

User's Manual for the FID GUI 3.5

FID is a tool used to generate Fuzzy Decision Trees based on input data and many different options. With version 3.5, a GUI is available that shows a graphical view of the input data, eases setting runtime options, allows manual modification of input data, and shows a tree view of the output.

Table of Contents

1	System Requirements.....	2
2	Installation.....	3
3	Launching the Application.....	4
4	Generating a New Fuzzy Decision Tree.....	5
4.1	File Definition Page.....	5
4.1.1	Command-Line Options.....	8
4.2	Input Display Page.....	14
4.3	Output View.....	16
4.3.1	Saving Output.....	18
4.3.2	Loading Previously Saved Output.....	18
5	Running the FID GUI – Unpartitioned Attributes.....	20
5.1	File Definition Page.....	20
5.2	Input Display Page.....	21
5.3	Output View.....	21
6	Manually Modifying Attributes.....	23
6.1	File Definition Page.....	23
6.2	Manual Discretization Dialog.....	23
6.2.1	Modifying Values.....	24
6.2.2	Inserting New Definitions.....	26
6.2.3	Deleting Definitions.....	27
6.3	Input Display Page.....	28
6.4	Output View.....	29
7	Troubleshooting.....	30

1 System Requirements

Because the FID GUI is written in Java, it is platform-independent. The same executable will work equally well with the Windows, Linux, or Solaris FID executable.

The FID GUI requires the Java J2SE version 1.4.2 or higher. If needed, this free download can be obtained at Sun's website at <http://java.sun.com/>.

The installation requires less than 4 MB of hard drive space. Memory requirements have not been calculated, but there should not be a problem on any machine that has Java installed.

2 Installation

Installing the FID GUI is extremely simple. Download the fidGUI35.jar file, the relevant executable file for your operating system, and the example data from the project website: <http://www.cs.umsl.edu/~janikow/fid/>.

That's it!

3 Launching the Application

To run the FID GUI, there are two options:

1. On a command line (like the DOS console in Windows), change directories to the directory where you installed the application. Then, use this command: `java -jar fidGUI35.jar`
2. In a graphical user interface (like Windows Explorer), navigate to the directory where you installed the application. Then, double-click on the `fidGUI35.jar` icon. If your machine has Java installed properly, then the application should launch. If not, you will have to associate `jar` files with `javaw` or choose `javaw` from the context-sensitive menu (the right-click menu).

We recommend using the first option because some error messages from the C engine are sent to the console. When the FID GUI is launched by double-clicking on an icon, the information is lost.

Upon launching, you will see the following view:

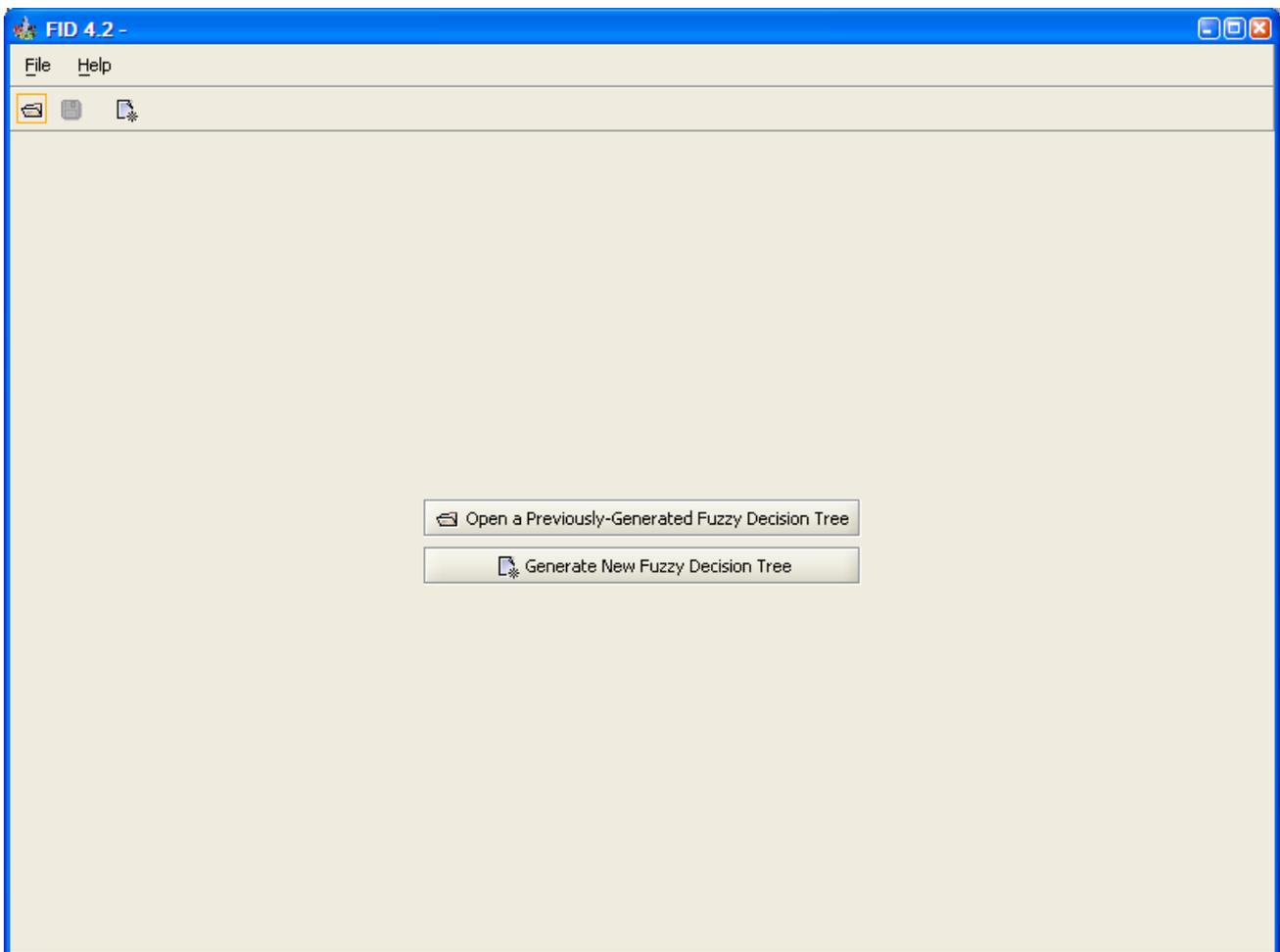


Figure 1 Initial View

The possible actions here are to open a previously-generated fuzzy decision tree or to generate a new fuzzy decision tree (in addition to the large buttons in the center of the view, those options are available from the File menu or from the tool bar buttons).

4 Generating a New Fuzzy Decision Tree

4.1 File Definition Page

Upon launching the Wizard, the first view you see is of the File Definition Page. This page is where you specify the File Directory, the Attributes File, the Events File, the Parameters File, the C Executable File, Command-Line Options, whether to use Cross Validation, and the K-Value and Label for Output Files if Cross Validation is used.

The screenshot shows a window titled "Generate Fuzzy Decision Tree" with a "File Definition" section. The "Use K-fold Cross Validation" checkbox is checked. The "File Directory" field is empty with a file browser icon. The "Attributes File", "Events File", and "Parameters File" fields are dropdown menus. The "Command-Line Options" field is empty with a file browser icon. The "FID Executable File" field contains the path "/Users/eryncantrell/Dropbox/fid/FID35/fid35" with a file browser icon. The "Label for Output Files" and "K-Value" fields are empty. At the bottom, there are buttons for "Previous", "Next", "Last", "Finish", and "Cancel".

The Attributes File, the Events File, and the Parameters File can be anywhere in your file system, but they must be in the same directory. To specify the directory, click on the file open button across from File Directory. The following File Chooser Dialog will be opened:

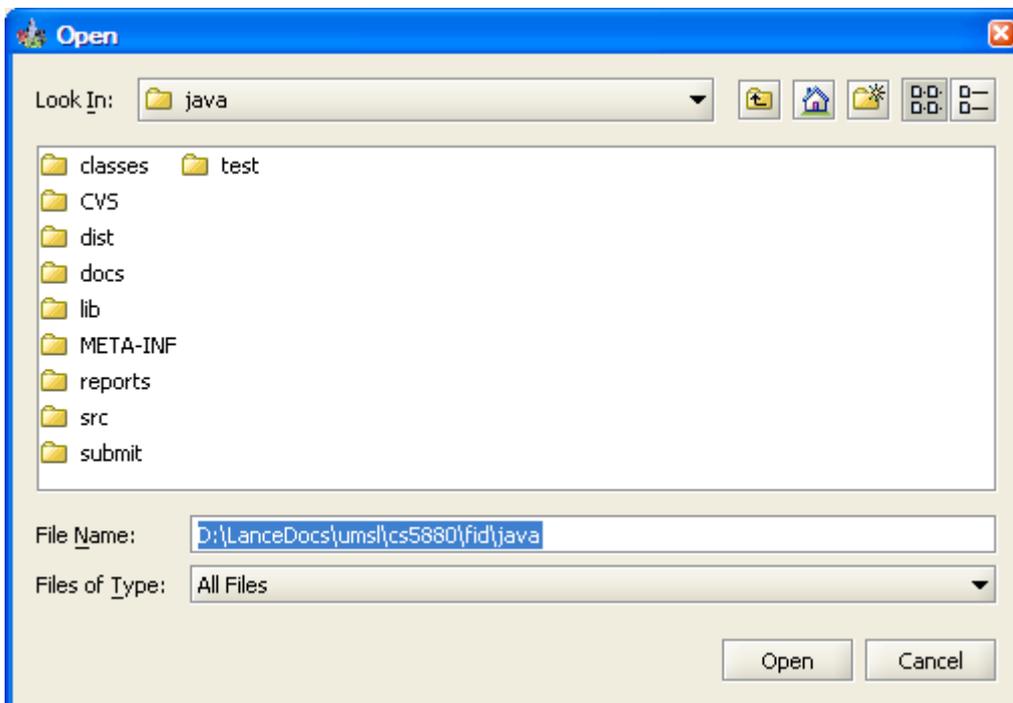


Figure 3: Working Directory Chooser

Select the directory that has your input files and click `Open`. The directory chosen here will also be the directory in which the output of the C Engine will be placed.

The Attributes File, the Events File, and the Parameters File are all defined the same way. The drop-down lists all of the files in the selected directory that have a name no more than 40 characters (a restriction inherited from the command-line version, which is indirectly used by the FID GUI). You should select an appropriate file for each of the three fields.

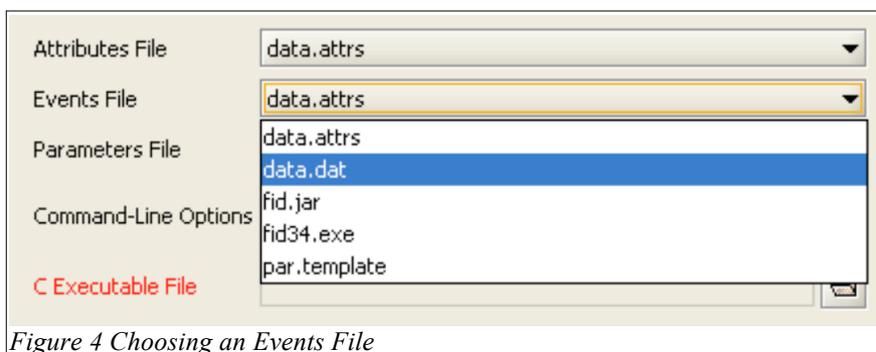


Figure 4 Choosing an Events File

If you select a file that does not have the right format, you will receive an error dialog similar to this one upon clicking `Next`:

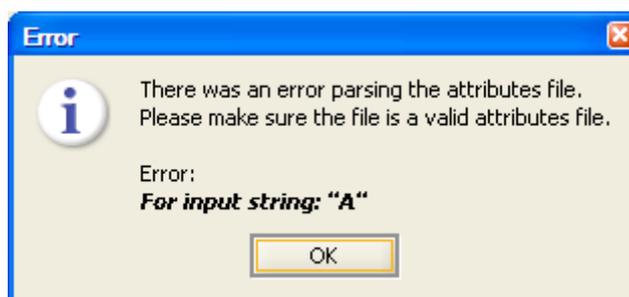


Figure 5 Attributes File Error Message

This error message results from trying to use an Events File as an Attributes File. If you get a message like this, please select a valid file instead.

