Expert Data Mining:

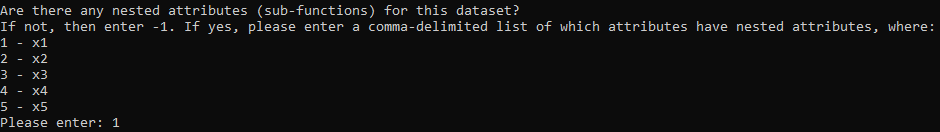
Nested Functions

# Introduction

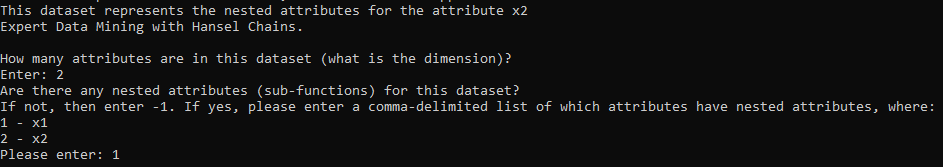
The Expert Data Mining software is now able to use nested functions. Essentially, any attributes can have sub-attributes, meaning that there can be multiple sequences of asking questions to the user. A common function that we like to use for testing is “x1x3x5 v x2x3x5 v x4x5.” Any one of these attributes can have a couple sub-attributes. Let’s say that attribute x3 has two sub-attributes, x3.1 and x3.2. These two attributes will have their own Boolean function which makes the attribute x3 true or false. The program uses both a top-down approach (identifying sub-attributes) and a bottom-up approach (identifying groups of attributes). The top-down approach is easier to use due to the nature of the user input. Both of the implementations are recursive depth-wise.

# Top-down Approach

The implementation is quite simple: before we ask the user any questions, we ask if there are any sub-attributes.

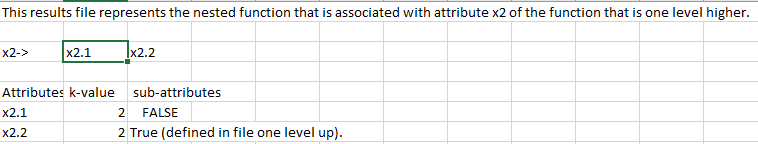


In this example, we entered one so that the attribute x1 would have sub-attributes. Multiple attributes can be designated to have sub-attributes. After this, the sequence of questions will be the exact same as it was in its previous implementations. Once all the questions have been answered, we move onto asking questions about the sub-attributes, starting with how many of them there are.



We entered two so that that there are 2 attributes in the dataset (dimension of 2), and then specified that there are sub-sub-attributes. The process is iterative, so any depth of sub-attributes could be used, although it would likely be better to group the attributes in a more efficient way if the depth gets too deep.

Furthermore, the output file of the results uses a slightly different nomenclature and format to account for the sub-functions and sub-attributes. First, the file is named “results\_of\_function\_*n*.csv” where n starts at the number 0 and increases. If n is equal to 0, then that results file represents the parent Boolean function. If *n* is equal to 1, it is for the sub-function. If it is equal to 2, it is for the sub-sub-function, and the pattern continues. If there are multiple attributes with sub-attributes in the same depth, then the results will be printed in the same file. An example of what is new in the output file is shown below



This file is associated with the function that is in the results file that is labeled with *n* – 1, so if this file is “results\_of\_function\_*1*.csv” (which it is in the attached files), then the parent will be “results\_of\_function\_0.csv.” The attributes are named slightly differently as well. Since the attributes in the picture represent its parent attribute “x2,” they are given the name “x2.1” and “x2.2.” for sub-sub-attributes, this can be quite confusing, so multiple levels aren’t considered in this aspect. For a sub-attribute of “x1.2,” within the results file, the parent attribute will just be called “x2,” for example.

# Bottom-up Approach

The bottom-up approach is similar in that it can achieve the same result as the top-down approach, but it takes a bit more work and pilot questions to get there. Note: using a top-down approach and bottom-up approach at the same time isn’t allowed for groups. If we create a grouping of attributes, it doesn’t make sense to ask for sub-attributes of the attribute that correlates to that group (they are the attributes in that group). However, it is possible and allowed to create sub-attributes before creating any groups.

