Chi-Square **A**utomatic **I**nteraction **D**etection (CHAID)

Psychology 315t
Justin Mary, Dale Berger

Topics:

Overview of CHAID
Introduction to CHAID using R

Sources:

Kass, G.V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistics Society, Series C, 29(2)*, 119-127.

R resources for CHAID: http://chaid.r-forge.r-project.org/ .

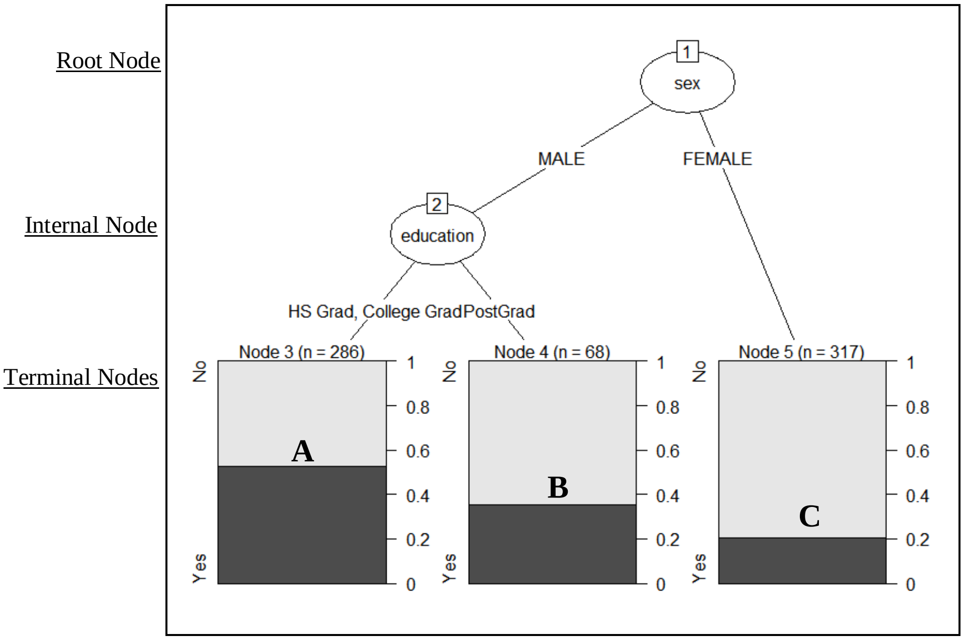
Overview:

Researchers are often tasked with mining large datasets to find the most useful predictors of an outcome. For example, federal agencies may be interested in identifying which subgroups of drivers are most and least likely to drink and drive so targeted campaigns can be developed. A univariate survey of the data might lead researchers to conclude that women drivers can be excluded from the campaign, given their low overall rates of drinking and driving. However, a multivariate analysis might find that a sub-group of women, those who prefer beer, are actually quite likely to drink and drive.

This example underscores the importance of identifying important subgroups. An apparent challenge with this goal is that there are potentially thousands of subgroups that can be assessed depending on the number of predictors. A tool that automatically identifies distinctly different subgroups is CHAID (Chi-Square Automatic Interaction Detection). The CHAID algorithm parses a predictor variable into its most distinct subgroups with respect to the dependent variable, while preserving parsimony and adjusting for alpha inflation (Type I error). This packet provides a general introduction of CHAID and a guide to implementing the CHAID algorithm using R.

CHAID: An application

Let's assume that researchers are interested in identifying sub-groups of drivers that are most and least likely to drink and drive. Data were collected from 671 respondents concerning their intention to drink and drive within twelve months of completing the survey (DD; 0 = No, 1 = Yes). Along with their intention to drink and drive, several demographic variables were also reported. For the purposes of this example, let's assume that only sex (1 = Male, 2 = Female) and educational status (1 = High School diploma, 2 = College Grad, 3= Postgraduate) were measured. Broadly speaking, the CHAID algorithm produces a series of subgroup merges and splits. Subgroups with the most similar DD intentions are merged while the most dissimilar subgroups are split into separate nodes.