You may have noticed that the label for the C Executable File is red at launch. This is true because a file must be selected for the `Next` button to be enabled. To fill in a value here, click on the file open button. The following File Chooser Dialog will be opened:

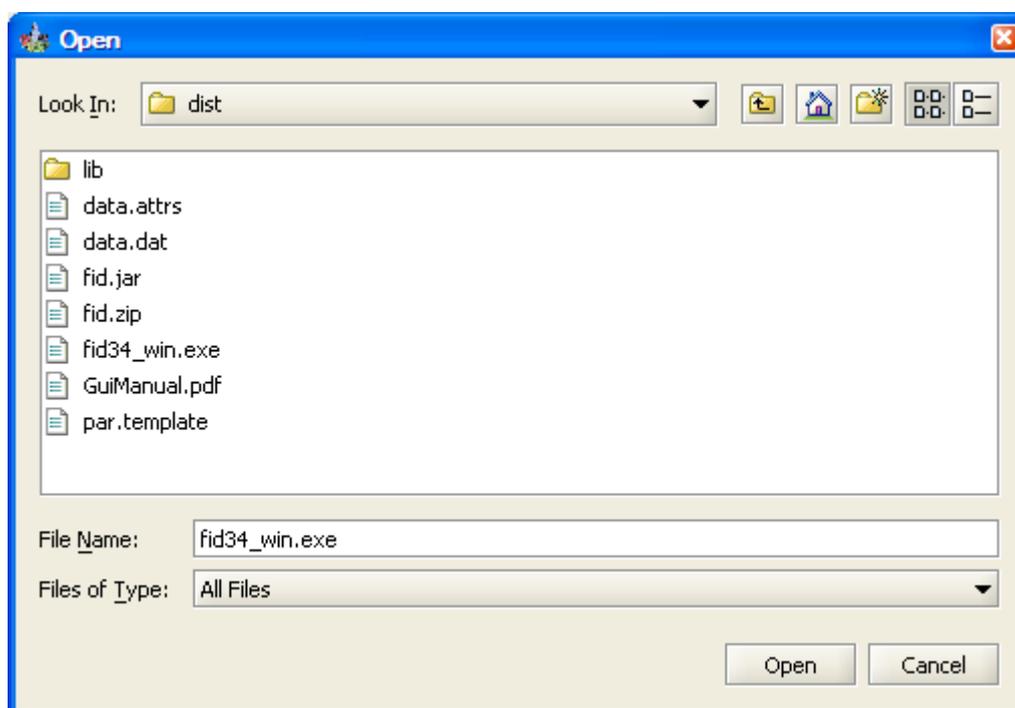


Figure 6 C Executable File Chooser

In this dialog, select `fid34_win.exe` (for Windows; choose the appropriate file for Linux or Solaris) and click `Open`.

You may be wondering why you can use a file chooser to select any file for the C Executable File, but you are limited to files in the installation directory for the other three files. This is a consequence of the 40-character limit on file names in the command-line version. The Attributes File, the Events File, and the Parameters File file locations are passed to the command-line version, but the location of the C Executable File is used only by the GUI. So, the GUI does not need to protect the command-line version from paths that are particularly long, like

`D:\MyDocs\umsl\fid\java\src\fid\dist\fid34_win.exe`, which would not be acceptable for the Attributes File, the Events File, and the Parameters File.

Your selection for the C Executable File will be saved by the FID GUI, so you will have to set this value only once.

With these four settings completed, the first page is complete and we can move on to the next page:

Figure 7: File Definition Page with Files Defined

However, before we move on, we should look at the Command-Line Options.

4.1.1 Command-Line Options

The Command-Line Options dialog allows you to override the values set in the Parameters File. These are all the options available with the command-line version of FID, except for `-p`, `-pruningoff`, and `-tree`.

To open the Command-Line Options dialog, click on the button to the far right of the “Command-Line Options” label. The following dialog will launch:

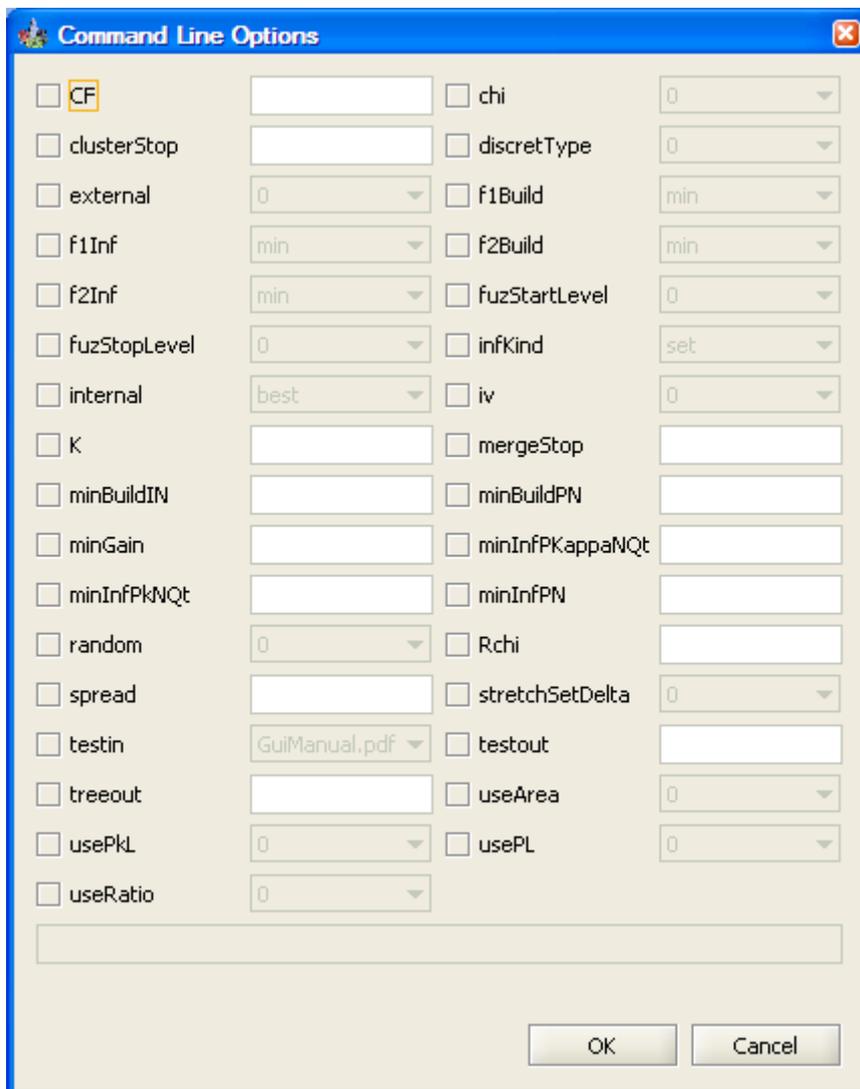


Figure 8 Initial Command-Line Options Dialog

All of the legal command-line options (except `-p`, `-pruningoff`, and `-tree`) are named here. There are four types of options:

Numeric – a real number within the range defined in the FID manual is acceptable

`CF`, `clusterStop`, `K`, `mergeStop`, `minBuildIN`, `minBuildPN`, `minGain`, `minInfPkappaNQt`, `minInfPkNQt`, `minInfPN`, and `spread`

Discrete – a specific list of values defined in the FID manual is acceptable

`chi`, `discretType`, `external`, `f1Build`, `f1Inf`, `f2Build`, `f2Inf`, `fuzStartLevel`, `infKind`, `internal`, `iv`, `random`, `stretchSetDelta`, `useArea`, `usePkL`, `usePL`, and `useRatio`

Input Files – a file in the directory selected on the File Definition Page with a name less than 40 characters is acceptable

`testin`

Output Files – any text is acceptable as long as the length does not exceed 40 characters

`testout` and `treeout`

Further definition of these options is contained in the FID manual.

When the check box next to an option name is selected, then the input is enabled. Here is an example of what the Command-Line Options dialog looks like with a few options selected:

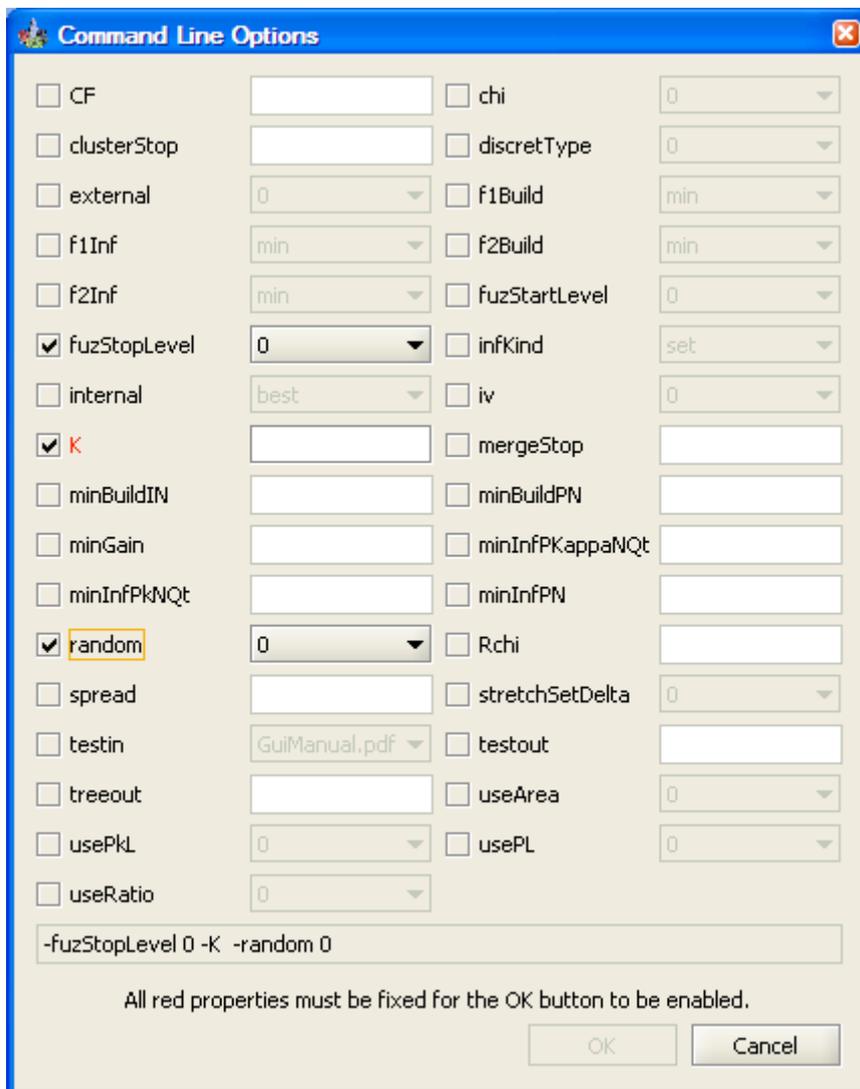


Figure 9 Command-Line Options Dialog with Empty K

There are several important things to notice here:

- Because `internal`, `K`, and `random` are selected, the summary field near the bottom shows what would be passed to the command-line version of FID, as though command-line options were being passed. The summary has the format `-key1 value1 -key2 value2 ...`, summarizing the information from the dialog. Note that there is no value between `-K` and `-random` because no value has been entered for `K`.
- The label for `K` is red. This happens because any selected text field must have some value in it, in addition to the other restrictions each field has. So, `K` is invalid.
- A line saying “All red properties must be fixed for the OK button to be enabled.” has appeared and the `OK` button is disabled. Both of these changes happened because one of the properties has an invalid value.

