E+01 0.54112E+01 0.55477E+01 0.56649E+01 0.57545E+01
12 \(0.60000E-02\) \(0.12700E+02\) 0.50837E+01 0.52324E+01 0.53388E+01 0.55076E+01 0.56314E+01 0.57727E+01
13 \(0.12200E+00\) \(0.11990E+02\) 0.56361E+01 0.58877E+01 0.61037E+01 0.62417E+01 0.64134E+01 0.65516E+01
14 \(0.10000E+00\) \(0.12700E+02\) 0.56387E+01 0.58837E+01 0.61037E+01 0.62417E+01 0.64134E+01 0.65516E+01
15 \(0.20000E-02\) \(0.12045E+02\) 0.57487E+01 0.59247E+01 0.60944E+01 0.62441E+01 0.64134E+01 0.65516E+01

Wei-Yin Loh 109 GUIDE manual
Figure 17: Lowess-smoothed mean wage curves in the terminal nodes of Figure 16.
5.10 Subgroup identification

If there is a treatment variable in the data, GUIDE can fit a tree model find subgroups with differential treatment effects. The dependent variable can be censored or not.

1. The treatment variable is designated as \( R \).

2. If there is no censoring, the response variable is designated as \( D \) as usual.

3. If there is censoring, then the survival time is designated as \( T \) and the event (typically death) indicator is designated as \( D \), taking value 1 for death and 0 for censored.

GUIDE has two methods for solving this problem, called \( gi \) and \( gs \). The former is more sensitive to predictive variables (i.e., variables that interact with treatment) and the latter is equally sensitive to prognostic and predictive variables. The methods are documented in Loh et al. (2015).

We illustrate the \( gi \) method with data from a breast cancer trial (Schmoor et al., 1996). The data are in the file cancer.txt from the ipred R package (Peters and Hothorn, 2012). In the description file cancerdsc.txt below, the treatment variable is hormone therapy, horTh. The variable time is censored survival time and death is the event indicator (1=death, 0=censored).

```
cancer.txt
NA
c1 c2 c3
1 horTh r
2 age s
3 menostat c
4 tsize s
5 tgrade c
6 pnodes s
7 progres c
8 estrec s
9 time t
10 death d
```

Input file generation

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning
Input your choice: 1
Name of batch input file: cancerin.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: cancerout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Whatever option chosen here will automatically be changed to multiple linear
when the program detects the presence of a ‘‘R’’ variable in the description file.

Input name of data description file (maximum 100 characters; enclose within quotes
if it contains spaces or non alphanumeric characters): cancerdsc.txt
Reading data description file ...
Training sample file: cancer.txt
Missing value code: NA
R variable present
Warning: model fit changed to linear in treatment variable
Dependent variable is death
Length of longest data entry = 4
Total number of cases = 686
Col. no. Categorical variable #levels #missing values
1 horTh 2 0
3 menostat 2 0
5 tgrade 3 0
Checking data ...
Smallest uncensored T = 72.000000000000000
No. cases dropped due to missing D or T or censored T < smallest uncensored T = 14
No. complete cases excluding censored T < smallest uncensored T = 672
No. cases used to compute baseline hazard = 672
No. cases with D=1 and T >= smallest uncensored = 299
The program will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Sum of chi-squares (Gs)
2 = Treatment interactions (Gi)
Input your choice ([1:2], <cr>=2):
Option 2 is generally more sensitive to detecting treatment interactions.
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var #R-var
Wei-Yin Loh 112 GUIDE manual
Survival time variable in column: 9
Censoring indicator variable in column: 10
Proportion of uncensored among nonmissing T and D variables = 0.445
No. cases used for training = 672
Warning: interaction tests skipped

Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): cancer.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node IDs and fitted values: cancer.fit
Input file is created!

Results  The following results show that the tree splits once on variable progres.

Proportional hazards regression with relative risk estimates
Pruning by cross-validation
Data description file: cancerdsc.txt
Training sample file: cancer.txt
Missing value code: NA
R variable present
Warning: model fit changed to linear in treatment variable
Dependent variable is death
Piecewise linear model
Length of longest data entry = 4
Smallest uncensored T = 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T = 14
No. complete cases excluding censored T < smallest uncensored T = 672
No. cases used to compute baseline hazard = 672
No. cases with D=1 and T >= smallest uncensored = 299
Number of dummy variables created = 1

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
t=survival time variable

For categorical variables, #categories include one for missing values

<table>
<thead>
<tr>
<th>Column number</th>
<th>Variable name</th>
<th>Type</th>
<th>Minimum value</th>
<th>Maximum value</th>
<th>Number of categories</th>
<th>Number of missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>horTh</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>age</td>
<td>s</td>
<td>2.1000E+01</td>
<td>8.0000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>menostat</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>tsize</td>
<td>s</td>
<td>3.0000E+00</td>
<td>1.2000E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>tgrade</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>pnodes</td>
<td>s</td>
<td>1.0000E+00</td>
<td>5.1000E+01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>progres</td>
<td>s</td>
<td>0.0000E+00</td>
<td>2.3800E+03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Wei-Yin Loh 113 GUIDE manual
5.10 Subgroup identification

8 estrec s 0.0000E+00 1.1440E+03
9 time t 7.2000E+01 2.6590E+03
10 death d 0.0000E+00 1.0000E+00

=================== Constructed variables ===================
11 lnbasehaz0 z -6.5103E+00 5.8866E-02
12 horTh.yes f 0.0000E+00 1.0000E+00