In the implementation, before we ask any questions, and only if there are more than 5 attributes, we ask the user if there are any groups of attributes. Then, if those groups of attributes still have more than 5 attributes, we ask the user to group them further, creating more than 2 levels of attributes. In the implementation, we don’t ask for specific attributes from the user because the program uses generic names in the first place, although this is definitely possible. Instead, we simply ask how attributes are in the group for simplicity of user input. For example, let’s say that we have 8 attributes, x1-x8. The user splits these into 2 groups, one group has attributes x1-x6 (y1), and the other group has attributes x7-x8 (y2). Now we have 2 groups of attributes. Since we have 2 groups, we can start asking questions based on those groups. Let’s say that those 2 groups are represented by attributes y1 and y2. Once all questions about y1 and y2 have been asked (the overall Boolean function), we start asking questions the sub-attributes of y1 and y2. Since the first group still has more than 5 attributes, we ask the user to split them further. The user splits these into two equal groups, x1-x3 (z1) and x3-x6 (z3). These 2 groups have less than 5 attributes, so we can start asking questions based on z1 and z3. Once that process is finished, we start asking questions on x1-x3 and x4-6. Finally, we move onto the sub-attributes of y2, which are x7 and x8, and start asking questions based on those attributes. Here is a depiction of each grouping of attributes:

Parent function: y1-y2

y1: z1-z2

z1: x1-x3

z2: x4-x6

y2: x7-x8

We end up with 5 functions that contain attributes y1 and y2, z1 and z2, x1 to x3, x4 to x6, and x7 and x8.

In the program, we use generic named attributes, so the results print-out would be less clear than this example. When this program gets a nicer UI and is implemented with visualization, it would be beneficial to require that users name all of their attributes and groups of attributes. The process will of course be longer, but results would be easier to read.

Below is an exact transcript of user input that created the example that is depicted above. It is a lengthy process, and creating the same hierarchy of functions might be easier to execute with the top-down approach.

How many attributes are in this dataset (what is the dimension)?

Enter: 8

Are there any nested attributes (sub-functions) for this dataset?

If not, then enter -1. If yes, please enter a comma-delimited list of which attributes have nested attributes, where:

1 - x1

2 - x2

3 - x3

4 - x4

5 - x5

6 - x6

7 - x7

8 - x8

Please enter: -1

Are there any groupings of attributes?

If not, then enter -1. If yes, please enter how many groups there are (please keep this to 2-5 groups):

2

Please enter the number of attributes that are in group 1: 6

Please enter the number of attributes that are in group 2: 2

Since parent attributes have been specified (the groupings), the expert data mining process will start with those attributes first.

Expert Data Mining with Hansel Chains.

What order to use for the Hansel Chains?

0 - Shortest Hansel Chain First

1 - Longest Hansel Chain First

2 - Manual Hansel Chain Order

3 - Default Hansel Chain Algorithm Order

Enter: 0

Is there any attribute which must be true for the datapoint to be true?

Enter the number assigned to each attribute or -1 if there is no such attribute.

If there are multiple attributes, separate them with a comma.

a - 0

b - 1

Enter: 1

Do you want to use chain jumping (1/0)?

Enter: 0

Use majority flag(1/0)?

Enter: 0

Enter the class for this data point:

a = 0

b = 1

Enter Class: 0

Enter the class for this data point:

a = 1

b = 1

Enter Class: 1

Checking for f-changes:

Number Datapoint Class

1.1 1, 0 0

2.1 0, 0 0

2.2 0, 1 0

2.3 1, 1 1

Please enter the number of any vectors which need any changes in a comma-separalited list (e.g. 1.1, 3.2, ..., 7.4):

The vector specified does not exist, so it has been skipped.

This dataset represents the nested attributes for the attribute x1

Expert Data Mining with Hansel Chains.

Are there any groupings of attributes?

If not, then enter -1. If yes, please enter how many groups there are (please keep this to 2-5 groups):

2

Please enter the number of attributes that are in group 1: 3

Please enter the number of attributes that are in group 2: 3

Since parent attributes have been specified (the groupings), the expert data mining process will start with those attributes first.

Expert Data Mining with Hansel Chains.

What order to use for the Hansel Chains?

0 - Shortest Hansel Chain First

1 - Longest Hansel Chain First

2 - Manual Hansel Chain Order

3 - Default Hansel Chain Algorithm Order

Enter: 0

Is there any attribute which must be true for the datapoint to be true?

Enter the number assigned to each attribute or -1 if there is no such attribute.