Now, let's change the values:

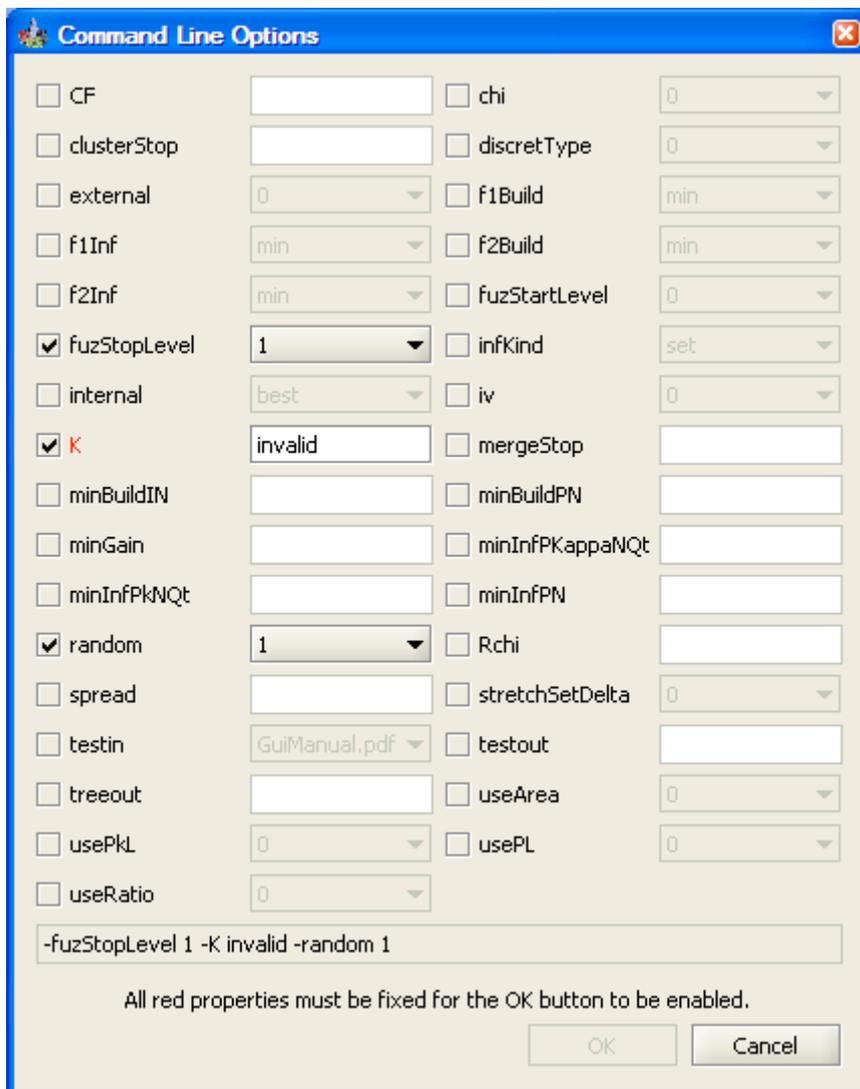


Figure 10 Command-Line Options Dialog with Non-Numeric K

As you can see, all three values have changed. This is not a problem for `internal` or `random`, because their values are limited to legal choices. However, `K` is still invalid, because the word “invalid” cannot be parsed into a number, so we cannot accept the command-line option.

Note, though, that the summary has changed to incorporate the new values.

Let's try again:

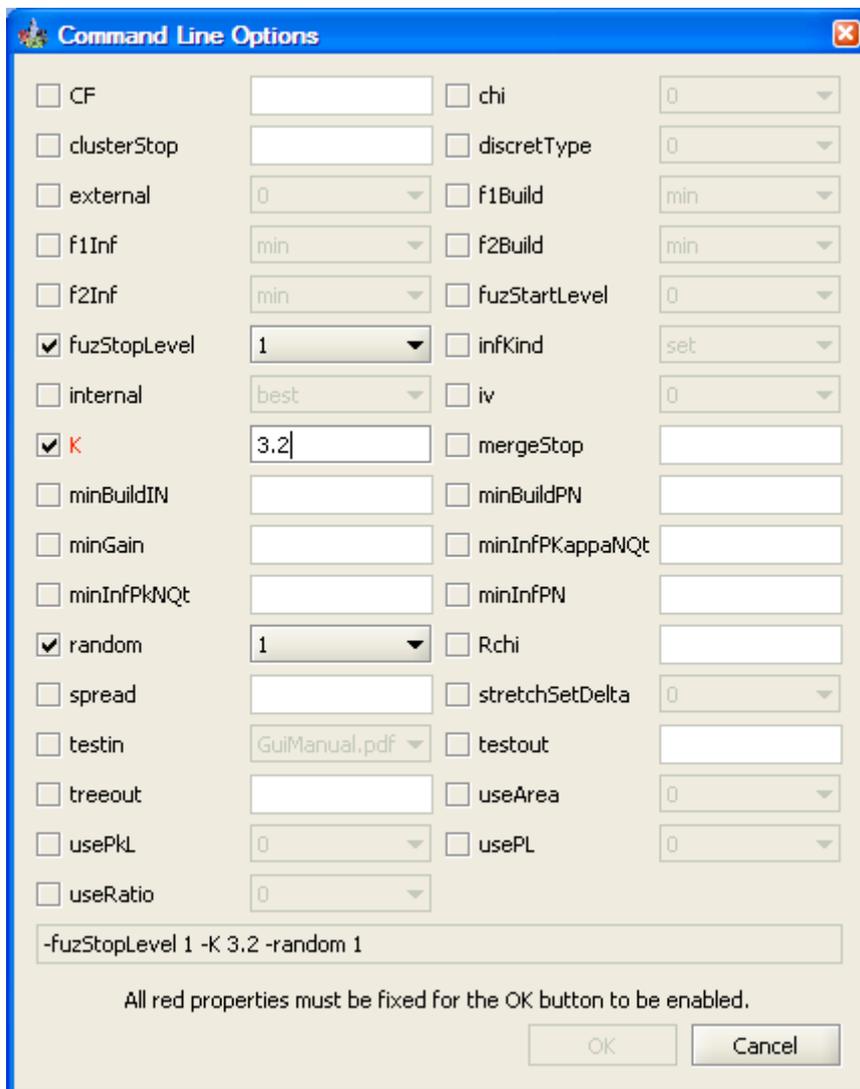


Figure 11 Command-Line Options Dialog with K Out of Range

We're getting closer, but we still cannot accept these options. Why? Because K is restricted to the range from 0 to 1, and 3.2 is outside the range (see the FID manual for more details).

One last try:

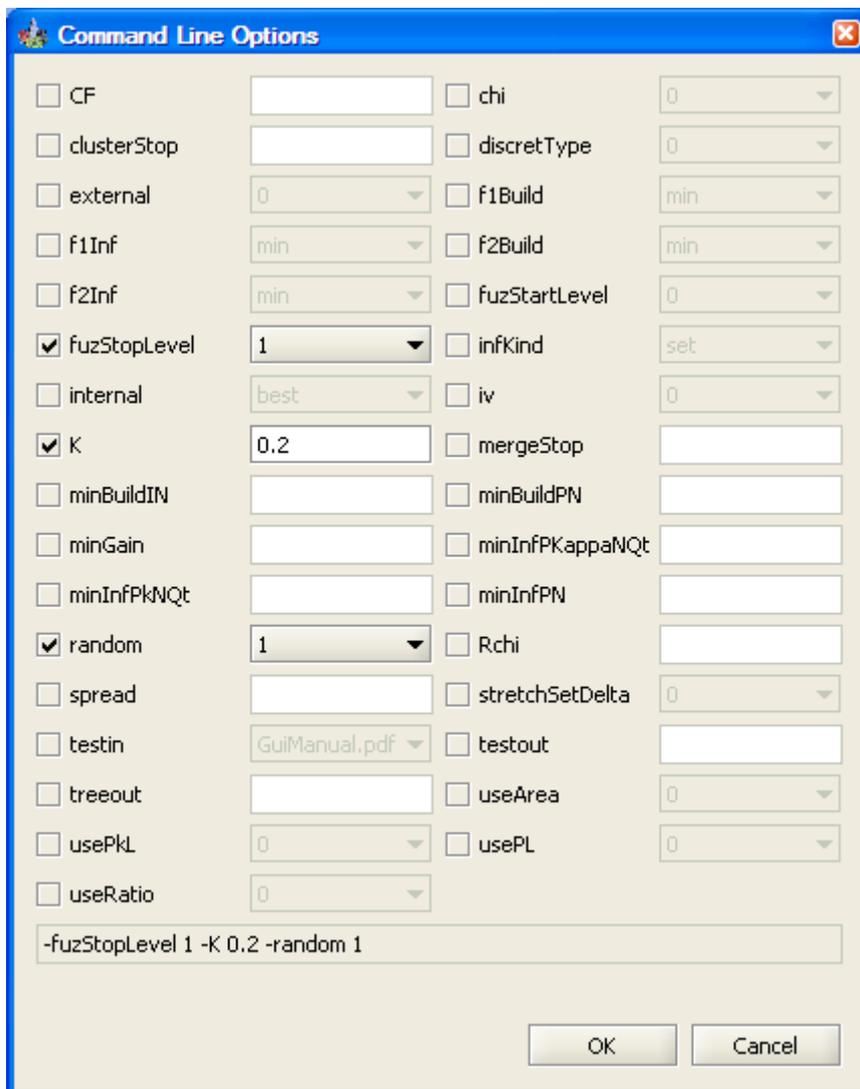


Figure 12 Command-Line Options Dialog with Valid Values

Success! We now have valid values for all selected properties. No labels are red, the OK button is enabled, and the nagging text at the bottom has disappeared.

This is what the File Definition Page looks like after clicking the OK button on the Command-Line Options dialog:

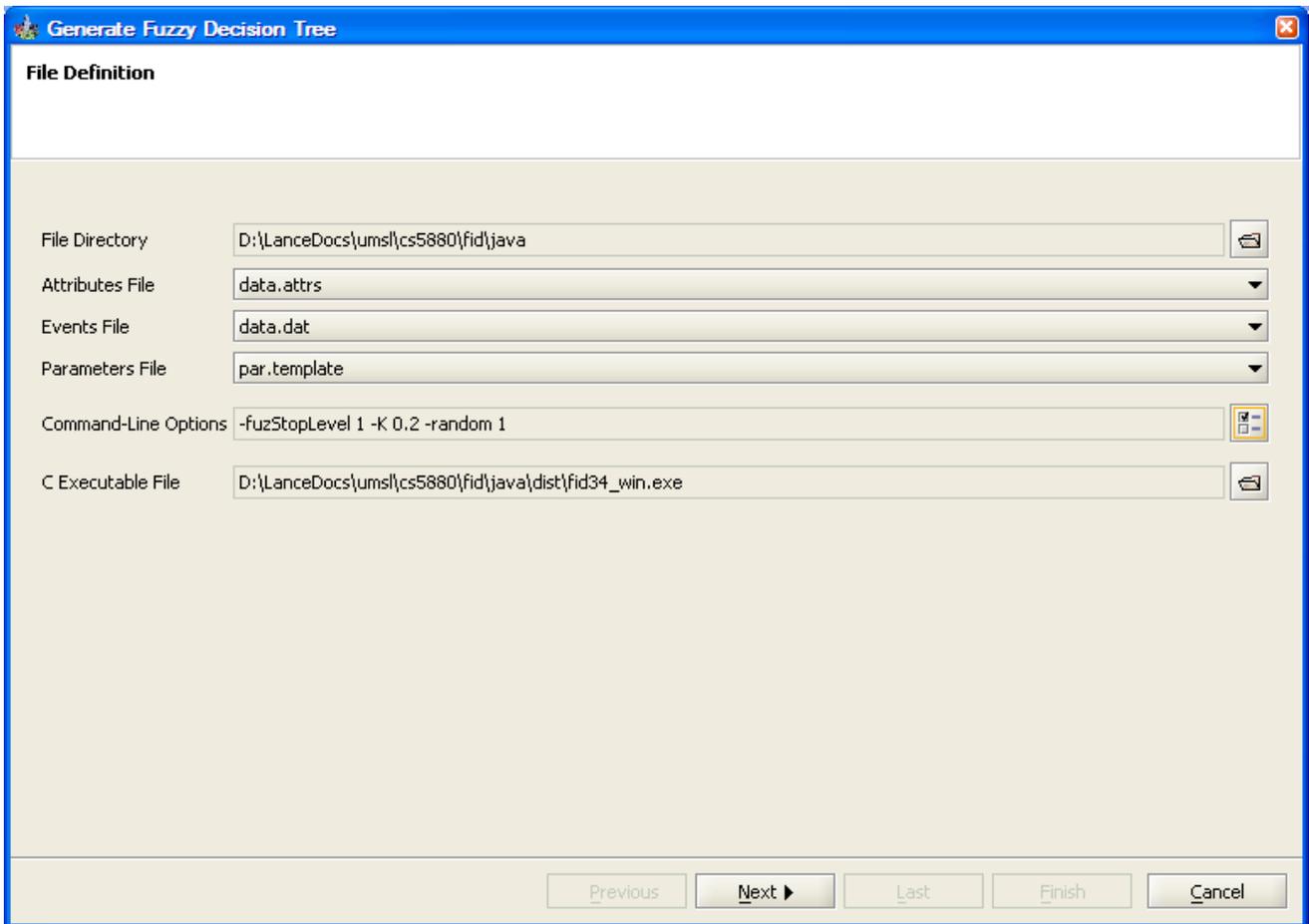


Figure 13: Complete File Definition Page

Now we can click on the `Next` button and move on to the Input Display Page.