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
686 0 0 0 0 0 5 0 2

Survival time variable in column 9
Censoring indicator variable in column 10
Proportion of uncensored among nonmissing T and D variables = 0.445
No. cases used for training = 672

Warning: interaction tests skipped
Treatment interactions (Gi)
No interaction tests
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 1.0000
Max number of split levels = 10
Minimum node size = 34
Number of iterations = 5
No calibration needed: no N variables
Number of SE’s for pruned tree = 5.0000E-01

Size and CV Loss and SE of subtrees:

<table>
<thead>
<tr>
<th>Tree</th>
<th>#Tnodes</th>
<th>Mean Loss</th>
<th>SE(Mean)</th>
<th>BSE(Mean)</th>
<th>Median Loss</th>
<th>BSE(Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>1.453E+00</td>
<td>5.891E-02</td>
<td>4.830E-02</td>
<td>1.408E+00</td>
<td>5.079E-02</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>1.453E+00</td>
<td>5.885E-02</td>
<td>4.827E-02</td>
<td>1.408E+00</td>
<td>5.061E-02</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>1.441E+00</td>
<td>5.676E-02</td>
<td>4.165E-02</td>
<td>1.395E+00</td>
<td>5.337E-02</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>1.445E+00</td>
<td>5.640E-02</td>
<td>4.195E-02</td>
<td>1.405E+00</td>
<td>5.353E-02</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>1.444E+00</td>
<td>5.632E-02</td>
<td>4.201E-02</td>
<td>1.405E+00</td>
<td>5.235E-02</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1.445E+00</td>
<td>5.627E-02</td>
<td>4.149E-02</td>
<td>1.405E+00</td>
<td>5.190E-02</td>
</tr>
<tr>
<td>7**</td>
<td>2</td>
<td>1.408E+00</td>
<td>5.211E-02</td>
<td>3.686E-02</td>
<td>1.386E+00</td>
<td>3.980E-02</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1.442E+00</td>
<td>5.157E-02</td>
<td>1.216E-02</td>
<td>1.450E+00</td>
<td>1.474E-02</td>
</tr>
</tbody>
</table>

0-SE tree based on mean is marked with *
0-SE tree based on median is marked with +
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same
Following tree is based on mean CV with naive SE estimate (**).

Structure of final tree. Each terminal node is marked with a T.

Rel. risk is mean risk relative to sample average ignoring covariates.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node

<table>
<thead>
<tr>
<th>Node</th>
<th>Total</th>
<th>Cases</th>
<th>Matrix</th>
<th>Node</th>
<th>Node</th>
<th>Split</th>
</tr>
</thead>
<tbody>
<tr>
<td>label cases fit rank rel.risk deviance variable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>672</td>
<td>672</td>
<td>1</td>
<td>1.00E+00</td>
<td>1.414E+00</td>
<td>progrec</td>
</tr>
<tr>
<td>2T</td>
<td>274</td>
<td>274</td>
<td>1</td>
<td>1.588E+00</td>
<td>1.584E+00</td>
<td>menostat</td>
</tr>
<tr>
<td>3T</td>
<td>398</td>
<td>398</td>
<td>1</td>
<td>7.095E-01</td>
<td>1.172E+00</td>
<td>menostat</td>
</tr>
</tbody>
</table>

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3

Regression tree:

Node 1: progres <= 21.000000
    Node 2: Risk relative to sample average ignoring covariates = 1.58824
    Node 1: progres > 21.000000 or NA
        Node 3: Risk relative to sample average ignoring covariates = 0.70947

**************************************************************
Node 1: Intermediate node
A case goes into Node 2 if progres <= 2.100000E+01
progres mean = 1.1092E+02

Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val Min Mean Max
Constant 1.2903E-01 1.85 0.0651
horTh.yes -3.6984E-01 -2.97 0.0031 0.0000E+00 3.6012E-01 1.0000E+00
Predicted relative risk = 1.0000000000000000
-----------------------------
Node 2: Terminal node

Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val Min Mean Max
Constant 5.0439E-01 5.04 0.0000
horTh.yes -1.1775E-01 -0.71 0.4786 0.0000E+00 3.6131E-01 1.0000E+00
Predicted relative risk = 1.5882374130670138
-----------------------------
Node 3: Terminal node

Coefficients of log-relative risk function:
Regressor Coefficient t-stat p-val Min Mean Max
Constant -1.3184E-01 -1.35 0.1775

Wei-Yin Loh 115 GUIDE manual
Figure 18: GUIDE Gi proportional hazards regression tree for differential treatment effects. At each intermediate node, an observation goes to the left branch if and only if the condition is satisfied. Numbers beside terminal nodes are estimated relative risks (relative to average for sample ignoring covariates) corresponding to treatment levels no and yes; numbers below are sample sizes.

```
horTh.yes  -6.5011E-01  -3.40  0.0007  0.0000E+00  3.5930E-01  1.0000E+00
Predicted relative risk =  0.70947399586773086
```

Observed and fitted values are stored in cancer.fit
LaTeX code for tree is in cancer.tex
Elapsed time in seconds:  12.6887312

The \LaTeX tree diagram and the Kaplan-Meier survival functions estimated from the data in the terminal nodes of the tree are shown in Figures 18 and 19, respectively.

