

In this tutorial we explain how Amnesia can help the user to generate a hierarchy from an existing dataset and how the user can further edit it, using a graphical editor.

Amnesia helps the user generate a hierarchy for an existing dataset on a specific attribute. The basic idea is that the user will upload a dataset and will instruct Amnesia to create a hierarchy on one of its attributes.

For distinct values, e.g., strings, Amnesia will organize the active domain of the attribute, i.e., all the different values appearing in the selected attribute, into groups that will comprise the leaves of a generalization hierarchy. The user must further define the fanout of the generalization tree, i.e., the number of children of each node. Amnesia will organize the active domain to groups of fanout size and choose a random value as their generalization. Then it will organize these random values to groups of fanout size and choose new random values as their generalization. This process will be repeated until there is a single value (the root of the generalization tree) as a generalization of every value appearing in the dataset.

For continuous domains, the hierarchy generation process is more natural and avoids the generation of random values. The hierarchy generated by Amnesia organizes the domain and not just the active domain of the selected attribute, i.e., the hierarchy does not organize only the values that appear in the selected attributes into groups, but instead defines ranges in the whole domain that covers every appearing value. For example, it indicates that values 2,3 and 6 will be replaced by the range [0-10]. To this end, the user must define the step, i.e., the size of ranges in which the domain is partitioned and the fanout of the generalization hierarchy. The following example we demonstrate how to create a hierarchy.

Hierarchy autogeneration

The user selects the automatic hierarchy generalization by clicking “Autogenerate hierarchy”.



To autogenerate a hierarchy a dataset must be loaded in advance and then the user has to choose which attribute that will be used in the hierarchy generation. Furthermore, the user must choose the type of the hierarchy (distinct or range) and the variable type.


1. Choose Attribute and Hierarchy Type 2. Hierarchy Info

Choose Attribute

On Attribute

Type

VarType



By clicking next, the user moves to the next screen where she/he has to give more information about the construction of hierarchy depending of the hierarchy type.

If the hierarchy is distinct type, the user has to choose the sorting (numeric, alphabetical or random) which defines how the leaf values are grouped. The values are first sorted according to the defined sorting order and then placed into groups of <fanout> size, starting from the first. The user has also to provide the name of hierarchy and the fanout (the number of children of each node).

1. Choose Attribute and Hierarchy Type 2. Hierarchy Info

Hierarchy Information

Sorting

Name

Fanout

If the hierarchy is range type, the user has to define the range of each leaf node, the name of hierarchy, the domain size (start-end limits) and the fanout.

Hierarchy Autogenerate

1. Choose Attribute and Hierarchy Type | **2. Hierarchy Info**

Hierarchy Information

Step	<input type="text" value="10"/>
Name	<input type="text"/>
Domain start End limit	<input type="text" value="18-100"/>
Fanout	<input type="text" value="10"/>

Previous Finish Cancel

© 2016, All Rights Reserved.

If the variable type is date, the user must provide additional information about the hierarchy's properties, since we do not measure time in the decimal system but rather in terms of days, months and years. The user must define separate steps for the aforementioned levels of granularity (days, months, years) and the fanout to build the hierarchy over years.

Hierarchy Autogenerate

1. Choose Attribute and Hierarchy Type | **2. Hierarchy Info**

Hierarchy Information

Name	<input type="text"/>
Domain start End limit	<input type="text" value="1940-2016"/>
Range step in terms of years	<input type="text" value="1"/>
Range step in terms of months	<input type="text" value="2"/>
Range step in terms of days	<input type="text" value="7"/>
Fanout of years nodes	<input type="text" value="0"/>

Previous Finish Cancel

© 2016, All Rights Reserved.