4.2 Input Display Page

In the Input Display Page, we see the advantage having a GUI provides us over the command-line version. This page displays the information in the Attributes File and Events File graphically. It also allows for modifying the data in the Attributes File and manual partitioning of Attributes.

Let's take a look at what the Input Display Page shows for the sample input:



Figure 14 Input Display Page

The Class definition is displayed at the left. The two decision possibilities, Yes and No, are displayed. In addition, the decision values of the events are shown on the graph (the location along the x-axis is meaningful to indicate the value; the location along the y-axis is useful only to show if there are multiple events at the same value). There were two events listed with non-linear decisions of No, and those are indicated below the graph.

As you can see, there are two Attributes, X and Size.

X is a nominal Attribute, so the possible values are listed: A, B, and C. In addition, the information from the Events File is displayed: there are six events in the input, 2 each for A and B, and 2 listed as unknown.

Size is a linear Attribute with a range of 0.0 to 1.0, and a graph of the three divisions is displayed. In addition, five of the events were at discrete locations and one was listed as Large. The five discrete values are indicated as black dots on the graph (the location along the x-axis is meaningful to indicate the value; the location along the y-axis is useful only to show if there are multiple events at the same value). The value of Large is noted below the graph.

Let's scroll down on the attributes to see what else there is:



Figure 15 Input Display Page Scrolled to the Bottom

Below each linear attribute's graph, there are two buttons: `Manually Discretize` and `Revert`. We will discuss using these buttons in a later section.

Please click the `Finish` button to process this input. What happens now is that the input files specified on the File Definition Page are passed to the C Engine (the same application that runs command-line FID). The results are then viewed within the main application window.

4.3 Output View

Upon clicking the `Finish` button, the C Engine processed the input defined in the Wizard and printed the long tree results to `tree.file` (or whatever was defined as `treeOut` in the Command-Line Options Dialog or the Parameters File). The FID GUI then parsed that file to create the output tree view in the right half of this view.

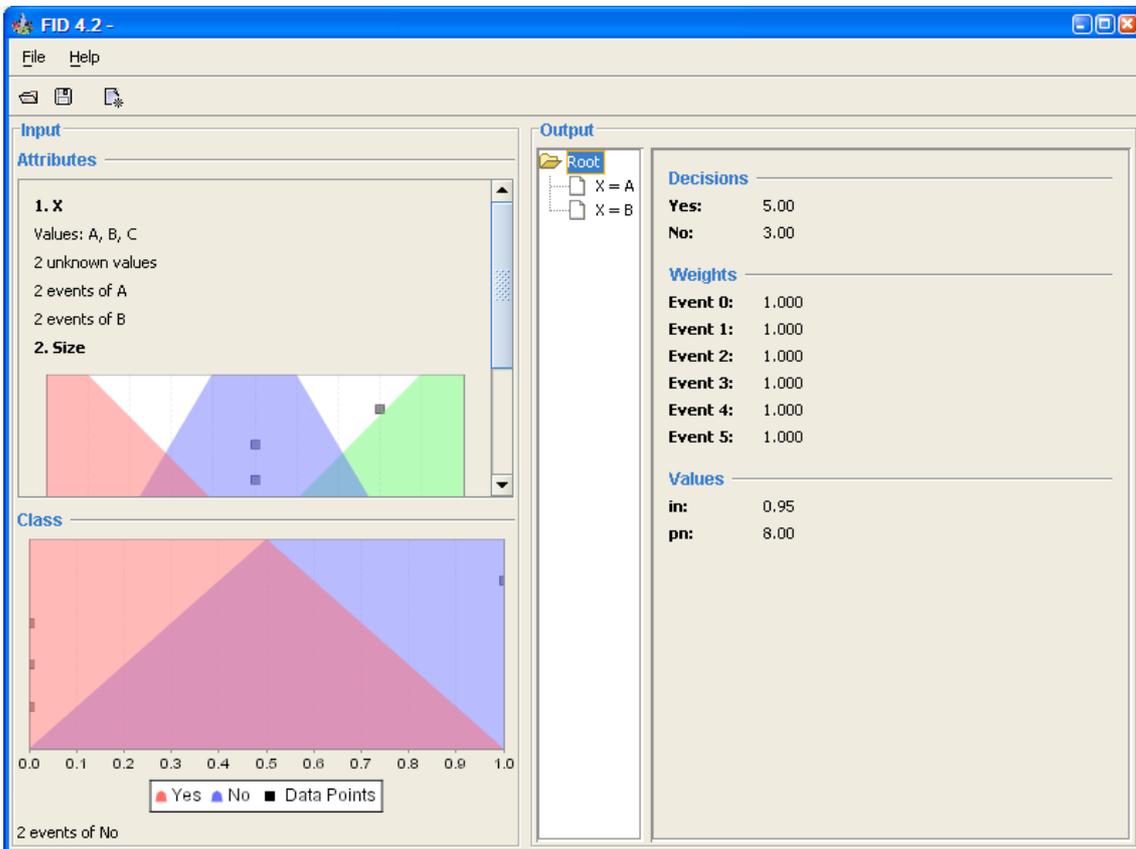


Figure 16 Output View

The right part of the Output section updates when different parts of the output tree are selected:

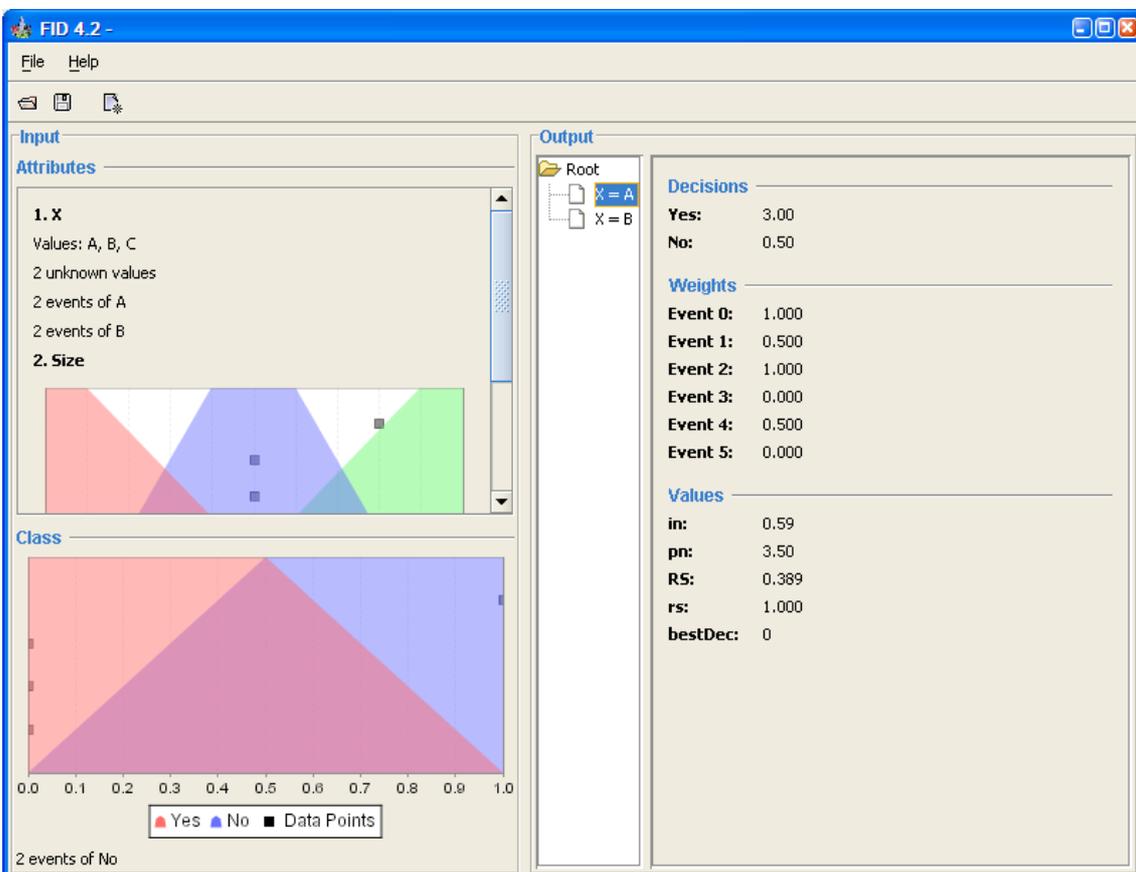


Figure 17 Selecting Tree Leaf

The information show in the right section is all the information available in the long-form tree output file (this is why the `-tree` option is disabled in the Command-Line Options Dialog).

The left part of the view displays the Attributes and the Class information, similar to the displays in the Input Display Page.

If Cross Validation was used, the results text file is displayed instead of the standard output view, since it is not possible to display the results of each fold as a tree.

4.3.1 Saving Output

At this point, you can save the input and output for later viewing. Using either the Save button in the tool bar or the Save option in the File Menu, you can request to save the current state. The following dialog box will appear:

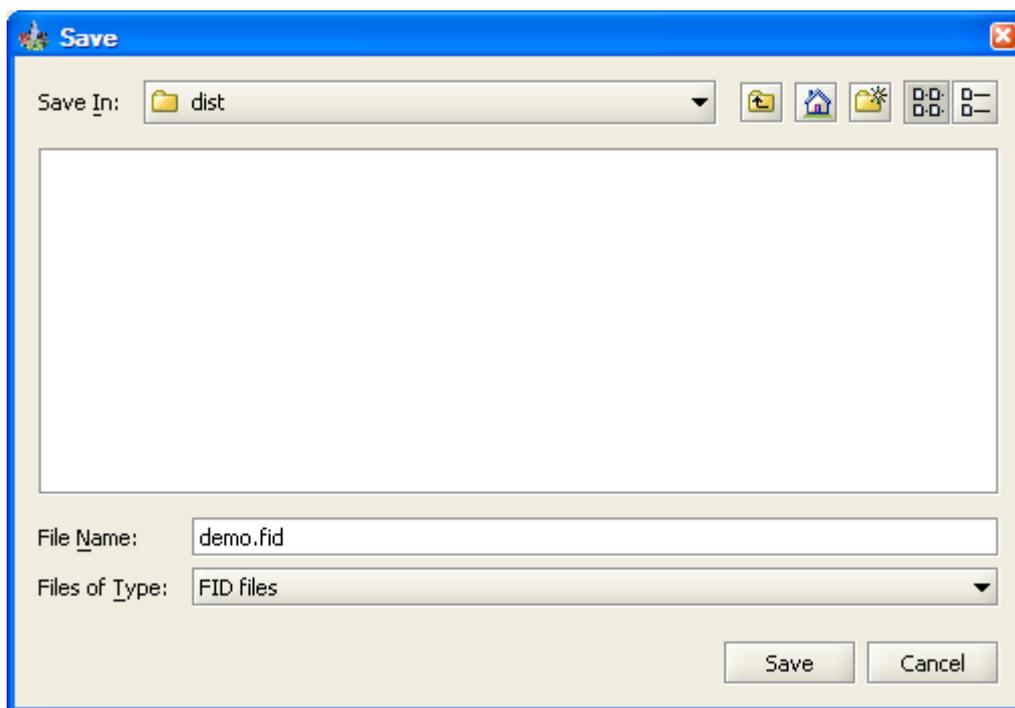


Figure 18 Saving State

You can specify a filename with a `.fid` extension. If you do not, `.fid` will be appended to the name of the file.