**Estimated relative risks and survival probabilities** The file cancer.fit gives the terminal node number, estimated survival time, log baseline cumulative hazard, relative risk (relative to the average for the data, ignoring covariates), survival probability, and median survival time of each observation in the training sample file cancer.txt. The results for the first few observations are shown below. See Section 5.7 for definitions of the terms.

```
train node survivaltime logbasecumhaz relativerisk survivalprob mediansurvtimetime
y 3  1.814000E+03  -3.317667E-01  8.787636E-01  2.014420E+03
y 3  2.018000E+03  -2.024282E-01  4.587030E-01  2.659000E+03
y 3  7.120000E+02  -1.300331E+00  4.587030E-01  2.659000E+03
y 3  1.807000E+03  -3.550694E-01  4.587030E-01  2.659000E+03
y 2  4.480000E+02  -2.105688E+00  1.660293E+00  1.038277E+03
```
5.11 Differential item functioning

GUIDE has an experimental option to identify important predictor variables and items with differential item functioning (DIF) in a data set with two or more item (dependent variable) scores. We illustrate it with a data set from Broekman et al. (2011, 2008) and Marc et al. (2008). It consists of responses from 1978 subjects on 15 items. There are 3 predictor variables (age, education, and gender). The data and description files are GDS.dat and GDS.dsc. Although the item responses in this example are 0-1, GUIDE allows them to be in any ordinal (e.g., Likert) scale. The contents of GDS.dsc are:

GDS.dat
NA
c1 c2 c3
1 SATIS d
2 SPIRIT d
3 HAPPY d
4 ALIVE d
5 ENERGY d
6 DROP d
7 EMPTY d
8 BORED d
9 AFRAID d
10 HELP d

Figure 19: Estimated survival probability functions for breast cancer data
Here is the session log to create an input file for identifying DIF items and the important predictor variables:

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1

Name of batch input file: GDSimp.in
Input 1 for model fitting, 2 for importance or DIF scoring,
3 for data conversion ([1:3], <cr>=1): 2

Name of batch output file: GDSimp.out
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2

Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5

Choose option 5 for item response data.
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: GDS.dsc
Reading data description file ...
Training sample file: GDS.dat
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 15
D variables are:
SATIS
SPIRIT
HAPPY
ALIVE
ENERGY
DROP
EMPTY
BORED
Choose multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables; otherwise choose univariate
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=2):
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=2):
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Length of longest data entry = 2
Total number of cases: 1978
Col. no. Categorical variable #levels #missing values
 16 MALE 2 0
Checking data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
Choose the default because DIF scoring is desired.

#cases w/ miss. D = number of cases with all D values missing
Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
1978 0 0 0 0 0 2 0 1
No. cases used for training: 1978
Warning: interaction tests skipped
Input expected fraction of noise variables erroneously selected ([0.00:0.99], <cr>=0.01):
Input 1 to save p-value matrix for differential item functioning (DIF),
2 otherwise ([1:2], <cr>=1):
Input file name to store DIF p-values: GDSimp.pv
You can create a description file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1): 2
Input 1 to keep only selected variables, 2 to exclude selected variables ([1:2], <cr>=1):
Input file name: GDSsub.dsc
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: GDSimp.scr
Input file is created!

The importance scores in the output file GDSimp.scr shows that all three predictor variables are distinguishable from noise (because their scores are above 1.0):

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.702</td>
<td>AGE</td>
</tr>
</tbody>
</table>
5.11 Differential item functioning

The last column of \texttt{GDSimp.pv} below shows that three items (\#4, 10, 13) have DIF.

<table>
<thead>
<tr>
<th>Item</th>
<th>Itemname</th>
<th>EDUCATION</th>
<th>AGE</th>
<th>GENDER</th>
<th>DIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SATIS</td>
<td>0.310E-01</td>
<td>0.946E-01</td>
<td>0.476E-01</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>SPIRIT</td>
<td>0.988E+00</td>
<td>0.456E+00</td>
<td>0.437E-02</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>HAPPY</td>
<td>0.938E+00</td>
<td>0.930E-01</td>
<td>0.375E-01</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>ALIVE</td>
<td>0.129E+00</td>
<td>0.282E-01</td>
<td>0.382E+00</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>ENERGY</td>
<td>0.721E+00</td>
<td>0.845E+00</td>
<td>0.573E-06</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>DROP</td>
<td>0.107E-01</td>
<td>0.117E+00</td>
<td>0.951E+00</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>EMPTY</td>
<td>0.369E-02</td>
<td>0.194E-02</td>
<td>0.315E-01</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>BORED</td>
<td>0.750E-07</td>
<td>0.166E+00</td>
<td>0.416E+00</td>
<td>yes</td>
</tr>
<tr>
<td>9</td>
<td>AFRAID</td>
<td>0.106E+00</td>
<td>0.323E-02</td>
<td>0.287E-02</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>HELP</td>
<td>0.928E-01</td>
<td>0.678E+00</td>
<td>0.148E-02</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>HOME</td>
<td>0.128E+00</td>
<td>0.826E+00</td>
<td>0.779E-03</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>MEMORY</td>
<td>0.434E+00</td>
<td>0.000E+00</td>
<td>0.440E-01</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>WORTH</td>
<td>0.934E+00</td>
<td>0.573E+00</td>
<td>0.624E+00</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>HOPE</td>
<td>0.653E+00</td>
<td>0.799E+00</td>
<td>0.109E+00</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>BETTER</td>
<td>0.956E+00</td>
<td>0.525E+00</td>
<td>0.747E+00</td>
<td>no</td>
</tr>
</tbody>
</table>

The following output file \texttt{GDSsub.dsc} can be used to fit a model to the selected item and predictor variables:

```
"GDS.dat"
"NA"
colnumber varname vartype
1 SATIS x
2 SPIRIT x
3 HAPPY x
4 ALIVE x
5 ENERGY d
6 DROP x
7 EMPTY x
8 BORED d
9 AFRAID x
10 HELP x
11 HOME x
12 MEMORY d
13 WORTH x
14 HOPE x
15 BETTER x
16 MALE c
```