If there are multiple attributes, separate them with a comma.

a - 0

b - 1

Enter: 0

Do you want to use chain jumping (1/0)?

Enter: 0

Use majority flag(1/0)?

Enter: 0

Enter the class for this data point:

a = 1

b = 0

Enter Class: 0

Enter the class for this data point:

a = 1

b = 1

Enter Class: 1

Checking for f-changes:

Number Datapoint Class

1.1 1, 0 0

2.1 0, 0 0

2.2 0, 1 0

2.3 1, 1 1

Please enter the number of any vectors which need any changes in a comma-separalited list (e.g. 1.1, 3.2, ..., 7.4):

The vector specified does not exist, so it has been skipped.

This dataset represents the nested attributes for the attribute x1

Expert Data Mining with Hansel Chains.

What order to use for the Hansel Chains?

0 - Shortest Hansel Chain First

1 - Longest Hansel Chain First

2 - Manual Hansel Chain Order

3 - Default Hansel Chain Algorithm Order

Enter: 0

Is there any attribute which must be true for the datapoint to be true?

Enter the number assigned to each attribute or -1 if there is no such attribute.

If there are multiple attributes, separate them with a comma.

a - 0

b - 1

c - 2

Enter: 2

Do you want to use chain jumping (1/0)?

Enter: 0

Use majority flag(1/0)?

Enter: 0

Enter the class for this data point:

a = 1

b = 0

c = 1

Enter Class: 1

Enter the class for this data point:

a = 0

b = 0

c = 1

Enter Class: 0

Enter the class for this data point:

a = 0

b = 1

c = 1

Enter Class: 1

Checking for f-changes:

Number Datapoint Class

1.1 0, 1, 0 0

1.2 1, 1, 0 0

2.1 1, 0, 0 0

2.2 1, 0, 1 1

3.1 0, 0, 0 0

3.2 0, 0, 1 0

3.3 0, 1, 1 1

3.4 1, 1, 1 1

Please enter the number of any vectors which need any changes in a comma-separalited list (e.g. 1.1, 3.2, ..., 7.4):

The vector specified does not exist, so it has been skipped.

This dataset represents the nested attributes for the attribute x2

Expert Data Mining with Hansel Chains.

What order to use for the Hansel Chains?

0 - Shortest Hansel Chain First

1 - Longest Hansel Chain First

2 - Manual Hansel Chain Order

3 - Default Hansel Chain Algorithm Order

Enter: 0

Is there any attribute which must be true for the datapoint to be true?

Enter the number assigned to each attribute or -1 if there is no such attribute.

If there are multiple attributes, separate them with a comma.

a - 0

b - 1

c - 2

Enter: 1

Do you want to use chain jumping (1/0)?

Enter: 0

Use majority flag(1/0)?

Enter: 0

Enter the class for this data point:

a = 0

b = 1

c = 0

Enter Class: 1

Checking for f-changes:

Number Datapoint Class

1.1 0, 1, 0 1

1.2 1, 1, 0 1

2.1 1, 0, 0 0

2.2 1, 0, 1 0

3.1 0, 0, 0 0

3.2 0, 0, 1 0

3.3 0, 1, 1 1

3.4 1, 1, 1 1

Please enter the number of any vectors which need any changes in a comma-separalited list (e.g. 1.1, 3.2, ..., 7.4):

The vector specified does not exist, so it has been skipped.

This dataset represents the nested attributes for the attribute x2

Expert Data Mining with Hansel Chains.

What order to use for the Hansel Chains?

0 - Shortest Hansel Chain First

1 - Longest Hansel Chain First

2 - Manual Hansel Chain Order

3 - Default Hansel Chain Algorithm Order

Enter: 0

Is there any attribute which must be true for the datapoint to be true?

Enter the number assigned to each attribute or -1 if there is no such attribute.

If there are multiple attributes, separate them with a comma.

a - 0

b - 1

Enter: 1

Do you want to use chain jumping (1/0)?

Enter: 0

Use majority flag(1/0)?

Enter: 0

Enter the class for this data point:

a = 0

b = 1

Enter Class: 1

Checking for f-changes:

Number Datapoint Class

1.1 1, 0 0

2.1 0, 0 0

2.2 0, 1 1

2.3 1, 1 1

Please enter the number of any vectors which need any changes in a comma-separalited list (e.g. 1.1, 3.2, ..., 7.4):

The vector specified does not exist, so it has been skipped.