Click the `Save` button, and the file will be created. The only visible change in the primary view will be the addition of the name of the file to the title bar:



Figure 19 Filename in the Title Bar

If you don't explicitly save the state, you will be asked whether you want to save unsaved state after any of the following actions:

- Launching the wizard to generate a new fuzzy decision tree.
- Loading a previously-saved file.
- Closing the application.

4.3.2 Loading Previously Saved Output

There are three ways to load previously-saved output: the Open button on the tool bar, the

Open option in the File Menu, or the button labeled “Open a Previously-Generated Fuzzy Decision Tree” on the initial startup page. Through any of these means, the following dialog box is launched:

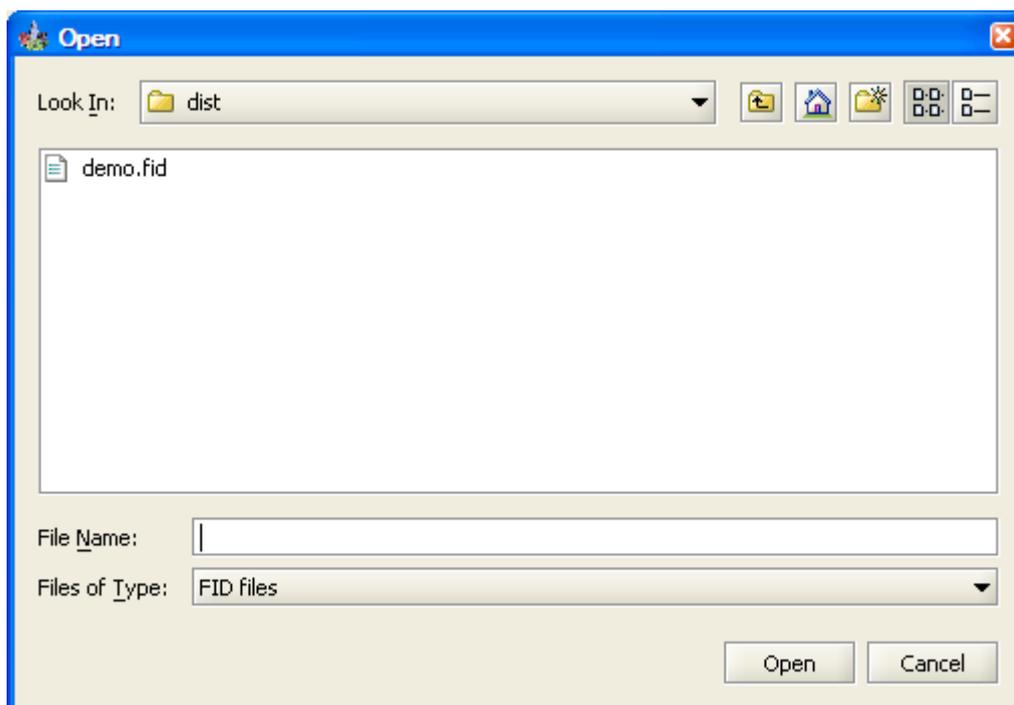


Figure 20 Open Dialog

Only files with a `.fid` extension are displayed. Select an appropriate file and click the Open button, and the previous session will be loaded into the application.

If an unsaved session exists when you request to open a previous session, you will be asked whether you want to save the current session first.

5 Running the FID GUI – Unpartitioned Attributes

The FID GUI can also help you with unpartitioned Attributes (also called undiscretized Attributes). To demonstrate these capabilities, we created versions of `data.attrs` and `data.dat` that did not define the `Size` attribute (and added a `Density` attribute for later examples). These files are called `disc.attrs` and `disc.dat` in the following example (these files are not included in the distribution).

5.1 File Definition Page

The File Definition Page works the same as before, except we select the new files and select the file `disc.dat` for the `-testin` command-line option:

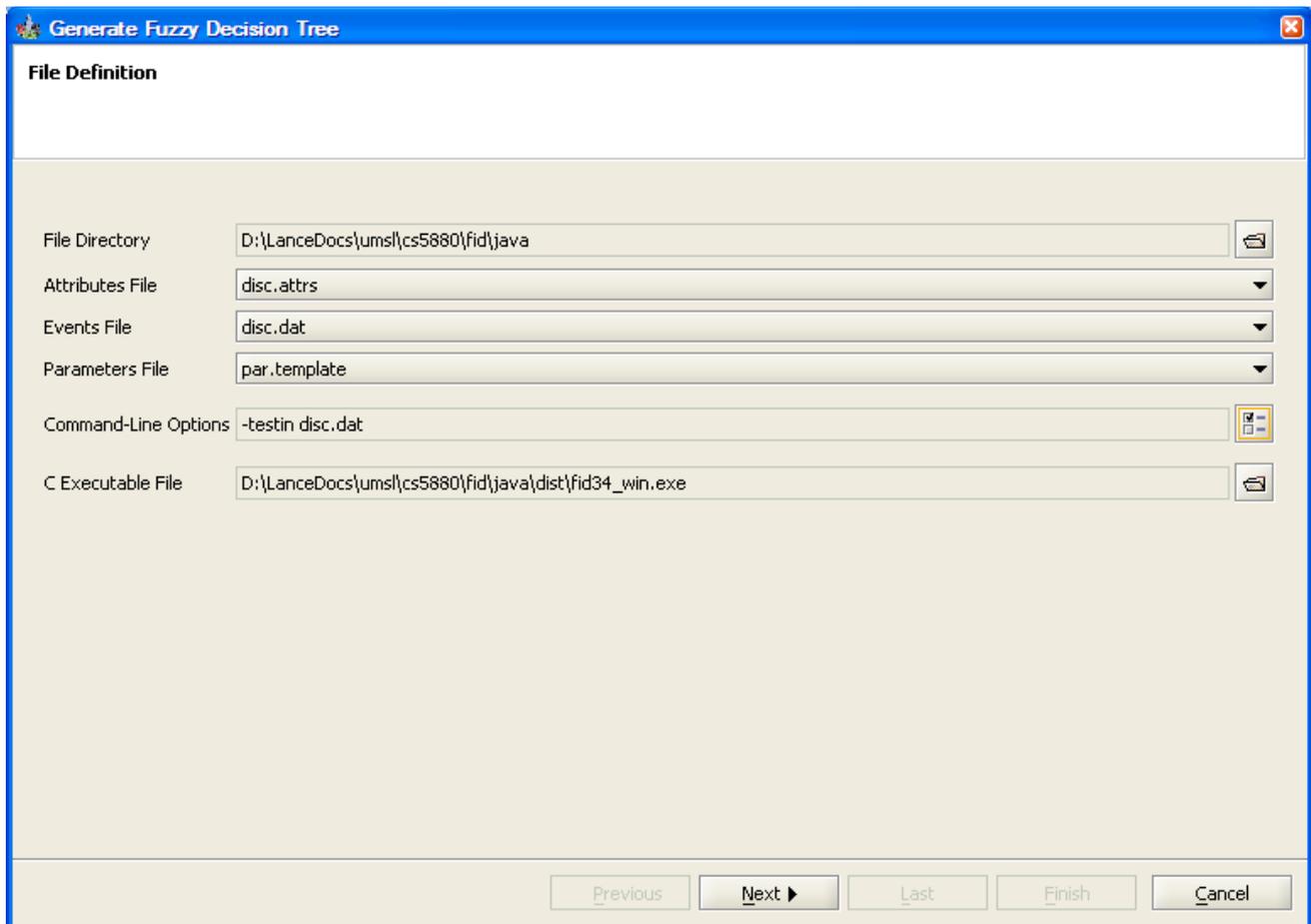


Figure 21: File Definition Page with Unpartitioned Attributes

When you click `Next`, the GUI executes the command-line version with the `-p` option. This creates a new file called `disc.attrs.gen` that has the `Size` Attribute discretized. The file is saved in the directory specified by `File Directory`.

5.2 Input Display Page

When the Input Display Page is shown, things are slightly different than before:

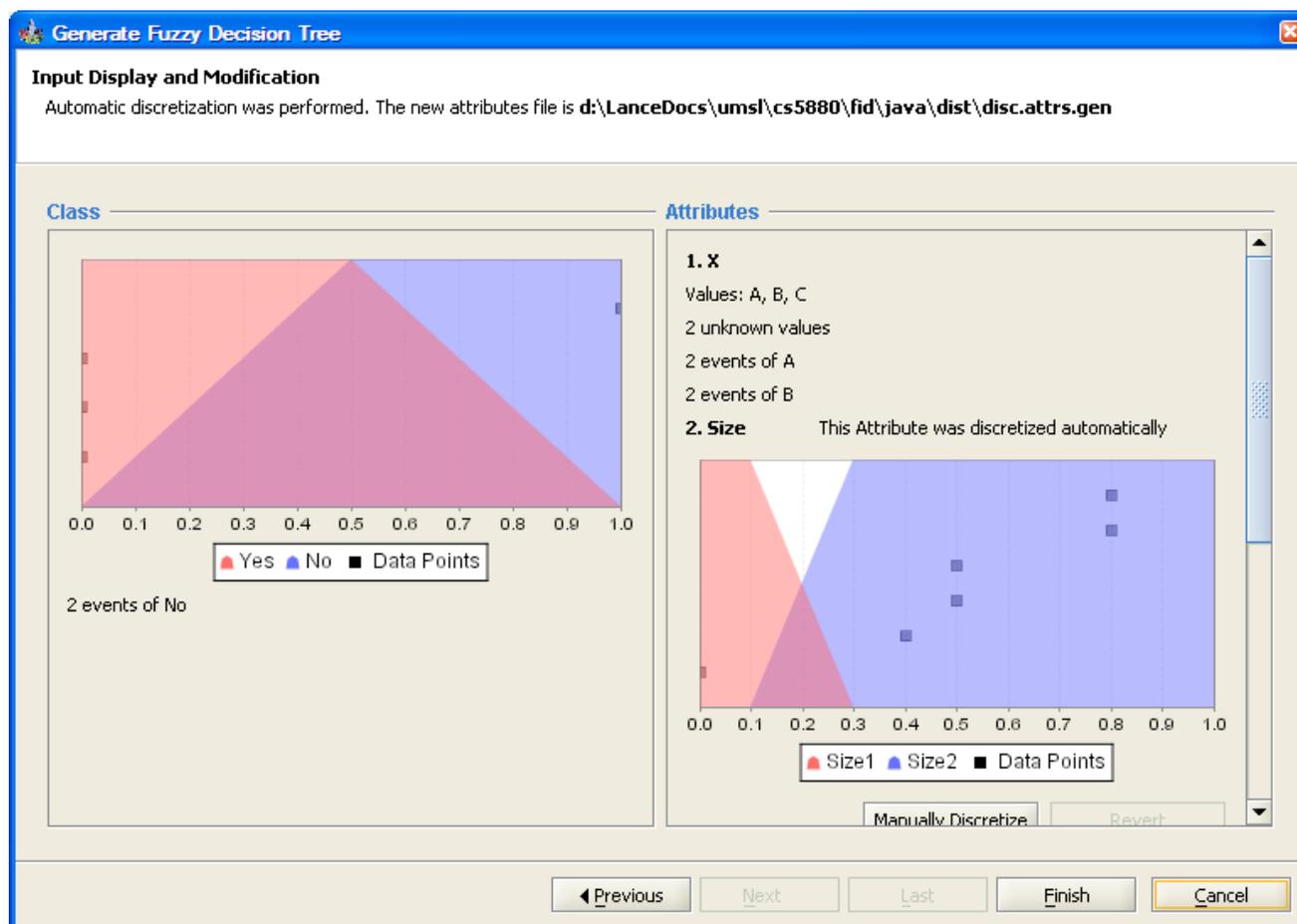


Figure 22 Input Display Page with an Automatically-Discretized Attribute

At the top of the page, you are informed that automatic discretization was performed, and you are given the location and name of the generated Attributes File. Also, the Attributes that were discretized are marked by their names with the following text: “This Attribute was discretized automatically”. Finally, the actual Attributes for Size are different than they were before.

Let's click Finish again.

5.3 Output View

The output view is similar to how it was before. As for Input Display Page, though, the Attributes that were discretized are marked by their names with the following text: “This Attribute was discretized automatically”.