Wei-Yin Loh 120 GUIDE manual
5.11 Differential item functioning

17 EDUCATION n
18 AGE n

Following is the input file creation log that uses GDSsub.dsc.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: GDSsub.in
Input 1 for model fitting, 2 for importance or DIF scoring,
   3 for data conversion ([1:3], <cr>=1):
Name of batch output file: GDSsub.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
   1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
   5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: GDSsub.dsc
Reading data description file ...
Training sample file: GDS.dat
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 3
D variables are:
   ENERGY
   BORED
   MEMORY
Choose multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables; otherwise choose univariate
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
   Choose 2 because items are not ordered.
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=2):
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Length of longest data entry = 2
Total number of cases: 1978
Col. no. Categorical variable #levels #missing values
   16 MALE 2 0
Checking data ...
PCA can be used for variable selection

Wei-Yin Loh 121 GUIDE manual
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
#cases w/ miss. D = number of cases with all D values missing

<table>
<thead>
<tr>
<th>Total</th>
<th>#cases w/ #missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#cases</td>
<td>miss. D ord. vals</td>
</tr>
<tr>
<td>1978</td>
<td>0</td>
</tr>
</tbody>
</table>

No. cases used for training: 1978
Warning: interaction tests skipped
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): GDSsub.tex
Input 2 to save node IDs of individual cases, 1 otherwise ([1:2], <cr>=2):
Input name of file to store terminal node ID of each case: GDSsub.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node fitted values: GDSsub.fit
Input file is created!

The result of running this input file produces a tree with one split on \( \text{AGE} \leq 64 \). The file \text{GDSsub.nid} gives the terminal node number that each case resides in the tree. The first 10 casea are shown below.

<table>
<thead>
<tr>
<th>case</th>
<th>train</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>y</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>2</td>
</tr>
</tbody>
</table>

The file \text{GDSsub.fit} gives the sample mean values of the dependent (item) variables in each terminal node:

<table>
<thead>
<tr>
<th>node</th>
<th>GDBORED</th>
<th>GDMEMORY</th>
<th>GDENERGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.13109E+00</td>
<td>0.64419E+00</td>
<td>0.75655E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.70175E-01</td>
<td>0.37953E+00</td>
<td>0.77485E+00</td>
</tr>
</tbody>
</table>

Figure 20 shows the tree and the mean item responses in each terminal node by item.

Wei-Yin Loh 122 GUIDE manual
Figure 20: GUIDE 0.50-SE regression tree for predicting response variables \text{ENERGY}, \text{BORED}, and \text{MEMORY}. PCA not used. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample sizes and predicted values of \text{ENERGY}, \text{BORED}, and \text{MEMORY} are printed below nodes.
6 Tree ensembles

A tree ensemble is a collection of trees. GUIDE has two methods of constructing an ensemble. One is called “bagged GUIDE”, which fits *pruned* GUIDE trees to bootstrap samples of the training data (Breiman, 1996). The other is called “GUIDE forest”; it is similar to random forest (Breiman, 2001), which fits *unpruned* trees to bootstrap samples but randomly selects a small subset of variables for split selection at each node. There is some empirical evidence that, if there are many variables of which only a few are useful for prediction, bagged GUIDE tends to be more accurate than GUIDE forest (Loh, 2009, 2012). But GUIDE forest is computationally faster.

6.1 Bagged GUIDE

We first demonstrate bagged GUIDE on the car data.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: bagin.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):

Name of batch output file: bagout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1): 2

*This is where an ensemble method is selected.*
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2): 1

Option 1 is bagged GUIDE, option 2 is GUIDE forest.
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): drivedsc.txt

Reading data description file ...
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Total number of cases = 428
Number of classes = 3

<table>
<thead>
<tr>
<th>Col. no.</th>
<th>Categorical variable</th>
<th>#levels</th>
<th>#missing values</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Make</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Type</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Wei-Yin Loh 124 GUIDE manual
Checking data ...

Class name        Num. cases  Proportion
4wd               94         0.21962617
fwd               224        0.52336449
rwd               110        0.25700935

Total #cases w/ #missing
#cases  miss. D ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
428          0        0     10      0     0      11      0     2

No. cases used for training = 428

Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Input name of file to store predicted class and probability: bagfit.txt
Input file is created!

Results

Ensemble of bagged classification trees
Pruning by cross-validation
Data description file: drivedsc.txt
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Number of classes = 3

Class    #Cases  Proportion
4wd      94     0.21962617
fwd      224    0.52336449
rwd      110    0.25700935

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

Column Name    Minimum          Maximum          #Categories          #Missing
3  Make          c             38               
5  Type          c             6               
6  Drive         d             3               
14 Rprice       s  1.0280E+04  1.9246E+05  
15 Dcost        s  9.8750E+03  1.7356E+05  
16 Enginsz      s  1.3000E+00  8.3000E+00  
17 Cylin        s -1.0000E+00  1.2000E+01  
18 Hp           s  7.3000E+01  5.0000E+02  
19 City         s  1.0000E+01  6.0000E+01  

Wei-Yin Loh  125  GUIDE manual
6.1 Bagged GUIDE

No. cases used for training: 428

Univariate split highest priority
Interaction splits 2nd priority; no linear splits
Number of trees in ensemble = 100
Pruning by v-fold cross-validation, with v = 5
Selected tree is based on mean of CV estimates
Simple node models
Estimated priors
Unit misclassification costs
Fraction of cases used for splitting each node = 0.23364
Max number of split levels = 7
Minimum node size = 10
Number of SE’s for pruned tree = 5.0000E-01

Mean number of terminal nodes = 8.020

Classification matrix for training sample:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>True class</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>4wd</td>
</tr>
<tr>
<td>4wd</td>
<td>61</td>
</tr>
<tr>
<td>fwd</td>
<td>19</td>
</tr>
<tr>
<td>rwd</td>
<td>14</td>
</tr>
<tr>
<td>Total</td>
<td>94</td>
</tr>
</tbody>
</table>

Number of cases used for tree construction = 428
Number misclassified = 57
Resubstitution est. of mean misclassification cost = 0.13317757009345793

Note: The above results will likely differ slightly from one run to another due to the randomness of bagging.