CHAID produces three subgroups with the lowest and highest drinking and driving rates.

- **A)** Male HS grads/College grads: 53% reported DD intention
- **B)** Male post grads: 35% reported DD intention
- **C)** Females: Only 20% of women reported DD intention.

Note that the CHAID analysis merged male HS Grads and College grads, given that they had similar DD intentions and split males and females because they were quite different in DD intentions.

CHAID in R

We will now provide a demonstration of CHAID in R for the drinking and driving example from page 1 using CHAIDe2.sav.

Section 1: Installing and loading the necessary packages.

Step 1: Downloading the CHAID package

Open the R command line. Connect to the Internet. Copy and paste the following code into the R command line:

```
install.packages("CHAID", repos="http://R-Forge.R-project.org").
```

The download may take some time.

Step 2: Load the packages for CHAID, HMISC, foreign, and Rcmdr:

To load the relevant packages, copy and paste the following code:

```
library(foreign)
library(Hmisc)
library(CHAID)
library(Rcmdr)
```

Section 2: Loading Data from SPSS

Step 1: Load data: R-commander will automatically start-up after loading the packages. From the R-Commander window, click Data→ Import data→from SPSS data set. Select your file CHAIDe2.sav. You will be given an opportunity to name your dataset. Note this decision and keep in mind that the name used in this example is 'Dataset'. Further note that the data set name is different from the file name of your .sav file and is unique to R.

Step 2: Verify that the data structure is correct: Sometimes R fails to create the factors for your variables correctly. To check, click Data → Manage variables in active data set→ convert numeric variables into factors. If there are any relevant variables in the left column, click the radio button "use numbers," select every relevant variable and click OK. A series of prompts may appear and you should click YES.

Section 3: Conducting the CHAID Analysis

There are two parts of the CHAID code. The first section of code handles all of the settings for splitting and merging subgroups. The second section of code handles the variables of interest and the data set.

Step 1: CHAID Settings: Copy and paste the following code into the R syntax window:

```
ctrl <- chaid_control(minbucket = 100, minsplit = 100, alpha2=.05, alpha4 = .05)
```

ctrl: A variable to hold all of the settings. You can name this anything you would like.

<- : Assignment operator. This piece of code tells R that we want to assign all of these parameters to the variable **ctrl** for safe keeping. This makes it easy for us to reference our settings in the future.

chaid_control: A function to set all of the parameters.

minbucket: Minimum number of observations in final (terminal node). A split will be performed only on subgroups that have at least as many cases as specified here.

minsplit: Minimum number of observations for subgroup split. If a variable meets all the other criteria for a split but has a sample size less than minsplit, the split will not be performed.

alpha2: *Merging threshold*: The threshold *p* – value at which groups will be merged if the observed *p* –value for a test of their difference is larger than the threshold. In our example, male high school grads and college grads had similar intentions to drink and drive (*p* = .680), so they are merged.

alpha4: *Splitting threshold*: The threshold *p* – value (Bonferroni corrected) at which groups will be split if the observed *p* – value is smaller than the threshold. In our example, male and female respondents were very different in their drinking and driving habits (*p* < .001) and a split was performed.

For more information and a list of additional parameters, see the CHAID documentation (p. 6).

Step 2: Specify the dataset, variables, and run the analysis: Copy and paste the following code into the R syntax window.

```
chaidUS <- chaid(dd ~ education+sex, data = Dataset, control = ctrl)
```

chaidUS: A variable holding the final analysis. You can name this variable whatever you wish.

chaid: Activates the program ‘chaid’. This is similar to the ‘plot’ function in SPSS syntax.

dd: Your categorical dependent variable for drinking and driving.

~ : Operator to separate DV from predictor variables.

education+sex: Your predictor variables of interest. Alternatively, this line can be replaced with a period and the program will use every variable in your dataset as a predictor except for the DV.

data = Dataset: Specify the name of your data set.

control = ctrl: Tells the program to use the specifications in your ctrl variable from above.
NOTE: No output will appear- nothing should happen even if you successfully execute the code.