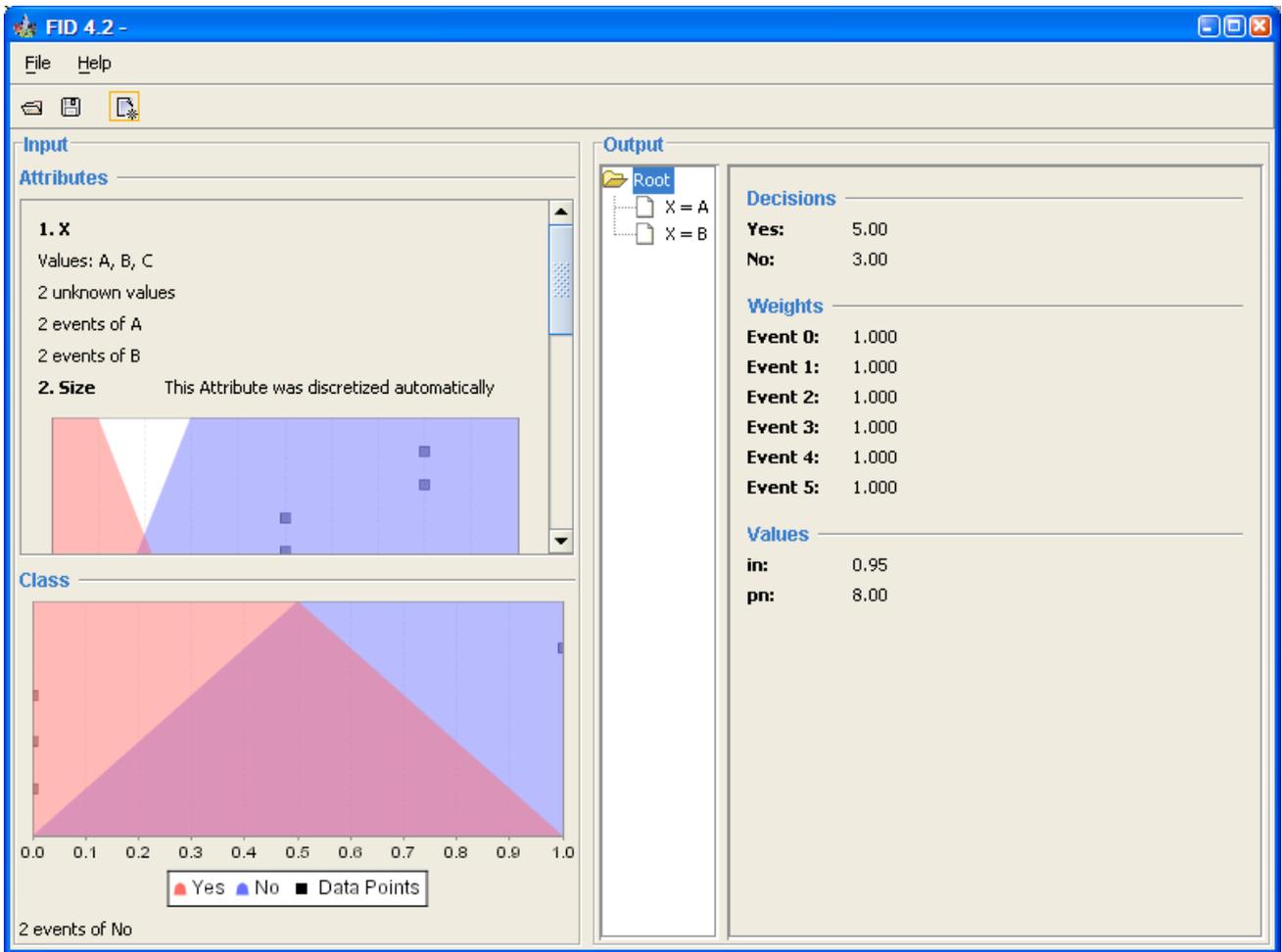


Figure 23 Output View with an Automatically-Discretized Attribute

6 Manually Modifying Attributes

In addition to automatic discretization, the FID GUI allows you to manually modify the attributes. To demonstrate these capabilities, let's use the same files we used in the previous section: `disc.attrs` and `disc.dat`.

6.1 File Definition Page

The File Definition Page works the same as in section 5.1.

6.2 Manual Discretization Dialog

On the Input Display Page, let's modify the attribute `Density`. To do this, click the `Manually Discretize` button below the `Density` graph (you will have to scroll down to the button). The following dialog box is displayed:

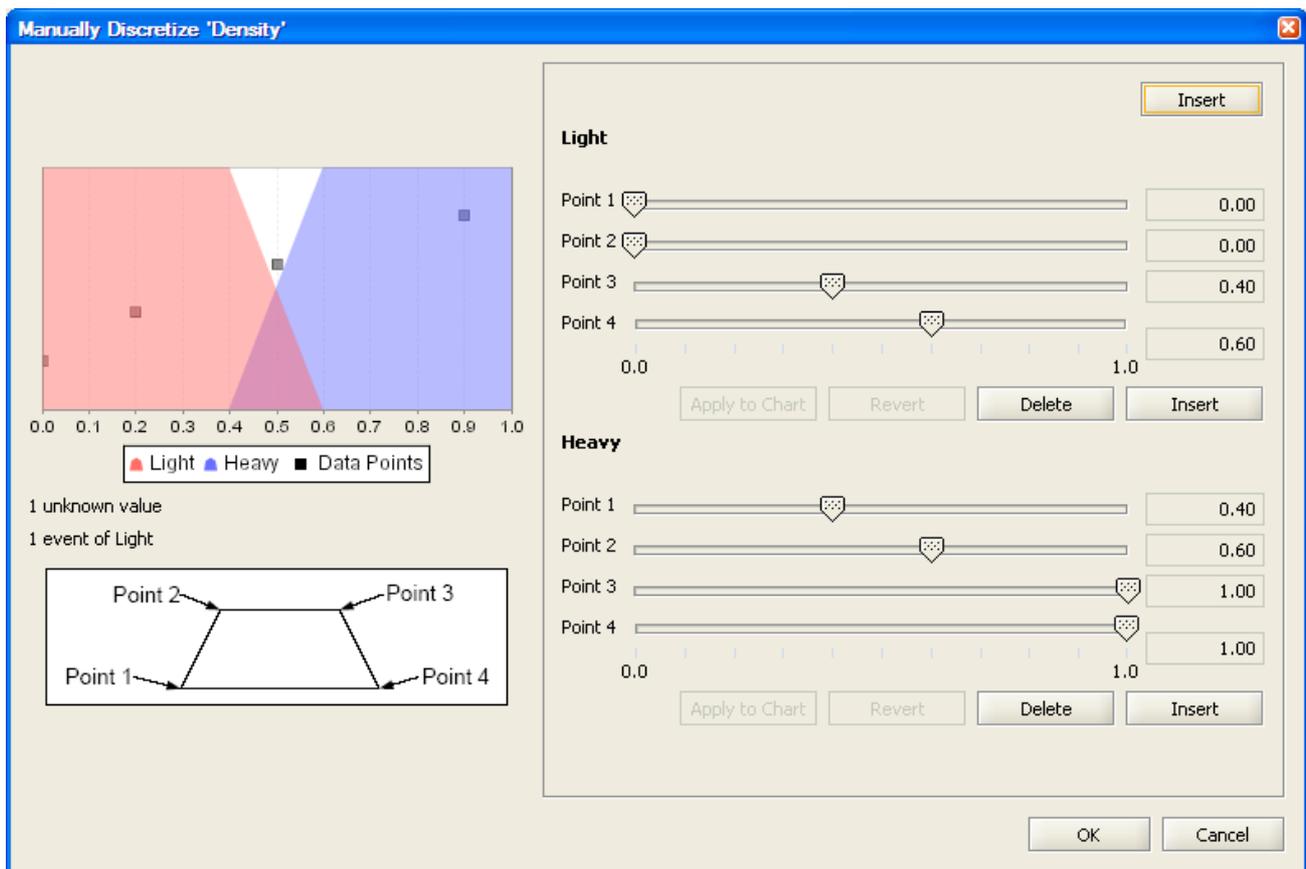


Figure 24 Initial Values for Density

There are two sections on the left and one section on the right. The top left section shows the same graph that was visible on the Input Display Page. It shows the definitions of the `Density` attribute and the data points. The unknown and nominally-defined values are specified below the image.

The bottom left section is a static image showing the meaning of `Point 1`, `Point 2`, `Point 3`, and `Point 4` as they relate to the attribute sections. Specifically, looking at the `Light` definition above, `Point 1` is the point at the bottom left of the pink trapezoid and the graph (noted as 0.0 on the slider on the right). `Point 2` is the point at the top left of the pink trapezoid and the graph (noted as 0.0 on the slider on the right). `Point 3` is the point at the top right of the pink trapezoid (noted as 0.4 on the slider on the right). `Point 4` is the point at the bottom right of the pink trapezoid (noted as 0.6 on the slider on the right).

Point 1 can be no bigger than Point 2, which can be no bigger than Point 3, which can be no bigger than Point 4.

6.2.1 Modifying Values

The sliders on the right are used to modify the shape of the trapezoid, and therefore the definition of the attribute. For example, slide `Light`'s `Point 1` slider all the way to the right.

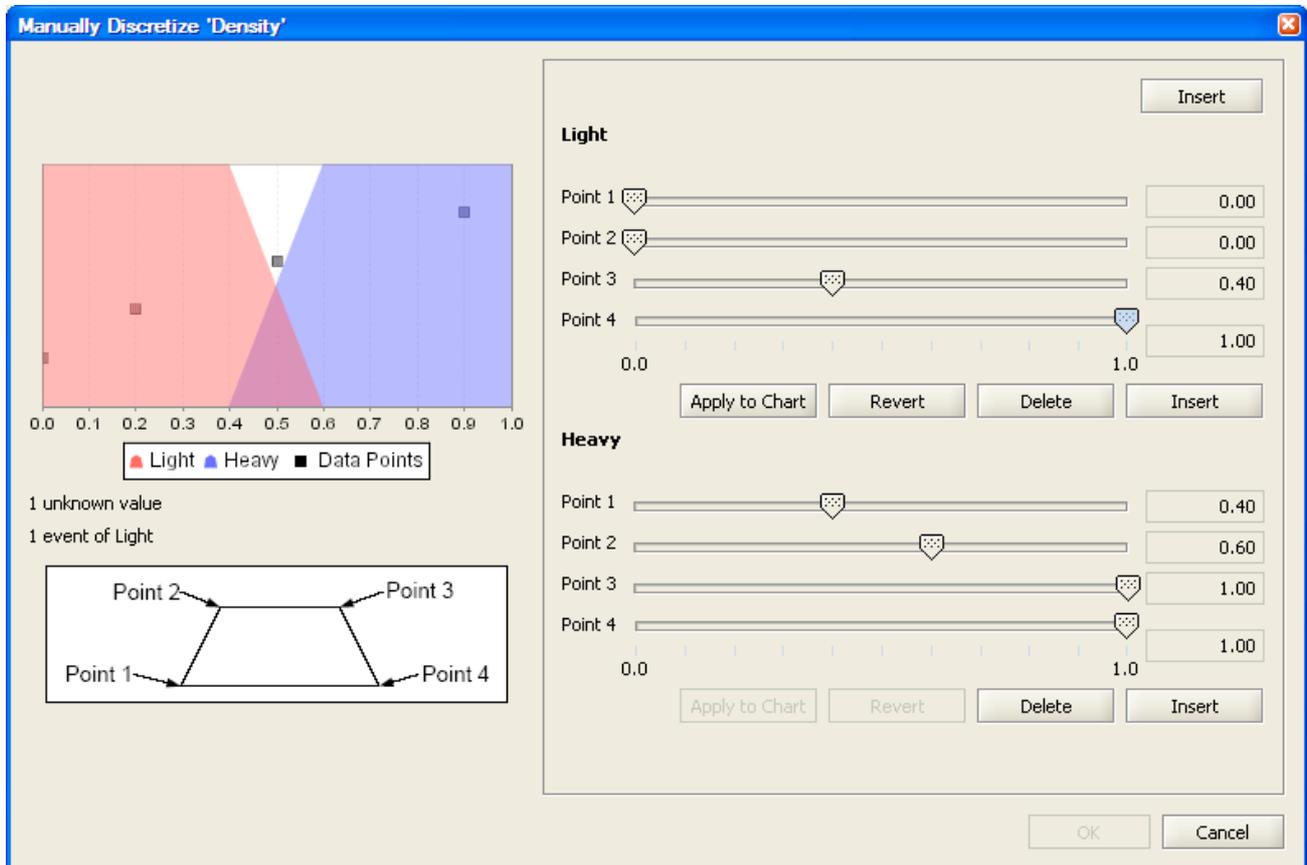


Figure 25 Valid Slider Change

With this action, four things changed:

- The value in the box on the right for `Point 1` changed to 1.00 to match the value of the slider.
- The `Revert` button became enabled. This happens whenever a change is applied to the sliders. Clicking the button will erase the change (and enable the `OK` button if this is the only definition that has been changed).
- The `Apply to Chart` button became enabled. Since this change is a valid change, you could apply the change to the definition as seen on the chart.
- The `OK` button became disabled. This button is disabled whenever there are unapplied changes.