Predicted class probability estimates are stored in bagfit.txt
Elapsed time in seconds: 7.46616936

The following lines from the top of the file bagfit.txt give the estimated class probabilities and the predicted class of the observations.
"4wd" "fwd" "rwd" predicted observed
0.50866E-01 0.90605E+00 0.43088E-01 "fwd" "fwd"
0.50866E-01 0.90605E+00 0.43088E-01 "fwd" "fwd"
0.50866E-01 0.90605E+00 0.43088E-01 "fwd" "fwd"
0.50866E-01 0.90605E+00 0.43088E-01 "fwd" "fwd"
0.64426E-01 0.85457E+00 0.81001E-01 "fwd" "fwd"
0.64426E-01 0.85457E+00 0.81001E-01 "fwd" "fwd"

6.2 GUIDE forest

GUIDE forest differs from bagged GUIDE in two respects:

1. At each node a random subset of the variables is used for split selection.

2. The trees in GUIDE forest are not pruned.

These are the same principles in random forest. GUIDE forest differs from the latter in using GUIDE’s unbiased variable selection method instead of greedy search. GUIDE forest typically requires many more trees than bagged GUIDE to achieve similar accuracy, but because the former does not prune the trees, the former is still faster to compute.

Input file creation

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1

Name of batch input file: forestin.txt
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: forestout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1): 2
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2):
Input 1 for random splits of missing values, 2 for nonrandom: ([1:2], <cr>=2):
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): drivedsc.txt
Reading data description file ...
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Total number of cases = 428
Number of classes = 3

Col. no. Categorical variable #levels #missing values
 3 Make 38 0
 5 Type 6 0

Checking data ...
Class name Num. cases Proportion
 4wd 94 0.21962617
 fwd 224 0.52336449
 rwd 110 0.25700935

Total #cases w/ #missing
 #cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
 428 0 0 10 0 0 11 0 2

No. cases used for training = 428
Choose 1 for estimated priors, 2 for equal priors, 3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Input name of file to store predicted class and probability: forest.fit
Input file is created!

Results
Random forest of classification trees
Data description file: drivedsc.txt
Training sample file: drive.txt
Missing value code: *
Warning: N variables changed to S
Dependent variable is Drive
Length of longest data entry = 26
Number of classes = 3
Class #Cases Proportion
 4wd 94 0.21962617
 fwd 224 0.52336449
 rwd 110 0.25700935

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
Column Name Minimum Maximum #Categories #Missing

Wei-Yin Loh 128 GUIDE manual
6.2 GUIDE forest 6 TREE ENSEMBLES

3 Make c 38
5 Type c 6
6 Drive d 3
14 Rprice s 1.0280E+04 1.9246E+05
15 Dcost s 9.8750E+03 1.7356E+05
16 Enginsz s 1.3000E+00 8.3000E+00
17 Cylin s -1.0000E+00 1.2000E+01
18 Hp s 7.3000E+01 5.0000E+02
19 City s 1.0000E+01 6.0000E+01
20 Hwy s 1.2000E+01 6.6000E+01
21 Weight s 1.8500E+03 7.1900E+03
22 Whlbase s 8.9000E+01 1.4400E+02
23 Length s 1.4300E+02 2.2800E+02
24 Width s 6.4000E+01 8.1000E+01

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
428 0 0 10 0 0 11 0 2

No. cases used for training: 428

Univariate split highest priority
No interaction and linear splits
Number of trees in ensemble = 500
Number of variables used for splitting = 5
Simple node models
Estimated priors
Unit misclassification costs
Fraction of cases used for splitting each node = 0.23364
Max number of split levels = 10
Minimum node size = 5
Mean number of terminal nodes = 29.82

Classification matrix for training sample:

Predicted True class
class 4wd fwd rwd
4wd 80 6 0
fwd 4 216 3
rwd 10 2 107
Total 94 224 110

Number of cases used for tree construction = 428
Number misclassified = 25
Resubstitution est. of mean misclassification cost = 5.8411214953271021E-002

Predicted class probability estimates are stored in forest.fit
Elapsed time in seconds: 3.62463903

Wei-Yin Loh 129 GUIDE manual
The above results are not particularly useful because it is impossible to analyze the individual trees. The results mostly provide a record of the parameter values chosen to construct the forest. The most interesting results are the predicted values in the file `forest.fit`, the top few lines of which are shown below.

```
"4wd" "fwd" "rwd" predicted observed
0.62238E-02 0.98398E+00 0.97993E-02 "fwd" "fwd"
0.63508E-02 0.98035E+00 0.13298E-01 "fwd" "fwd"
0.55558E-02 0.98198E+00 0.12466E-01 "fwd" "fwd"
0.53931E-02 0.98608E+00 0.85241E-02 "fwd" "fwd"
0.56024E-02 0.98205E+00 0.12348E-01 "fwd" "fwd"
0.92692E-02 0.96697E+00 0.23756E-01 "fwd" "fwd"
```

7 Importance scores

GUIDE can rank the variables in order of their importance for predicting the dependent variable. In addition, it provides a threshold score for distinguishing the important variables from the unimportant ones.