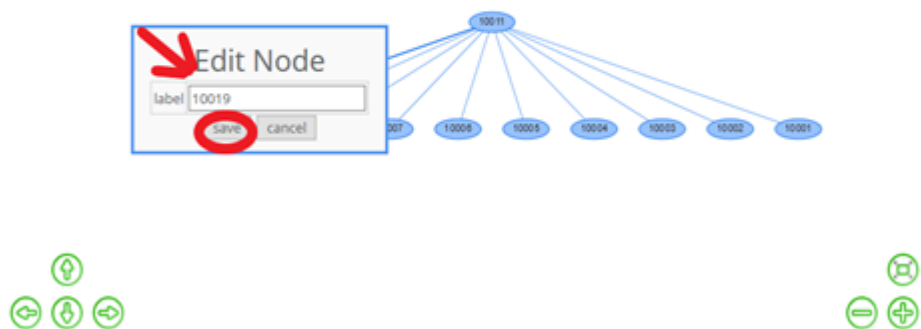
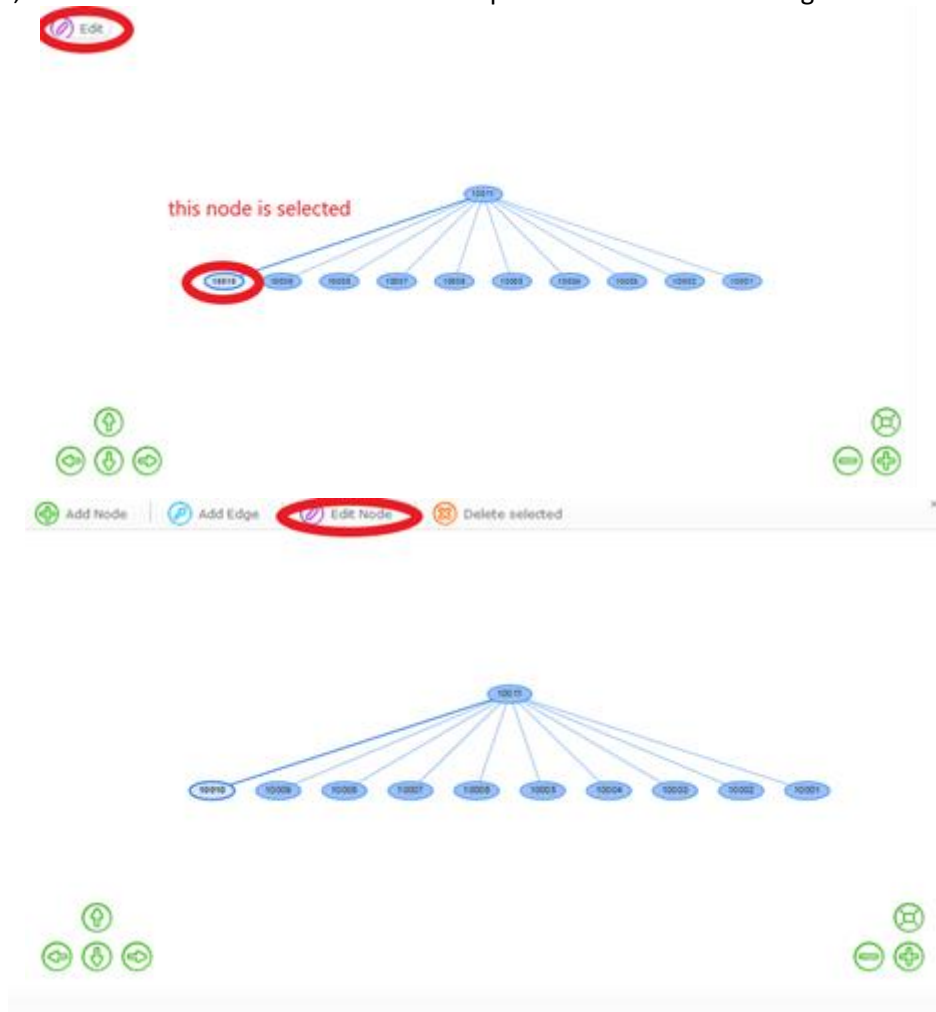
Finally, the hierarchy is ready to be used.

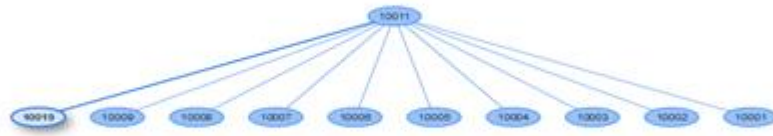
Editing a hierarchy

After uploading or autogenerating your hierarchy the user edit it, by changing a node's value or deleting or adding nodes.

Change node value

In distinct hierarchies the value of the root node cannot be altered. However, the rest of the nodes can be changed, deleted and additional nodes can be added. To edit a node, the user has to select the node by clicking on it and then click the “edit” button. An edit menu will appear, where she or he has to the edit node option as we see in the images below.





Delete node

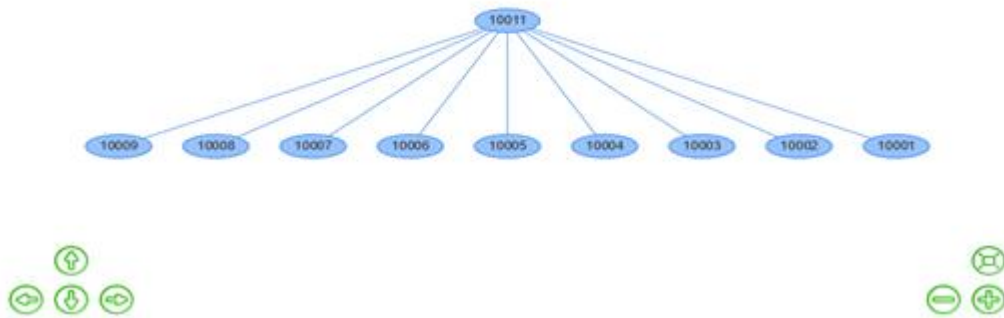
Deleting a node is simpler; the user only needs to select the node, click edit and then select "delete node".