Let's click on `Apply to Chart`:

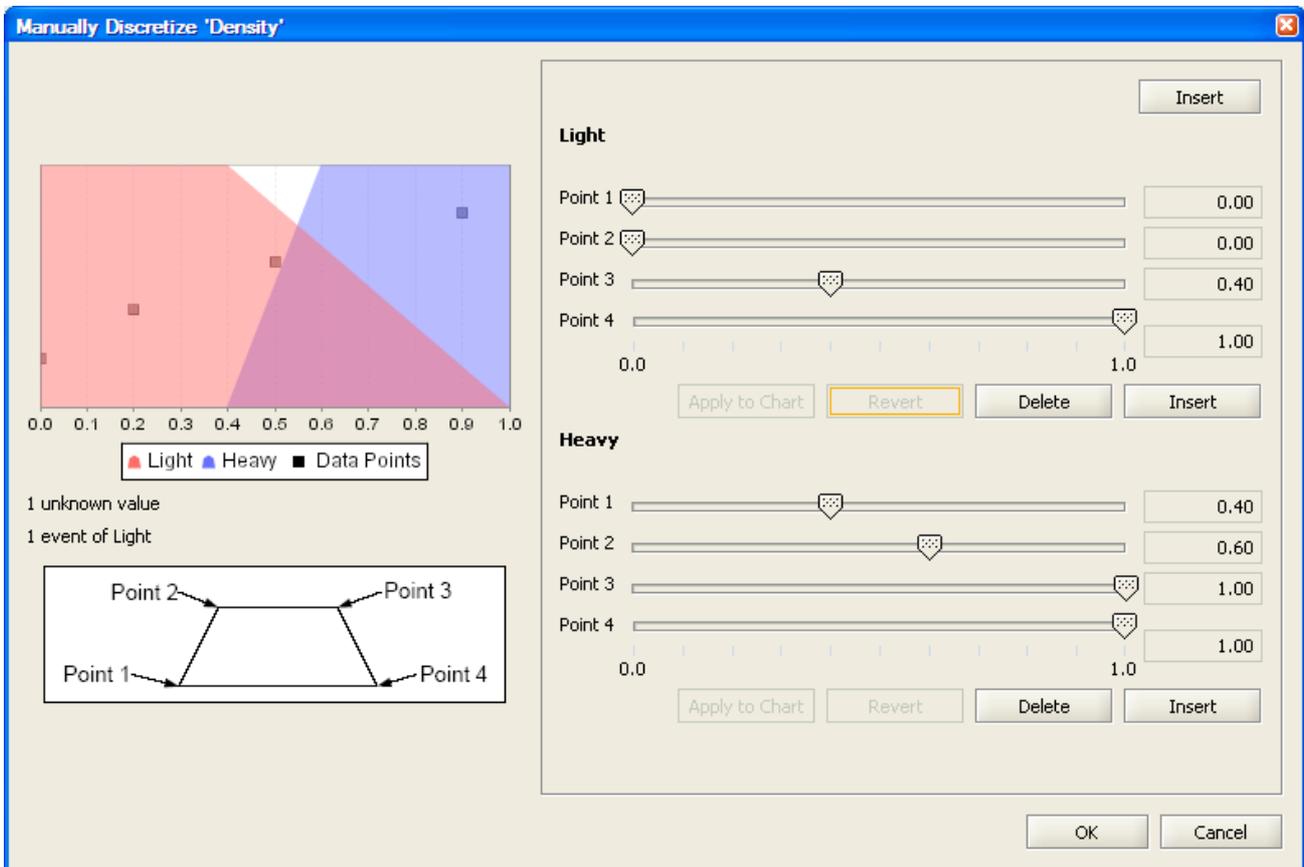


Figure 26 Slider Change Applied

With this action, four things changed:

- The shape of the trapezoid in the chart now reflects the values in the sliders.
- The `Revert` button became disabled because there are no changes to revert.
- The `Apply to Chart` button became disabled because there are no changes to apply.
- The `OK` button became enabled.

Now, slide `Light's Point 1` slider all the way to the left.

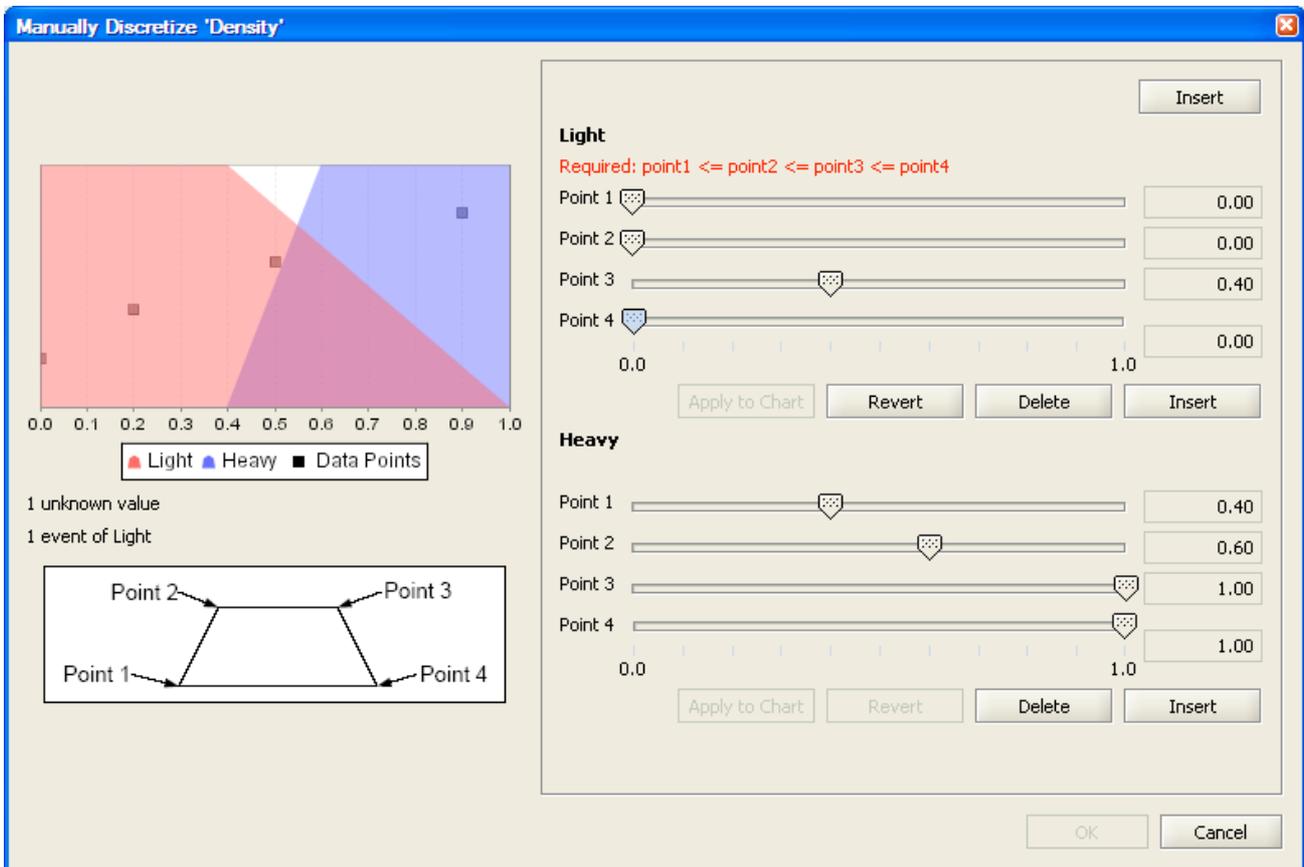


Figure 27 Invalid Slider Change

With this action, four things changed:

- The value in the box on the right for Point 1 changed to 0.00 to match the value of the slider.
- The `Revert` button became enabled.
- A reminder showed up that the points must increase in size.
- The `OK` button became disabled.

Because Point 4 cannot be smaller than Point 3, this is an invalid condition. We cannot change the attribute to these values. So, let's revert by clicking the `Revert` button. We return to the last saved condition.

6.2.2 Inserting New Definitions

Let's add a definition called `Medium` between `Light` and `Heavy`. So do so, click the `Insert` button below the `Light` sliders. The following Dialog box pops up:

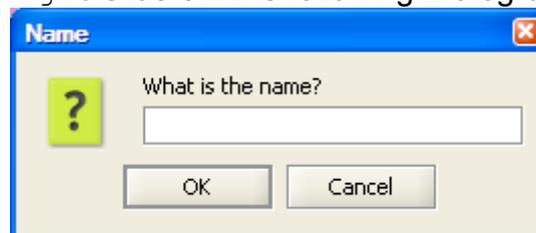


Figure 28 Insert Definition

Enter `Medium` and click `OK`. Valid names must have 1 to 39 characters, no spaces, and must be unique for the attribute.

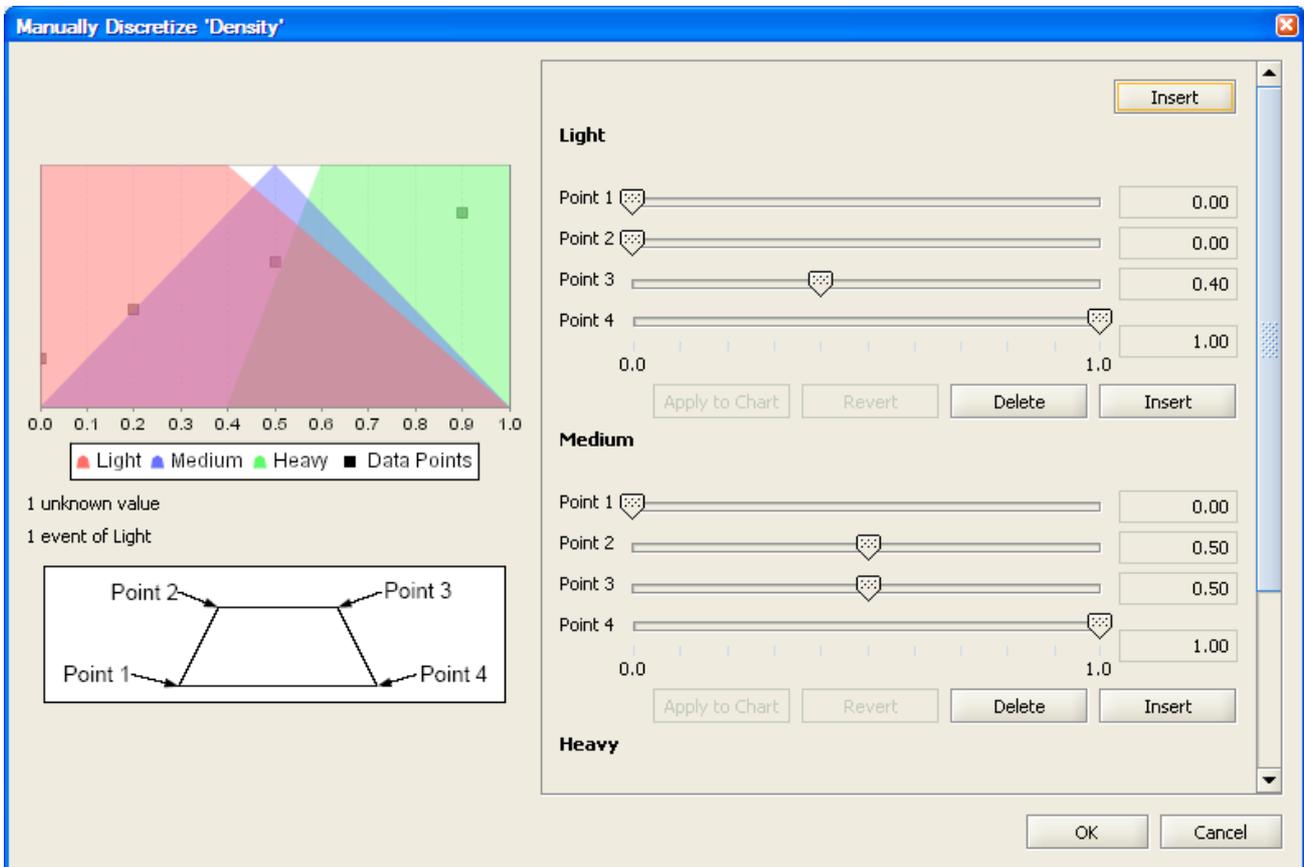


Figure 29 After Adding Medium

Medium is now added between Light and Heavy with default values, and it appears on the chart. The points for Medium can be modified as discussed above.