7.1 Baseball data example

We demonstrate this capability with the baseball data below.

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 1
Name of batch input file: `bbimp.in`
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1): 2
Option 2 yields importance scores.
Name of batch output file: `bbimp.out`
Input 1 for classification tree, 2 for regression tree ([1:2], <cr>=1): 2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):

Wei-Yin Loh 130 GUIDE manual
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters); 
enclose with matching quotes if it has spaces: bbdsc.txt
Reading data description file ...
Training sample file: bbdat.txt
Missing value code: NA
Warning: N variables changed to S
Warning: B variables changed to C
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases: 263

Col. no. Categorical variable #levels #missing values
  16 Leag86    2  0
  17 Div86    2  0
  18 Team86  24  0
  19 Pos86   23  0
  24 Leag87   2  0
  25 Team87  24  0

Checking data ...

No weight variable in data file
No. cases used for training: 263
Input expected fraction of noise variables erroneously selected 
([0.00:0.99], <cr>=0.01):

This sets the ‘‘alpha’’ value such that, under the null hypothesis that all
variables are noise, the proportion erroneously selected is alpha.

You can create a description file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1): 2

This option lets GUIDE automatically write a new description file with the
unimportant variables given the X designation.
Input 1 to keep only selected variables, 2 to exclude selected variables ([1:2], <cr>=1):
Input file name: bbsub.dsc
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: bbimp.scr
A file by that name already exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input file is created!

Results  The importance scores are given at the end of the output file bbimp.out.
### Predictor variables sorted by importance scores

<table>
<thead>
<tr>
<th>Importance Scores</th>
<th>Scaled</th>
<th>Unscaled</th>
<th>Rank</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0</td>
<td>1.93217E+01</td>
<td>1</td>
<td>Hitcr</td>
<td></td>
</tr>
<tr>
<td>95.4</td>
<td>1.84423E+01</td>
<td>2</td>
<td>Batcr</td>
<td></td>
</tr>
<tr>
<td>85.8</td>
<td>1.65721E+01</td>
<td>3</td>
<td>Runcr</td>
<td></td>
</tr>
<tr>
<td>82.0</td>
<td>1.58486E+01</td>
<td>4</td>
<td>Rbcr</td>
<td></td>
</tr>
<tr>
<td>74.7</td>
<td>1.44374E+01</td>
<td>5</td>
<td>Yrs</td>
<td></td>
</tr>
<tr>
<td>66.6</td>
<td>1.28632E+01</td>
<td>6</td>
<td>Wlkr</td>
<td></td>
</tr>
<tr>
<td>41.8</td>
<td>8.06758E+00</td>
<td>7</td>
<td>Hit86</td>
<td></td>
</tr>
<tr>
<td>41.0</td>
<td>7.91880E+00</td>
<td>8</td>
<td>Hr86</td>
<td></td>
</tr>
<tr>
<td>32.0</td>
<td>6.17371E+00</td>
<td>9</td>
<td>Run86</td>
<td></td>
</tr>
<tr>
<td>31.0</td>
<td>5.98389E+00</td>
<td>10</td>
<td>Bat86</td>
<td></td>
</tr>
<tr>
<td>30.8</td>
<td>5.95530E+00</td>
<td>11</td>
<td>Rb86</td>
<td></td>
</tr>
<tr>
<td>27.6</td>
<td>5.32659E+00</td>
<td>12</td>
<td>Wlk86</td>
<td></td>
</tr>
<tr>
<td>17.6</td>
<td>3.39316E+00</td>
<td>13</td>
<td>Hr86</td>
<td></td>
</tr>
<tr>
<td>8.8</td>
<td>1.70322E+00</td>
<td>14</td>
<td>Pos86</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>1.40858E+00</td>
<td>15</td>
<td>Puto86</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>1.02078E+00</td>
<td>16</td>
<td>Team87</td>
<td></td>
</tr>
</tbody>
</table>

------------------- cut-off -------------------
| 3.7 | 7.07834E-01 | 17 | Err86 |
| 3.4 | 6.53864E-01 | 18 | Asst86 |
| 2.3 | 4.43732E-01 | 19 | Team86 |
| 1.9 | 3.61595E-01 | 20 | Leag87 |
| 1.4 | 2.77317E-01 | 21 | Leag86 |
| 1.3 | 2.51640E-01 | 22 | Div86  |

Variables with unscaled scores above 1 (the cut-off line) are deemed important.