Step 3: Display your output:

print(chaidUS): This function will print a table.

Plot(chaidUS): This function will create a plot like on page 2.

The CHAID Algorithm:

There are five total steps in the CHAID algorithm. CHAID often loops, so steps will be repeated on multiple occasions.

Step 1: Create separate crosstabs for each predictor (gender, education) with your dependent variable (DD).

Step 2: For each predictor, compute a chi-square for every possible 2x2 sub-table. For instance, you would compare HS grad to college grad on DD (yes or no), then HS grad to Postgrad, then college grad to post grad, then men to women (see handout). Find the **least** significant relationship and merge the subgroups if the *p* –value is larger than a predetermined threshold (alpha2). Keep repeating this step until all possible combinations have a *p* –value smaller than the threshold. In our example, high school grads and college grads have very similar views on drinking and driving and they are merged into one category (all high school students and college graduates would be coded as 1). Note that gender cannot be merged because there are only two levels, and constants are not permitted in the program.

Step 3: It is worth noting that Step 3 exists, given that it is referenced in the R documentation and in the CHAID source paper (Kauss, 1980). However, Step 3 is redundant with Step 2 and for this reason it is optional and often omitted.

Step 4: Re-run chi-squares on all predictor variables using the optimally merged predictors (your new educational variable has only two levels) and gender. Select the most significant variable and then split if the *p* –value is smaller than a set threshold (alpha4). Sex had the smallest *p* –value and it was smaller than .05, so you initiate the split, and then return the educational variable to its original state for further steps.

Step 5: Go back to Step 1 and run the algorithm for each new subgroup identified in Step 4. You would now look at the educational variables within males and females separately.

Note: CHAID takes parsimony into account as well as maximum distinctiveness. This is why you create composite variables to begin with. Further, CHAID can be extended to include dependent variables that are ordinal and variables with more than two categories.

CHAID Documentation

Description

Creates a classification tree by using the CHAID algorithm. The bolded arguments are the most important. Many of the other options have defaults and can be ignored.

Usage

```
chaid(formula, data, subset, weights, na.action = na.omit,
    control = chaid_control())
chaid_control(alpha2 = 0.05, alpha3 = -1, alpha4 = 0.05,
         minsplit = 20, minbucket = 7, minprob = 0.01,
         stump = FALSE, maxheight = -1)
```

Note. Please see the example equations on pages 3-4 for more information.

Arguments

formula The response variable and all covariates are assumed to be categorical (either ordered or not). **('formula' is a fancy way of saying 'your variables of interest').**

data an optional data frame containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `chaid` is called. **(Your R dataset name)**

`subset` You can specify a specific range of cases to be included if you would like.

`weights` an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. You can give some variables more weight in the algorithm.

`na.action` a function which indicates what should happen when the data contain `NA`s. The default is `na.omit`. This is a built-in cleaning mechanism for empty cells.

`control` hyper parameters of the algorithm as returned by `chaid_control`.

alpha2 *Merging threshold:* The level of significance used for merging of predictor categories **(step 2).**

`alpha3` If set to a positive value, this is the level of significance used for splitting formerly merged categories of the predictor (step 3). Otherwise, step 3 is omitted (the default). **(You can likely ignore this) -1 indicates that you want to ignore this variable.**

alpha4 *Splitting threshold*: The level of significance used for splitting of a node in the most significant predictor (step 5).

minsplit Number of observations in non-terminal node at which no further split is desired.

minbucket Minimum number of observations in terminal nodes. (Number of cases)

minprob The threshold for the proportion of observations in a newly split node. If the new node contains a smaller proportion of observations of the original cell than the threshold, do not split. Otherwise, split. E.g. Threshold = .10. If gender (n= 100) is to be split, but females (n = 8) make up less than the threshold (.10) do not split. In this case females make up .08 of the original sample, so no split would be performed.

`stump` only root node splits are performed. (Root nodes are nodes that have no incoming connections) http://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf

`maxheight` Maximum height for the tree. (Maximum number of levels). -1 indicates no maxheight.