this node is selected

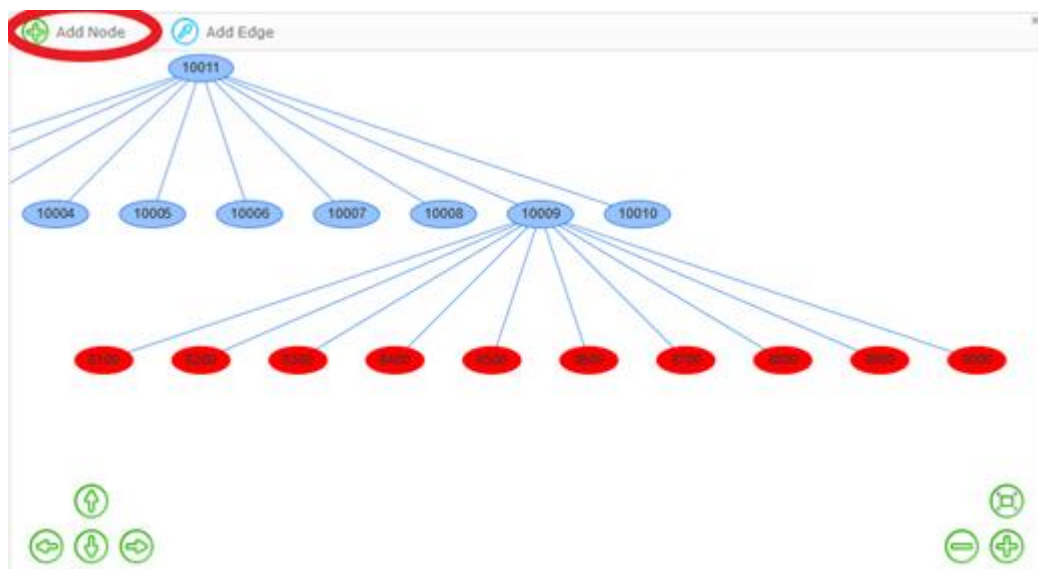


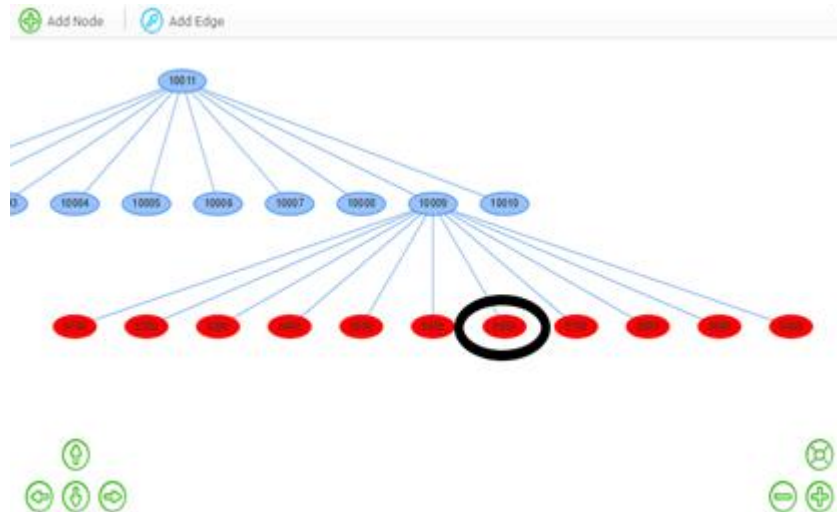
Edit



Add node

To add a node in a hierarchy the user has first to click the “Edit” button and then the “add node” button. Then by clicking at any empty point (i.e., a point that does not fall inside an existing node) in the hierarchy canvas, a pop-up dialog will appear. By filling out the value of the node, the node is created, but it is not yet integrated in the hierarchy. To add it at the right parent node, the user must click the “add edge” button of the edit menu, and connect the parent node to the new one using the pointer.





An edited hierarchy must be saved in order to be available for future use. If the user wants to discard then she or he must click the “Remove Hierarchy” button.

Hierarchy

version:1.0.7 beta

Load or create a new generalization hierarchy. A generalization hierarchy will define how values are mapped to more generalized concepts. A generalization hierarchy can be defined independently of any dataset.