Number of important splitting variables = 16

Number of unimportant splitting variables = 6

Here are the contents of the file bbimp.scr:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.322</td>
<td>Hitcr</td>
</tr>
<tr>
<td>2</td>
<td>18.442</td>
<td>Batcr</td>
</tr>
<tr>
<td>3</td>
<td>16.572</td>
<td>Runcr</td>
</tr>
<tr>
<td>4</td>
<td>15.849</td>
<td>Rbcr</td>
</tr>
<tr>
<td>5</td>
<td>14.437</td>
<td>Yrs</td>
</tr>
<tr>
<td>6</td>
<td>12.863</td>
<td>Wlkr</td>
</tr>
<tr>
<td>7</td>
<td>8.068</td>
<td>Hit86</td>
</tr>
<tr>
<td>8</td>
<td>7.919</td>
<td>Hr86</td>
</tr>
<tr>
<td>9</td>
<td>6.174</td>
<td>Run86</td>
</tr>
<tr>
<td>10</td>
<td>5.984</td>
<td>Bat86</td>
</tr>
<tr>
<td>11</td>
<td>5.955</td>
<td>Rb86</td>
</tr>
<tr>
<td>12</td>
<td>5.327</td>
<td>Wlk86</td>
</tr>
<tr>
<td>13</td>
<td>3.393</td>
<td>Hr86</td>
</tr>
</tbody>
</table>
And here are the contents of the file `bbsub.dsc`:

```
"bbdat.txt"
"NA"

<table>
<thead>
<tr>
<th>colnumber</th>
<th>varname</th>
<th>vartype</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Id</td>
<td>x</td>
</tr>
<tr>
<td>2</td>
<td>Name</td>
<td>x</td>
</tr>
<tr>
<td>3</td>
<td>Bat86</td>
<td>n</td>
</tr>
<tr>
<td>4</td>
<td>Hit86</td>
<td>n</td>
</tr>
<tr>
<td>5</td>
<td>Hr86</td>
<td>n</td>
</tr>
<tr>
<td>6</td>
<td>Run86</td>
<td>n</td>
</tr>
<tr>
<td>7</td>
<td>Rb86</td>
<td>n</td>
</tr>
<tr>
<td>8</td>
<td>Wlk86</td>
<td>n</td>
</tr>
<tr>
<td>9</td>
<td>Yrs</td>
<td>n</td>
</tr>
<tr>
<td>10</td>
<td>Batcr</td>
<td>n</td>
</tr>
<tr>
<td>11</td>
<td>Hitcr</td>
<td>n</td>
</tr>
<tr>
<td>12</td>
<td>Hrcr</td>
<td>n</td>
</tr>
<tr>
<td>13</td>
<td>Runcr</td>
<td>n</td>
</tr>
<tr>
<td>14</td>
<td>Rbcr</td>
<td>n</td>
</tr>
<tr>
<td>15</td>
<td>Wlkr</td>
<td>n</td>
</tr>
<tr>
<td>16</td>
<td>Leag86</td>
<td>x</td>
</tr>
<tr>
<td>17</td>
<td>Div86</td>
<td>x</td>
</tr>
<tr>
<td>18</td>
<td>Team86</td>
<td>x</td>
</tr>
<tr>
<td>19</td>
<td>Pos86</td>
<td>c</td>
</tr>
<tr>
<td>20</td>
<td>Puto86</td>
<td>n</td>
</tr>
<tr>
<td>21</td>
<td>Asst86</td>
<td>x</td>
</tr>
<tr>
<td>22</td>
<td>Err86</td>
<td>x</td>
</tr>
<tr>
<td>23</td>
<td>Salary</td>
<td>x</td>
</tr>
<tr>
<td>24</td>
<td>Leag87</td>
<td>x</td>
</tr>
<tr>
<td>25</td>
<td>Team87</td>
<td>c</td>
</tr>
<tr>
<td>26</td>
<td>Logsalary</td>
<td>d</td>
</tr>
</tbody>
</table>
```
8 Other features

8.1 Pruning with test samples

GUIDE typically has three pruning options for deciding the size of the final tree: (i) cross-validation, (ii) test sample, and (iii) no pruning. Test-sample pruning is available only when there are no derived variables, such as creation of dummy indicator variables when ‘b’ variables are present. If test-sample pruning is chosen, the program will ask for the name of the file containing the test samples. This file must have the same column format as the training sample file. Pruning with test-samples or no pruning are non-default options.

8.2 Prediction of test samples

GUIDE can produce R code to predict future observations from all except kernel and nearest neighbor classification and ensemble models. This is also a non-default option.

Predictions of the training data for all models can be obtained, however, at the time of tree construction. This feature can be used to obtain predictions on “test samples” (i.e., observations that are not used in tree construction) by adding them to the training sample file. There are two ways to distinguish the test observations from the training observations:

1. Use a weight variable (designated as \( W \) in the description file) that takes value 1 for each training observation and 0 or each test observation.

2. Replace the D values of the test observations with the missing value code.

For tree construction, GUIDE does not use observations in the training sample file that have zero weight.

8.3 GUIDE in R and in simulations

GUIDE can be used in simulations or used repeatedly on bootstrap samples to produce an ensemble of tree models. For the latter,

1. Create a file (with name data.txt, say) containing one set of bootstrapped data.

2. Create a data description file (with name desc.txt, say) that refers to data.txt.
3. Create an input file (with name input.txt, say) that refers to desc.txt.

4. Write a batch program (Windows) or a shell script (Linux or Macintosh) that repeatedly:
   (a) replaces the file data.txt with new bootstrapped samples;
   (b) calls GUIDE with the command: guide < input.txt; and
   (c) reads and processes the results from each GUIDE run.

In R, the command in step 4b depends on the operating system. If the GUIDE program and the files data.txt and input.txt are in the same folder as the working R directory, the command is:

Linux/Macintosh: system("guide < input.txt > log.txt")
Windows: shell("guide < input.txt > log.txt")

If the files are not all in the same folder, full path names must be given. Here log.txt is a text file that stores messages during execution. If GUIDE does not run successfully, errors are also written to log.txt.

8.4 Generation of powers and products

GUIDE allows the creation of certain powers and products of regressor variables on the fly. Specifically, variables of the form $X_1^pX_2^q$, where $X_1$ and $X_2$ are numerical predictor variables and $p$ and $q$ are integers, can be created by adding one or more lines of the form

0 i p j q a

at the end of the data description file. Here $i$ and $j$ are integers giving the column numbers of variables $X_1$ and $X_2$, respectively, in the data file and $a$ is one of the letters $n$, $s$, or $f$ (corresponding to a numerical variable used for both splitting and fitting, splitting only, or fitting only).