To add a new definition at the top, use the Insert button at the top.

6.2.3 Deleting Definitions

You can delete definitions as well. Let's delete our new Medium definition by clicking the Delete button below the Medium sliders. Your request will be confirmed:

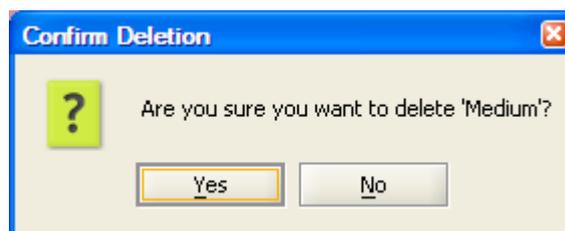


Figure 30 Confirm Definition Deletion

Click Yes, and you will see that Medium has been removed.

Next try to delete Light by clicking the Delete button below the Light sliders. You will see the following message:



Figure 31 Undeletable Definition

Light cannot be deleted because it is explicitly used as a nominal value in the data file.

Note: You can delete all the definitions within an attribute (if none are used nominally). However, you will not be able to click the OK button and save the modification because an attribute must have at least one definition.

6.3 Input Display Page

To see the results of our work, we will have to click OK and return to the Input Display Page. First, add the Medium definition back in. Then click OK.

The Input Display Page now looks like this:

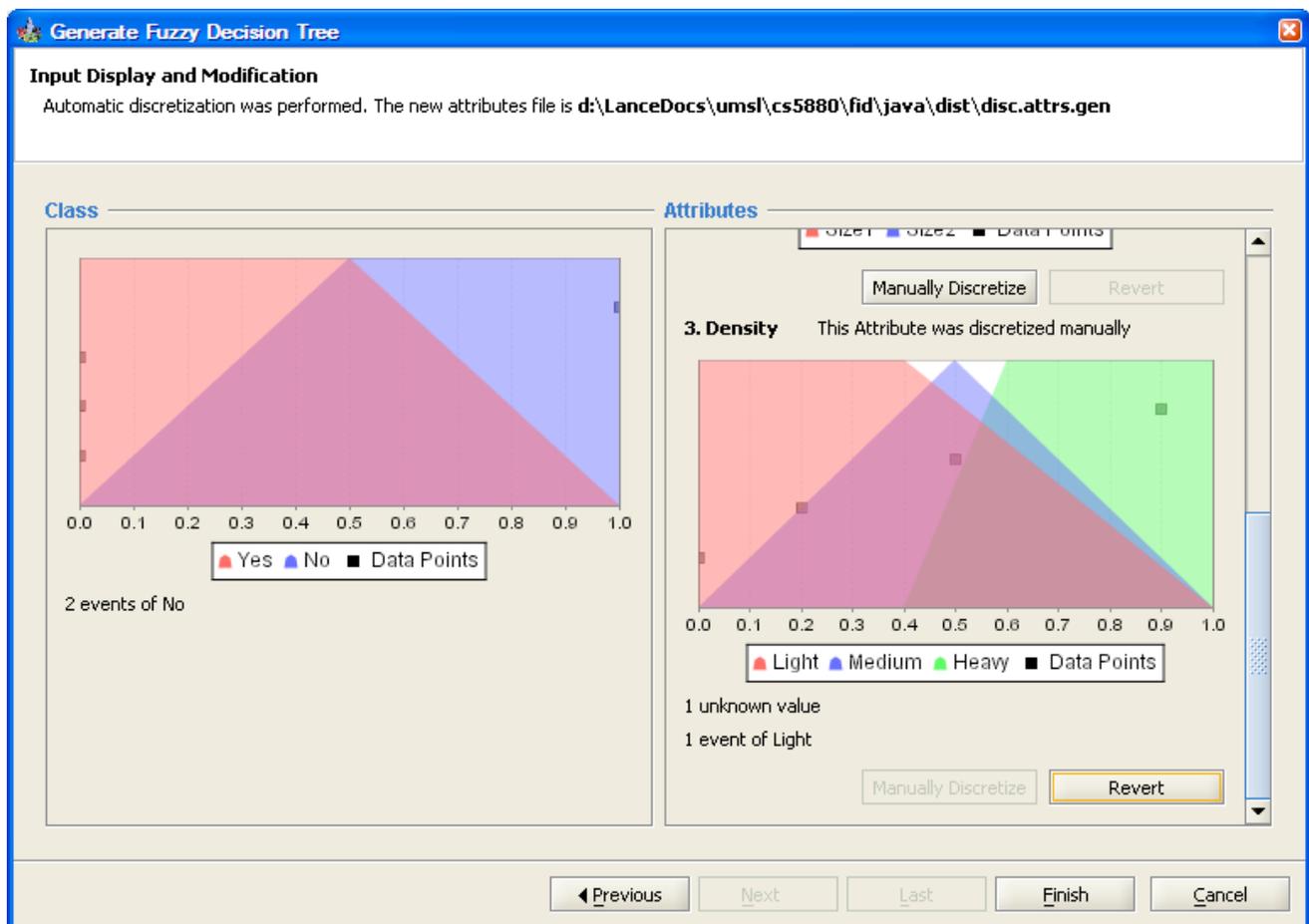


Figure 32 Input Display Page After Modification

Our changes to Light have been transferred here, and Medium has been added. Additionally, the attribute that was modified is marked by its name with the following text: “This Attribute was discretized manually”. If an automatically-discretized attribute is modified, only this message is displayed; the original message is no longer relevant. However, the note about automatic discretization stays in the header.

Let's click `Finish` again. At this point, the FID GUI writes an attributes file with the information you have specified (similar to how the command-line application writes an attributes file after automatic discretization). This file will be created in the same directory as the file it modifies, with “.man” appended to the name. So, for our example, we modified the automatically generated `disc.attrs.gen`, so the new filename is `disc.attrs.gen.man`.

6.4 Output View

The output view is similar to how it was before. As for Input Display Page, though, the Attributes that were modified are marked by their names with the following text: “This Attribute was discretized manually”.

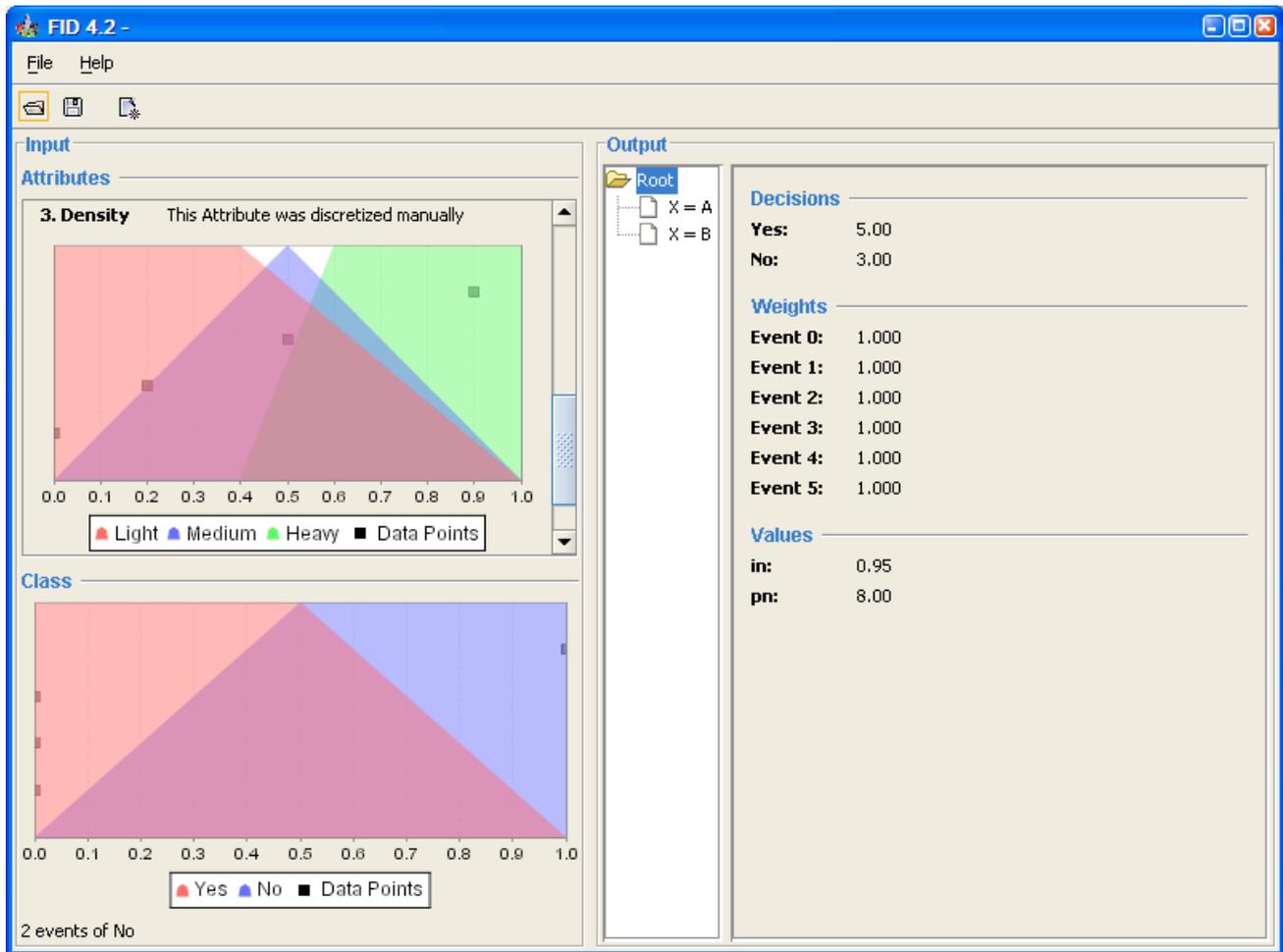


Figure 33 Output View with a Manually-Modified Attribute

7 Troubleshooting

Sometimes, the parameters or options selected are incompatible with the input data. When that happens, the command-line version, which acts as an engine for the FID GUI, exits with a message to the console. The FID GUI handles this by displaying a dialog like this:

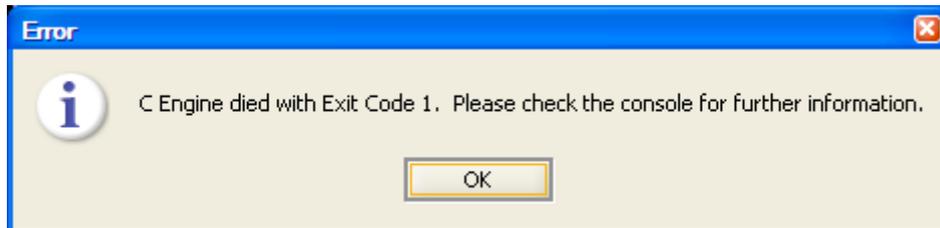


Figure 34 Error Message

If this happens, you will not be able to proceed to the next page.

If you see a message like this, click the `Cancel` or `Finish` button and look at the console to see the error message. If you ran the FID GUI by clicking on an icon, either rerun the FID GUI from a DOS console to see the error, or directly run the command-line version with the same input, parameters, and options.