To illustrate, suppose we wish to fit a piecewise quadratic model in the variable $Y_{rs}$ for the baseball data. This is easily done by adding one line to the file bbds.txt. First we assign the $s$ (for splitting only) designator to every numerical predictor except $Y_{rs}$. This will prevent all variables other than $Y_{rs}$ from acting as regressors in the piecewise quadratic models. To create the variable $Y_{rs}^2$, add the line

0 9 2 9 0 f
8.5 Data formatting functions

The program includes a utility function for reformatting data files into forms required by some statistical software packages:

When the program is given this description file, the output will show the regression coefficients of $\text{Yrs}$ and $\text{Yrs}^2$ in each terminal node of the tree.
8.5 Data formatting functions

1. **R/Splus**: Fields are space delimited. Missing values are coded as **NA**. Each record is written on one line. Variable names are given on the first line.

2. **SAS**: Fields are space delimited. Missing values are coded with periods. Character strings are truncated to eight characters. Spaces within character strings are replaced with underscores (_).

3. **TEXT**: Fields are comma delimited. Empty fields denote missing values. Character strings longer than eight characters are truncated. Each record is written on one line. Variable names are given on the first line.

4. **STATISTICA**: Fields are comma delimited. Commas in character strings are stripped. Empty fields denote missing values. Each record occupies one line.

5. **SYSTAT**: Fields are comma delimited. Strings are truncated to eight characters. Missing character values are replaced with spaces, missing numerical values with periods. Each record occupies one line.

6. **BMDP**: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are indicated by asterisks. Variable names longer than eight characters are truncated.

7. **DataDesk**: Fields are space delimited. Missing categorical values are coded with question marks. Missing numerical values are coded with asterisks. Each record is written on one line. Spaces within categorical values are replaced with underscores. Variable names are given on the first line of the file.

8. **MINITAB**: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are coded with asterisks. Variable names longer than eight characters are truncated.

9. **NUMBERS**: Same as **TEXT** option except that categorical values are converted to integer codes.

10. **C4.5**: This is the format required by the C4.5 (Quinlan, 1993) program.

11. **ARFF**: This is the format required by the WEKA (Witten and Frank, 2000) programs.

Following is a sample session where the iris data are reformatted for R or Splus.
8.5 Data formatting functions

0. Read the warranty disclaimer
1. Create an input file for batch run
2. Fit a model without creating input file
3. Convert data to other formats
4. Variable importance scoring and differential item functioning

Input your choice: 3
Input name of log file: log.txt
Input 1 if D variable is categorical, 2 if real, 0 if none ([0:2], <cr>=1):

Input name of data description file (maximum 100 characters; enclose within quotes if it contains spaces or non alphanumeric characters): irisdsc.txt

Reading data description file ...
Training sample file: irisdata.txt
Missing value code: ?
Warning: N variables changed to S
Dependent variable is class
Length of longest data entry = 11
Total number of cases = 150
Number of classes = 3
Choose one of the following data formats:

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Separ</th>
<th>char.</th>
<th>numer.</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R/Splus</td>
<td>space</td>
<td>NA</td>
<td>NA</td>
<td>1 line/case, var names on 1st line</td>
</tr>
<tr>
<td>2</td>
<td>SAS</td>
<td>space</td>
<td>.</td>
<td>.</td>
<td>strings trunc., spaces -&gt; ' '</td>
</tr>
<tr>
<td>3</td>
<td>TEXT</td>
<td>comma</td>
<td>empty</td>
<td>empty</td>
<td>1 line/case, var names on 1st line</td>
</tr>
<tr>
<td>4</td>
<td>STATISTICA</td>
<td>comma</td>
<td>empty</td>
<td>empty</td>
<td>1 line/case, commas stripped</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>var names on 1st line</td>
</tr>
<tr>
<td>5</td>
<td>SYSTAT</td>
<td>comma</td>
<td>space</td>
<td>.</td>
<td>1 line/case, var names on 1st line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>strings trunc. to 8 chars</td>
</tr>
<tr>
<td>6</td>
<td>BMDP</td>
<td>space</td>
<td>*</td>
<td>*</td>
<td>cat values -&gt; integers (alph. order)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>strings trunc. to 8 chars</td>
</tr>
<tr>
<td>7</td>
<td>DATADESK</td>
<td>space</td>
<td>?</td>
<td>*</td>
<td>1 line/case, var names on 1st line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>spaces -&gt; ' '</td>
</tr>
<tr>
<td>8</td>
<td>MINITAB</td>
<td>space</td>
<td>*</td>
<td></td>
<td>cat values -&gt; integers (alph. order)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>var names trunc. to 8 chars</td>
</tr>
<tr>
<td>9</td>
<td>NUMBERS</td>
<td>comma</td>
<td>NA</td>
<td>NA</td>
<td>1 line/case, var names on 1st line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cat values -&gt; integers (alph. order)</td>
</tr>
<tr>
<td>10</td>
<td>C4.5</td>
<td>comma</td>
<td>?</td>
<td>?</td>
<td>1 line/case, dependent variable last</td>
</tr>
<tr>
<td>11</td>
<td>ARFF</td>
<td>comma</td>
<td>?</td>
<td>?</td>
<td>1 line/case</td>
</tr>
</tbody>
</table>

0 abort this job

Input your choice ([0:11], <cr>=1):
Input name of new data file: iris.rdata
Follow the commented lines in "iris.rdata" to read the data into R or Splus
References


